ONLINE APPOINTMENT FOR HEALTHCARE CLINIC

Business Requirements Document (BRD)

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Version | Date | Description |
| Zohaib Waqar | 1.0 |  |  |

Table of Contents

**System Architecture Overview 2**

**Component Breakdown 3**

**Data Design 3**

**Technology Stack4**

**Assumptions & Constraints 4**

**Any Other4**

Sub heading4

1. **System Architecture Overview**

The system is structured as a traditional MVC (Model-View-Controller) web application. It is organized as follows:

* 1. **Frontend:** Server-rendered web pages using HTML/CSS/JavaScript using Razor syntax.
  2. **Backend:** A monolithic backend using a web framework that adheres to MVC architecture (e.g. ASP.NET Core MVC).
  3. **Database:** Microsoft SQL Server used for data persistence.
  4. **APIs:** Currently, no RESTful APIs are exposed externally.

1. **Component Breakdown**

|  |  |
| --- | --- |
| Module | Responsibility |
| Register | Allows new users (patients, doctors, staff) to create accounts. |
| Login | Authenticates users and provides role-based access to the system. |
| Appointment Booking | Allows users (patients) to view available slots and book appointments. |
| Appointment History | Enables logged-in users to view past and upcoming appointments. |
| Profile | Displays and allows editing of user information (available only after login). |
| Schedule Availability | Enables doctors to set their available times for appointments. |
| Create Schedule | Enables staff to define time slots and manage availability calendars for doctors. |
| Reports | Allows admin/staff to generate reports (e.g., total appointments, no-shows, etc.). |

1. **Data Design**

|  |  |
| --- | --- |
| Key Entities | Fields |
| User | UserID, Name, Email, PasswordHash, Role (Patient/Doctor/Staff) |
| Profile | ProfileID, UserID (FK), Phone, Address, Gender, DOB, Specialization (for doctors) |
| Appointment | AppointmentID, PatientID (FK), DoctorID (FK), ScheduleID (FK), Date, Time, Status |
| Schedule | ScheduleID, DoctorID (FK), Date, StartTime, EndTime, AvailabilityStatus |
| Report | ReportID, ReportType, GeneratedDate, FiltersUsed, CreatedBy (StaffID) |

* 1. **Entity Relationships (ERD Summary):**
     1. A **User** can have one **Profile**
     2. A **Doctor (User)** can have many **Schedules**
     3. A **Patient (User)** can book many **Appointments**
     4. An **Appointment** links a **Patient**, a **Doctor**, and a **Schedule**

1. **Technology Stack**

|  |  |
| --- | --- |
| ****Layer**** | ****Technology**** |
| Frontend | HTML, CSS, JavaScript (or Razor if ASP.NET) |
| Backend | ASP.NET Core MVC (or similar MVC Framework) |
| Database | Microsoft SQL Server |
| ORM (if any) | Entity Framework or ADO.NET |
| Reporting Tools | In-app export features |
| Authentication | Form-based login with role-based access |

1. **Assumptions & Constraints**
   1. Only registered users (Patients/Doctors/Staff) can access core features like Appointment History, Doctors Scheduling & Report, Finance Reports.
   2. System roles are predefined: Patient, Doctor, Staff and Admin.
   3. Doctors can only set their own availability.
   4. Appointment conflicts (double-booking) will be prevented via backend validation.
   5. All times are assumed to be in a single time zone unless extended.
   6. No external API integration is currently planned (e.g., SMS/email notification).
   7. Reports are assumed to be downloadable in PDF/Excel formats.

**Appendix ‘A’**

**HEADING OF APPENDIX A**