Software Requirement Specification

Hospital Management System

1. Project Overview

1.1 Introduction

The Hospital Management System (HMS) is a WordPress-based web application designed to manage various hospital activities, including patient registration, appointment scheduling, doctor management, billing, and reporting. The system provides an interface for administrators, doctors, patients, and support staff to streamline hospital processes.

1.2 Objectives

- To automate patient registration and appointment processes.
- To improve patient data management and security.
- To create a system for efficient resource allocation and management.
- To enhance patient-doctor interaction through a user-friendly portal.

1.3 Scope

The system will handle key hospital operations, including:

- Patient information management
- Doctor and staff scheduling
- Billing and reporting
- Appointment booking and management
- Role-based access for different user types (Administrator, Doctor, Patient, Staff)

1.4 Technology Stack

- **Frontend**: WordPress with custom themes and plugins
- **Backend**: PHP (via WordPress)
- **Database**: MySQL
- **Design**: Figma

- **Workflow Management**: Lucidchart

2. Functional Requirements

2.1 User Roles

1. **Administrator**
   - Manage doctors, patients, appointments, and billing.
   - Assign roles and permissions.
   - Generate reports.
2. **Doctor**
   - View appointments.
   - Manage patient records and treatment history.
   - Update availability and schedules.
3. **Patient**
   - Register and log in to view and manage their profile.
   - Book, cancel, or reschedule appointments.
   - View medical history and billing information.
4. **Staff**
   - Assist in managing appointments and patient records.
   - Handle patient billing and reports.

2.2 Key Functionalities

1. **Patient Registration and Management**
   - Secure registration and profile management.
   - Personal data protection in compliance with data security standards.
2. **Appointment Scheduling**
   - Calendar view for scheduling.
   - Appointment rescheduling and cancellation options.
3. **Doctor Availability and Scheduling**
   - Manage doctor schedules and availability.
   - Allow for real-time schedule updates.
4. **Billing and Invoice Generation**
   - Generate bills based on services and treatments.
   - Secure payment gateway for online transactions.
5. **Reporting and Analytics**
   - View patient statistics and hospital reports.
   - Export data for further analysis.

3. Design Documentation

3.1 UI/UX Design (Reference Figma)

The user interface is designed in Figma with the following sections:

- **Dashboard**: An overview page for each role, showcasing essential functionalities.
- **Patient Management**: Forms and tables for managing patient details.

- **Appointment Management**: A calendar and booking interface for managing appointments.
- **Billing**: Invoice templates and billing summary views.

3.2 Workflow Design (Reference Lucidchart)

The system workflow, as created in Lucidchart, includes:

- User registration and authentication flow.
- Appointment booking and confirmation process.
- Billing and payment flow.

Each component in the workflow is linked to specific user roles, ensuring clear role-based operations.

4. Database Structure

4.1 Database Tables

The system uses a MySQL database with the following primary tables:

1. **Users**
   - Stores user details like role, contact, and login information.
2. **Patients**
   - Holds patient-specific information, including medical history and personal data.
3. **Doctors**
   - Stores details of doctors, including specialization and availability.
4. **Appointments**
   - Manages appointment details with references to patient and doctor IDs.
5. **Billing**
   - Records financial transactions and invoice details for patients.

4.2 Entity-Relationship Diagram (ERD)

Refer to the **Lucidchart link** for a detailed ERD of the system. It outlines relationships such as:

- **One-to-Many** between Doctors and Appointments.
- **One-to-Many** between Patients and Appointments.
- **One-to-One** between Patients and Billing.

5. Testing and Quality Assurance

5.1 Testing Methodology

The system will undergo:

- **Unit Testing**: To validate individual components.
- **Integration Testing**: To ensure components interact as expected.
- **System Testing**: For complete system evaluation.
- **User Acceptance Testing (UAT)**: To verify user requirements are met.

5.2 Test Cases

1. **User Registration**
   - **Expected Outcome**: Users can register with valid information.
   - **Result**: Pass/Fail
2. **Appointment Booking**
   - **Expected Outcome**: Patients can book, reschedule, or cancel appointments.
   - **Result**: Pass/Fail
3. **Billing Process**
   - **Expected Outcome**: Accurate invoice generation and secure payment processing.
   - **Result**: Pass/Fail
4. **Data Access and Security**
   - **Expected Outcome**: Role-based access control is maintained.
   - **Result**: Pass/Fail

5.3 Performance and Load Testing

- **Objective**: Ensure system stability under high load.
- **Metrics**: Response time, error rate, and resource utilization.

6. Conclusion

This documentation outlines the functionality, design, and structure of the Hospital Management System developed on WordPress. Following this will aid in future maintenance, scaling, and potential upgrades.