**Project Specification: Medical Registration and Consultation Platform**

**Executive Summary**

The Medical Registration and Consultation Platform is designed to streamline patient registration, booking, and consultation processes while maintaining compliance with medical and regulatory requirements. The platform will cater to patients, nurses, and administrators through a web-based application built with React.js (frontend) and Node.js (backend).

This document integrates elements from the previous project specification for 420Doctors and the new requirements to deliver a comprehensive, functional system.

**Objectives**

1. **Simplify User Onboarding:** Multi-step registration process with payment integration.
2. **Role-Based Access:** Clear separation of permissions for users, nurses, and administrators.
3. **Seamless Workflows:** Enable collaboration between patients and healthcare providers.
4. **Automation:** Generate PDFs for forms and email delivery.
5. **Third-Party Integrations:** Scheduling, payment, and email services.
6. **Scalability and Compliance:** Support high user volumes and adhere to security standards.
7. **Logging**: Maintain detailed logs for all critical system actions for auditing and troubleshooting.

**Functional Requirements**

**1. User Registration Process**

**Step 1: Medical Information Form**

**Fields:**

* Personal Details: First Name, Last Name, Address, Postal Code.
* Medical Details: ID Number, Ailment, Ailment Notes.
* Confirmation: Terms acceptance checkbox.
* Digital Signature Field.

**Business Rules:**

* If an ailment is flagged as severe:
  + Automatically mark the subject line of emails as "Severe Case: [Ailment Name]".
  + Ensure the doctor is CC’d on the first email notification sent to the nurse.
* Enforce a validation process for all severe cases to include a mandatory note explaining the severity.

**Booking System:**

* Integrated with Calendly (or similar).
* Appointment intervals: 20 minutes.
* Availability: Monday to Saturday, 10:00 AM – 10:00 PM.
* Buffer times to prevent overlap.
* Send notifications via email for appointments.

**Step 2: Payment**

* Payment gateway: Local provider in South Africa, such as AddPay.
* Secure payment gateway integration to process registration fees.
* Trigger actions:
  + Convert registration form to PDF.
  + Email PDF to nurse and doctor (if marked severe).

**Step 3: Platform Access**

Unlock consultation form upon successful payment.

**Step 4: Registration Form Storage and Regeneration**

- Store completed registration forms in the database.

- Allow regeneration of the form as a PDF for both users and administrators.

**2. Consultation Form Workflow**

* **Editable Form:** Accessible by users and nurses.
* **Approval Process:**
  + Nurses approve and submit the form.
  + Upon approval:
    - Form becomes uneditable.
    - Registration and consultation forms merge into one PDF.
    - Email sent to the doctor, nurse notified.
  + Two weeks after submission, nurses receive the user's license to add to their account.
  + Upon adding the license, the user receives an email notification with the license attached.
  + Users must have the ability to download their license from their platform account.
  + Admins must track all licenses and view pending licenses for follow-up.

**3. Admin Functionality**

* Manual user registration without payment.
* Manage user access and accounts.
* Oversee bookings and submissions.
* Track and manage pending licenses.

**Logging and Monitoring**

**Logging Requirements**

**Event Logs:**

* User registration initiation and completion.
* Payment processing status (success or failure).
* Severe case flagging and notifications.
* Consultation form approval and merging.
* License issuance and addition to user accounts.

**Access Logs:**

* Role-based access actions, including admin overrides.

**Audit Logs:**

* Track all data changes for compliance, including form edits, approvals, and license updates.

**Email Templates**

**Registration Email**

**Subject Line:** "Welcome to Our Medical Platform – Registration Complete"

**Body:**

Dear [Patient Name],

Your registration has been successfully completed. Attached is a PDF copy of your registration form for your records. Our nurse team will review your submission and contact you soon.

Thank you,

[Platform Name]

**CC:** Nurse Email

**Attachment:** Registration Form PDF

**Severe Registration Email**

**Subject Line:** "Severe Case Notification: [Ailment Name]"

**Body:**

Dear [Patient Name],

Your registration has been flagged as a severe case. Our medical team has been notified and will prioritize your review.

Please find your registration form attached for reference. Our nurse and doctor will reach out to you for further steps.

Thank you,

[Platform Name]

**CC:** Nurse Email, Doctor Email

**Attachment:** Registration Form PDF

**Complete Consultation Email**

**Subject Line:** "Consultation Complete – Next Steps"

**Body:**

Dear [Patient Name],

Your consultation form has been successfully reviewed and approved. Attached is the finalized consultation and registration form for your reference.

