جامعة ٦ أكتوبر
October 6 University

علوم الحاسب ونظم المعلومات
Info Sys & Comp Science

Practical Project

Hospital Management System

Prepared By :

Mohab Nasser Abdelkader	NT
Mahmoud Mohamed Ahmed	CS
Mahmoud Hany Mahmoud	CS
Menna Khaled Abdelrehim	CS
Ali Mohamed Mahmoud	NT

Supervised By :

Dr. Ayman Hassanein

Co-Supervisor :

Eng. Mohamed Abdelmoneim

Eng. Ahmed Abdelmoneim

Egypt 2025

Acknowledgment

Table of Contents

By the grace of Allah, we have successfully reached the completion of this practical project, which represents an important step in our learning journey. Over the past month, we worked on analyzing the idea, gathering the required information, and developing a clear and functional model of the system.

Today, we present the project and we welcome any comments or suggestions that may help us improve and enhance it.

We would like to express our sincere appreciation to **Dr. Nabila Mohamed Hassan** for her guidance and support throughout the preparation of this work.

We also extend our thanks to **Dr. Ayman Hassanein**, **Eng. Mohamed Abdelmonem**, and **Eng Ahmed Abdelmonem** for their valuable assistance and encouragement during the development of this project.

Their support played an important role in helping us complete this work in the best possible way.

Table of Contents

Table of Contents

Table of Figures

Table of Figures

Abstract

Abstract

Chapter One

"Introduction"

Chapter One

Introduction

1.1 What's The Hospital Management System?

The Hospital Management System (HMS) is a comprehensive digital solution designed to automate and integrate all hospital operations into one unified platform. The system facilitates the management of patient records, doctor schedules, appointments, billing, and real-time notifications.
Its main purpose is to enhance the efficiency of healthcare services by providing fast, accurate access to information for doctors, staff, and patients, resulting in a smoother and more effective workflow inside the hospital.

1.2 Problem of Traditional Hospital

Many hospitals still rely on manual processes and paper-based documentation, which leads to several challenges, including:

- Difficulty accessing patient files quickly when needed.
- High risk of losing or damaging important medical records.
- Poor coordination and communication between hospital departments.
- Frequent human errors in scheduling, billing, and data entry.
- Lack of automated notifications or patient updates about appointments or results.
- Long waiting times and slow registration procedures.

These issues negatively impact the quality of medical services and increase the workload on healthcare staff.

1.3 Importance Of Solving The Problem

- Enhance the overall quality of healthcare services.
- Reduce human errors significantly.
- Speed up daily operations and improve hospital productivity.
- Improve patient experience and satisfaction.
- Provide accurate, up-to-date information for better medical decision-making.
- Save time and effort for doctors, nurses, and administrative staff.
- Ensure secure and well-organized storage of patient data.

1.4 Project Objectives

The main objective of this project is to develop a fully integrated Hospital Management System that:

- Digitally manages and stores patient records.
- Allows doctors to view patient histories, diagnoses, and updates instantly.
- Organizes patient appointments and sends automated reminders.
- Improves communication through a centralized notification system.
- Automates billing and administrative procedures to minimize errors.
- Creates a unified environment that connects all hospital departments efficiently.

1.5 Proposed Solutions

To address the existing challenges, several solutions can be proposed:

- Implementing an electronic system to store and manage patient data.
- Developing an appointment system with automated reminders and notifications.
- Providing doctors with dashboards for quick and easy access to patient information.
- Creating a centralized database containing medical history, appointments, and treatment details.
- Introducing a notification system to inform patients about updates, results, and schedule changes.
- Automating administrative operations such as billing, payments, and reporting.

1.6 Chosen Solution

The chosen approach is to develop a **fully integrated Hospital Management System** that includes:

- **Patient Management:** Registration, record updates, medical history, and case tracking.

- **Doctor Management:** Schedules, patient files, diagnoses, and prescription management.
- **Appointment System:** Booking, modifying, and canceling appointments with automatic reminders.
- **Notification System:** Instant alerts and updates sent to patients.
- **Billing Module:** Generating, storing, and tracking bills and financial records.
- **Centralized Database:** Connecting all modules to ensure fast, secure access to information.

