**HOSPITAL MANAGEMENT SYSTEM**

**Developed by-**

**Narendhar Kumar G**

**ABSTRACT**

The project entitled as “Hospital Management Process”. It can be used to keep track of the patients registering in a hospital or clinic. Also this system supports accessing the previous visit histories of any patient, search for patients by name and other properties etc. patient care system Management will support registering patients. Developed this software by using Asp.Net is used as the front-end tool and SQL Server is used as the back-end tool.

Users of this software can search for patients by name, admission date, discharge date etc. Users can view the previous visit histories of any patient. This system can maintain the list of doctors in the hospital and also can maintain the list of beds/rooms available in the hospital. Patients are categorized into "In Patients" and "Out Patients". The objectives of it are to make more efficient and convenient storage and retrieval of large volume of data relating to advance. The system should be totally menu driven, interactive and highly user friendly.

Hospital Management Process is a website application. This application is designed for multi specialty hospitals, to cover a wide range of hospital administration and management processes. It can be used anywhere at any time. Patient Registration can be possible through website. After registration patient password send to mail. Then Patients can see the available doctors and fixed appointment through website.

In Previous application bills are generated by recording price for each facility provided to Patient on a separate sheet and at last they all are summed up. But in this application provide easy calculations and bills also accept through online transactions. The system provides an attractive and easy to handle user interface which follows computerized standards. Considerations have been given at the design and development stages of the package in order to ensure data integrity and security. It is highly user friendly. No deep knowledge of computers is required to operate this system.

Functions such as data entry addition, deletion, modification, navigation etc., have been made simpler and interactive. The system is fully modular in design and protected from unauthorized access.

**INTRODUCTION**

Hospital Management process is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. Patient Record System is designed for multi specialty hospitals, to cover a wide range of Patient administration and management processes.

It is an integrated end-to-end Patient Management System that provides relevant information across the Patient to support effective decision making for patient care, Patient administration and critical financial accounting, in a seamless flow.

Hospital Management Process is a software product suite designed to improve the quality and management of Patient management in the areas of clinical process analysis and activity-based costing. Patient Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

**Objective:-**

1. Define hospital
2. Recording information about the Patients that come.
3. Generating bills.
4. Recording information related to diagnosis given to Patients.
5. Keeping record of the Immunization provided to children/patients.
6. Keeping information about various diseases and medicines available to cure them.

These are the various jobs that need to be done in a Hospital by the operational staff and Doctors. All these works are done by system. Most of the hospitals are using manual and computer methods for managing various types of works which is time taking process. In order to reduce work we designed simple software application through which doctors , patients, billing, employees details are managed.

**MODULES**

* **Administration**
* **Hospital Module**
* **Doctor Details**
* **Employee Details**
* **Appointment Fix**
* **Patient Registration**
* **Inpatient**
* **Outpatient**
* **Report**

**MODULES DESCRIPTION**

**Administration**

In this module Authorized users are allowed to access. User enters the User type, Username and password. Checks User type, Username and password are valid or not. User is directed to next page or shows the message box “Login Failed”.

**Hospital Module**

This module manages hospital activities and functions of the all. It also displays the list of departments present in hospital. We can search a hospital by using id and then view hospital details.

**Doctor Details:**

This interface provides us the facilities such as viewing doctor's details including his/her id, qualification and specialization, address etc.. It also displays the list of doctors present in various departments. We can search a doctor by his id and then view his details. This interface also provides us the facility to add new doctors in.

**Employee Details:**

This Module contains Employee Details Like Employee Id, Name, Designation, Address etc. Employee can add Inpatient Details. Employee Maintained Inpatient details and Outpatient Details.

**Patient Registration through website**

Registration of patients involves accepting certain general and demographic information about the patient. Patients can register through website. The Patient ID will remain same for his all subsequent visits to the hospital whereas he will be allocated a new registration number on every visit. Patient Demographic Details like Name, Age, Gender, Address, Contact number, etc.. Patients can register through website. Then Patient receive password through mail.

**Appointment Fix**

This Module provides Doctors Available information. Then The patient can Fix An Appointment through the website.

**Inpatient**

This module manages day to day activities and functions of the Inpatient. It also provides Data for Managerial and Administrative functions pertaining to Admission of Patients, Ward Management and also Dietary Management.  
**Outpatient**

This module manages requests and visits of Outpatients. Tracking of consultant Doctors for OPD /Indoor procedures. It also provides the option for defining consultation based on Procedures/Department. Also Classifies visit of the patient as New/Old for that consultation. The system manages the consultation Date & Time for the doctors and the patients.

**Report Module:**

In this module only Admin can access the page and check the report. Enter Username, from date, to date and then get the reports. It retrieve’s all the information stored from different tables. Requested report is generated.

