Chapter 1. Systems Introduction

* 1. Description of the project:

Hospital are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep the hospital running smoothly & successfully.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper.

Thus keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as “Hospital Management System”.

The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and also provides robust & reliable storage and backup facilities.

Objectives of the system:

The project “Hospital management system” is aimed to develop to maintain the day –to-day state of admission/discharge of patients, list of doctors, reports generation, and etc.It is designed to achieve the following objectives:

1. To computerize all details regarding patient details & hospital details.
2. Scheduling the appointment of patient with doctors to make it convenient for both.
3. Scheduling the services of specialized doctors and emergency properly so that facilities provided by hospital are fully utilized in effective and efficient manner.
4. If the medical store issues medicines to patients, it should reduce the stock status of the medical store and vice-versa.
5. It should be able to handle the test reports of patients conducted in the pathology lab of the hospital.
6. The inventory should be updated automatically whenever a transaction is made.
7. The information of the patients should be kept up to date and there record should be kept in the system for historical purposes.
   1. Methodologies for Data Collection
      1. Primary data collection

* Raw data (also known as primary data) is a term for data collected from a source. Raw data has not been subjected to processing or any other manipulation, and are also referred to as primary data.
* Primary data is a type of information that is obtained directly from first-hand sources by means of surveys, observation or experimentation. It is data that has not been previously published and is derived from a new or original research study and collected at the source such as in marketing.
* Primary data collection are observed and recorded directly from respondents. The information collected is directly related to the specific research problem identified. All the questions that one asks the respondents must be totally unbiased and formulated so that all the different respondents understand it.
  + 1. Secondary data collection.

Secondary data is data collected by someone other than the user. Common sources of secondary data for social science include censuses, organizational records and data collected through qualitative methodologies or qualitative research. Primary data, by contrast, are collected by the investigator conducting the research.

* <http://www.apollohospitals.com/>

Apollo is a private healthcare provider in [Asia](http://en.wikipedia.org/wiki/Asia) with hospitals in India, [Sri Lanka](http://en.wikipedia.org/wiki/Sri_Lanka), [Bangladesh](http://en.wikipedia.org/wiki/Bangladesh) and many more countries. It Provides world class services for knee, hip replacement, heart, etc.

* <http://www.maxhealthcare.in/>

Max Hospital is a multi-specialty hospital owned and managed by Max Healthcare Institute Limited (MHIL). The company provides patient services including [nuclear medicine](http://en.wikipedia.org/wiki/Nuclear_medicine) and [cardiac imaging](http://en.wikipedia.org/wiki/Cardiac_imaging),labs and[electrophysiology](http://en.wikipedia.org/wiki/Electrophysiology), [neurosciences](http://en.wikipedia.org/wiki/Neurosciences), [orthopedics](http://en.wikipedia.org/wiki/Orthopedics), ,generalsurgery,urology,nephrology, gastroenterology, mental health and behavioral sciences, rehabilitative services, and pulmonology

In this project all the information has been gathered from secondary sources that is internet.

* 1. Software Requirement Specification

A Software requirements specification (SRS), a requirements specification for a software system, is a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains non-functional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

1. Introduction

The following subsections of Software Requirement Specifications Document should facilitate in providing the entire overview of the Information system “Hospital Management System” under development. This document aims at defining the overall software requirements for admin. Efforts have been made to define the requirements of the Information system exhaustively and accurately.

1.1 Purpose

The main purpose of Software Requirement Specifications Document is to describe in a precise manner all the capabilities that will be provided by the Software Application “Hospital Management System”. It also states the various constraints which the system will be abide to. This document further leads to clear vision of the software requirements, specifications and capabilities. These are to be exposed to the development, testing team and end users of the software.

1.2 Scope

The proposed software product is the Hospital Management System (HMS). The system will be used in any Hospital, Clinic, Dispensary or Pathology labs in any Hospital, Clinic, Dispensary or Pathology labs to get the information from the patients and then storing that data for future  usage

The current system in use is a paper-based system. It is too slow and cannot provide updated lists of patients within a reasonable timeframe. The intentions of the system are to reduce over-time pay and increase the number of patients that can be treated accurately. Requirements statements in this document are both functional and non-functional.

* 1. Definitions,Acroynyms and Abbreviations

CFD: – Context Flow Diagram

DFD: – Data Flow Diagram

IDE: – Integrated Development Environment

SQL: – Structured Query Language

SRS: – Software Requirement Specification.

GUI: - Graphical User Interface

* 1. References

1. Software Engineering by K.K.Aggrawal, Singh, Yogesh.
2. Ian Somerville, Software Engineering, Third Edition.
3. Programming In Visual Basic 6.0 by Julia Case Bradley.
4. Introduction to Visual Basic 6.0 by Gary Haggard, Wade Hutschan.
5. Overall Description of the Proposed System.
   1. Product Perspective

The application will be windows-based, self contained and independent software product.

Visual Basic 6.0

MS-Access

* + 1. Interfaces

The application will have a user friendly and menu based interface. Following screens will be provided.

