VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

A Report on

Medical Management System

Submitted in partial fulfilment of the requirements as a part of the DBMS Laboratory for the award of degree of

Bachelor of Engineering
in
Information Science and Engineering

Submitted by

Nishaanth Govardhan
1RN16IS058

Aishwarya Arvind
1RN16IS007

Under the Guidance of:

Faculty Incharge
Mrs. Harshitha K Raj
Asst. Professor
Dept. of ISE, RNSIT

Lab Incharge
Mr. R Rajkumar
Asst. Professor
Dept. of ISE, RNSIT

Department of Information Science and Engineering
RNS Institute of Technology
Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

2018 – 2019

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the internship report entitled ***MEDICAL MANAGEMENT SYSTEM*** has been successfully completed by **Nishaanth Govardhan** bearing USN **1RN16IS058** and **Aishwarya Arvind** bearing USN **1RN16IS007** presently V semester student of **RNS Institute of Technology** in partial fulfilment of the requirements as a part of the DBMS Laboratory for the award of the degree ***Bachelor of Engineering in Information Science and Engineering*** under **Visvesvaraya Technological University, Belagavi** during academic year 2018 – 2019. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements as a part of DBMS Laboratory for the said degree.

Mrs.Harshitha K Raj	**Mr. R Rajkumar**	**Dr. M V Sudhamani**
Faculty Incharge	Lab Incharge	Professor and HOD
Assistant Professor	Assistant Professor	

External Viva

Name of the Examiners	**Signature with date**
1. _____	_____
2. _____	_____

ABSTRACT

The medical record system is a database management system that uses database technology to construct, maintain and manipulate various kinds of data about a person's medical history and care across time.

The DBMS can track and update all the information of registered patients in the medical Centre during a particular time span.

Records may include the patients: Patient's personal history,doctor's information,medications prescribed, diagnosis and treatment details and billing details.

The medical record is a key instrument used in planning, evaluating, and coordinating patient care in both the inpatient and the outpatient settings.

We seek to expand the project by having a fully equipped inventory, complete with all possible information that a patient requires. In the end we seek to completely mimic the hospital websites present with a slight focus on the various patients and the requirements. Any staff using the website will be able to insert all the details, update and and delete them when needed.The UI has been made very simple to provide ease of access for all types of users.

The mini project has been implemented using C# to and SQLite.

ACKNOWLEDGMENT

The fulfilment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavours with success.

I would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment to carry out this Project work.

I would like to thank our beloved Director **Dr. H N Shivashankar** for his confidence feeling words and support for providing facilities throughout the course.

I would like to express my thanks to our Principal **Dr. M K Venkatesha** for his support and inspired me towards the attainment of knowledge.

I wish to place on record my words of gratitude to **Dr. M V Sudhamani,** Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind my Project work.

I would like to express my profound and cordial gratitude to my Faculty incharge **Mr. R Rajkumar**, Assistant Professor, Department of Information Science and Engineering for their valuable guidance, constructive comments throughout the Project work.

I would like to express my profound and cordial gratitude to my guide **Mrs Harshitha K Raj** , Assistant Professor, Department of Information Science and Engineering for their valuable guidance in preparing Project report.

I would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped me to carry out the project work.

And lastly, I would hereby acknowledge and thank my parents who have been a source of inspiration and also instrumental in carrying out this Project work.

AISHWARYA ARVIND **NISHAANTH GOVARDHAN**

1RN16IS007 **1RN16IS058**

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

BOOTP	-	Bootstrap Protocol
BGP	-	Border Gateway Protocol
CMC	-	C Model Checker
DNS	-	Domain Name Service
DHCP	-	Dynamic Host Control Protocol
DART	-	Directed Automated Random Testing
D3S	-	Debugging Deployed Distributed Systems
DNSSD	-	DNS Service Discovery
D-ITG	-	Distributed Internet Traffic Generator
DNV	-	Declarative Network Verifier
IETF	-	Internet Engineering Task Force
IOT	-	Interoperability Testing
LLVM	-	Low Level Virtual Machine
MPE-SE	-	Multiple Packet Exchange – Symbolic Execution
PPP	-	Pont-to-Point Protocol

Chapter 1

INTRODUCTION

1.1 Background

Databases and database technology have a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic commerce, engineering, medicine, genetics, law, education, and library science. The word database is so commonly used that we must begin by defining what a database is. Our initial definition is quite general. A database is a collection of related data.1 By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using a personal computer and software such as Microsoft Access or Excel. This collection of related data with an implicit meaning is a database. The preceding definition of database is quite general; for example, we may consider the collection of words that make up this page of text to be related data and hence to constitute a database. However, the common use of the term database is usually more restricted.

A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse (UoD). Changes to the mini world are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases

among various users and applications. Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world, and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

HISTORY OF DBMS

In 1959, the TX-2 computer was developed at MIT's Lincoln Laboratory. The TX-2 integrated a number of new man-machine interfaces. A light pen could be used to draw sketches on the computer using Ivan Sutherland's revolutionary Sketchpad software. [4] Using a light pen, Sketchpad allowed one to draw simple shapes on the computer screen, save them and even recall them later. The light pen itself had a small photoelectric cell in its tip. This cell emitted an electronic pulse whenever it was placed in front of a computer screen and the screen's electron gun fired directly at it. By simply timing the electronic pulse with the current location of the electron gun, it was easy to pinpoint exactly where the pen was on the screen at any given moment. Once that was determined, the computer could then draw a cursor at that location. Also in 1961 another student at MIT, Steve Russell, created the first video game, E. E. Zajac, a scientist at Bell Telephone Laboratory (BTL), created a film called "Simulation of a two-giro gravity attitude control system" in 1963.

During 1970s, the first major advance in 3D computer graphics was created at UU by these early pioneers, the hidden-surface algorithm. In order to draw a representation of a 3D object on the screen, the computer must determine which surfaces are "behind" the object from the viewer's perspective, and thus should be "hidden" when the computer creates (or renders) the image. In the 1980s, artists and graphic designers began to see the personal computer, particularly the Commodore Amiga and Macintosh, as a serious design tool, one that could save time and draw more accurately than other methods.