This solution was selected because it offers the highest effectiveness in addressing the operational, administrative, and communication problems hospitals face.

1.7 Tools & Technologies

The tools and technologies proposed for building the system include:

1.7.1 Programming Languages:

- C#

1.7.2 User Interface Technologies:

- .NET Windows Forms
- Web UI (HTML, CSS, JavaScript, React JS)

1.7.3 Database Systems:

- MySQL
- SQL Server
- Oracle Database

1.7.4 Additional Tools:

- Git & GitHub for version control
- Lucid chart / Draw.io for diagrams
- Microsoft Word or Google Docs for documentation
- MVC design pattern for structured and maintainable development

Chapter Two

"Design and Methodology"

Chapter Two

Design and Methodology

2.1 User Requirements

2.1.1. For Patients

- Patients want to book appointments easily.
- Patients want access to medical history, prescriptions, and test results.
- Patients want to view their bills and make online payments.

2.1.2. For Doctors

- Doctors want quick access to patient records.
- Doctors want to view lab results instantly.
- Doctors want an easy way to write prescriptions and notes.

2.1.3. For Nurses & Staff

- Nurses want to track patient vitals and treatment schedules.
- Staff want to manage rooms, beds, and internal tasks.

2.1.4. For Administrators

- Admins want full system control.
- Admins want to modify departments, staff accounts, and schedules.
- Admins want access to analytics and financial reports.

2.2 System Requirements

2.2.1. Hardware Requirements

- Central server with at least: 16GB RAM, Multi-core CPU, SSD storage.
- Backup server or cloud backup storage.

- Workstations or tablets for staff.
- Printer and barcode scanner (optional for pharmacy and lab).

2.2.2. Software Requirements

- Database: MySQL / PostgreSQL / SQL Server.
- Backend: Java / Python / .NET / Node.js.
- Frontend: React / Angular / HTML5.
- OS: Linux/Windows Server.
- Web server: Apache / Nginx.
- API support: REST/GraphQL.

2.3 Functional Requirements

2.3.1. User Management

- The system shall allow patients, doctors, nurses, and admins to create accounts.
- The system shall authenticate users through a secure login.
- The system shall allow admins to assign user roles.

2.3.2. Patient Management

- The system shall maintain electronic patient records (EHR).
- The system shall store patient demographics, medical history, allergies, lab results, and prescriptions.
- The system shall allow staff to update or review patient records.

2.3.3. Appointment Management

- The system shall allow patients to book, reschedule, or cancel appointments.
- The system shall display available time slots for each doctor.

- The system shall notify doctors and patients of appointment changes.

2.3.4. Doctor & Staff Management

- The system shall allow doctors to view patient files prior to appointments.
- The system shall allow doctors to create prescriptions and medical notes.
- The system shall allow staff to manage schedules and tasks.

2.3.5. Pharmacy Management

- The system shall manage drug inventory.
- The system shall track medication stock levels.
- The system shall notify when medicine stock is low.
- The system shall allow pharmacists to dispense medicine.

2.4 Non-Functional Requirements

2.4.1. Performance

- Appointment booking must load within 2 seconds.
- The system shall support at least 500 concurrent users.
- Transactions must be processed in under 3 seconds.

2.4.2. Security

- All patient data must be encrypted.
- Role-based access must restrict unauthorized access.
- All login attempts must be logged.
- The system shall implement two-factor authentication for admins.

2.4.3. Usability

- User interface must be simple and easy to navigate.
- The system must support both desktop and mobile interfaces.

- Medical record screens should be optimized for fast data entry.

2.4.4. Reliability & Availability

- System uptime must be at least 99.5%.
- Automated backups must run daily.
- The system must automatically recover from failures without data loss.

2.4.5. Scalability

- The system must support expansion to more departments or branches.
- Database must handle increasing patient records smoothly.

2.4.6. Maintainability

- The system must follow modular design principles.
- Code must be documented for future updates.

Chapter Three

"Conclusion & Recommendations"

Chapter Three

Conclusion & Recommendations

References

References