1. A Login Screen for entering username, password and role (Administrator, operator) will be provided. Access to different screens will be based upon the role of the user.
2. A Form for Search the details of a patient.
3. The Form for creating a new patient record will contain text fields where the Patient ID will be machine generated and the rest of the details will have to be filled up.
4. A Form for generating the tests reports.
5. The Form to produce a bill will create fields such as Patient ID, Appointment No., Doctor’s charges, Hospital charges etc. which will need to be filled up.

The following reports will be generated:

1. Tests reports
   * 1. Hardware Interfaces

Processor: Pentium IV AND motherboard

RAM: 512MB or above

Hard Disk: 40GB or above

Input Devices: Keyboard, Mouse

Output Devices: Monitor; -14” VGA

* + 1. Software Interfaces

OPERATING SYSTEM: Windows XP

FRONT END: Microsoft Visual Basic 6.0

BACK END: Microsoft Access

* + 1. Operations

This product will not cover any automated housekeeping aspects of database. The DBA at client site will be manually deleting old/ non required data. Database backup and recovery will also have to be handled by DBA.

* 1. Product Functions

The system will allow access only to authorized users with specific roles (Administrator, Operator). Depending upon the user’s role, he/she will be able to access only specific modules of the system.

A summary of the major functions that the software will perform:

1. A login facility for enabling only authorized access to the system.
2. When a patient is admitted, the front-desk staff checks to see if the patient is already registered with the hospital. If he is, his/her Name is entered into the computer.  Otherwise a new Patient ID is given to this patient.
3. If a patient checks out, the administrative staff shall delete his patient ID from the system.
4. The system generates reports on the following information: List of detailed information regarding the patient who has admitted in the hospital.

2.3 User Characteristics

1. Educational Level: At least graduate and should be comfortable with English language.

2. Technical Expertise: Should be a high or middle level employee of the organization comfortable with using general purpose applications on a computer.

* 1. System Features
* Login module

Description

This module records only user and password of the user.

* Patient module

Description

It keeps track of all details about both in-patient and out-patient. Patient id, patient name, address, admitted date, doctor name, and room no are entered in a form and stored for future reference. Also particular patient details can be viewed in the table using a separate form with an attribute patient id.

* Inpatient module

Description

Admission request will be made here. Request for admission is made before patient admitting the hospital.

* Outpatient module

Description

This module manages activities related to patient who visits the Hospital or Resident Doctor or Consultant Doctor for Medical Consultations, diagnosis and treatment.

* Pathology module

Description

This module Generates reports which will be done in pathology lab of the Hospital.

* Billing module

Description

This module bills the both inpatient and outpatient who comes to hospital.  It also includes Payment details of patients. Depending on the payments bill report is generated.

1. Logical Database Requirements

The proposed information system contains the following data tables in its database collection.

1. Patient Details table
2. Doctor Details Table
3. Room Details Table
4. Bill Details table
5. Software System Attributes

Reliability

This application is a reliable product that produces fast and verified output of all its processes.

Availability

This application will be available to use and help them to carry out their operations conveniently.

Security

The application will be password protected. User will have to enter correct username, password and role in order to access the application.

Maintainability

The application will be designed in a maintainable manner. It will be easy to to incorporate new requirements in the individual modules.

Portability

The application will be easily portable on any windows-based system that has oracle installed.

Chapter-2 System Design

2.1 Introduction

The purpose of Design phase is to plan a solution for problem specified by the requirements. System design aims to identify the modules that should be in the system, the specification of those modules and how the interact with each other to produce the results. The goal of the design process is to produce a model that can be used later to build that system. The produced model is called design of the system.

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. Normally, the design proceeds in two stages:

* Physical design
* Database design

Physical Design

The physical design is a graphical representation of a system showing the system’s internal and external entities and the flow of data into and out of these entities. An internal entity is an entity within the system that transforms data.

To represent the physical design of the system, we use diagrams like data flow diagrams, use case diagrams, etc.

2.2 Data Flow Diagrams

The Data Flow Diagram (DFD) is a graphical representation of the flow of data through an information system. Data flow diagrams are used by systems analysts to design information-processing systems but also as a way to model whole organizations. The main merit of DFD is that it can provide an overview of what data a system would processes, what transformations of data are done, what data are stored and which stored data is used, and where the result is flow.

Standard Symbols used in DFD:

|  |  |  |
| --- | --- | --- |
| Symbol | Name | Function |
|  | Data Flow | Used to connect processes to each other. The arrowhead indicates direction of data flow. |
|  | Process | Performs some transformation to input data to output data. |
|  | Source or sink.(external entity) | A source of system inputs or sink of system outputs. |
| Data Store | A repository of data. Arrowheads indicate net inputs or net outputs to the store. |

Table no.2.1

Level 0 DFD

A context diagram is a top level (also known as Level 0) data flow diagram. It only contains one process node (process 0) that generalizes the function of the entire system in relationship to external entities. In level 0 dfd, system is shown as one process.