In the late 1980s, SGI computers were used to create some of the first fully computer-generated short films at Pixar. The Macintosh remains a highly popular tool for computer graphics among graphic design studios and businesses. Modern computers, dating from the 1980s often use graphical user interfaces (GUI) to present data and information with symbols, icons and pictures, rather than text. Graphics are one of the five key elements of multimedia technology.

3D graphics became more popular in the 1990s in gaming, multimedia and animation. In 1996, Quake, one of the first fully 3D games, was released. In 1995, Toy Story, the first full-length computer-generated animation film, was released in cinemas worldwide. Since then, computer graphics have only become more detailed and realistic, due to more powerful graphics hardware and 3D modelling software.

1.2 INTRODUCTION ABOUT THE PROJECT

The purpose of the project entitled as "MEDICAL MANAGEMENT SYSTEM" is to computerise the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost – effective. It deals with the collection of patient's information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details, while system output is to get these details on to the screen. The Hospital Management System can be entered using a username and password. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast. The project Medical Management system includes registration of patients, storing their details into the system, and also computerised billing in the pharmacy, and labs. The software has the facility to give a unique id for every patient and stores the details of every patient. Since we are entering details of the patients electronically in the" Hospital Management System", data will be secured. Using this application we can retrieve patient's history with a single click. Thus processing information will be faster. It guarantees accurate maintenance of Patient details. It easily reduces the book keep- ing task and thus reduces the human effort and increases accuracy speed. Just like any other developer this project is the most basic website built using simple tools. We seek to increase the dynamic of the project by adding various other innovations to it. Such innovations would seem possible only with time, which we lack but regardless we strive to complete what we started.

Chapter 2

Schema Diagram

A database schema is a skeleton structure that represents the logical view of the entire database. It defines how the data is organised and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive by means of schema diagrams. Its the database designers who design the schema to help programmers understand the database and make it useful

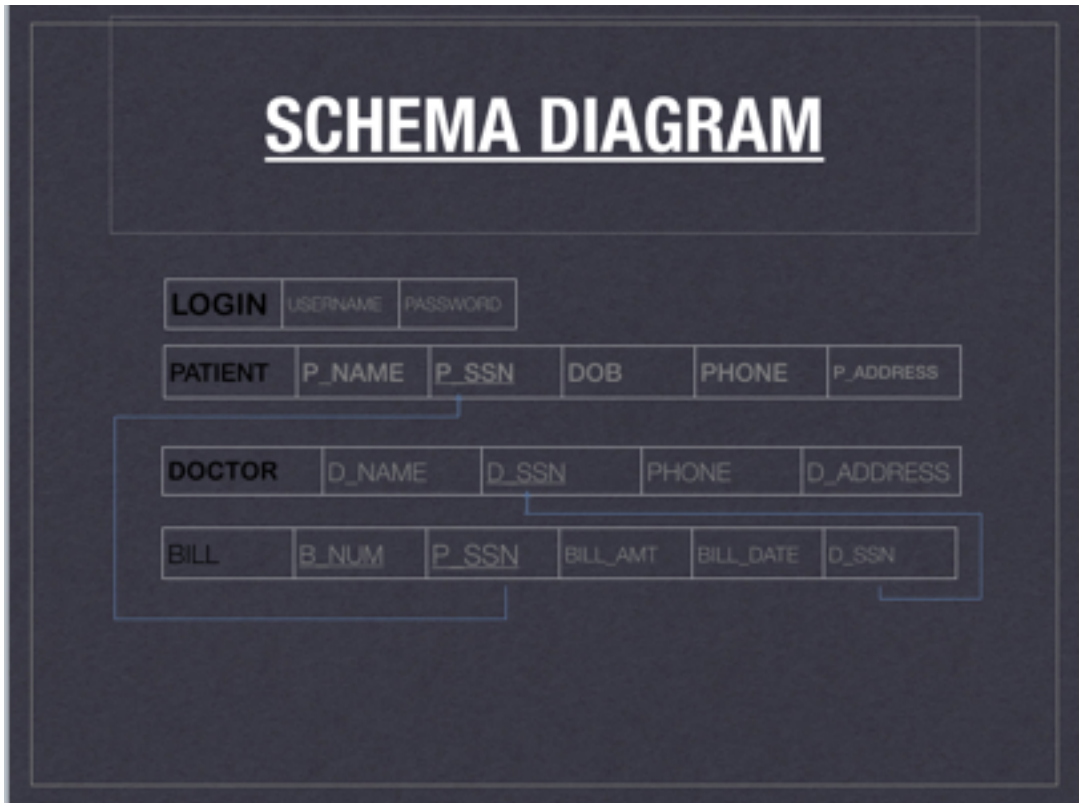

Figure. 2.1: Relational schema of medical record system

2.1 Entity Relationship Diagram

An Entity Relationship Diagram describes the structure of a database with the help of a diagram, which is known as ER Diagram. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of ER diagram are entity set and relationship set.

An ER Diagram shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.