**SYSTEM SPECIFICATION**

**HARDWARE SPECIFICATION**

|  |  |  |
| --- | --- | --- |
| PROCESSOR | : | Intel Pentium Dual Core |
| MOTHERBOARD | : | Intel 915GVSR chipset board |
| RAM | : | 2 GB DDR3 RAM |
| HARD DISK DRIVE | : | 500 GB HDD |
| KEY BOARD | : | 104 KEYS STANDARD |
| MOUSE | : | OPTICAL MOUSE |
| MONITOR | : | 15”COLOR MONITOR |

**SOFTWARE SPECIFICATION**

|  |  |  |
| --- | --- | --- |
| DEVELOPMENT ENVIRONMENT | : | WINDOWS 7 / 10 |
| FRAMEWORK | : | .NET FRAMEWORK 4.5 |
| DEVELOPMENT TOOL | : | VISUAL STUDIO 2012 |
| FRONT END | : | ASP.NET |
| CODING LANGUAGE | : | C# |
| CONNECTVITY TOOL | : | ADO.NET |
| BACK END | : | SQL SERVER 2008 |
| THIRD PARTY TOOLS | : | CSS, AJAX, JAVASCRIPT |
| REPORT | : | GRID / EXCEL |

**EXISTING SYSTEM**

The system, which is followed at present, is a computerized system. The type of manual entry made so many corrections in the reporting time and also difficult and frequent occurrence of error. Important drawback of existing system is time factor and data storage. It would not be help Management to precede the problem in entry time. It will take long time to prepare reports or else. The data loss also most important factor in the existing system.

**Drawbacks of existing system**

* Lack of immediate retrievals
* Lack of prompt updating and immediate information storage
* Lack of data security
* Error prone manual stock taking
* Preparation of accurate and prompt reports

**PROPOSED SYSTEM**

The present Hospital Management System is designed based on the customer reviews and ideas as this has to fit for both large and even to small size hospitals around the world. This application records everything right from the registration of the patient to what kind of problem they have and what kind of treatment they are suggesting and what test they have conduction on the patient what was the result what medicine’s they have give and later what is the condition of the patient. Everything in detail will be available with this application. A network will be needed to connect the computers within the facility. Implementation of the automated system will require buying of computers that will act as a platform upon which the system well run. Implementation of the system calls for employment a person has knowledge about computers.

**Advantage**

* Planned approach towards working
* Reliability and No Redundancy
* Accuracy and correct data
* Data Security
* Time Saving
* Immediate retrieval of information and storage of information
* Easy to Operate

**LANGUAGE SPECIFICATION**

**What is .NET?**

When .NET was announced in late 1999, Microsoft positioned the technology as a platform for building and consuming Extensible Markup Language (XML) Web services. XML Web services allow any type of application, be it a Windows- or browser-based application running on any type of computer system, to consume data from any type of server over the Internet.

The reason this idea is so great is the way in which the XML messages are transferred: over established standard protocols that exist today. Using protocols such as SOAP, HTTP, and SMTP, XML Web services make it possible to expose data over the wire with little or no modifications to your existing code.Figure presents a high-level overview of the .NET Framework and how XML Web services are positioned.

**Introduction .NET Framework**

Now that you are familiar with the major goals of the .NET Framework, let's briefly examine its architecture. As you can see in Figure 1-2, the .NET Framework sits on top of the operating system, which can be a few different flavors of Windows and consists of a number of components .NET is essentially a system application that runs on Windows.

Conceptually, the CLR and the JVM are similar in that they are both runtime infrastructures that abstract the underlying platform differences. However, while the JVM officially supports only the Java language, the CLR supports any language that can be represented in its Common Intermediate Language (CIL). The JVM executes bytecode, so it can, in principle, support many languages, too. Unlike Java's bytecode, though, CIL is never interpreted. Another conceptual difference between the two infrastructures is that Java code runs on any platform with a JVM, whereas .NET code runs only on platforms that support the CLR. In April, 2003, the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC) recognized a functional subset of the CLR, known as the Common Language Interface (CLI), as an international standard. This development, initiated by Microsoft and developed by ECMA International, a European standards organization, opens the way for third parties to implement their own versions of the CLR on other platforms, such as Linux or Mac OS X. For information on third-party and open source projects working to implement the ISO/IEC CLI and C# specifications

The layer on top of the CLR is a set of framework base classes. This set of classes is similar to the set of classes found in STL, MFC, ATL, or Java. These classes support rudimentary input and output functionality, string manipulation, security management, network communications, thread management, text management, reflection functionality, collections functionality, as well as other functions.

On top of the framework base classes is a set of classes that extend the base classes to support data management and XML manipulation. These classes, called ADO.NET, support persistent data management—data that is stored on backend databases. Alongside the data classes, the .NET Framework supports a number of classes to let you manipulate XML data and perform XML searching and XML translations.

Classes in three different technologies (including web services, Web Forms, and Windows Forms) extend the framework base classes and the data and XML classes. Web services include a number of classes that support the development of lightweight distributed components, which work even in the face of firewalls and NAT software. These components support plug-and-play across the Internet, because web services employ standard HTTP and SOAP.

Web Forms, the key technology behind ASP.NET, include a number of classes that allow you to rapidly develop web Graphical User Interface (GUI) applications. If you're currently developing web applications with Visual Interdev, you can think of Web Forms as a facility that allows you to develop web GUIs using the same drag-and-drop approach as if you were developing the GUIs in Visual Basic. Simply drag-and-drop controls onto your Web Form, double-click on a control, and write the code to respond to the associated event.

