**COVER PAGE**

A logo of a university

Description automatically generated

EEC 521/CIS 534: Software Engineering

Professor: Dr. Yongjian Fu

|  |  |  |
| --- | --- | --- |
| **Team Members** | | |
| Team Member 1 | LIKITH REKAPALLI | CSU ID: 2845684 |
| Team Member 2 | KRISHNA SAI SALLA | CSU ID: 2887720 |

**Developer's Guide**

**1.0 Introduction & Background**

**1.1 Introduction**

The **Hospital Management System (HMS)** is a web-based platform designed to automate and streamline hospital administrative processes, including patient management, staff management, billing, room assignments, and medical record handling. The system aims to facilitate seamless interactions between hospital staff (doctors, nurses, administrative personnel) and patients. Patients can book appointments, view their medical records, and make secure payments online. Staff can manage patient data, view medical histories, and generate invoices for services rendered. The system integrates multiple hospital departments, ensuring data consistency and improving operational efficiency.

**1.2 Objectives**

The primary objectives of the Hospital Management System are:

1. **Patient Management**:
   * Enable hospitals to manage patient records, including personal information, medical history, and current health status.
2. **Staff Management**:
   * Manage staff details (doctors, nurses, admin staff), including scheduling.
   * Enable role-based access control to ensure proper access to information based on user roles.
3. **Billing and Invoicing**:
   * Generate bills for various hospital services (consultations, treatments, medical tests).
   * Enable online payments and integration with external financial systems for payment tracking.
4. **Room and Bed Management**:
   * Keep track of room availability and assign patients to rooms as needed.
   * Ensure efficient utilization of hospital beds and resources.
5. **Medical Records**:
   * Maintain a secure and centralized repository of medical records for each patient, including prescriptions, test results, and treatment histories.
   * Allow doctors and authorized staff to access and update records as necessary.
6. **Reporting and Analytics**:
   * Provide comprehensive reports for hospital administrators on various metrics, including patient volumes, staff performance, and financial data.
7. **Security and Privacy**:
   * Implement robust security measures to protect sensitive patient and hospital data.
   * Ensure compliance with relevant data protection laws such as HIPAA or GDPR.
8. **User Interface**:
   * Design a simple, intuitive interface for hospital staff and patients to interact with the system.

**1.3 Project Scope**

The scope of the Hospital Management System includes:

* **Patient Features**: Patient registration, viewing medical records, billing and payment.
* **Staff Features**: Staff registration, schedule management, access to patient records.
* **Administration Features**: Managing patient data, room assignments, generating reports, system configuration.
* **System Features**: Integration of various modules (billing, medical records, etc.) into a cohesive system.
* **Platform Requirements**: Cross-platform support, accessible via web browsers and mobile devices.

**Exclusions**:

* Integration with external hospital equipment or systems (such as diagnostic machines).
* Advanced features like telemedicine (which may be added in future versions).

**2.0 Architecture**

**2.1 Software Architecture & Requirements Specifications**

**1. Technical Requirements**

* **Hardware**:
  + Servers for hosting the application and database.
  + Sufficient computing resources for concurrent user access (at least 1000 users).
* **Software**:
  + Front-end: HTML5, CSS3, JavaScript, ReactJS or AngularJS.
  + Back-end: PHP.
  + Database: Microsoft SQL Server or PostgreSQL.
  + Operating System: Windows Server 2016 or higher or Linux-based systems.
* **Platform Compatibility**:
  + Web browsers: Chrome, Firefox, Safari, Edge.
  + Mobile responsiveness for tablets, and desktops.

**2. Functional Requirements**

* **User Authentication**: Secure login system for patients, hospital staff, and administrators.
* **Patient Management**: Registration, profile management, medical history.
* **Staff Management**: Staff scheduling, assignment tracking.
* **Billing & Payment**: Generate invoices.
* **Medical Record Management**: Store and manage medical records securely, with access controlled by roles.
* **Room Management**: Assign and track available hospital rooms and beds.

**3. Performance Requirements**

* **Response Time**: The system must respond within 3 seconds for user actions (booking appointments, viewing medical records).
* **System Uptime**: 99% uptime to ensure availability and minimal downtime.
* **Concurrent User Capacity**: The system should handle at least 1000 concurrent users without significant performance degradation.

**4. Design Constraints**

* **UI/UX**: The user interface should be simple, intuitive, and designed for hospital staff with varied technical expertise.
* **Compliance**: The system should meet healthcare data protection regulations (e.g., HIPAA, GDPR).

**5. Database Requirements**

* **Patient Data**: Store sensitive data securely, including medical history, test results, and prescriptions.
* **Staff Data**: Manage staff profiles, roles, and schedule details.
* **Room Management Data**: Track room availability and assignment.
* **Billing Data**: Maintain invoices.

**6. Security Requirements**

* **Data Encryption**: All sensitive data must be encrypted using SSL/TLS for secure communication.
* **Role-Based Access Control (RBAC)**: Access to different parts of the system (patient data, billing, etc.) should be restricted based on roles (admin, doctor, nurse).
* **Authentication**: Use multi-factor authentication (MFA) for sensitive system access.

**7. Maintainability Requirements**

* **Codebase**: The code should be modular, well-commented, and easy to extend or modify.
* **Version Control**: Use Git for version control to manage updates and collaborations.

**8. Usability Requirements**

* **User Interface**: Design should be easy for hospital staff and patients, with clear navigation and minimal training required.
* **Mobile Responsiveness**: Ensure a responsive design for patients and staff accessing the system on mobile devices.

**2.2 Database Design**

The following tables are used in the database schema:

* **Patient Table**: Stores patient details (ID, name, DOB, contact info, medical history).
* **Staff Table**: Stores staff details (ID, name, role, department, schedule).
* **Appointment Table**: Stores patient appointment records (appointment ID, patient ID, doctor, date, time).
* **Billing Table**: Stores transaction data (invoice ID, patient ID, services rendered, amount).
* **Room Table**: Stores room availability and assignment data (room ID, patient ID, room type, bed status).

**2.3 Software Interfaces**

* **IDE**: Visual Studio Code or Visual Studio 2019 for .NET development (for back-end and front-end).
* **Database**: Microsoft SQL Server for relational database management.
* **Front-End**: ReactJS for dynamic front-end development.

**3.0 Conclusion**

**3.1 Remarks on Implementation**

The development of the **Hospital Management System** has successfully automated critical hospital processes, ensuring smoother operations and improving service delivery. The system integrates various modules, including patient management, billing, and staff management, into a unified platform. Key technologies such as PHP, SQL Server, HTML, CSS and JS were used to build a robust, scalable, and responsive application.

**3.2 Future Improvements**

* **Telemedicine Integration**: Future versions could include video consultation features.
* **Advanced Analytics**: Implement AI-based tools to predict patient outcomes and optimize hospital resource allocation.
* **Mobile App**: Develop a mobile app for patients to manage their appointments, payments, and health records on the go.
* **Data Backup and Disaster Recovery**: Implement a robust backup system and disaster recovery plan to ensure data integrity and availability.