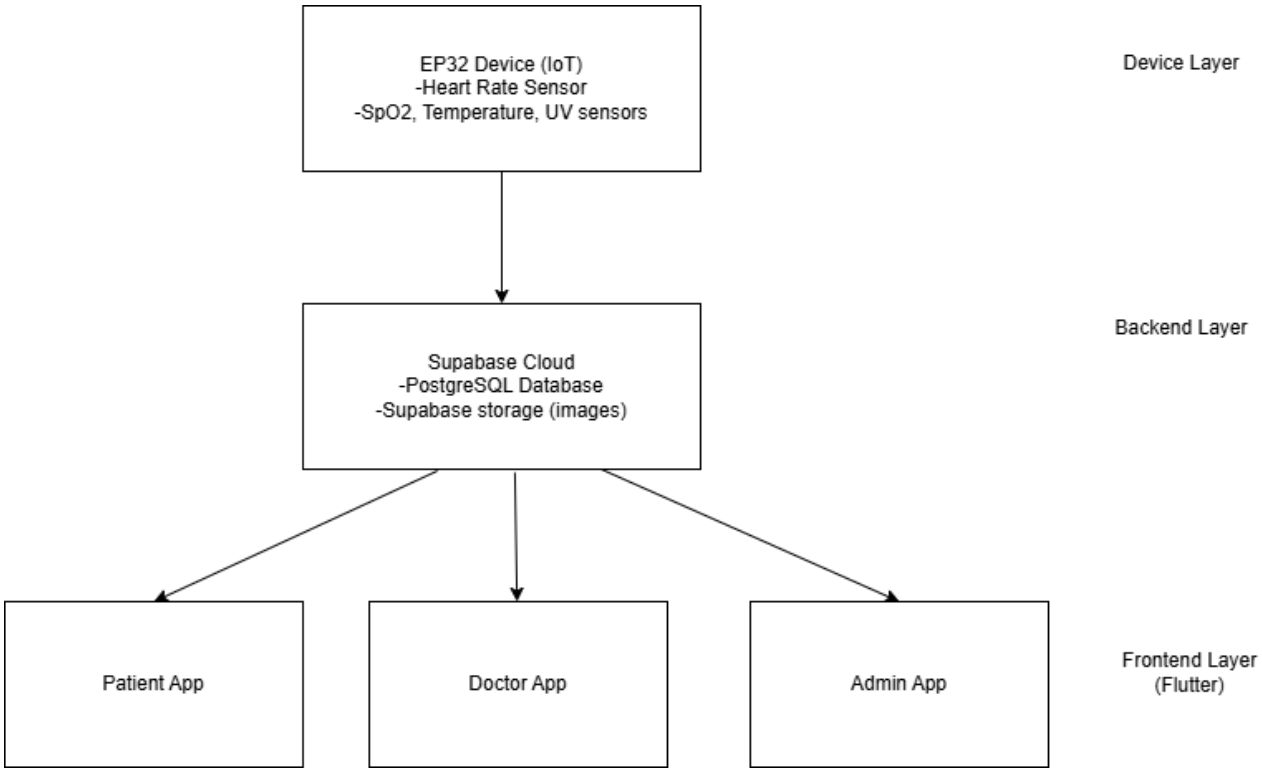
2. SYSTEM ARCHITECTURE AND DESIGN:

2.1. HIGH LEVEL ARCHITECTURE:

2.1.1. 3-Tier CLIENT SERVER ARCHITECTURE:



The HealthLab system follows a three-tier IoT-integrated client-server architecture, consisting of a wearable device layer, a cloud backend, and a mobile frontend. This structure ensures real-time health data collection, secure cloud storage, and user-friendly access through a mobile application.