**VISUAL STUDIO.NET**

VS.NET is the [application](http://itxt.vibrantmedia.com/al.asp?ipid=111&cc=us&ai=15598777&di=64340&ts=20040122055531&redir=http://www61.overture.com/d/sr/?xargs=02u3hs9yoaj12POzCDhBDN%2F2iWg22LsXwtpDwpdI%2Bu8YyWSRDAUTDv8kHDj3FW1d2jb5m%2Ftv5mp%2BUW6Iff7%2FQw8AXnl0KBiJYjPCNwSLtwcIxLtX6KR6H26DTYDu729occtttvfr%2FvNuwfpGxWgwrAsqUP4oyeXOywFwBXZXRoxyWcHFONk7znnTghwpuPFNiC2WhdcqxJwWoDGy7DyuaMtLPp7%2FEzjzP1noyY8IjWqrpUlSw1CSu1LXSCBZlTdYL%2FcvWGJZ5ZrOnfLB8kRls%3D" \t "_blank) development tool to develop [applications](http://itxt.vibrantmedia.com/al.asp?ipid=111&cc=us&ai=15598777&di=23001&ts=20040122055531&redir=http://www60.overture.com/d/sr/?xargs=02u3hs9yoaj12POzCDhBDN%2F2iWwP3CjwWFSkHil8j6RxgEsoCGLon%2B%2FxI4jylW1d2jb5lP1nCzYw9z8mLoOv7N3IlpZtDAQ8IbbhFRhpvwcIxzcmFOtzZu2HmxHNj53Oh368%2B%2F8rhdtbgvz6FWsqmqwkuKUlk%2BVslgzg5bZPRqIpLvB5jSxq5TTb0fcct7TDxB0WhdUKwJRmoWD036yuSENkfot%2BPlj%2FsZhxNwYMafurKVqg0Ui8xaYQVoOPnTMd2Olh3J4ywQoXiCCw%2F0refhWw%3D%3D" \t "_blank) for .NET. It supports development of all types of applications that .NET supports. It also provides support for [VB.NET](http://itxt.vibrantmedia.com/al.asp?ipid=111&cc=us&ai=15598777&di=23269&ts=20040122055531&redir=http://www60.overture.com/d/sr/?xargs=02u3hs9yoaj1VuzTCDhBBm30Vvb2vgbAew1wduUvQ5AR4rdn4kMFJ5KBL7qVNq6qtOzNHrN7D%2F8nYr4W%2BMPbtTGa5kmhECyNZasIhJMT1ohEs9azHHvMzNdRZe4If6uxj22%2Fq8usPkvWfx9tCPkUgjk3Z9rPJwjKIHuEgIUgpUCQl62sac3m8Xmhp2TuTd4itIKbQirKCrn9%2BXlNSVWOpr%2BKo%2Buj4wIXOmbIaLZSqCF5hSpgWXy8S4nfyZlhOVy2YFrKcIzAB%2FaRzZUQ%3D%3D" \t "_blank) ,C#, Visual C++.Net and Visual J# languages. VS.NET is a single environment that provides all tools required to develop and debug applications.

The following are key features of Visual Studio.NET.

**Languages supported** - VS.NET supports application development using the language of your choice. It also allows mixed language solutions.

**IntelliSence** - IntelliSense provides options that make programming in VS.NET easier than ever before.

The following are important functions of intellisense.

Allows you to see the syntax of the method that you are calling

Completes the variable, command, or function name once you have entered enough characters to disambiguate the term.

Displays the list of valid members for the class, structure or namespace you type so that you can select one form the list. It places the selected member in your code.

Automatically brace matching allows whether ending brace is given after opening brace is given.

**WHAT IS ASP.NET?**

For creating dynamic web pages using server-side scripting, Microsoft introduced ASP. ASP.NET is the .NET version of ASP. ASP.NET is a standard HTML file that contains embedded server-side scripts.

* ASP.NET offers a whole new approach to developing web applications
* ASP.NET offers a richer programming model that emulates an event-driven programming environment.
* ASP.NET helps create dynamic web pages with ease
* ASP.NET allows us to combine standard HTML elements (such as Tables, Text, etc) with controls (such as Labels, Datagrids, DropDownLists, etc) along with event-driven code to produce a web page that is dynamically generated every time your page is requested from a browser.

**ASP.NET provides the following advantages of server-side scripting:**

* ASP.NET enables you to access information from data sources, such as back-end databases and text files that are stored on a Web Server or a computer that is accessible to a Web Server.
* ASP.NET enables you to use a set of programming code called templates to create HTML documents. The advantage of using templates is that you can dynamically insert content retrieved from data sources, such as back-end databases and text files, into an HTML document before the HTML document is displayed to users. Therefore, the information need not be changed manually as and when the contents retrieved from data source change.
* ASP.NET also enables you to separate HTML design from the data retrieval mechanism. Therefore, changing the HTML design does not affect the programs that retrieve data from the databases. Similarly, server-side scripting ensures that changing data sources does not require a change in HTML documents.

