Second Opinion App - Architecture Document

1. Application Architecture

Second Opinion follows a **Microservices Architecture** to ensure modularity and scalability. The system is divided into the following services:

- **User Service**: Handles authentication, user profiles, and medical history preferences.
- **Medication Information Service**: Manages the database of medications, their effects, interactions, and usage guidelines.
- **Consultation Service**: Processes user queries about medications and provides information.
- **Notification Service**: Sends medication reminders and important health alerts.
- **API Gateway**: Manages requests between the frontend and backend services.

2. Database

- **ER Diagram**: Represents relationships between users, medications, consultations, and reminders.

  [Placeholder for ER Diagram showing relationships between users, medications, interactions, and consultations]

- **Schema Design**:
  - **Users** (id, name, email, password_hash, age, medical_conditions, allergies, current_medications)
  - **Medications** (id, name, generic_name, category, usage, side_effects, interactions, contraindications, dosage_info)

    - **UserMedications** (id, user_id, medication_id, prescribed_date, dosage, frequency, notes)
    - **Consultations** (id, user_id, medication_id, question, response, created_at)
    - **Reminders** (id, user_medication_id, reminder_time, notification_sent)

3. Data Exchange Contract

- **Frequency of data exchanges**:
    - Medication information is updated regularly from trusted pharmaceutical databases.
    - User consultations are processed in real-time.
    - Medication reminders are scheduled according to user preferences.
- **Data Sets**:
    - User data (for authentication, medical history, and preferences)
    - Medication data (comprehensive information about medicines)
    - Consultation data (user queries and system responses)
- **Mode of Exchanges**:
    - **API**: RESTful APIs for medication information and user management.
    - **WebSockets**: For real-time consultation responses.
    - **Push Notifications**: For medication reminders and important alerts.

4. Deployment Architecture

- **Frontend**: React Native mobile app and React-based web application.
- **Backend**: Node.js (Express.js) hosted on AWS Lambda or DigitalOcean.

- **Database**: PostgreSQL for structured medical data with MongoDB for unstructured consultation data.

- **Authentication**: JWT-based authentication with additional security for healthcare data.

- **Storage**: HIPAA-compliant cloud storage for user medical information.

- **Third-party Integrations**: APIs for medication databases (like RxNorm, FDA, etc.)