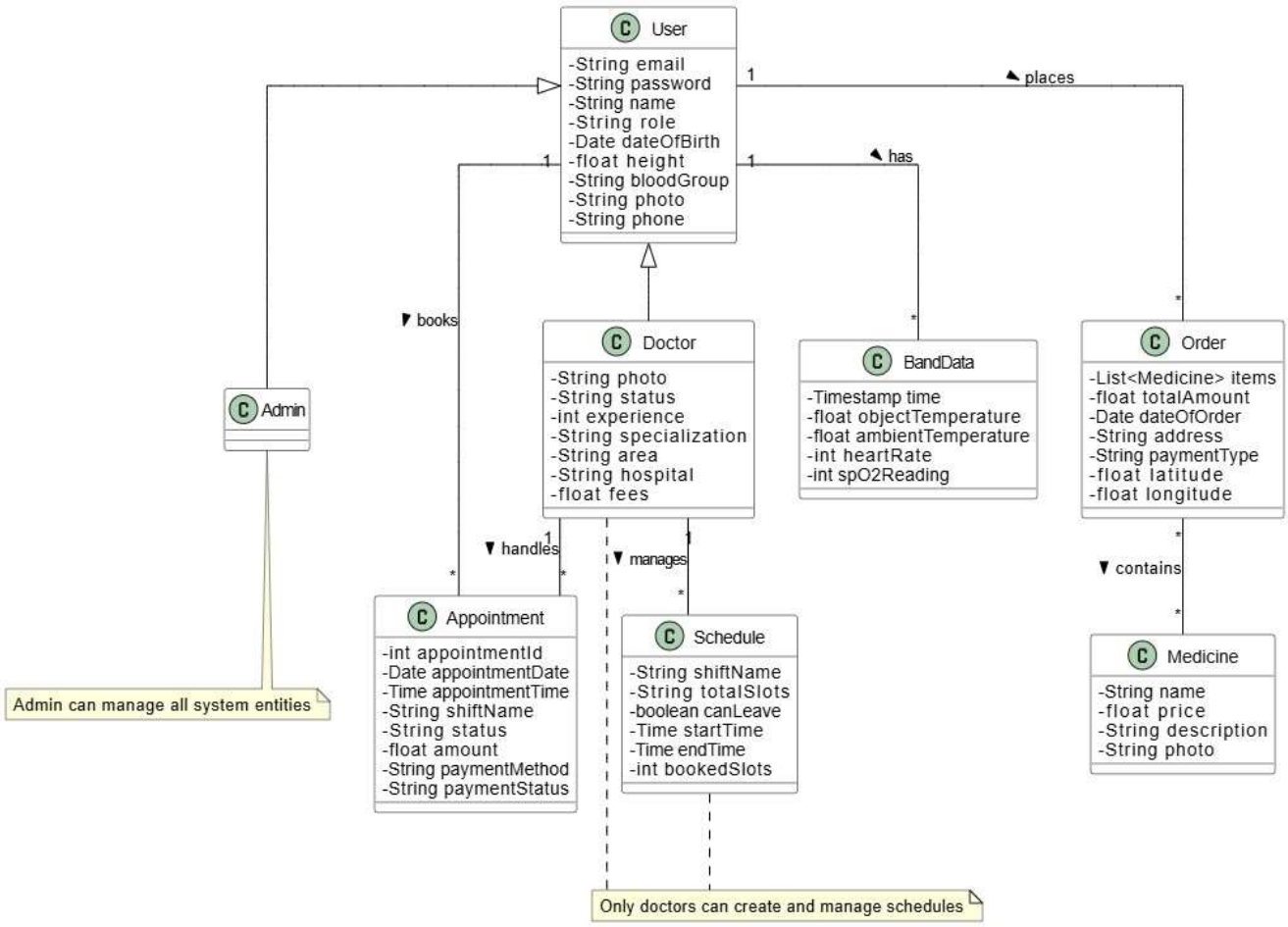
- **Device Layer (IoT / Edge Layer)**:
  At the core of the system is an ESP32-C3 microcontroller integrated with multiple health sensors, including:
  - Heart rate and SpO_2 sensor (MAX30102)
  - Temperature sensor (MLX90614)
  - UV Index sensor
    The device collects real-time health metrics and transmits the data to the cloud using HTTP POST requests over Wi-Fi, formatted as JSON.

- **Backend Layer (Cloud Services)**:
  The backend is powered by Supabase, a platform offering:
  - A PostgreSQL database for storing all application data (users, appointments, orders, etc.)
  - Authentication with role-based access control (Patients, Doctors, Admins)
  - File storage for profile images and product pictures
    Supabase ensures real-time synchronization and scalability.

- **Frontend Layer (Mobile Application)**:
  Built using Flutter, the mobile application provides tailored experiences for each user role:
  - Patients can monitor vitals, book appointments, and order medicines.
  - Doctors can manage schedules, appointments, and view patient data.
  - Admins can oversee users, manage doctors and pharmacy inventory.
    The app interfaces with Supabase via secure API calls, fetching and updating data in real time.

  This layered architecture ensures separation of concerns, promotes maintainability, and supports future scalability such as BLE integration or cloud-based analytics.

2.2. CLASS DIAGRAM:



ASSUMPTIONS:

User Role Generalization:

- The User class acts as a superclass for Doctor and Admin.
- A user can only have one role at a time (Patient, Doctor, or Admin).
- Every Doctor and Admin is also a User and shares base user attributes.

Band Data Monitoring:

- Each User can have multiple BandData records (1:N relationship).
- Band data includes biometric and environmental parameters.
- Data is time-stamped and uniquely associated with the user.

Appointment Booking:

- A User (as a patient) can book multiple Appointments.
- Each Appointment must be linked to both a Doctor and a Schedule.

- Appointments contain payment, timing, and status information.

Doctor Specialization:

- The Doctor class extends User and includes additional attributes like:
- Specialization, hospital, experience, and consultation fees.
- Each Doctor can manage multiple Schedules.

Schedule Management:

- Only Doctors can create and manage Schedules.
- A Schedule contains time slots and availability for appointments.
- A schedule may span multiple days or shifts.