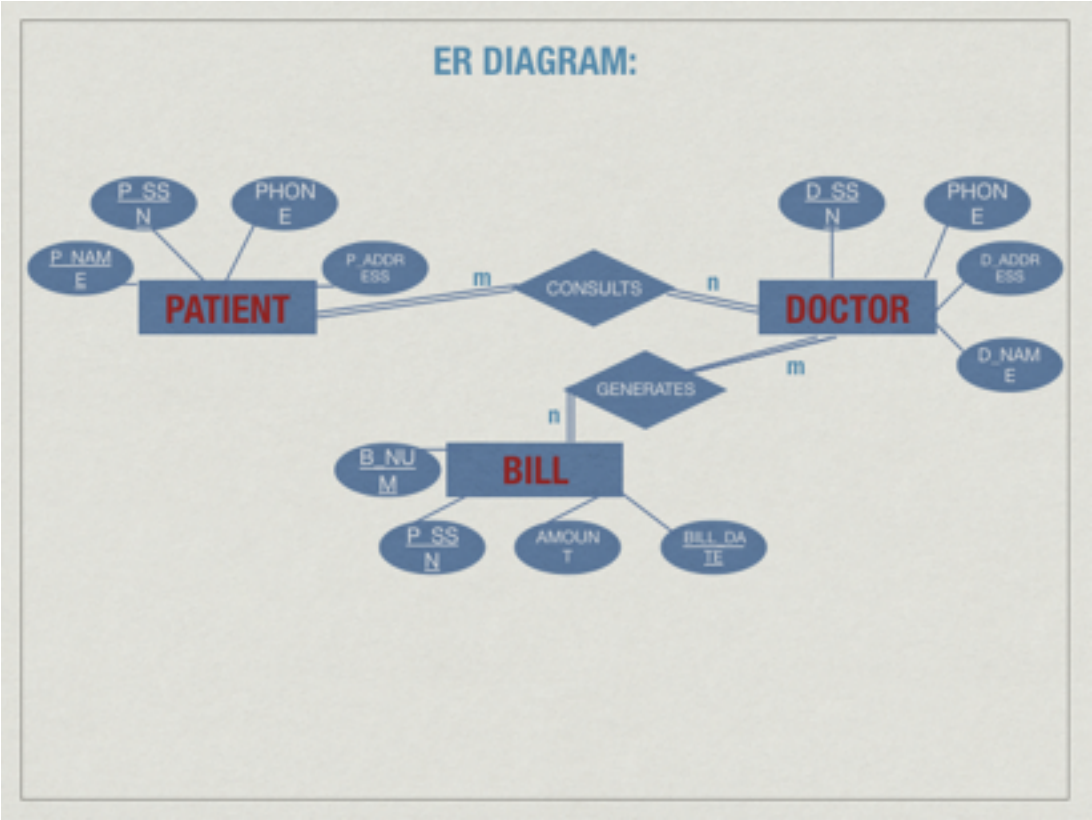

Figure 2.2 ER Diagram for medical record system.

CHAPTER 3

SYSTEM DESIGN

The system design is used to indicate a brief overview of our database management system. The four major tables used and its respective attributes are shown in the following tables. The entire project is designed based on the tables as shown below: The four tables are: Doctor, Patient, Bill and System

DOCTOR: Account in which the details of all patients be stored.Details of the doctors in the hospital.

Figure 3.1: Doctor information

Field	Type	Null	Key	Default
d_name	varchar(15)	No		Null
d_ssn	varchar(15)	No	Primary	Null
phone	varchar(15)	No		Null
d_address	varchar(20)	No		Null

PATIENT : A detailed description of patient details along with few functionalities.

Figure 3.2: Patient information

Field	Type	Null	Key	Default
p_ssn	varchar(15)	No	Primary	Null
p_name	varchar(15)	No		Null
p_address	varchar(20)	No		Null
DOB	Date	No		Null
phone	varchar(20)	No		Null

BILL: A bill for a visitor is done in this table.

Figure 3.3: Bill information

Field	Type	Null	Key	Default
b_no	varchar(20)	No	Primary	Null
bill_amount	int	No		Null
p_ssn	varchar(15)	No	Foreign	Null
bill_date	Date	No		Null

SYSTEM: This table contains login information of user

Figure 3.4: System table

Field	Type	Null	Key	Default
username	varchar(15)	No	Primary	Null
password	varchar(15)	No		Null

Chapter 4

IMPLEMENTATION

4.1 Software Used

Front End Used: Microsoft Visual Studio

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 Form applications, websites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Window, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight. Microsoft Visual Studio simplifies the basic tasks of creating, debugging and deploying applications.

Microsoft Visual Studio comes with .NET Framework and supports applications targeting Windows. It supports IBM DB2 and Oracle databases, in addition to Microsoft SQL Server. It has integrated support for developing Microsoft Silverlight applications, including an interactive designer. Microsoft Visual Studio users several tools to make parallel programming simpler: in addition to the Parallel Extensions for the .NET Framework and the Parallel Patterns Library for native code, Visual Studio includes fools for debugging parallel applications. The Visual Studio code editor now highlights references; whenever a symbol is selected; all other usages of the symbol are highlighted. It also users a Quick Search feature to incrementally search across all symbols in C++, C# and VB.NET projects. Quick Search supports substring matches and camel Case searches. The Call Hierarchy feature allows the developers to see all the methods that are called from a current method as well as the methods that call the current one. Visual Studio supports a consumer best mode which developers can opt into.

4.2 Back End Used:Microsoft SQL Server

Microsoft SQL Server is an application used to create computer databases for the Microsoft Windows family of server operating systems. Microsoft SQL Server provides an environment used to generate database that can be accessed from workstations, the

Internet, or other media such as a personal digital assistant (PDA). Microsoft SQL Server is used to create desktop, enterprise, and web-based database applications. It is used at different levels and with various goals. SQL Server makes simpler and easier to deploy, manage, and optimise enterprise data and analytical applications. An enterprise data management platform, it performs a single management console that enables data administrators anywhere in your organisation to monitor, manage, and tune all of the databases and associated services across your enterprise. It provides an extensible management infrastructure that can be easily programmed by using SQL.

Enabling users to customise and extend their management environment and independent software vendors to build additional tools and functionality to further extend the capabilities that come out of the box. SQL Server simplifies management by providing integrated management console to monitor and manage the SQL Server relational database as well as integration services, analysis services, reporting services, notification services and SQL Server Mobile Edition across large number of distributed servers and databases. Database administrator can perform several tasks at the same time, such as authorising and executing a query, viewing server objects, managing an object, monitoring system activities, and viewing online help. SQL Server expose more than 70 new measures of internal database performance and resource usages, ranging from memory, locking, and scheduling to transactions and network and disk I/O. these dynamic management views provide greater transparency and visibility into the database and a powerful infrastructure for proactive monitoring of database health and performance. The major characteristics are listed below: Reliability: achieve a more secure deployment.