**C# Language**

**C#** (pronounced C Sharp) is a [multi-paradigm programming language](http://en.wikipedia.org/wiki/Multi-paradigm_programming_language" \o "Multi-paradigm programming language) that encompasses [functional](http://en.wikipedia.org/wiki/Functional_programming" \o "Functional programming), [imperative](http://en.wikipedia.org/wiki/Imperative_programming" \o "Imperative programming), [generic](http://en.wikipedia.org/wiki/Generic_programming" \o "Generic programming), [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming" \o "Object-oriented programming) ([class-based](http://en.wikipedia.org/wiki/Class_%28computer_science%29" \o "Class (computer science))), and [component-oriented](http://en.wikipedia.org/wiki/Component-based_software_engineering" \o "Component-based software engineering) programming disciplines. It was developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft" \o "Microsoft) as part of the [.NET](http://en.wikipedia.org/wiki/Microsoft_.NET" \o "Microsoft .NET) initiative and later approved as a standard by [ECMA](http://en.wikipedia.org/wiki/Ecma_International" \o "Ecma International) (**ECMA-334**) and [ISO](http://en.wikipedia.org/wiki/International_Organization_for_Standardization" \o "International Organization for Standardization) (**ISO/IEC 23270**). C# is one of the 44 programming languages supported by the [.NET Framework](http://en.wikipedia.org/wiki/.NET_Framework" \o ".NET Framework)'s [Common Language Runtime](http://en.wikipedia.org/wiki/Common_Language_Runtime" \o "Common Language Runtime).

**Features of C#**

By design, C# is the programming language that most directly reflects the underlying Common Language Infrastructure (CLI). Most of C#'s intrinsic types correspond to value-types implemented by the CLI framework. However, the C# language specification does not state the code generation requirements of the compiler: that is, it does not state that a C# compiler must target a Common Language Runtime (CLR), or generate Common Intermediate Language (CIL), or generate any other specific format. Theoretically, a C# compiler could generate machine code like traditional compilers of C++ or FORTRAN; in practice, all existing C# implementations target CIL.

Some notable C# distinguishing features are:

* There are no global variables or functions. All methods and members must be declared within classes. It is possible, however, to use static methods/variables within public classes instead of global variables/functions.
* Local variables cannot shadow variables of the enclosing block, unlike C and C++. Variable shadowing is often considered confusing by C++ texts.

**Overview of ADO.NET**

Most applications require some kind of data access. Desktop applications need to integrate with central databases, Extensible Markup Language (XML) data stores, or local desktop databases. ADO.NET data-access technology allows simple, powerful data access while maximizing system resource usage.

Different applications have different requirements for data access. Whether your application simply displays the contents of a table, or processes and updates data to a central SQL server, ADO.NET provides the tools to implement data access easily and efficiently.

**Disconnected Database Access**

Previous data-access technologies provided continuously connected data access by default. In such a model, an application creates a connection to a database and keeps the connection open for the life of the application, or at least for the amount of time that data is required. However, as applications become more complex and databases serve more and more clients, connected data access is impractical for a variety of reasons, including the following:

* Open database connections are expensive in terms of system resources. The more open connections there are, the less efficient system performance becomes.
* Applications with connected data access are difficult to scale. An application that can comfortably maintain connections with two clients might do poorly with 10 and be completely unusable with 100.
* Open database connections can quickly consume all available database licenses, which can be a significant expense. In order to work within a limited set of client licenses, connections must be reused whenever possible.

ADO.NET addresses these issues by implementing a disconnected data access model by default. In this model, data connections are established and left open only long enough to perform the requisite action. For example, if an application requests data from a database, the connection opens just long enough to load the data into the application, and then it closes. Likewise, if a database is updated, the connection opens to execute the UPDATE command, and then closes again.

**ADO.NET Data Architecture**

Data access in ADO.NET relies on two entities: the DataSet, which stores data on the local machine, and the Data Provider, a set of components that mediates interaction between the program and the database.

**The DataSet**

The DataSet is a disconnected, in-memory representation of data. It can be thought of as a local copy of the relevant portions of a database. Data can be loaded into a DataSet from any valid data source, such as a SQL Server database, a Microsoft Access database, or an XML file. The DataSet persists in memory, and the data therein can be manipulated and updated independent of the database. When appropriate, the DataSet can then act as a template for updating the central database.

The DataSet object contains a collection of zero or more DataTable objects, each of which is an in-memory representation of a single table. The structure of a particular DataTable is defined by the DataColumns collection, which enumerates the columns in a particular table, and the Constraint collection, which enumerates any constraints on the table. Together, these two collections make up the table schema. A DataTable also contains a DataRows collection, which contains the actual data in the DataSet.

The DataSet contains a DataRelations collection. A DataRelation object allows you to create associations between rows in one table and rows in another table. The DataRelations collection enumerates a set of DataRelation objects that define the relationships between tables in the DataSet. For example, consider a DataSet that contains two related tables: an Employees table and a Projects table. In the Employees table, each employee is represented only once and is identified by a unique EmployeeID field. In the Projects table, an employee in charge of a project is identified by the EmployeeID field, but can appear more than once if that employee is in charge of multiple projects. This is an example of a one-to-many relationship; you would use a DataRelation object to define this relationship. Additionally, a DataSet contains an Extended Properties collection, which is used to store custom information about the DataSet

**The Data Provider**

The link to the database is created and maintained by a data provider. A data provider is not a single component, rather it is a set of related components that work together to provide data in an efficient, performance-driven manner. The first version of the Microsoft .NET Framework shipped with two data providers: the SQL Server .NET Data Provider, designed specifically to work with SQL Server 7 or later, and the OleDb .NET Data Provider, which connects with other types of databases. Microsoft Visual Studio .NET 2010 added two more data providers: the ODBC Data Provider and the Oracle Data Provider. Each data provider consists of versions of the following generic component classes:

* The Connection object provides the connection to the database.
* The Command object executes a command against a data source. It can execute non-query commands, such as INSERT, UPDATE, or DELETE, or return a DataReader with the results of a SELECT command.
* The DataReader object provides a forward-only, read-only, connected recordset.
* The DataAdapter object populates a disconnected DataSet or DataTable with data and performs updates.