The Level 0 DFD shows how the system is divided into 'sub-systems' (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

Personal details IPD &OPD

Patient

Admin

Patient

Admin

Bill, Report Detail report, bills generate

Fig. No.2.1

Level 1 DFd

Admin

Personal details of patient. Update Patient db

Update

Admin

Personal Details of doctor Update Doctor db

Patient details

Update Reports DB

R\_charges

Doctor name & fees

Patient details

Fig. No.2.2

2.3 E-R Diagram

Entity-Relationship Diagram is a graphical representation of entities and their relationship to each others. It describes how data is related to each other. An entity is a piece of data- an object or a concept about which data is stored. A relationship is how the data is shared between entities. In E-R Diagram, there are 3 main Components:

|  |  |  |
| --- | --- | --- |
| Symbol | Name | Description |
|  | Entity | An entity can be any object, place, person or anything. |
|  | Attribute | An Attribute Describes a property or characteristics of an entity. |
|  | Relationship | A Relationship Describes relation between entities. |

Table No.2.2

E-R Diagram of Hospital Management System

Room

Bill

Issued

Assign

Patient

Treats

Doctor

1 M

1

1

M

1

Fig. No.2.3

2.4 Data Dictionary

1. Doctor Details

|  |  |  |  |
| --- | --- | --- | --- |
| Name-Doctor Details | | | |
| Name | Type | Size | Description |
| ID | Integer |  | ID of the Doctor |
| Name | Varchar | 50 | Name of the Doctor |
| Address | Varchar | 150 | Address of the Doctor |
| Phone Number | Varchar | 90 | Contact number of the doctor |
| Qualification | Varchar | 100 | Qualification of the Doctor |
| Gender | Varchar | 30 | Gender of the doctor |

Table No.2.3

1. Room Details

|  |  |  |  |
| --- | --- | --- | --- |
| Name-Room Details | | | |
| Name | Type | Size | Description |
| Room\_no | Integer |  | Id of the Room |
| Room Type | Varchar | 50 | General or Private Room |

Table No.2.4

1. patient Details

|  |  |  |  |
| --- | --- | --- | --- |
| Name- patient Details | | | |
| Name | Type | Size | Description |
| Patient\_no | Integer | 20 | ID of the Patient |
| Name | Varchar | 60 | Name of the patient |
| Age | Integer | 20 | Age of the Patient |
| Gender | Varchar | 30 | Gender of the Patient |
| Address | Varchar | 90 | Address of the Patient |
| Date | Datetime | 30 | Date of admission |
| Contact Number | Varchar | 90 | Contact number of the patient |
| Room No | Varchar | 50 | Admitted patient room no.(in case of IP) |

Table No.2.3

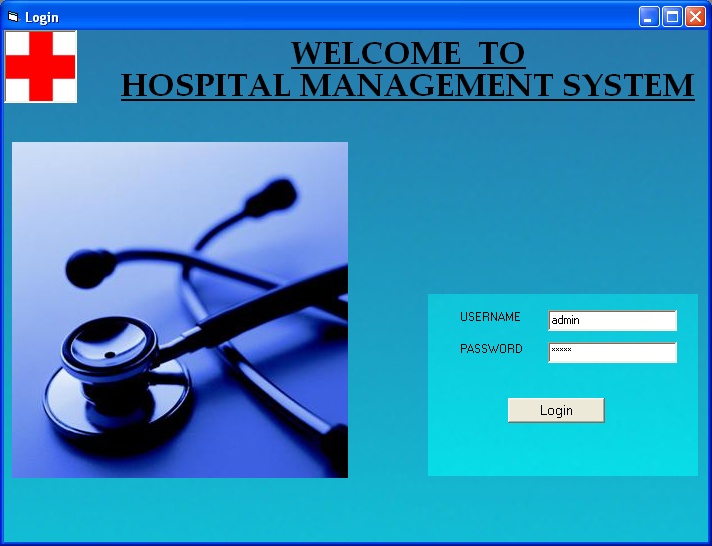
1. Bill Details

|  |  |  |  |
| --- | --- | --- | --- |
| Name-Bill Details | | | |
| Name | Type | Size | Description |
| Bill\_no | Integer | 20 | Number of the Bill |
| Date | Datetime | 20 | Date at which bill is generated |
| Patient Id | Varchar | 50 | Id of the patient |
| Name | Varchar | 50 | Name of the Patient |
| Age | Varchar | 50 | Age of the patient |
| Gender | Varchar | 50 | Gender of the Patient |
| Date of Admission | Varchar | 50 | Date on which patient is admitted into Hospital |
| Date of Discharge | Varchar | 50 | Date on which patient is Discharged from the Hospital |
| Room Charges | Varchar | 50 | Charges of the room |
| Pathology fees | Varchar | 50 | Laboratory report Charges |
| Doctor Fees | Varchar | 50 | Doctor Checkup Fees |
| Miscellaneous | Varchar | 50 | Other Charges |
| Total Amount | Varchar | 100 | Total amount of the bill |

Chapter-3 Systems Development & Implementation

3.1 Screenshot and Source Code

1. Login form



Private Sub Command1\_Click()