SQLServer provides rich security features to protect data and network resources. Condentiality: Protect your data. SQL Server clustering supports Kerberos authentication on a virtual Server and Microsoft-style policies on standard logins that a consistent policy is applied across all accounts in the domain. Integrity: SQL Server supports encryption capabilities within database itself, fully integrated with a key management infrastructure. By default, client server communications are in encrypted.administrator can perform several tasks at the same time, such as authorising and executing a query, viewing server objects, managing an object, monitoring system activities, and viewing online help. SQL Server expose more than 70 new measures of internal database performance and resource usages, ranging from memory, locking, and scheduling to transactions and network and

disk I/O. these dynamic management views provide greater transparency and visibility into the database and a powerful infrastructure for proactive monitoring of database health and performance. The major characteristics are listed below: Achieve a more secure deployment. SQL Server provides rich security features to protect data and network resources.

4.3 Back End Used:SQL Database Connectivity

The first thing you will need to do when interacting with a database is to create a connection. The connection tells the rest of the .NET code which database it is talking to. It manages all of the low-level logic associated with the specific database protocols. This makes it easy for you because the most work you will have to do in code instantiates the connection object, open the connection, and then close the connection when you are done.

Creating SQL Connection object

A SqlConnection is an object, just like any other C# object. Most of the time, you just declare and instantiate the SqlConnection all at the same time.

The SqlCommand Object

A SqlCommand object allows you to specify what type of interaction you want to perform with a database. For example, you can do select, insert, modify, and delete commands on rows of data in a database table. The SqlCommand object can be used to support disconnected data management scenarios, but in this lesson, we will only use the SqlCommand object alone. A later lesson on the SqlDataAdapter will explain how to implement an application that uses disconnected data. This lesson will also show you how to retrieve a single value from a database, such as the number of records in a table.

Creating SQL Command Object

Similar to other C# objects, you instantiate a SqlCommand object via the new instance declaration, as follows:

```
SqlCommand cmd = new SqlCommand("select CategoryName from Categories", conn);
```

The line above is typical for instantiating a SqlCommand object. It takes a string parameter that holds the command you want to execute and a reference to a SqlConnection object.

4.4 Discussion of code segment

The sample code snippets for Doctor user interface is as shown below:

Delete doctor data

```
{
   try
   {
      SqlCommand cmd = new SqlCommand("DocDelete_SP", con);
      cmd.CommandType = CommandType.StoredProcedure;
      cmd.Parameters.AddWithValue("@d_ssn", d_ssn.Text);
      con.Open();
      try
      {
         cmd.ExecuteNonQuery();

      }
      catch (Exception ex)
      {
         MessageBox.Show("<<<INVALID SQL OPERATION>>>" + ex);

      }
      con.Close();
      refresh_DataGridView();
   }
   catch(Exception ex)
   {
      MessageBox.Show("" + ex);
   }

}
Refresh data grid view
{
   try
   {
      SqlCommand cmd = new SqlCommand("ShowAllDocData_SP", con);
      cmd.CommandType = CommandType.StoredProcedure;

      SqlDataAdapter DA = new SqlDataAdapter(cmd);
      DataSet DS = new DataSet();
      DA.Fill(DS);

      con.Open();
      try
      {
         cmd.ExecuteNonQuery();
      }
      catch(Exception ex)
```

```
            {
               MessageBox.Show("        <<<INVALID SQL OPERATION>>>: \n" + ex);
            }
            con.Close();

            dataGridView1.DataSource = DS.Tables[0];

                        this.dataGridView1.Columns[0].AutoSizeMode = DataGridViewAuto-
SizeColumnMode.DisplayedCells;
                        this.dataGridView1.Columns[1].AutoSizeMode = DataGridViewAuto-
SizeColumnMode.Fill;
                        this.dataGridView1.Columns[2].AutoSizeMode = DataGridViewAuto-
SizeColumnMode.Fill;
                        this.dataGridView1.Columns[3].AutoSizeMode = DataGridViewAuto-
SizeColumnMode.Fill;
         }
         catch(Exception ex)
         {
            MessageBox.Show("" + ex);
         }
      }

      Add doctor data
      {
         SqlCommand cmd = new SqlCommand("DocAdd_SP", con);
         cmd.CommandType = CommandType.StoredProcedure;

         cmd.Parameters.AddWithValue("@d_ssn", d_ssn.Text);
         cmd.Parameters.AddWithValue("@d_name", d_name.Text);
         cmd.Parameters.AddWithValue("@phone", phone.Text);
         cmd.Parameters.AddWithValue("@d_address", d_address.Text);
         con.Open();
         try
         {
            cmd.ExecuteNonQuery();

         }
         catch(Exception ex)
         {
            MessageBox.Show("<<<INVALID SQL OPERATION>>>"+ex);

         }
         con.Close();
         refresh_DataGridView();

      }
```

```
        private void dataGridView1_CellContentClick(object sender, DataGridView-
CellEventArgs e)
    {

    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {

    }

    private void clear_Click(object sender, EventArgs e)
    {
      d_name.Text = "";
      d_ssn.Text = "";
      phone.Text = "";
      d_address.Text = "";
    }
  }
}
```

Search User details

```
    {
      try
      {
        SqlCommand cmd = new SqlCommand("BillSearch_SP", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@p_ssn", textBox1.Text);
        SqlDataAdapter DA = new SqlDataAdapter(cmd);
        DataSet DS = new DataSet();
        DA.Fill(DS);

        con.Open();
        try
        {
          cmd.ExecuteNonQuery();
        }
        catch(Exception ex)
        {
          MessageBox.Show("Invalid SQL Operation:" + ex);
        }
        con.Close();
        dataGridView1.DataSource = DS.Tables[0];
      }
      catch(Exception ex)
      {
        MessageBox.Show("" + ex);
      }
```

```
        }
    }
}
CREATE PROCEDURE [dbo].DocAdd_SP

  @d_name varchar(20),
  @d_ssn varchar(20),
  @phone varchar(15),
  @d_address varchar(20)


AS
        insert into Doctor (d_name,d_ssn,phone,d_address) values
(@d_name,@d_ssn,@phone,@d_address)
RETURN 0


CREATE PROCEDURE [dbo].DocDelete_SP
        @d_ssn varchar(15)
AS
        delete from Doctor where d_ssn=@d_ssn
RETURN 0


CREATE PROCEDURE [dbo].ShowAllDocData_SP

AS
        SELECT d.d_name,d.d_ssn,d.d_address,d.phone
        from Doctor d
RETURN 0

CREATE PROCEDURE [dbo].BillSearch_SP
   @p_ssn varchar(15)
AS
        SELECT b_no,bill_amount,bill_date
        From Bill b
        Where b.p_ssn=@p_ssn
RETURN 0

CREATE PROCEDURE [dbo].VisitDocSearch_SP
        @p_ssn varchar(15),
        @d_ssn varchar(15)

AS
RETURN 0
```