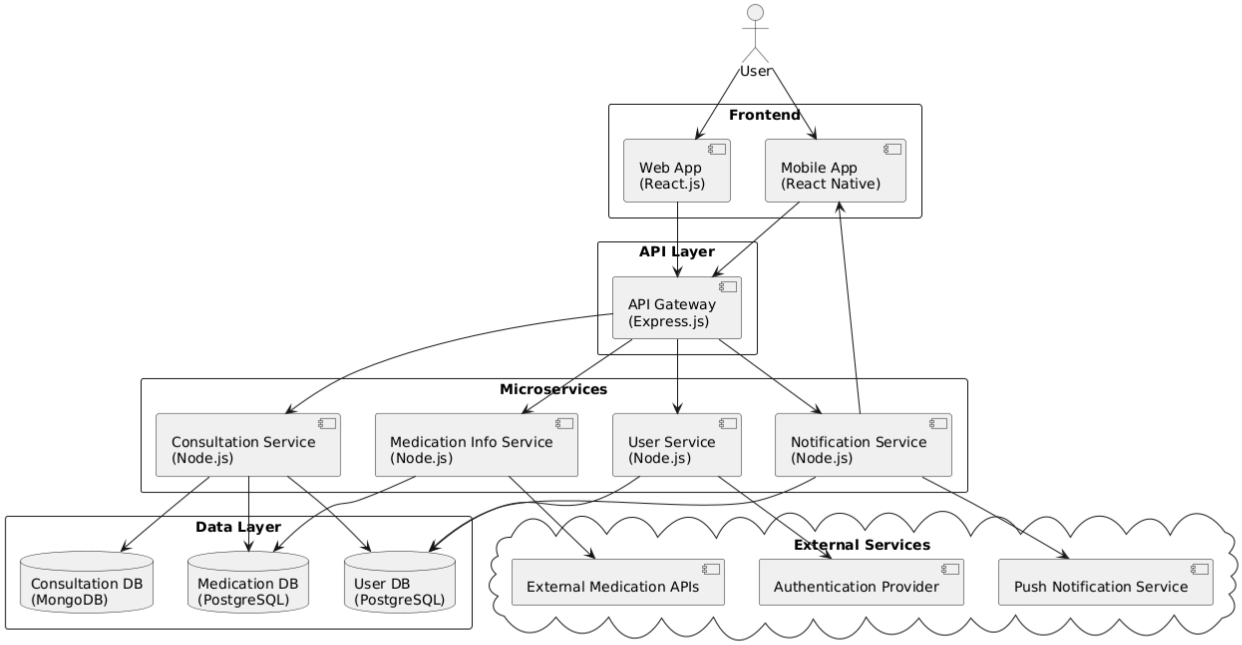
5. Security and Compliance

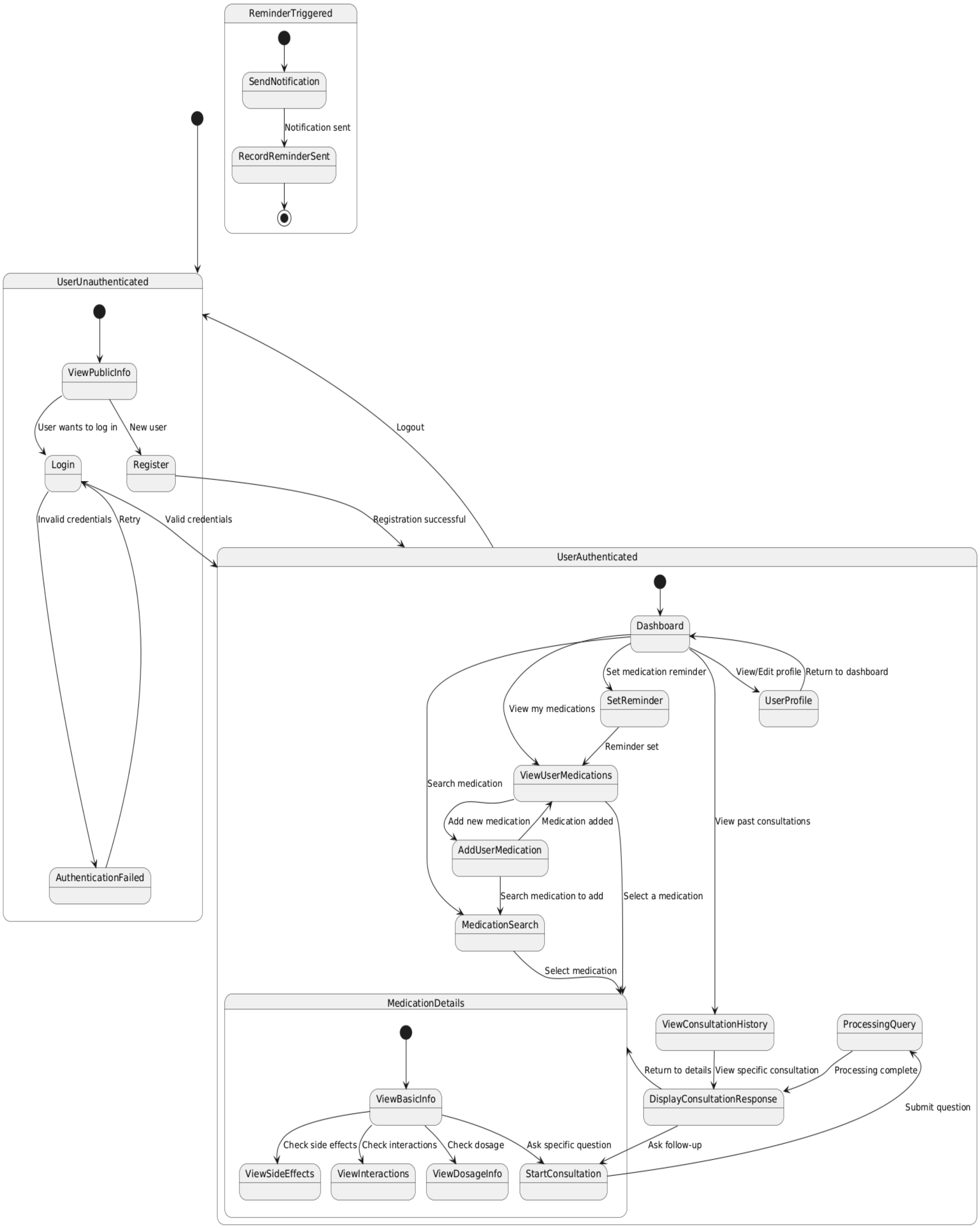
- HIPAA compliance for handling sensitive medical information
- End-to-end encryption for all user data
- Regular security audits and penetration testing
- Data anonymization for analytics purposes

UML Diagrams

Sequential Diagram for Medication Consultation Flow

[Placeholder for Sequential Diagram showing the flow from user query about a medication to system response with relevant information]




ReminderTriggered
SendNotification
Notification sent
RecordReminderSent
UserUnauthenticated
ViewPublicInfo
User wants to log in
New user
Login
Register
Invalid credentials
Retry
Valid credentials
Registration successful
Logout
AuthenticationFailed
UserAuthenticated
Dashboard
Set medication reminder
View/Edit profile
Return to dashboard
SetReminder
UserProfile
View my medications
Reminder set
ViewUserMedications
Search medication
Add new medication
Medication added
View past consultations
AddUserMedication
Search medication to add
Select a medication
MedicationSearch
Select medication
MedicationDetails
ViewConsultationHistory
ProcessingQuery
Return to details
View specific consultation
Processing complete
ViewBasicInfo
DisplayConsultationResponse
Submit question
Check side effects
Check interactions
Check dosage
Ask specific question
Ask follow-up
ViewSideEffects
ViewInteractions
ViewDosageInfo
StartConsultation