With DataEnvironment1.rscmd\_login 'opening the tbl\_login

.Open

.MoveFirst

While Not .EOF 'scan all entry under tbl\_login

If Text1 = !UserName And Text2 = !Password Then

a = MsgBox("Username and Password Accepted, Welcome!", vbInformation) 'login accepted

Me.Hide

Form3.Show

Exit Sub

Else

.MoveNext 'move to another row

End If

Wend

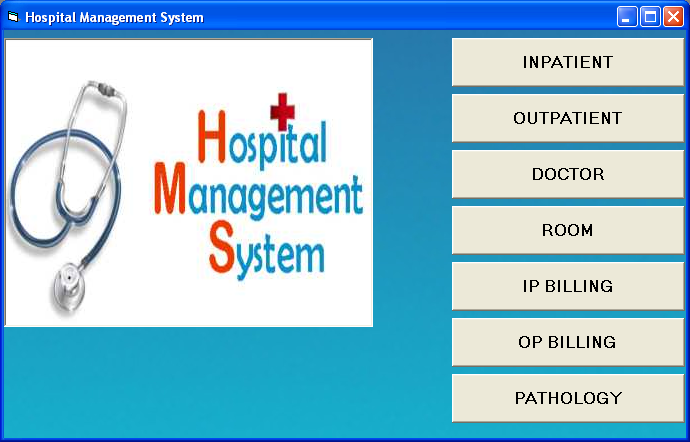
.Close 'close tbl\_login

a = MsgBox("You have entered an invalid username/password", vbCritical) 'login detaits not matched

End With

End Sub

1. Homepage of Hospital Management System



Private Sub Command1\_Click()

Me.Hide

Form2.Show

End Sub

Private Sub Command2\_Click()

Me.Hide

Form8.Show

End Sub

Private Sub Command3\_Click()

Me.Hide

Form6.Show

End Sub

Private Sub Command4\_Click()

Me.Hide

Form10.Show

End Sub

Private Sub Command5\_Click()

Me.Hide

form5.Show

End Sub

Private Sub Command6\_Click()

Me.Hide

Form7.Show

End Sub

Private Sub Command7\_Click()

Me.Hide

Form4.Show

End Sub

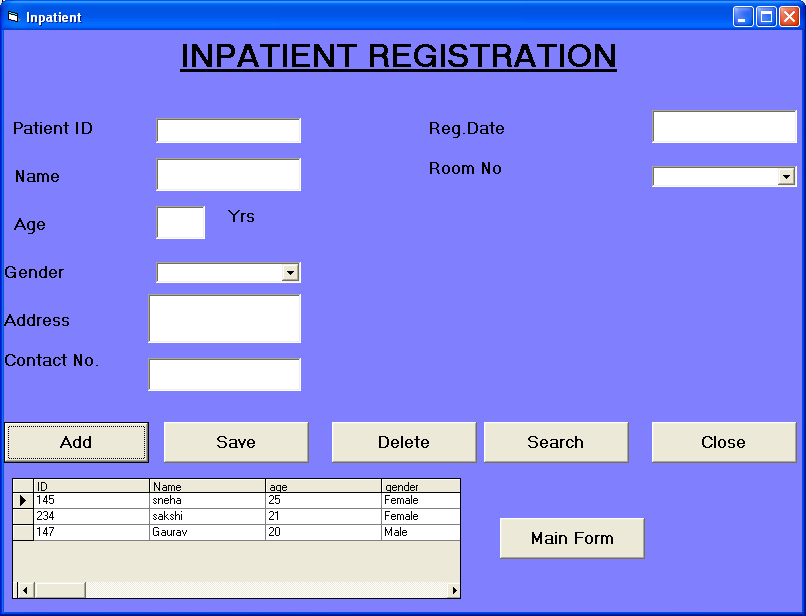
Private Sub Command8\_Click()

Me.Hide

Form9.Show

End Sub

1. Inpatient Registration Form



Public con As New ADODB.Connection

Public rs As New ADODB.Recordset

Sub main()

opendb

Form1.Show

End Sub

Sub opendb()

Set con = Nothing

Set con = New ADODB.Connection

con.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=patient.mdb;Persist Security Info=False"

con.CursorLocation = adUseClient

If con.State = 1 Then

MsgBox "succesfully connected"

Else

MsgBox "not connected"

End If

End Sub

Private Sub Form\_Load()

Set rs = Nothing

Set rs = New ADODB.Recordset

rs.Open "select \* from inpatient", con, adOpenDynamicmic, adLockOptimistic

Set DataGrid1.DataSource = rs

End Sub

Add:-

Private Sub Command6\_Click()

Text1.Text = Clear

Text2.Text = Clear

Text3.Text = Clear

Text4.Text = Clear

Text5.Text = Clear

Text6.Text = Clear

Combo1.Text = Clear

End Sub

Save:-

Private Sub Command1\_Click()

rs.AddNew

rs!ID = Text1.Text

rs!Name = Text2.Text

rs!age = Text3.Text

rs!gender = Combo1.Text

rs!address = Text6.Text

rs!contact\_no = Text5.Text

rs!Date = Text4.Text

rs!roomno = Combo2.Text

rs.Update

MsgBox " Record Added Successfully! ", vbOKOnly + vbInformation, "info"