4.5 Usages

- Managing the record system in hospitals
- Keeping track of patients
- Keeping track of doctors on a daily basis.
- Database of the Billing system.
- List of medicines prescribed to the patients.

The idea is to expand the application to as many as medical centres from small clinics to large hospitals.

4.6 DISCUSSION OF THE RESULTS

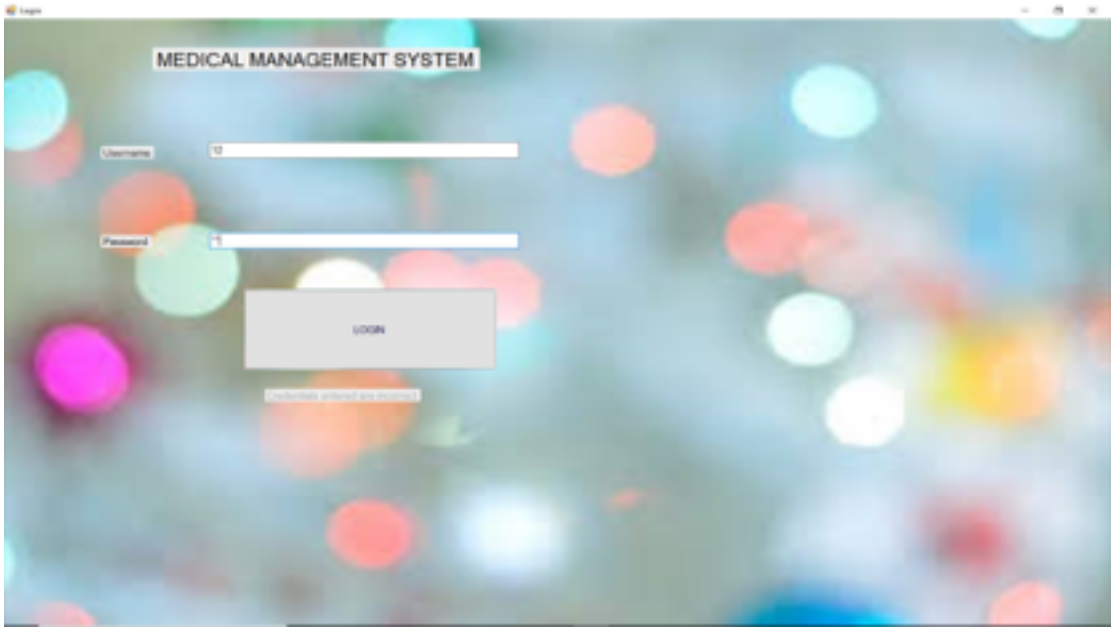

Figure. 5.1: Shows the login form

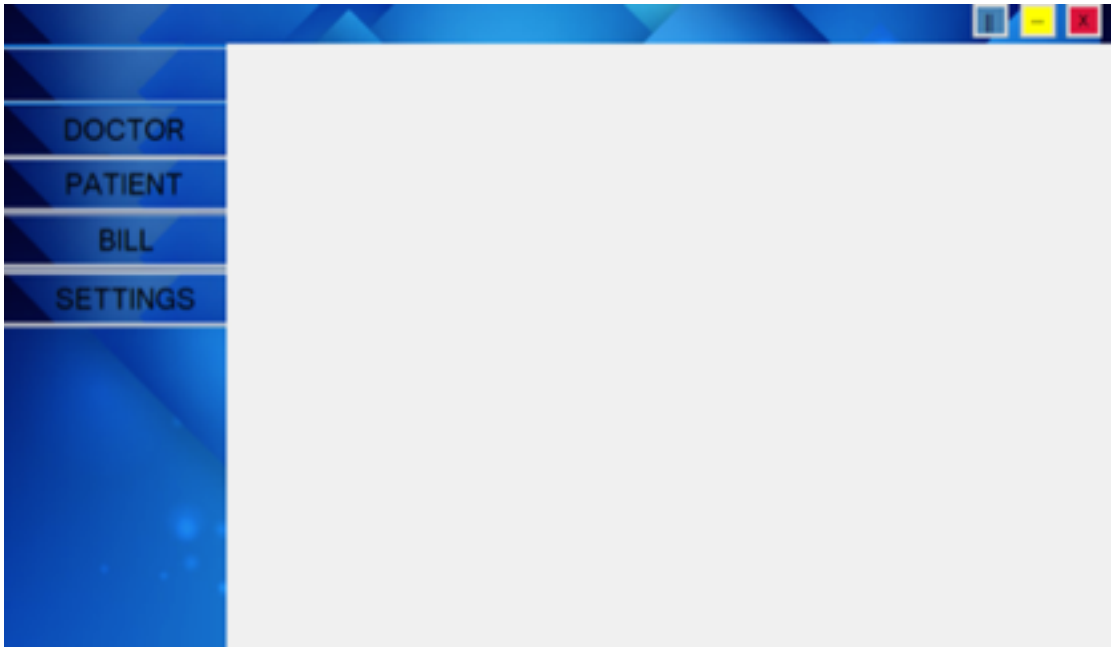

Figure. 5.2: Shows the application body/ main menu

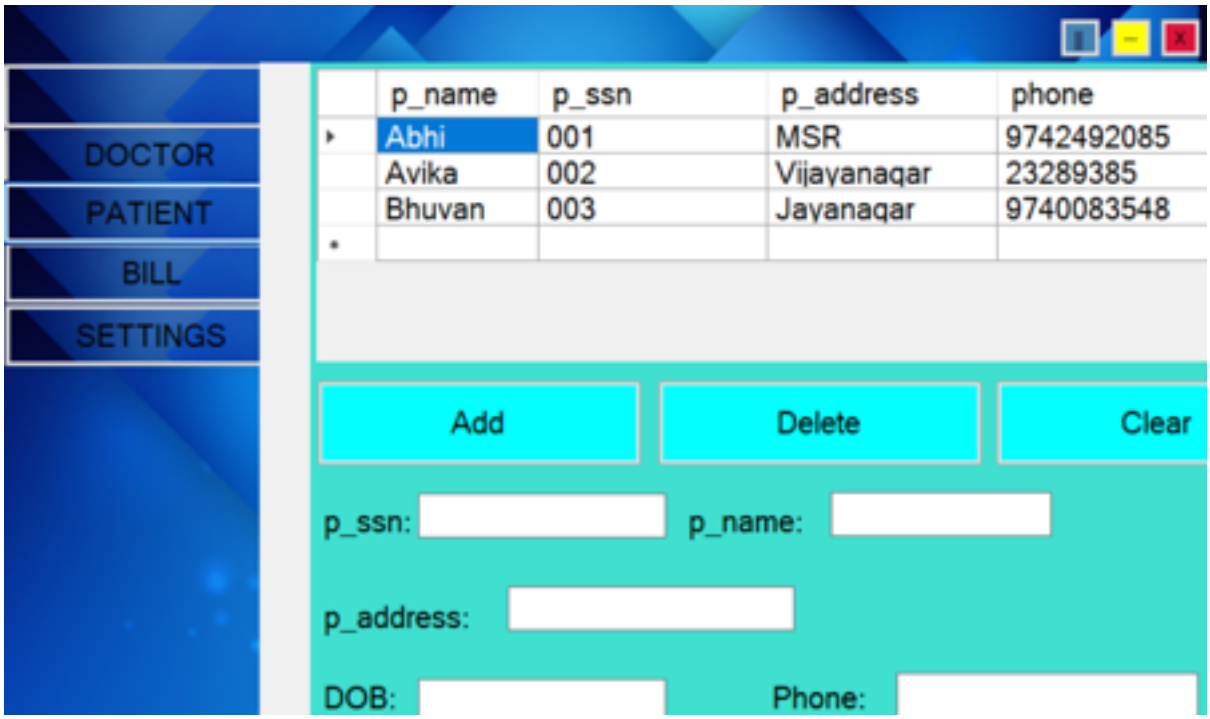

Figure. 5.3: Shows the patient form:

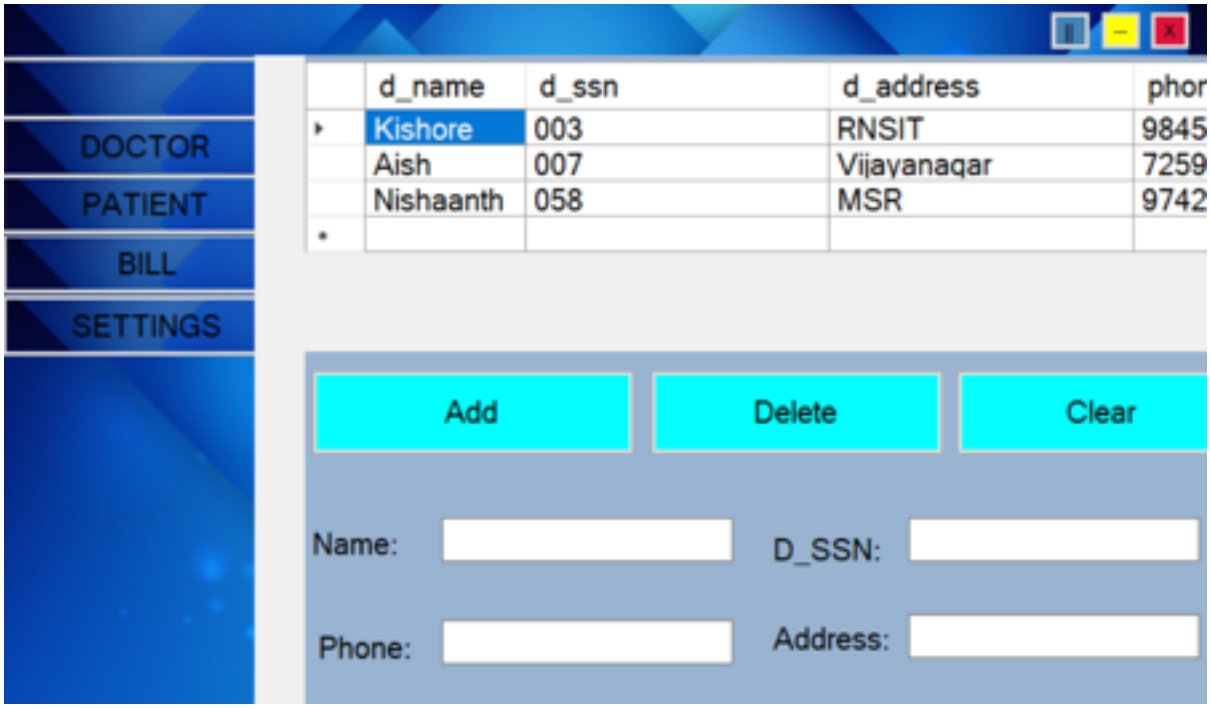

Figure. 5.4: Shows the doctor form

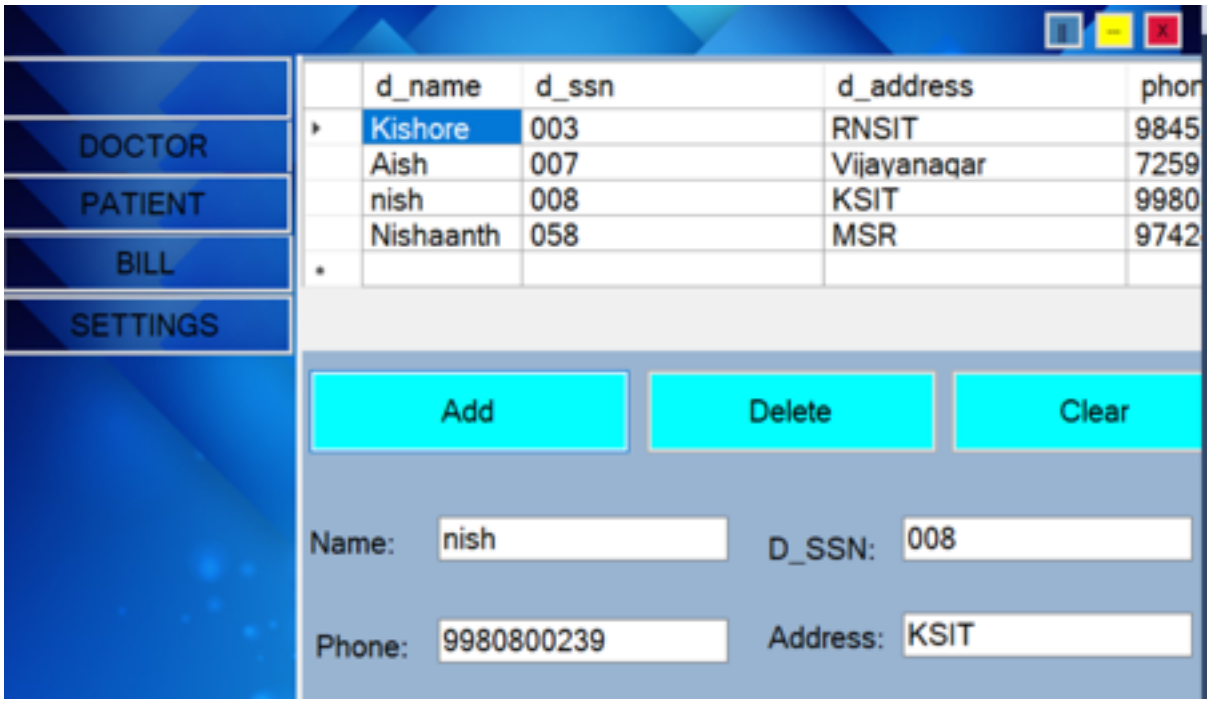

Figure 5.5: Shows the updated doctor record after adding a new doctor

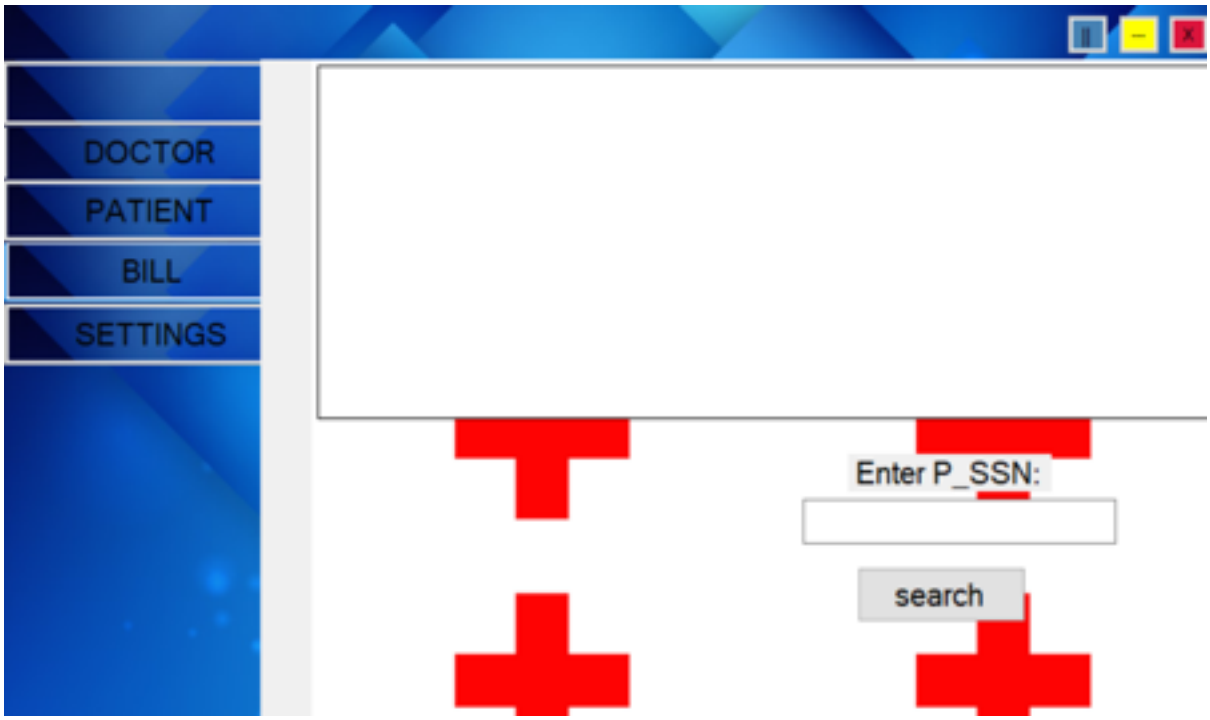

Figure 5.6: Bill menu

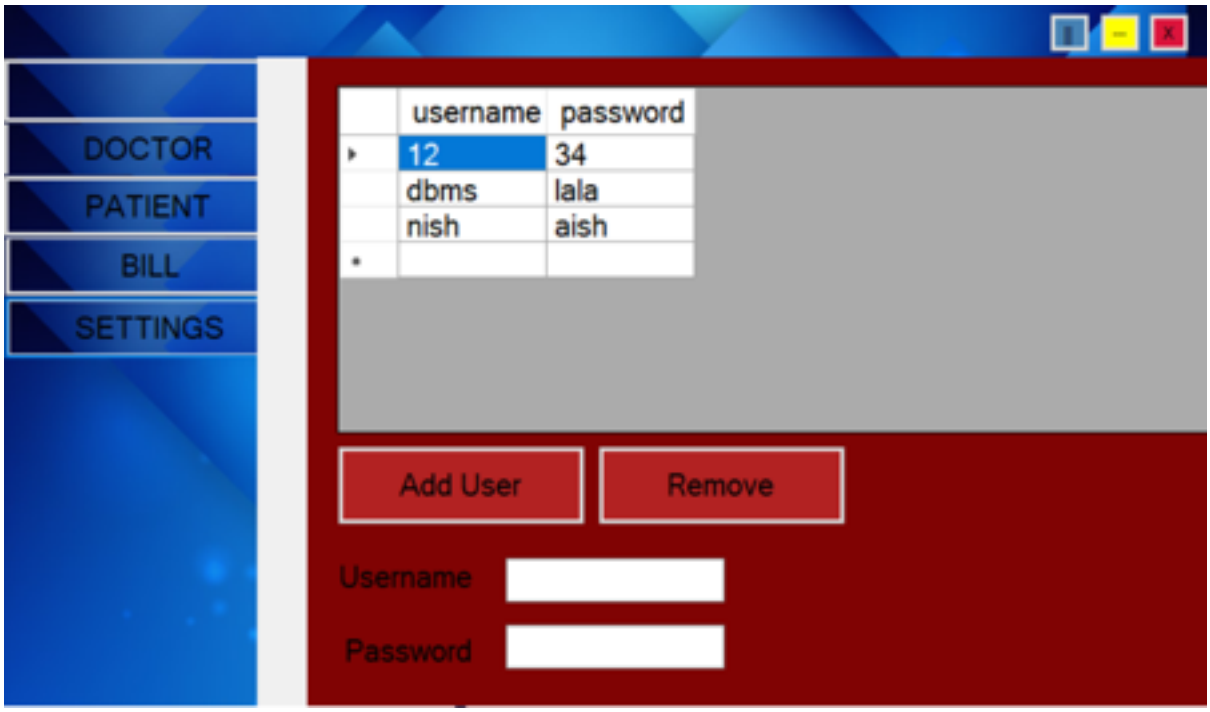

Figure 5.7: Shows Settings menu

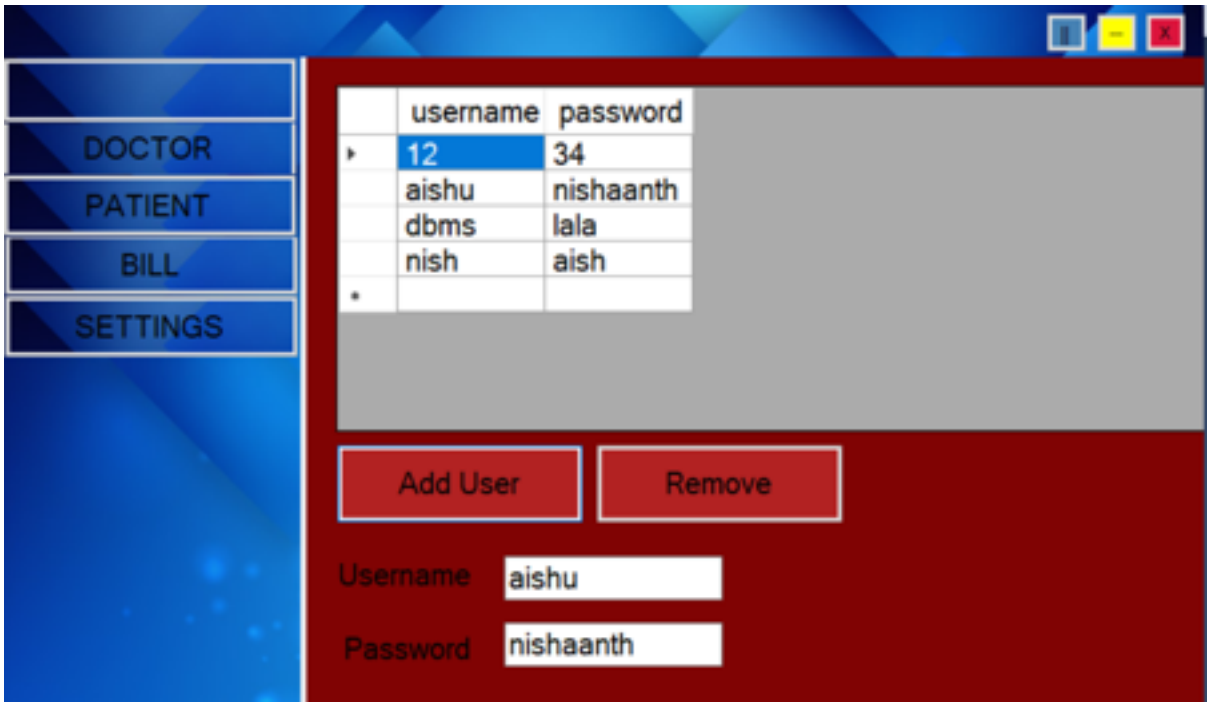

Figure 5.8: To add a new user in the settings menu

Chapter 5

CONCLUSION AND FUTURE ENCHANCEMENTS

The medical management system is developed using C# and SQL.It fully meets the objectives of the system which it has been developed.The system has reached a steady state where bugs all been eliminated.Merging the relations was made easy due to the fact that foreign keys existed that referenced the various tables that were merged.

The SQL statement became complex and more error-prone as the number of relations.

Printing generated bills,We can add more facilities like generating prescription, giving diagnosis details.

Hosting the platform on online servers for booking an appointment

Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.

Can be advanced to show due date of payment and bill amount can be made to pay over Paytm and other online transaction techniques.

REFERENCES

[1] Database Application Design by Dragomir R. Radev

[2] A Normal Form for relational databases that is not based on domains and keys(Article By Ronald Fagin.

http://www.wikipedia.com

http://www.books.google.com

http://www.w3schools.com

http://www.stackoverflow.com