# **Software Design Document (SDD)**

**Project Name:** Clinic Appointment Booking System

**Author(s):** Group 1 – BSc. Information Systems, Year 3

**Version:** 1.0

**Date:** 2025-10-13

## **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Summary of Changes** |
| 1.0 | 2025-10-13 | BIS Group | Initial version for university project |

## **Table of Contents**

1. Introduction
2. System Overview
3. System Architecture
4. Database Design
5. Module Descriptions
6. User Interface Design
7. Security Considerations
8. Testing Strategy
9. Deployment
10. Conclusion

## **1. Introduction**

### **Purpose**

This document provides the software design for the **Clinic Appointment Booking System**, a university-level project developed by third-year Information Systems students. It describes the structure, components, and functionality of the system in a simplified and practical way.

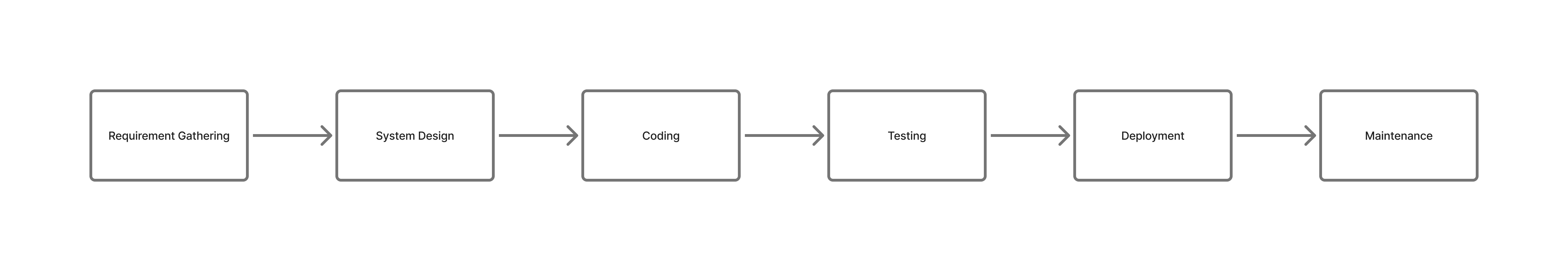
### **Scope**

The system allows patients to book medical appointments online, doctors to manage their schedules, and administrators to oversee clinic operations. The project is limited to basic booking, user management, and report generation features.

### **Objectives**

* To digitize the appointment booking process.
* To reduce manual paperwork in clinics.
* To enable patients to schedule appointments easily.
* To allow doctors to manage appointments effectively.

**The Software Development Life Cycle**



## **2. System Overview**

### **Description**

The Clinic Appointment Booking System enables patients to register, log in, and book appointments with doctors. Doctors can log in to view appointments, confirm availability, and manage bookings. The administrator manages users, schedules, and reports.

### **Users**

* **Administrator:** Manages system users and overall settings.
* **Doctor:** Views and manages their appointments.
* **Patient:** Books and views appointments.