End Sub

Delete:-

Private Sub Command4\_Click()

rs.Delete

MsgBox " Record Deleted Successfully! ", vbInformation, "Delete"

rs.MoveNext

End Sub

Close:-

Private Sub Command2\_Click()

If MsgBox("Close the form ?", vbQuestion + vbYesNo, "Confirm") = vbYes Then

End

End If

End Sub

Search:-

Private Sub Command5\_Click()

Me.Hide

form5.Show

End Sub

Main form:-

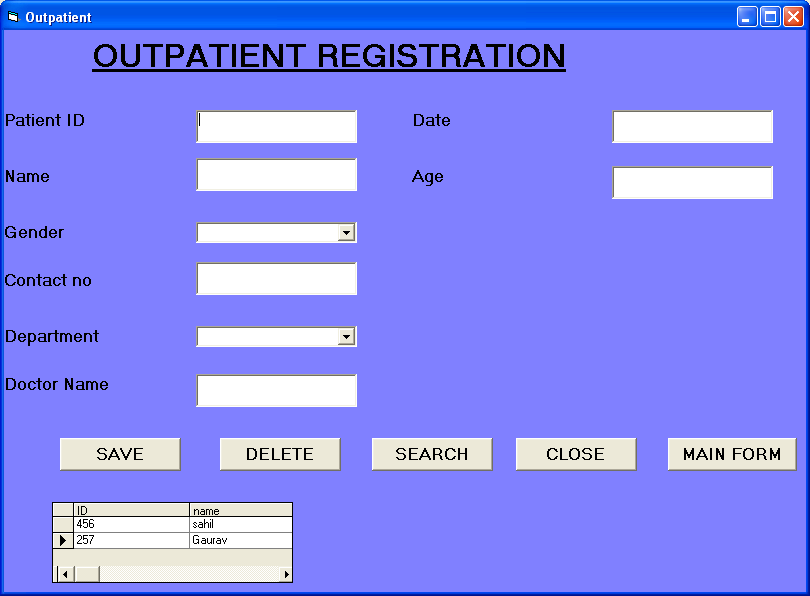
Private Sub Command3\_Click()

Me.Hide

Form3.Show

End Sub

1. Outpatient Registration Form



Save:-

Private Sub Command1\_Click()

rs.AddNew

rs!ID = Text1.Text

rs!Name = Text2.Text

rs!age = Text6.Text

rs!gender = Combo1.Text

rs!contact\_no = Text3.Text

rs!dept = Combo2.Text

rs!doctorname = Text5.Text

rs!Date = Text4.Text

rs.Update

MsgBox " Record Added Successfully! ", vbOKOnly + vbInformation, "info"

Text1.Text = ""

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text5.Text = ""

Text6.Text = ""

Combo1.Text = ""

Combo2.Text = ""

End Sub

Delete:-

Private Sub Command2\_Click()

On Error Resume Next

If MsgBox("Data Deleted Successfully!", vbExclamation + vbOKCancel, "Confirm Delete") = vbOK Then

rs.Delete

End If

End Sub

Search:-

Private Sub Command5\_Click()

Me.Hide

Form11.Show

End Sub

Close:-

Private Sub Command3\_Click()

If MsgBox("Close the form ?", vbQuestion + vbYesNo, "Confirm") = vbYes Then

End

End If

End Sub

Main Form:-

Private Sub Command4\_Click()

Me.Hide

Form3.Show

End Sub

Private Sub Form\_Load()

Set rs = Nothing

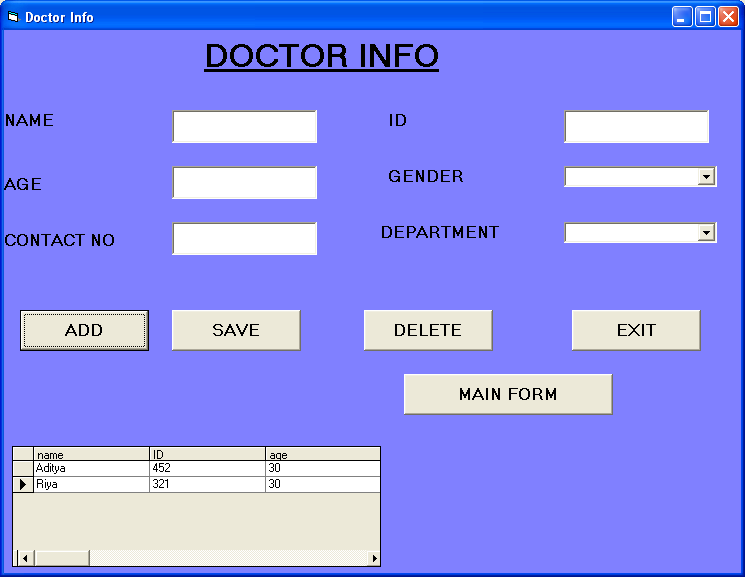
Set rs = New ADODB.Recordset

rs.Open "select \* from outpatient", con, adOpenDynamicmic, adLockOptimistic

Set DataGrid1.DataSource = rs

End Sub

1. Doctor Information form



Private Sub Form\_Load()