Data access in ADO.NET is facilitated as follows: a Connection object establishes a connection between the application and the database. This connection can be accessed directly by a Command object or by a DataAdapter object.

The Command object provides direct execution of a command to the database. If the command returns more than a single value, the Command object returns a DataReader to provide the data. This data can be directly processed by application logic. Alternatively, you can use the DataAdapter to fill a DataSet object. Updates to the database can be achieved through the Command object or through the DataAdapter.The generic classes that make up the data providers are summarized in the following sections.

**The Connection Object**

The Connection object represents the actual connection to the database. Visual Studio .NET 2010 supplies two types of Connection classes: the SqlConnection object, which is designed specifically to connect to SQL Server 7 or later, and the OleDbConnection object, which can provide connections to a wide range of database types.

Visual Studio .NET 2010 further provides a multipurpose ODBCConnection class, as well as an OracleConnection class optimized for connecting to Oracle databases. The Connection object contains all of the information required to open a channel to the database in the ConnectionString property. The Connection object also incorporates methods that facilitate data transactions.

**The Command Object**

The Command object is represented by two corresponding classes, SqlCommand and OleDbCommand. You can use Command objects to execute commands to a database across a data connection. Command objects can be used to execute stored procedures on the database and SQL commands, or return complete tables. Command objects provide three methods that are used to execute commands on the database:

* ExecuteNonQuery.
  + Executes commands that return no records, such as INSERT, UPDATE, or DELETE
* ExecuteScalar.
  + Returns a single value from a database query
* ExecuteReader.
  + Returns a result set by way of a DataReader object

**The DataReader Object**

The DataReader object provides a forward-only, read-only, connected stream recordset from a database. Unlike other components of a data provider, DataReader objects cannot be directly instantiated. Rather, the DataReader is returned as the result of a Command object’s ExecuteReader method. The SqlCommand.Execute­Reader method returns a SqlDataReader object, and the OleDbCommand.ExecuteReader method returns an OleDbDataReader object.

Likewise, the ODBC and Oracle Command.ExecuteReader methods return a DataReader specific to the ODBC and Oracle Data Providers respectively. The DataReader can supply rows of data directly to application logic when you do not need to keep the data cached in memory. Because only one row is in memory at a time, the DataReader provides the lowest overhead in terms of system performance, but it requires exclusive use of an open Connection object for the lifetime of the DataReader.

**The DataAdapter Object**

The DataAdapter is the class at the core of ADO.NET disconnected data access. It is essentially the middleman, facilitating all communication between the database and a DataSet. The DataAdapter fills a DataTable or DataSet with data from the database whenever the Fill method is called. After the memory-resident data has been manipulated, the DataAdapter can transmit changes to the database by calling the Update method. The DataAdapter provides four properties that represent database commands. The four properties are:

* SelectCommand**.**

Contains the command text or object that selects the data from the database. This command is executed when the Fill method is called and fills a DataTable or a DataSet.

* InsertCommand**.**

Contains the command text or object that inserts a row into a table.

* DeleteCommand**.**

Contains the command text or object that deletes a row from a table.

* UpdateCommand**.**

Contains the command text or object that updates the values of a database.

When the Update method is called, changes in the DataSet are copied back to the database, and the appropriate InsertCommand, DeleteCommand, or UpdateCommand is executed.

**Accessing Data**

Visual Studio .NET has many built-in wizards and designers to help you shape your data-access architecture rapidly and efficiently. With minimal actual coding, you can implement robust data access for your application. However, the ADO.NET object model is fully available through code to implement customized features or to fine-tune your program. In this lesson, you will learn how to connect to a database with ADO.NET and retrieve data to your application. You will learn to use the visual designers provided by Visual Studio .NET and direct code access.

**SQL SERVER 2008**

Microsoft SQL server lets you quickly build powerful and reliable database applications. SQL server highly scalable, fully relational, high performance, multi-user database server. That can be used by enterprise of any size to manage large amount of data for client\server applications.

The major new and improved features of SQL server include the multi-user support Multi platform support, added memory support, scalability, integration with MMC, Microsoft Management console and improved multiple server management. Parallel database backup and restore. Data replication, Data warehousing distributed queries, distributed transactions, Dynamic cocking Internet Access, Integrated windows security, Mail integration Microsoft English Query, ODBC Support.

SQL Server management is accomplished through a set of component applications. SQL Server introduces a number of new and improved management tools that are SQL Server Enterprise management, profiles, and Query Analyzer service manager wizards.