### **Tools and Technologies**

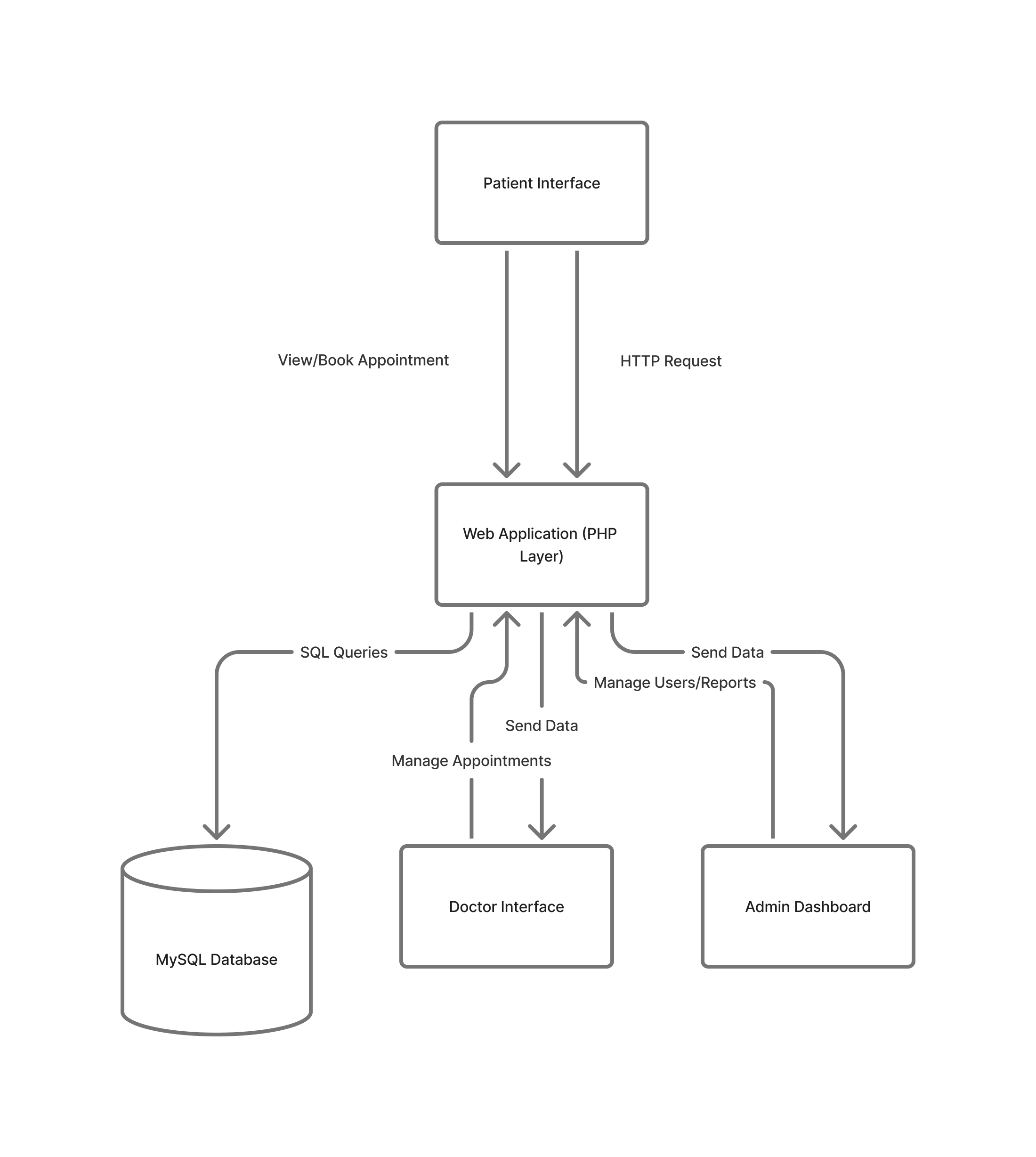
* **Frontend:** HTML, CSS, JavaScript, Tailwind
* **Backend:** PHP, Node.js
* **Database:** MySQL
* **Server Environment:** XAMPP (Apache, MySQL, PHP)

## **3. System Architecture**

The system follows a **three-tier architecture**:

* **Presentation Layer:** User interfaces for patients, doctors, and admin.
* **Application Layer:** Handles business logic and user requests (PHP scripts).
* **Data Layer:** Stores user and appointment data in MySQL.

**Architectural Diagram**



**4. Database Design**

### **Main Tables**

1. **patients** - stores information about patients
2. **appointments** - stores appointment details (date, time, doctor, patient)
3. **doctors** - stores doctor-specific details like specialization
4. **administrator** - stores information about the administrator

### **Example Table Structure: appointments**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| id | INT (PK) | Unique appointment ID |
| patient\_id | INT (FK) | Reference to user table |
| doctor\_id | INT (FK) | Reference to doctor table |
| date | DATE | Appointment date |
| time | TIME | Appointment time |
| status | VARCHAR(20) | e.g., booked, confirmed, cancelled |