Set rs = Nothing

Set rs = New ADODB.Recordset

rs.Open "select \* from doctor", con, adOpenDynamicmic, adLockOptimistic

Set DataGrid1.DataSource = rs

End Sub

ADD:-

Private Sub Command1\_Click()

Text1.Text = Clear

Text2.Text = Clear

Text3.Text = Clear

Text5.Text = Clear

Combo1.Text = Clear

Combo3.Text = Clear

End Sub

Save:-

Private Sub Command2\_Click()

rs.AddNew

rs!Name = Text1.Text

rs!ID = Text2.Text

rs!age = Text3.Text

rs!gender = Combo3.Text

rs!contactno = Text5.Text

rs!department = Combo1.Text

rs.Update

MsgBox " Record Added Successfully! ", vbOKOnly + vbInformation, "info"

End Sub

Delete:-

Private Sub Command5\_Click()

rs.Delete

MsgBox " Record Deleted Successfully! ", vbInformation, "Delete"

rs.MoveNext

End Sub

Exit:-

Private Sub Command3\_Click()

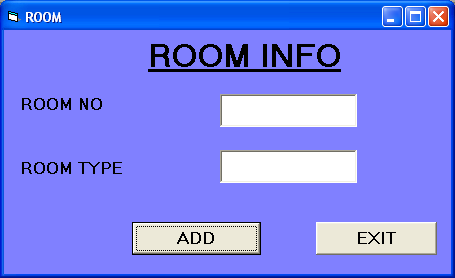
If MsgBox("Close the form ?", vbQuestion + vbYesNo, "Confirm") = vbYes Then

End

End If

End Sub

1. Room Information Form



Private Sub Form\_Load()

Set rs = Nothing

Set rs = New ADODB.Recordset

rs.Open "select \* from room", con, adOpenDynamicmic, adLockOptimistic

End Sub

Add:-

Private Sub Command1\_Click()

rs.AddNew

rs!no = Text1.Text

rs!Type = Text2.Text

rs.Update

MsgBox " Record Added Successfully! ", vbOKOnly + vbInformation, "info"

Text1.Text = ""

Text2.Text = ""

End Sub

Exit:-

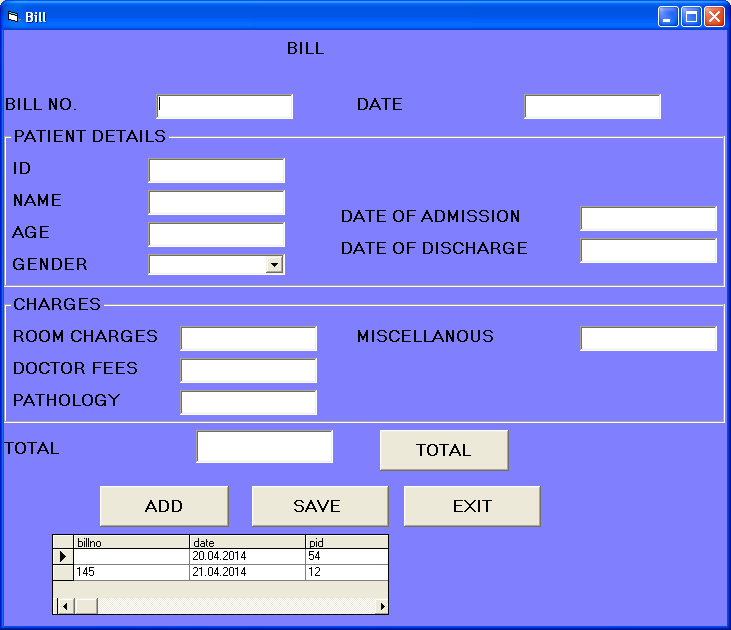
Private Sub Command2\_Click()

Me.Hide

Form3.Show

End Sub

1. Inpatient Billing Form



Add:-

Private Sub Command1\_Click()

Text1.Text = Clear

Text2.Text = Clear

Text3.Text = Clear

Text4.Text = Clear

Text5.Text = Clear

Text6.Text = Clear

Combo1.Text = Clear

Text7.Text = Clear

Text8.Text = Clear

Text9.Text = Clear

Text10.Text = Clear

Text11.Text = Clear

Text12.Text = Clear

End Sub

Save:-

Private Sub Command2\_Click()

rs.AddNew

rs!billno = Text1.Text

rs!Date = Text2.Text

rs!pid = Text12.Text

rs!Name = Text3.Text

rs!age = Text4.Text

rs!gender = Combo1.Text

rs!date\_of\_adm = Text5.Text

rs!date\_of\_discharge = Text6.Text

rs!roomcharges = Text7.Text

rs!doctorfees = Text8.Text

rs!pathology = Text9.Text

rs!miscellanous = Text10.Text

rs!total = Text11.Text

rs.Update

MsgBox " Record Saved Successfully! ", vbOKOnly + vbInformation, "info"