The OLAP Services feature available in SQL Server version is now called SQL Server 2008 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version is now called Microsoft SQL Server 2008 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects,

They are,

**1. TABLE**

**2. QUERY**

**3. FORM**

**4. REPORT**

**5. MACRO**

**TABLE:**

A database is a collection of data about a specific topic.

**VIEWS OF TABLE:**

We can work with a table in two types,

1. Design View

2. Datasheet View

**Design View**

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

**Datasheet View**

To add, edit or analyses the data itself we work in tables datasheet view mode.

**QUERY:**

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot(it cannot be edited).Each time we run query, we get latest information in the dynaset.Access either displays the dynaset or snapshot for us to view or perform an action on it ,such as deleting or updating.

**FORMS:**

A form is used to view and edit information in the database record by record .A form displays only the information we want to see in the way we want to see it. Forms use the familiar controls such as textboxes and checkboxes. This makes viewing and entering data easy.

**Views of Form:**

We can work with forms in several primarily there are two views,

They are,

**1. Design View**

**2. Form View**

**Design View**

To build or modify the structure of a form, we work in forms design view. We can add control to the form that are bound to fields in a table or query, includes textboxes, option buttons, graphs and pictures.

**Form View**

The form view which display the whole design of the form.

**REPORT:**

A report is used to vies and print information from the database. The report can ground records into many levels and compute totals and average by checking values from many records at once. Also the report is attractive and distinctive because we have control over the size and appearance of it.

**MACRO:**

A macro is a set of actions. Each action in macros does something. Such as opening a form or printing a report .We write macros to automate the common tasks the work easy and save the time.

**DATAFLOW DIAGRAM**

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

**Dfd Symbols:**

In the DFD, there are four symbols

* A square defines a source(originator) or destination of system data
* An arrow identifies data flow. It is the pipeline through which the information flows
* A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
* An open rectangle is a data store, data at rest or a temporary repository of data

Process that tran forms data flow.

Source or Destination of data

Data flow

Data Store

Level 0:

Efficient Doctor and Patient Portal

Admin/User

Validation

NO

YES

Home

**Level-1**

Login

Admin

No

Validation

Yes

Home

Response

Add Hospital

Stored Hospital Details

Hospital

Hospital Db

Add Doctor

Stored doctor Details

Doctor

Doctor Db

Add Employee

Stored employee details

Employee Db

Employee

Feedback Db

View Feedback and organs

**Level-2**

Login

Employee

No

Validation

Yes

Home

Add Room Details

Stored DB

Room Db

details

Add Inpatient Details

Stored DB

Add\_InPatient

details

Inpatient and Outpatient Maintenance

Stored DB

Out\_Patient Db

details

Feedback Db

View Feedback

**Level-3**

Login

Doctor

No

Validation

Yes

Home

View Appointment

Stored DB

Appointment Db

details

EmpAtt Db

Employee Attendance

Stored DB

details

Patient History

View Feedback

Feedback Db

**Level-4**

Login

User

No

Validation

Yes

Home

Doctor Available

Stored DB

Add Doctor Db

details

Appointment Fix

Stored DB

Appointment Db

details

Stored DB

Out Patient Db

History

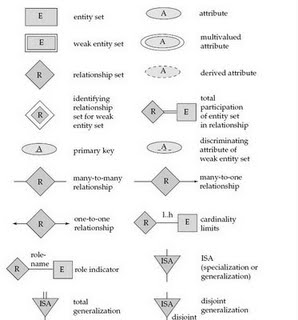
details

Feedback

**E-R DIAGRAM**

* + The relation upon the system is structure through a conceptual ER-Diagram, which not only specifics the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
  + The entity Relationship Diagram (ERD) depicts the relationship between the data objects. The ERD is the notation that is used to conduct the date modeling activity the attributes of each data object noted is the ERD can be described resign a data object descriptions.
  + The set of primary components that are identified by the ERD are
  + Data object
  + Relationships
  + Attributes
  + Various types of indicators.

The primary purpose of the ERD is to represent data objects and their relationships.



**ER Diagram**

Name

Id

username

Branch name

Password

M

1

Hospital

Hospital

Admin Login

Login

Branch id

id

name

Email

Mobile no

Doctor

nature problem

Outpatient

M

date

Prescription

OutPatient

name

Designation

id

Contact no

Doctor

Qualif

Address

Email

name

id

Addr

Employee

InPatient

nature problem

mobile

InPatient

name

id

Designation

EmailId

Employee

pwd

name

id

pwd

Contact no

Patient Registration

Patient

Registration

Address

Mobile

Mail

Dob

**SYSTEM FLOW DIAGRAM**

Start

Admin

Add Doctor Details

Login

Yes

No

Attendance

Add employee

Report

Employee

Login

No

Yes

Add InPatient Details

Maintain InPatient

History of Patient

User

Login

New Registration

Yes

No

Doctor Available

Appointment Fix

History of Patient

End

Logout

Report

Logout

Logout

**DATABASE DESIGN**

**Table Name : UserReg**

**primary key : Mail ID**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Size | Description |
| id | int | - | Identification |
| name | varchar | 50 | Name |
| password | varchar | 50 | password |
| natproblem | varchar | 50 | Nature problem |
| dob | varchar | 50 | Date of Birth |
| mailid | varchar | 50 | mailid |
| mobileno | varchar | 50 | Mobile Nmuber |
| age | varchar | 50 | age |