The doctor will review your case and follow up as needed.

Thank you,

[Platform Name]

**CC:** Nurse Email, Doctor Email

**Attachment:** Merged Registration and Consultation Form PDF

**License Notification Email**

**Subject Line:** "Your License Has Been Issued"

**Body:**

Dear [Patient Name],

Your license has been successfully added to your account. You can download it directly from your profile. Please find a copy of your license attached for reference.

Thank you,

[Platform Name]

**Attachment:** License PDF

**Technical Requirements**

**Frontend**

* **Technology:** React.js
* **Features:**
  + User-friendly, responsive design.
  + Multi-step forms with real-time validation.

**Backend**

* **Technology:** Node.js with Express.js
* **Features:**
  + Secure API endpoints.
  + PDF generation and email handling.
  + Integration with external services (Calendly, payment gateway).

**Database**

* **Technology**: PostgreSQL
* **Structure:**
  + **Users Tabl**e: Stores user details, roles, and access levels.
  + **Registration Forms Table:** Tracks completed registration forms and allows regeneration.
  + **Customers Table**: Tracks registration details and license statuses.
  + **Audit Table:** Logs actions for compliance.
  + **Consultation Forms Table:** Records consultation data.
  + **Licenses Table:** Tracks licenses and their association with users.

**Example Database Schema**

**Users Table**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| id | INT | Primary Key |
| username | VARCHAR | User’s Login Name |
| name | VARCHAR | User’s Full Name |
| role | VARCHAR | Role (Admin/Nurse) |
| Email | VARCHAR | User’s Email Address |
| created\_at | DATETIME | Account Creation Date |

**Registration Forms Table**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| id | INT | Primary Key |
| user\_id | INT | Reference to User Table |
| form\_data | JSON | Registration Form Content |
| created\_at | DATETIME | Account Creation Date |
| updated\_at | DATETIME | Form Update Date |

**Customers Table**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| id | INT | Primary Key |
| name | VARCHAR | Full Name |
| email | VARCHAR | Email Address |
| created\_at | DATETIME | Account Creation Date |
| license\_status | BOOLEAN | License Verified (Y/N) |

**Consultation Forms Table**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| id | INT | Primary Key |
| customer\_id | INT | Reference to Customer |
| form\_data | JSON | Consultation Form Content |
| created\_at | DATETIME | Account Creation Date |
| approved\_by | INT | Nurse ID |

**Licenses Table**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| id | INT | Primary Key |
| user\_id | INT | Reference to User |
| license\_data | JSON | License Content |
| created\_at | DATETIME | Account Creation Date |
| added\_by | INT | Nurse ID |
| issued\_date | DATETIME | Date License was Issued |

**User Roles and Permissions**

**1. User (Patient)**

* Complete registration.
* Book appointments.
* Access and edit consultation forms until approval.
* Receive and download license when issued.

**2. Nurse**

* Review and edit forms.
* Approve consultation forms.
* Trigger form merging and email submission to doctors.
* Add licenses to user accounts.
* Receive reminders for pending licenses.

**3. Admin**

* Register users manually.
* Manage accounts and permissions.
* Access system logs and analytics.
* Monitor and follow up on pending licenses.

**Workflow Summary**

1. **Registration**:
   1. User fills out the medical information form and books an appointment.
   2. Completes payment and receives access to the consultation form.
2. **Consultation:**
   1. User collaborates with the nurse to finalize the consultation form.
3. **Approval:**
   1. Nurse approves the form, triggers merging and email to the doctor.
   2. Severe cases ensure doctor is CC’d in all related communications.
4. **License Issuance:**
   1. Two weeks post-consultation, nurse receives the user's license.
   2. Nurse adds license to the user's account.
   3. User receives an email with the license and can download it.
   4. Admin tracks all pending and issued licenses for oversight.

**Deliverables**

1. Fully functional web application.

2. Integration with third-party tools.

3. Documentation for user roles, workflows, and admin functionalities.

**Appendix**

**Expanded Information for Client and Developer**

* Appointment intervals of 20 minutes to manage schedules effectively.
* Local South African payment gateway (AddPay) for secure transactions.
* Business rules for scheduling include preventing double bookings and ensuring timely reminders.
* Stored registration forms provide a way to regenerate PDFs for compliance or review.
* Severe cases flagged during registration automatically ensure doctor’s involvement from the first notification.
* License management system integrated to track pending and issued licenses efficiently.
* Logging ensures that all critical system interactions are traceable and compliant with auditing requirements.