1. **Database Design**:
   * You'll need a database to store patient information, appointment details, and any associated notes. Consider entities like Patient, Appointment, and Payment.
   * Each patient should have a unique identifier and fields for basic information like name, mobile number, and email.
   * Appointments should be linked to patients and contain information like appointment date, time, notes, and a reference to any associated payment.
2. **User Interface**:
   * Create interfaces for users to interact with the system. This includes forms for adding patients and appointments, as well as a dashboard for viewing patients and their appointments.
   * Implement a search functionality on the dashboard to allow users to easily find patients by name.
3. **Functionality Implementation**:
   * **Create Patient**: Implement a form where users can input basic patient information and save it to the database.
   * **List Patients**: Develop a dashboard that displays a list of patients retrieved from the database.
   * **Search Functionality**: Enable users to search for patients by name, filtering the results dynamically.
   * **View Patient**: Create a detailed view for each patient, showing all their information and any linked appointments.
   * **Create Appointment**: Allow users to schedule appointments for patients, associating them with the respective patient in the database. Ensure that appointments are displayed on the patient's detail view.
   * **Stripe Integration**: Integrate with the Stripe API to generate payment links for each appointment. When creating an appointment, generate a unique payment link and store it in the database. Provide users with the ability to access these payment links when viewing appointments.
4. **Stripe Integration**:
   * Sign up for a test account with Stripe to get API keys for integrating payment functionality.
   * Implement logic to generate a payment link for each appointment using the Stripe API.
   * Store the generated payment link in the database along with the appointment details.
5. **Testing and Deployment**:
   * Thoroughly test the system to ensure all functionalities work as expected.
   * Deploy the system to a suitable environment, ensuring that it's secure and accessible to users.
6. **Documentation and Support**:
   * Provide comprehensive documentation for users and developers on how to use the system and its various functionalities.
   * Offer support channels for users to report issues and seek assistance.

By following these steps, you can develop a Patient Appointment System that meets the outlined requirements and integrates Stripe for payment processing.