## The University of Azad Jammu & Kashmir, Muzaffarabad

*OOP's Project Proposal of 2nd Semester*

**Project Title:**

# Hospital Management system

*Course Title: Object Oriented Programming*

*Group no: 07*

**Instructor:**

**Engineer Awais Rathor**

**Department:**

**Software Engineering**

**Group Members**

**Roll No:**

**2024-SE-06**

**2024-SE-35**

**2024-SE-40**

# 1. Introduction

The Smart Hospital Management System (SHMS) is a modern, Object-Oriented C++ application aimed at automating and integrating core hospital operations to improve efficiency, accuracy, and patient satisfaction.

In traditional hospitals, processes like patient registration, doctor allocation, appointment scheduling, billing, and pharmacy management are often manual, time-consuming, and prone to human error. The SHMS will address these challenges by providing an all-in-one platform to manage hospital operations digitally.

By implementing the four pillars of Object-Oriented Programming Encapsulation, Inheritance, Polymorphism, and Abstraction SHMS will ensure clean architecture, reusability, scalability, and maintainability.

The system will also allow for future expansion, such as integration with telemedicine services, AI-based appointment allocation, and real-time analytics dashboards for hospital administrators.

# 2. Objectives

The primary goals of SHMS are:

- Automation: Streamline hospital operations to save time and minimize manual work.
- Error Reduction: Ensure accuracy in data handling, billing, and scheduling.
- Scalability: Support easy integration of new services and modules.
- Security: Protect sensitive patient and hospital data.
- User Experience: Provide an intuitive interface for patients, doctors, and administrators.
- Analytics & Reporting: Offer valuable insights for better hospital decision-making.

# 3. Scope of the Project

The SHMS will provide the following functionalities:

- Patient Management: Register new patients, update details, view medical history.
- Doctor Management: Maintain doctor profiles, specializations, and schedules.
- Automatic Doctor Allocation: Assign doctors based on availability and specialization.
- Appointment Management: Online booking, walk-in scheduling, cancellation, and reminders.
- Billing System: Generate bills for consultations, diagnostics, and medicines.
- Pharmacy Management: Track medicine stock, expiry dates, and issue prescriptions.
- Diagnostics Management: Record lab test results and reports.
- Emergency Services: Enable quick admission with minimal patient details.
- Reporting & Analytics: Generate monthly, quarterly, or yearly hospital performance reports.

# 4. OOP Concepts Coverage

## Encapsulation

Definition: Wrapping data and related methods into a single unit while restricting access.
Implementation in SHMS:

- Classes like Patient, Doctor, and Billing will have private attributes and public getters/setters to ensure secure data handling.
- Example: Patient details (name, age, contact) stored privately, accessible only through methods like set Details () and get Details ().

## Inheritance

Definition: Deriving new classes from existing classes to reuse and extend functionality.
Implementation in SHMS:

- Base class: Person with attributes (name, age, gender, contact).
- Derived classes: Patient, Doctor, Staff.
- Specialized doctors (e.g., Surgeon, Cardiologist) will inherit from doctor.

## Polymorphism

Definition: Ability for the same function name to behave differently based on the object.
Implementation in SHMS:

- Overriding: display Info () in Person will be overridden in Patient and Doctor for specific outputs.
- Overloading: Multiple schedule Appointment () methods for online and walk-in bookings.
- Virtual Functions: generate Bill () in Billing overridden for insured vs. non-insured patients.

## Abstraction

Definition: Hiding complex implementation details while exposing only necessary functionality.
Implementation in SHMS:

- Abstract class Hospital Service with pure virtual functions like perform Service ().
- Subclasses: Pharmacy, Diagnostics, Emergency Service will implement the abstract methods**.**

# 5. System Modules

## User Management

  - Register, update, and delete patients and doctors.
  - Role-based access control (Admin, Doctor, Patient).

## Appointment Management

  - Intelligent scheduling based on doctor availability.
  - Automated reminders via email/SMS.

## Billing & Payments

  - Itemized billing for consultations, pharmacy, and diagnostics.
  - Multiple payment methods (cash, card, online).

## Pharmacy Management

  - Medicine stock tracking with low-stock alerts.
  - Expiry date monitoring.

## Diagnostics Module

  - Record lab tests, upload results, and integrate with patient history.

## Emergency Services

  - Rapid patient admission with minimal registration steps.

## Reporting & Analytics

  - Patient flow statistics.
  - Doctor workload analysis.
  - Revenue reports.

## Security Module (Advanced)

  - Data encryption for sensitive information.
  - Secure login with password hashing.

# 6. Tools & Technologies

- Programming Language: C++
- IDE: Code::Blocks / Dev C++ / Visual Studio
- Database: MySQL / SQLite / File-based storage

- Paradigm: Object-Oriented Programming
- Optional Libraries:
  - SMTP for email notifications.
  - PDF generation libraries for billing and reports.

## 7. Advanced Features (Optional)

- Telemedicine Integration: Video consultations and remote diagnosis.
- AI-Based Appointment Allocation: Suggest doctors based on symptoms and availability.
- Inventory Prediction: Predict medicine demand using past usage trends.
- Multi-Language Support: English and local language for accessibility.
- Backup & Restore: Secure data backups to avoid data loss.

## 8. Expected Benefits

- Efficiency: Reduced waiting times and faster hospital operations.
- Accuracy: Minimized human errors in billing and record-keeping.
- Scalability: Ability to add new hospital services in the future.
- Security: Protected patient data through access control and encryption.
- User Satisfaction: Smooth hospital experience for patients and staff.

## 9. Project Timeline (Example)

| Phase | Duration | Description |
|---|---|---|
| Requirements Analysis | 1 Week | Gather requirements, finalize scope. |
| Design & Architecture | 1 Week | Create UML diagrams, database schema. |
| Development | 3 Weeks | Code implementation of modules. |
| Testing | 1 Week | Unit testing, integration testing. |
| Deployment | 1 Week | Deploy system, prepare documentation. |

# 10.Future Enhancements

- Integration with IoT-based patient monitoring systems.
- AI-powered health prediction based on patient history.
- Blockchain for secure patient data sharing between hospitals.