## **5. Module Descriptions**

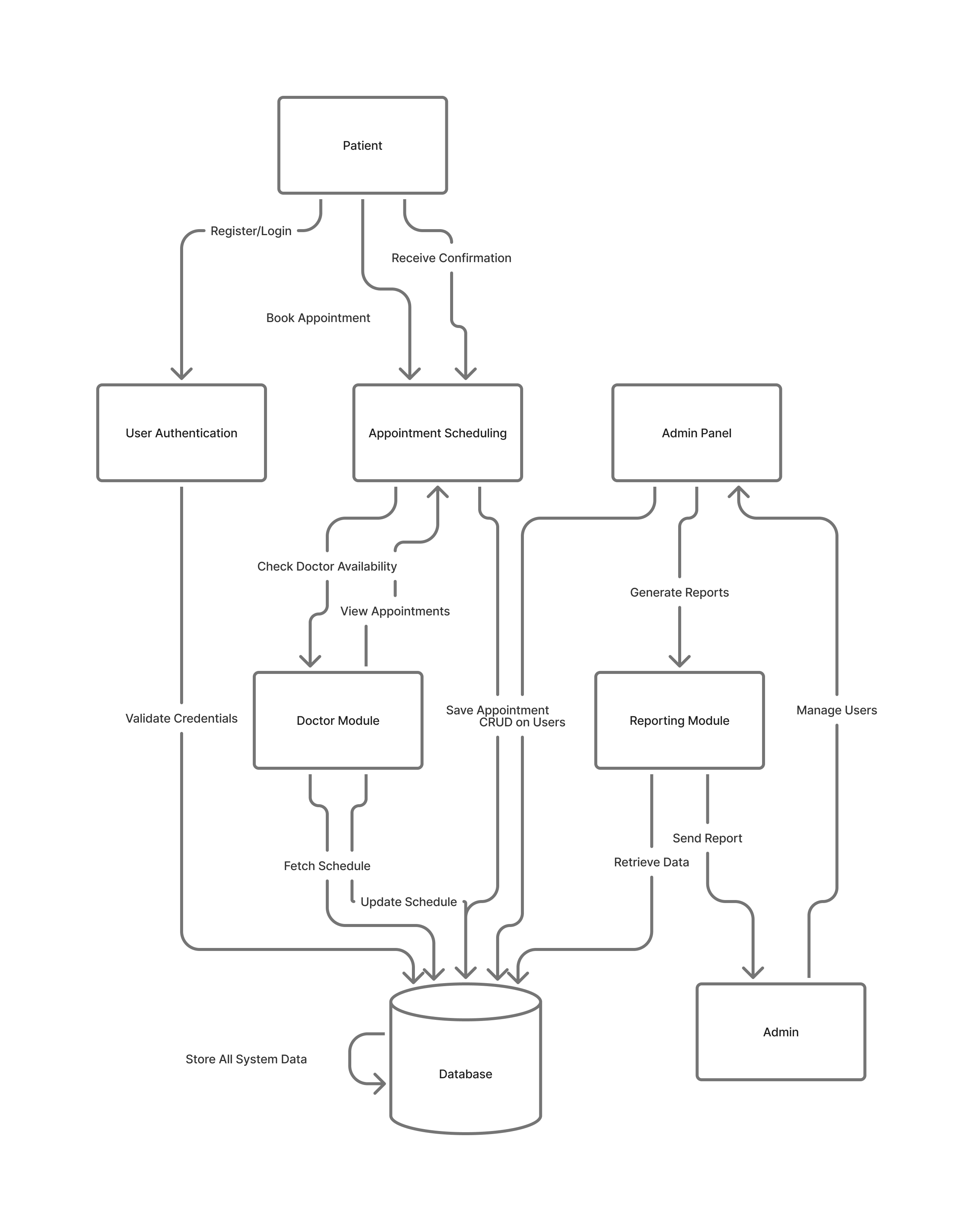
### **a) User Management Module**

Handles user registration, login, and roles (Doctor, Patient).

### **b) Appointment Management Module**

Allows patients to book appointments and doctors to manage their schedules.

## **Data Flow Diagram for the Clinic Appointment Booking System**



## **6. User Interface Design**

### **Pages Overview**

* **Login Page** – Users enter credentials.
* **Registration Page** – For new users to sign up.
* **Patient Dashboard** – View and book appointments.
* **Doctor Dashboard** – View scheduled appointments.

All pages are designed to be responsive and simple for usability.

## **7. Security Considerations**

* Passwords are hashed before storage.
* Input validation is applied to prevent SQL Injection.
* Access control ensures users can only access authorized pages.

## **8. Testing Strategy**

### **Testing Methods**

* **Unit Testing:** Individual PHP functions.
* **Integration Testing:** Booking flow from patient to doctor.
* **User Acceptance Testing (UAT):** Carried out by classmates or instructors.

### **Test Scenarios**

* Successful login and logout.
* Booking an appointment.
* Viewing doctor’s schedule.
* Admin managing users.

## **9. Deployment**

The project runs locally on XAMPP. The system can later be deployed to a live web server such as InfinityFree or Hostinger for demonstration.

## **10. Conclusion**

This SDD outlines the basic structure and components of the Clinic Appointment Booking System developed as part of a university project. It focuses on simplicity, usability, and functionality for small to medium clinics.

**Prepared by:** Group 1  
**Course:** Bachelor of Information Systems (Year 3)  
**Institution:** Kyambogo University  
**Date:** 2025-10-13