

Hospital Management System Documentation

Overview

The Hospital Management System is a C-based application providing a text-interface for managing hospital operations. It stores data in text files and includes role-based access for administrators and general users.

System Architecture

Main Component

- Hospital Management System.c: Main program containing all functionalities.

Data Storage Files

- admin passkey.txt: Stores admin login credentials.
- patientinfo.txt: Records patient data.
- doctorinfo.txt: Stores doctor details and availability.
- hospitalinfo.txt: Contains general hospital information.
- appointment_info.txt: Tracks appointment data.

Access Control System

Administrator Access

Admins can log in via admin passkey.txt and have access to:

Patient Management:

- Add, view, update, delete, and search patient records.

Doctor Management:

- Add, view, modify, delete, and search doctor details.

Hospital Information:

- View and update facility data.

Appointment Management:

Hospital Management System Documentation

- View all appointments and related statistics.

Customer/User Access

Users (without login) can:

Doctor Information:

- Browse and search doctors; check availability.

Hospital Information:

- View general details and contact information.

Appointment Management:

- Schedule, view, and cancel their appointments.

Data Management

- Uses text files for data persistence.
- Structured formatting with delimiters.
- Functions to read, parse, update, and write records.
- Validates inputs to maintain integrity.
- Search functionality across different data fields.

Key Features

- Dual Interface: Separate menus for admins and users.
- Data Persistence: All changes saved in text files.
- Search Capability: Multi-parameter record search.
- Access Control: Secure admin-only operations.
- Appointment System: Manages scheduling and cancellations.

System Requirements

- C compiler (GCC recommended)

Hospital Management System Documentation

- Text file read/write access
- Console/terminal environment

Implementation Notes

- Modular programming structure.
- Extensive file I/O operations.
- String parsing and manipulation.
- Input validation routines.
- Menu-driven navigation for ease of use.