End Sub

Search:-

Private Sub Command3\_Click()

Me.Hide

Form3.Show

End Sub

Total:-

Private Sub Command5\_Click()

Text11.Text = Val(Text7.Text) + Val(Text8.Text) + Val(Text9.Text) + Val(Text10.Text)

End Sub

Private Sub Form\_Load()

Set rs = Nothing

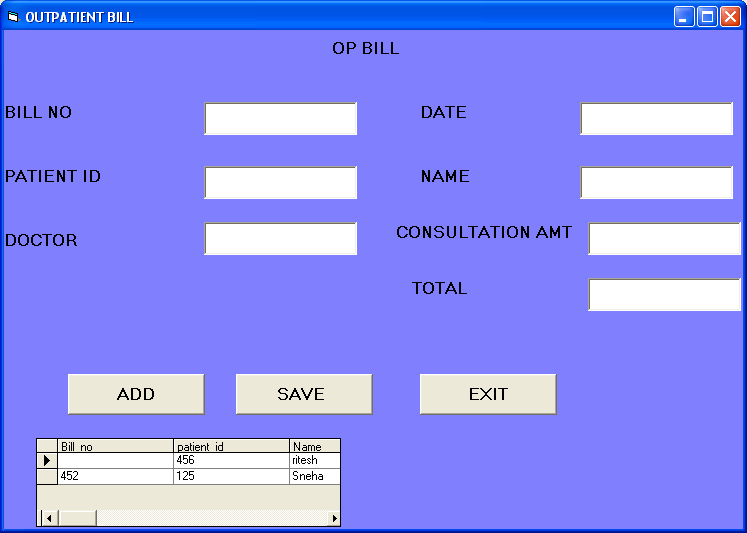
Set rs = New ADODB.Recordset

rs.Open "select \* from ipbill", con, adOpenDynamicmic, adLockOptimistic

Set DataGrid1.DataSource = rs

End Sub

1. Outpatient Billing Form



Add:-

Private Sub Command1\_Click()

Text1.Text = Clear

Text2.Text = Clear

Text3.Text = Clear

Text4.Text = Clear

Text5.Text = Clear

Text6.Text = Clear

Text7.Text = Clear

End Sub

Save:-

Private Sub Command2\_Click()

rs.AddNew

rs!Bill\_no = Text1.Text

rs!Date = Text2.Text

rs!patient\_id = Text3.Text

rs!Name = Text4.Text

rs!Doctor\_name = Text5.Text

rs!amount = Text6.Text

rs!total = Text7.Text

rs.Update

MsgBox " Record Saved Successfully! ", vbOKOnly + vbInformation, "info"

End Sub

Exit:-

Private Sub Command3\_Click()

Me.Hide

Form3.Show

End Sub

Private Sub Form\_Load()

Set rs = Nothing

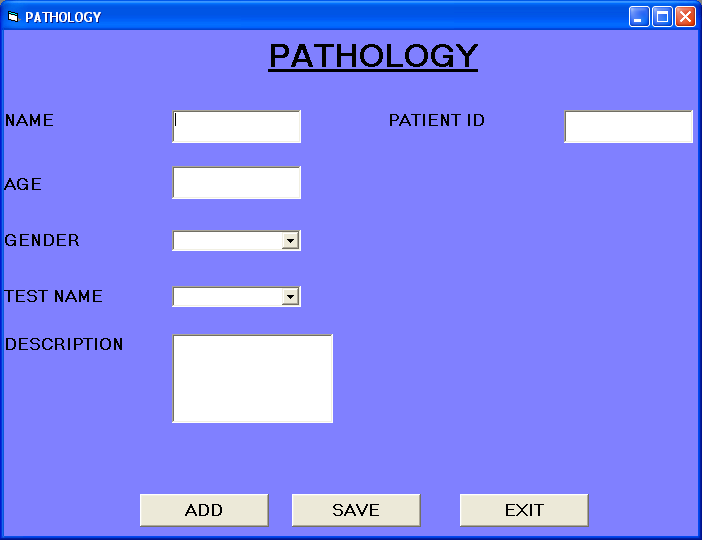
Set rs = New ADODB.Recordset

rs.Open "select \* from opbill", con, adOpenDynamicmic, adLockOptimistic

Set DataGrid1.DataSource = rs

End Sub

1. Pathology Form



Add:-

Private Sub Command1\_Click()

Text1.Text = Clear

Text2.Text = Clear

Text3.Text = Clear

Text4.Text = Clear

Combo1.Text = Clear

Combo2.Text = Clear

End Sub

Save:-

Private Sub Command2\_Click()

rs.AddNew

rs!Name = Text1.Text

rs!pid = Text2.Text

rs!age = Text3.Text

rs!gender = Combo1.Text

rs!testname = Combo2.Text

rs!Description = Text4.Text

rs.Update

MsgBox " Record saved Successfully! ", vbOKOnly + vbInformation, "info"

