Hospital Management System

A C++ console-based hospital appointment management system with priority-based triage, waiting list management, and appointment scheduling.

Author

Muhammed Mustafa Ozbey
Student ID: 5001230020

Features

Doctor Management

- Add doctors with ID, name, and specialty
- List all doctors with appointment counts
- View individual doctor schedules

Appointment System

- Schedule appointments with priority levels (1-5)
- Automatic time slot creation with configurable capacity
- Cancel appointments with automatic waiting list promotion
- List appointments by doctor and time slot
- Search for all appointments by patient ID

Triage Queue

- Priority-based patient queue (1 = highest, 5 = lowest)
- FIFO ordering for same priority levels
- Call next patient based on priority
- Undo last patient call

Waiting List

- Automatic waiting list when time slots are full
- Automatic promotion when appointments are cancelled

File Structure

```
├── run.cpp                # Main entry point
├── hospital_sys.h         # Hospital system class declaration
├── hospital_sys.cpp       # Hospital system implementation
├── doctor.h               # Doctor class declaration
├── doctor.cpp             # Doctor class implementation
├── patient.h              # Patient class declaration
├── patient.cpp            # Patient class implementation
├── appointment.h          # Appointment class declaration
├── appointment.cpp        # Appointment class implementation
├── timeslot_info.h        # Time slot information declaration
├── timeslot_info.cpp      # Time slot implementation
├── triage_entry.h         # Triage entry struct and comparator
└── triage_entry.cpp       # Triage entry implementation
```

Data Structures Used

- **std::map** - Store doctors by ID and time slots
- **std::priority_queue** - Triage queue with priority ordering
- **std::stack** - Call history for undo functionality
- **std::queue** - Waiting lists per time slot
- **std::vector** - Store appointments per time slot

Compilation

```
g++ -o hospital run.cpp hospital_sys.cpp doctor.cpp patient.cpp appointment.cpp timeslot_info.cpp triage_entry.cpp
```

Usage

Run the compiled program:

```
./hospital        # Linux/Mac
hospital.exe      # Windows
```

Menu Options

```
1. Add doctor
2. List all doctors
3. View doctor schedule
4. Schedule appointment
5. Cancel appointment
6. List appointments for a doctor and time slot
7. Call next patient (triage)
8. Undo last call
9. Find patient appointments
0. Exit
```

Example Workflow

1. **Add Doctors**

   ```
   Enter doctor ID: D001
   Enter doctor name: Dr. Ozbey
   Enter specialty: Cardiology
   ```

2. **Schedule Appointment**

   ```
   Enter patient ID: P001
   Enter patient name: Alice Bob
   Enter priority level (1=high, 5=low): 2
   Enter doctor ID: D001
   Enter time slot (type: 09:00): 09:00
   ```

3. **Call Next Patient (Triage)**

   ```
   Calling next patient
   Next patient:
     Patient ID: P001
     Name: John Doe
     Priority: 2
     Doctor: D001
     Time slot: 09:00
   ```

Class Descriptions

HospitalSystem

Main system controller managing doctors, appointments, and triage queue.

Doctor

Represents a doctor with schedule management capabilities.

Patient

Stores patient information including ID, name, and priority level.

Appointment

Represents a scheduled appointment linking patient and doctor.

TimeSlotInfo

Manages appointments and waiting lists for specific time slots.

TriageEntry

Entry in the priority queue with patient and appointment details.

Priority System

- **Priority 1**: Critical/Emergency cases
- **Priority 2**: Urgent cases
- **Priority 3**: Normal cases
- **Priority 4**: Low priority
- **Priority 5**: Routine checkups

Patients with the same priority are processed in FIFO order.

Configuration

Default time slot capacity: **2 patients per slot**
Can be modified in `HospitalSystem` constructor (`default_capacity` variable).

Notes

- Time slots are created automatically when first appointment is scheduled
- Waiting list patients are automatically promoted when appointments are cancelled
- All patient calls can be undone using the undo feature
- Patient IDs and Doctor IDs should be unique