**Table Name : Add\_InPatient**

**primary key : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Size | Description |
| id | int | - | Identification |
| name | nchar | 10 | Name |
| dob | varchar | 50 | password |
| age | varchar | 10 | Nature problem |
| gender | varchar | 50 | Date of Birth |
| mailid | varchar | 50 | mailid |
| mobileno | varchar | 12 | Mobile Nmuber |
| address | varchar | 50 | age |
| city | varchar | 50 | city |
| state | varchar | 50 | state |
| natproblem | varchar | 50 | natproblem |
| ad\_date | varchar | 50 | ad\_date |

**Table Name : Add\_Doctor**

**primary key : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Size | Description |
| ID | int | - | Identification |
| password | varchar | 50 | Password |
| docname | varchar | 50 | Doctor Name |
| dob | varchar | 50 | Date of Birth |
| mobileno | varchar | 12 | Mobile Number |
| email | varchar | 50 | Email-ID |
| Address | varchar | 50 | Address |
| city | varchar | 50 | City |
| state | varchar | 50 | State |
| doj | varchar | 50 | Date of Joining |
| qualification | varchar | 50 | Qualification |
| designation | varchar | 50 | Designation |
| Special | Varchar | 50 | Special |
| Description | Varchar | 50 | Description |

**Table Name : Add\_Employee**

**primary key : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Size | Description |
| id | int | - | Identification |
| name | Varchar | 50 | Name |
| password | Varchar | 50 | Password |
| dob | Varchar | 50 | Date of Birth |
| mobileno | Varchar | 50 | Mobile Number |
| mailid | Varchar | 50 | Mail ID |
| address | Varchar | 50 | Address |
| city | Varchar | 50 | City |
| state | Varchar | 50 | State |
| doj | Varchar | 50 | Date of Joining |
| Designation | varchar | 50 | Designation |

**Table Name : Add\_Hospital**

**primary key : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Size | Description |
| id | int | - | Identification |
| Name | varchar | 50 | Name |
| Branchid | varchar | 50 | Branch ID |
| Branchname | varchar | 50 | Branch Name |
| Address | varchar | 50 | Address |
| Mailid | varchar | 50 | Mail ID |
| Mobno | varchar | 50 | Mobile Number |

**Table Name : Add\_Appointment**

**primary key : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Size | Description |
| name | varchar | 50 | Name |
| mobile | varchar | 12 | Mobile |
| age | varchar | 50 | Age |
| select\_date | varchar | - | Select Date |
| select\_time | varchar | - | Select Time |

**Table Name : Doctor Attendance**

**Foreign key : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Size | Description |
| Id | int | - | Identification |
| Docid | int | - | Doctor ID |
| Donname | varchar | 50 | Doctor Name |
| Attdate | datetime | - | Attendance Date |
| Attmonth | varchar | 50 | Attendance Month |

**Table Name : Employee Attendance**

**Foreign key : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Size | Description |
| Id | int | - | Identification |
| Empid | int | - | Employee ID |
| Empname | varchar | 50 | Employee Name |
| Attdate | datetime | - | Attendance Date |
| Attmonth | varchar | 50 | Attendance Month |

**Table Name : Out\_Patient**

**Foreign key : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Size | Description |
| id | int | - | Identification |
| name | varchar | 50 | Name |
| natprobelm | varchar | 50 | Nature Problem |
| bloodtestno | varchar | 50 | Blood Test Number |
| xrayno | varchar | 50 | Xray Number |
| ctscan | varchar | 50 | CT Scan |
| prescription | varchar | 50 | Prescription |
| Commends | varchar | 50 | Commends |
| date | varchar | 50 | Date |

**Table Name : Patient Maintenance**

**Foreign key : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Datatype | Size | Description |
| id | int | - | Identification |
| name | varchar | 50 | Name |
| natproblem | varchar | 50 | Nature Problem |
| bloodtestno | varchar | 50 | Blood Test Number |
| xrayno | varchar | 50 | Xray Number |
| ctscan | varchar | 50 | CT Scan |
| prescription | varchar | 50 | Prescription |
| commends | varchar | 50 | Commends |
| date | varchar | 50 | Date |

**INPUT DESIGN**

Input design is the part of overall system design which requires very careful attention. Often the collection of input data is the most expensive part of the system, in terms of both the equipment used and the number of people involved; it is the point of most contact for the users with the computer system; and it is prone to error. If data going into the system are incorrect, then the processing and output will magnify these errors.

In this system inputs are given in two ways, the Existing users can directly enter into the system using login form, and new users have to register all their details in the registration form provided.Input design is the very important part in the project and should be concentrated well as it is prone to error. The data that are to be inserted are to be inserted with care as this plays a very important role. In order to get the meaningful output and to achieve good accuracy the input should be acceptable and understandable by the user.Here; I take input as a five different images with one as a grayscale image and four as a colored image.

**OUTPUT DESIGN**

Output design plays a very important role in a system. Getting a correct output is a task that has to be concentrated, as a system is validated as a correct one only if it gives the correct output according to the input.

