REPORT

HOSPITAL MANAGEMENT SYSTEM

REG NO: SEU/IS/20/MIT/032

INDEX No: MIT 1314

Table of Contents

Introduction

The Hospital Management System (HMS) is a C# Windows Forms application designed to streamline and automate various hospital functions. These include patient registration, diagnosis management, record tracking, and administrative data handling. The system simplifies data entry and retrieval, ensuring that hospital staff can quickly and accurately access patient records, treatment histories, and other critical hospital information.

The application is built using C# and SQL Server as the database. It supports multi-user functionalities and offers a clean, user-friendly interface that allows staff to manage operations efficiently. The goal of this project is to create a centralized platform for hospitals to reduce paperwork, prevent errors, and improve data management.

Login Page

Overview

The login page is a security feature of the Hospital Management System. This may include authorities such as hospital staff. The login form is designed using C# Windows Forms and serves as the first interaction point between the user and the system. It verifies the user's credentials (username and password) before granting access to the system's dashboard.

The login page improves system security by preventing unauthorized access to sensitive patient data and hospital management tools.

Functional Description

The login form consists of two main input fields:

- **Username**: A `TextBox` where the user enters their username.
- **Password**: A `TextBox` where the user enters their password (with the password characters hidden for privacy).

There is also a **Login Button** that triggers the authentication process when clicked.

How it Works

- **User Interaction**:
  - The user enters their credentials (username and password).
  - After inputting the details, the user clicks the **Login** button.
- **Authentication**:
  - The system compares the entered username and password with predefined values.

- In this example, the valid username is **"clinic"** and the password is "pass".
- If the credentials match, the system grants access by hiding the login form and showing the `Dashboard` form.
- If the credentials do not match, an error message is displayed.

Result

System Design

The system is designed with a modular structure, allowing for various functionalities to be accessed via different forms and panels. Key modules include patient registration, diagnosis entry, viewing patient history, and hospital information management.

- **Client-side:** C# Windows Forms are used for the user interface. The UI elements include text boxes, combo boxes, data grids, and buttons, making it easy for users to enter and view data.
- **Server-side**: SQL Server is used to store and manage data. Various tables like AddPatient, PatientMore, and others store the patient details and diagnosis information.

Features and Functionalities

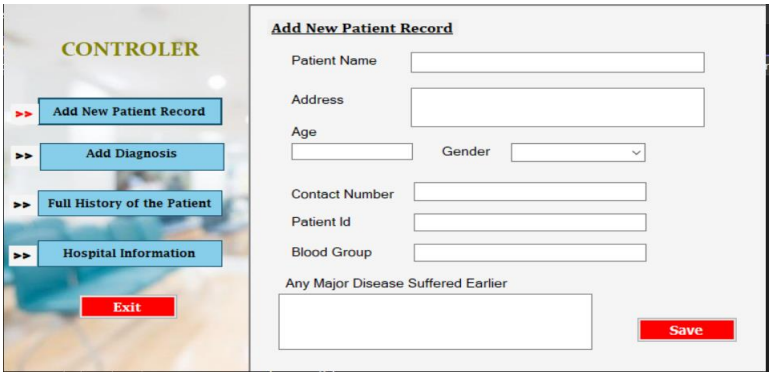
Add New Patient Record

This feature allows hospital staff to enter and store patient details, including the patient's name, age, gender, contact information, address, and medical conditions.

How it works:

1. The user clicks the "Add Patient" button, activating the corresponding form (panel1).
2. The form prompts the user to input patient details (name, contact, age, gender, blood group, etc.).
3. The system validates the input to ensure that all required fields are filled.
4. After clicking the "Save" button, the patient data is stored in the AddPatient table in the SQL Server database.

Result-



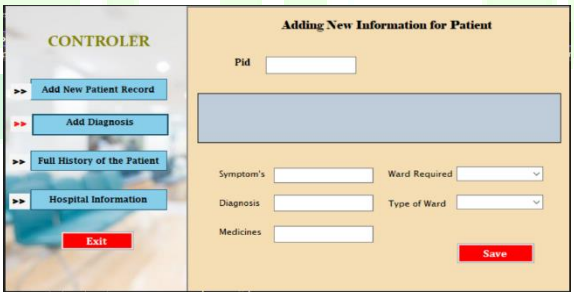
Add Diagnosis Information

After a patient's record is created, medical staff can enter the diagnosis information for the patient. This includes symptoms, diagnosis, medications, and ward details.

How it works:

1. The user selects a patient by entering the Patient ID and filling out diagnosis details (symptoms, diagnosis, medications, ward type).
2. After entering the data, the user clicks the "Save" button to store the diagnosis.
3. The diagnosis information is stored in the PatientMore table.

Result



Error message

After clicking the Add Diagnosis button, That Data is not Shown in my DataGrideView even if we type the pid number from the details in the previous form on the pid textBoxof the interface.

View Full History of the Patient

This feature allows users to view all records associated with a specific patient, including personal details and medical history.

How it works:

1. The user enters the Patient ID in the search box.
2. The system retrieves all records related to the patient, including diagnosis and history, by running a SQL join query between AddPatient and PatientMore tables.
3. The data is displayed in a DataGridView control for easy review.

Result -

Hospital Information Management

The hospital admin can enter or update general hospital information, including department names, services provided, and available staff.

How it works:

1. Users can access the hospital information section to input data about hospital departments and services.
2. This data is stored in a separate table (HospitalInfo).

Data Handling and Security

The system uses SQL Server for secure data storage, ensuring that patient data is safely stored in a relational database. The database is designed with tables such as:

- **AddPatient**: Stores patient information.
- **PatientMore**: Stores diagnosis and treatment details.

Data validation is done at the form level to prevent invalid data entry. Exception handling is also implemented to catch and display errors when users enter data in an incorrect format.

Result

	Name	Full_address	Contact	Age	Gender	Blood_Group	Major_Disease	pid
1	malithi	colombo	745896236	28	Female	B-	Yes	4
2	sachini	kandy	785412596	23	Female	B+	No	5
3	gihan	Mathara	785496523	23	Male	A-	No	6

	pid	Symptoms	Diagnosis	Medicines	Ward	Ward_Type
1	121					
2	121					
3	121	fever, high bp	140/120	electors 500mg	Yes	None
4	121					
5	244	high bp	180/120	losaton 500mg	Yes	AC
6	500	fever	170/100	vitamin C	No	None
7	244					
8	400	fever, high preshar	100/170	atvasting 500mg	Yes	Non-AC

User Interface

The user interface is designed using Windows Forms, offering an intuitive layout with various buttons and panels. Key UI elements include:

- **Buttons**: For adding patients, diagnosis, viewing history, and managing hospital info.
- **TextBoxes**: For entering patient and diagnosis details.
- **DataGridView**: To display data fetched from the database, such as patient records and histories.
- **Panels**: Used to switch between different forms (e.g., panel1 for adding patients, panel2 for diagnosis).

Conclusion

The Hospital Management System successfully streamlines patient data management and provides an efficient interface for handling hospital operations. By reducing manual paperwork and organizing patient information in a database, the system ensures quick and secure access to medical records, contributing to improved healthcare services. Future improvements could include adding more detailed analytics, integrating with billing systems, and enhancing the user interface.