End Sub

Exit:-

Private Sub Command3\_Click()

Me.Hide

Form3.Show

End Sub

Private Sub Form\_Load()

Set rs = Nothing

Set rs = New ADODB.Recordset

rs.Open "select \* from pathology", con, adOpenDynamicmic, adLockOptimistic

End Sub

1. Inpatient Search Form



Private Sub Command1\_Click()

rs.Filter = adFilterNone

rs.Requery

End Sub

Exit:-

Private Sub Command2\_Click()

Me.Hide

Form3.Show

End Sub

Private Sub Form\_Load()

Set rs = Nothing

Set rs = New ADODB.Recordset

rs.Open "select \* from inpatient", con, adOpenDynamicmic, adLockOptimistic

Set DataGrid1.DataSource = rs

End Sub

Private Sub Text1\_Change()

If Text1.Text = "" Then

Call Form\_Load

Me.Show

Else

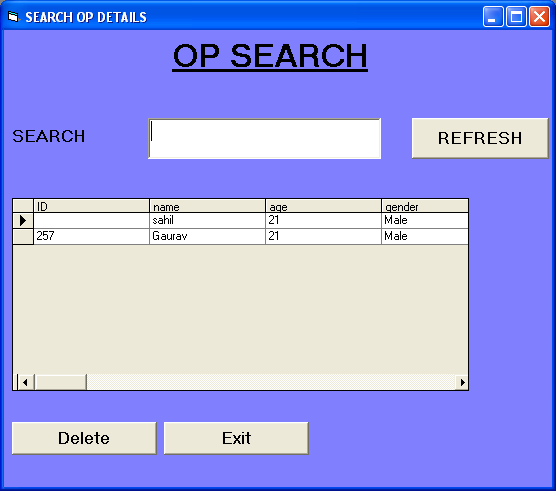
rs.Filter = "Name LIKE '" & Me.Text1.Text & "\*'"

Set DataGrid1.DataSource = rs

End If

End Sub

1. Outpatient Search Form



Private Sub Command1\_Click()

rs.Filter = adFilterNone

rs.Requery

End Sub

Private Sub Command2\_Click()

Me.Hide

Form3.Show

End Sub

Private Sub Form\_Load()

Set rs = Nothing

Set rs = New ADODB.Recordset

rs.Open "select \* from outpatient", con, adOpenDynamicmic, adLockOptimistic

Set DataGrid1.DataSource = rs

End Sub

Private Sub Text1\_Change()

If Text1.Text = "" Then

Call Form\_Load

Me.Show

Else

rs.Filter = "Name LIKE '" & Me.Text1.Text & "\*'"

Set DataGrid1.DataSource = rs

End If

End Sub

3.2 SYSTEM TESTING

1. Login Form Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test | Expected  Input | Expected  Output | Actual Input | Actual Output |
| 1. | Login Form | Username, Password | Homepage | admin,  admin | Homepage |

2. Doctor Information Form Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test | Expected  Input | Expected  Output | Actual Input | Actual Output |
| 2. | Doctor Form | Details of doctor | Msgbox appears(“data is saved”) | Detail of Doctor | Msgbox “Record Added Successfully!” |

3. Room Information Form Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test | Expected  Input | Expected  Output | Actual Input | Actual Output |
| 2. | Room Form | Room no and  Room type | Msgbox appears(“data is saved”) | Room no and  Room type | Msgbox “Record Added Successfully!” |

Chapter-4 Scope of Improvement, Summary and Conclusion

CONCLUSION

The project Hospital Management System (HMS) is for computerizing the working in a hospital. It is a great improvement over the manual system. The computerization of the system has speed up the process. In the current system, the front office managing is very slow. The hospital managing system was thoroughly checked and tested with dummy data and thus is found to be very reliable. The software takes care of all the requirements of an average hospital and is capable to provide easy and effective storage of information related to patients that come up to the hospital.

It generates test reports and also provides the facility for searching the details of the patient. It also provides billing facility on the basis of patient’s status whether it is an indoor or outdoor patient. The system also provides the facility of backup as per the requirement.

FUTURE ENHANCEMENTS

The proposed system is Hospital Management System. We can enhance this system by including more facilities like pharmacy system for the stock details of medicines in the pharmacy.

Providing such features enable the users to include more comments into the system.

LIMITATIONS:

* The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
* Training for simple computer operations is necessary for the users working on the system.

BIBLIOGRAPHY

1. Julia Case Bradley, Programming In Visual Basic 6.0,First Edition, Tata McGraw Hill Edition.
2. Michael Halvorson, Microsoft Visual Basic 6.0 Professional, Second Edition, Microsoft Press.
3. Bayross, Programming in Visual Basic 6, 2008 Edition, B.p.b Publications.
4. Gary Haggard, Wade Hutschan, Introduction to Visual Basic 6.0, Tata McGraw Hill Edition.
5. Mike Smart, Learn Access 2003 VBA with the Smart Method, The Smart Method Ltd.

Web References

1. [www.google.com](http://www.google.com)

2. [www.slideshare.com](http://www.slideshare.com)