Here in this work in all the three days of inductions if the employee has completed all his/her input, then the output shows the status as completed or his status will be pending. Here, I got output as same five different images with one as a grayscale image and four as a colored image.

**SYSTEM TESTING**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client/server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

Testing issues

 Client GUI considerations

 Target environment and platform diversity considerations

 Distributed database considerations

 Distributed processing considerations

Types of Testing

1. Unit Testing

2. Integration Testing

3. Validation Testing

4. User acceptance Testing

5. System Testing

**Unit Testing**

All modules were tested and individually as soon as they were completed were checked for their correct functionality. Unit testing is carried out by verify and recover errors within the boundary of the smallest unit or a module. In this testing step, each module was found to be working satisfactory per the expected output of the module. In the package development, each module is tested separately after it has been completed and checked with valid data.

**Integration Testing**

The entire project was split into small programs; each of these single programs gives a frame as an output. These programs were tested individually; at last all these programs where combined together by creating another program where all these constructions were used. It gives a lot of problem by not functioning in an integrated manner.

The user interface testing is important since the user has to declare that the arrangements made in the frames are convenient and it is satisfied. When the frames are the test, the end user gave suggestion. Since they were much exposed to do the work manually.

**Validation Testing**

At the culmination of the black box testing software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of test i.e., validation succeeds when the software functions in a manner that can be reasonably accepted by the customer.

**User Acceptance Testing**

User acceptance testing of the system is the key factor the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of development and making change whenever required. This is done with regard to the input screen design and output screen design.

**System Testing**

This is to verify that all the system elements have been properly integrated and perform allocated functions. Testing is executing a program to test the logic changes made in it and with intention of finding errors. Tests are also conducted to find discrepancies between system and its original objective, current specification and documents.

**SYSTEM IMPLEMENTATION**

Implementation is the stage of the project when the theoretical design is turned into a working system. The implementation stage is a system project in its own right. It includes careful planning, investigation of current system and its constraints on implementation, design of methods to achieve the changeover, training of the staff in the changeover procedure and evaluation of the changeover method.

The first task in implementation is planning deciding on the methods and time scale to be adopted. Once the planning has been completed the major effort is to ensure that the programs in the system are working properly when the staff has been trained, the complete system involving both computer and user can be executed effectively. Thus the clear plans are prepared for the activities.

Successful implementation of the new system design is a critical phase in the system life cycle. Implementation means the process of converting a new or a revised system design into an operational one.

**SYSTEM MAINTAINS**

This software can be modified as need occurs. Maintenance includes all the activities after installation of the software that is performed to keep the system operational. The process of maintaining involves.

 Understanding the existing software

 Understand the effect of change

 Test for satisfaction

Maintenance can be done to this project by simply adding the new requirements that are the form of database the system can be modified. The maintenance process also helps to remove an error that resides in the system even after testing process.

**FUTURE ENHANCEMENT**

In the future enhancement every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements with extra features can be made to the application, so that the web site functions very attractive and useful manner than the present one.

**CONCLUSION**

It is concluded that the application works well and satisfy the users in creating the account in online. The application is tested very well with security issues and errors are properly debugged. The site is simultaneously accessed by more than one system in online. Simultaneous login from more than one place is tested.

The site works according to the restrictions provided in their respective browsers. Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one. The speed of the transactions become very high compared with normal.

The account holder can view that information only with the unique user id and password provided by the bank. After those process completed successfully a message will be displayed to the user about the transaction. If the account holder provides the wrong user ID or Password it will provide an error.

We mainly concentrate on the user satisfaction and reduce the significance .they can create their bank account through online with distributed verification of ensure code data. Using homographic token improves procedure in terms of finding out misbehaving servers, on data blocks including security for updating, deleting and modifying data.

**BIBLIOGRAPHY**

**Books**

1. Q. Wang, K. Ren, W. Lou, and Y. Zhang, “Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance,” Proc. of IEEE INFOCOM, 2009.

2. Amazon.com, “Amazon Web Services (AWS),” Second Edition.

3. D. Carman, P. Kruus, and B. Matt, “Constraints and Approaches for Distributed Sensor Network Security,” Technical Report 00-010, NAI Labs, 2000.

4. Rick Kiessig, “Ultra-Fast ASP.NET,” web sites using ASP.NET and SQL Server.

5. Anne Boehm Murachs “ASP.NET 3.5 Web Programming with VB 2008”, July 21, 2008.

6. Israel B. Ocbina “Mastering VB.NET and C#”, Seventh Edition, October 22, 2004.

7. MacDonald, Matthew, Szpuszta, Mario (2005) “Pro ASP.NET 2.0 in C# 2005”, First Edition.

8. Stephen Walther “ASP.NET 3.5 Unleashed”, December 28, 2007.

9. Stephen Walther “Data Access in the ASP.NET 2.0 Framework”, September 26, 2007.

10. Douglas O.Reilly “Designing Microsoft ASP.NET Applications” TataMcGrawHill Edition, 2002.

**REFERENCES**

**Websites**

1. http://msdn.microsoft.com

2. http://www.asp.net

3. http:// www.amazons.com