Admin Privileges:

- Admin is a specialized user role with the ability to:
- Manage all users, appointments, orders, and pharmacy listings.
- Admin does not have additional attributes beyond User, but has system-level access.

Order Placement:

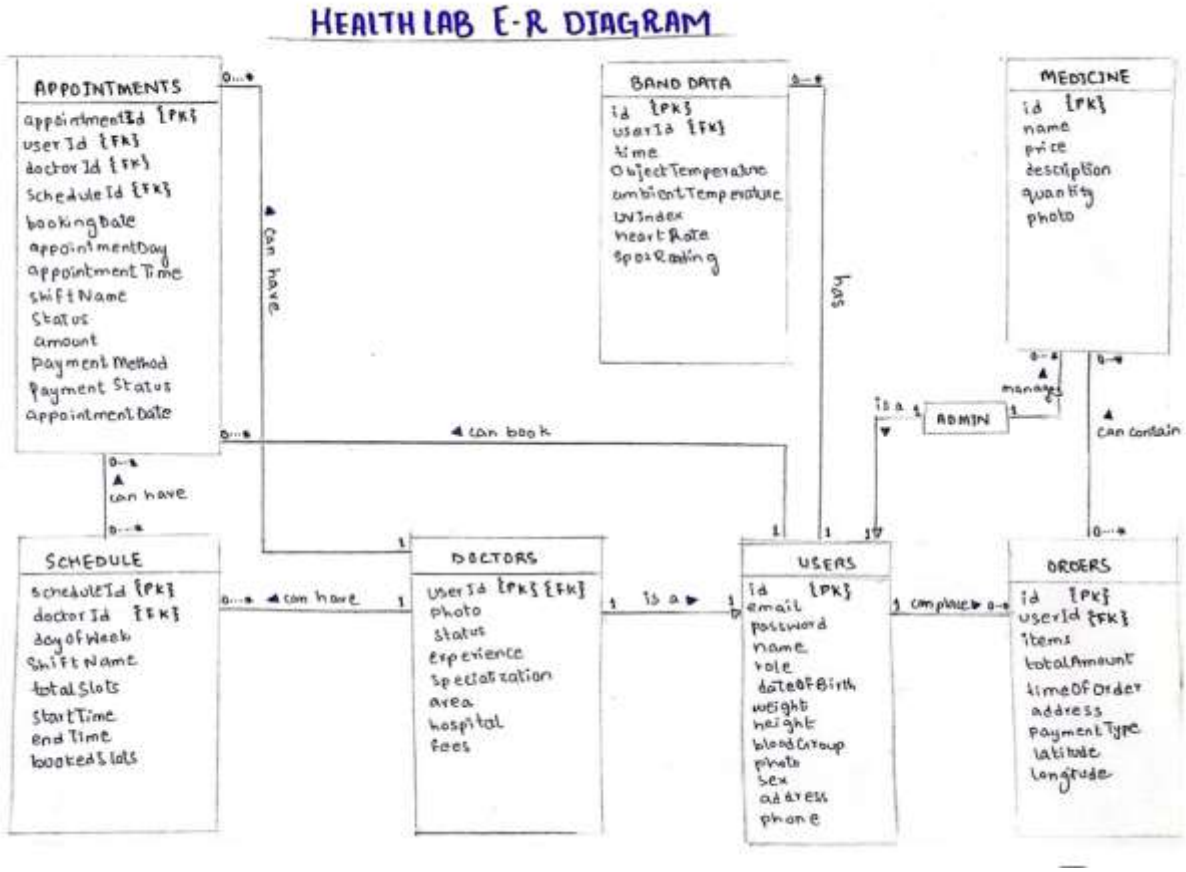
- Each User (as a patient) can place multiple Orders.
- Orders contain delivery and payment details including GPS location.
- An Order can contain multiple Medicine items (M:N relationship).

Medicine Management:

- Medicine entities store product catalog information: name, price, description, and image.
- Medicines are linked to orders through a "contains" relationship.

2.3. DATABASE SCHEMA:

2.3.1. ERD:



ASSUMPTIONS:

Users:

- Each user is uniquely identified by id.
- Users can be Patients, Doctors, or Admins, determined by the role attribute.
- Authentication is managed using email, password, and additional identity details such as phone and address.
- A user may:
  Book multiple appointments (if patient),
  Manage schedules and write prescriptions (if doctor),
  Or perform system-level operations (if admin).
- A user can have multiple Band Data entries for biometric tracking.
- A user can place multiple Orders for medicines through the pharmacy module.

Band Data:

- Each Band Data record is uniquely identified by id.
- Linked to a specific user via the userId foreign key.
- Stores real-time biometric metrics collected via wearable devices: objectTemperature, UVIndex, heartRate, SpO2Reading, and time of collection.
- A user can have zero or more Band Data records (1:N relationship).

Doctor:

- A Doctor is a specialized type of user, uniquely identified by userId (also the primary key and a foreign key to the Users entity).
- Each doctor may have multiple Schedules and Appointments.
- Stores attributes such as: photo, status, experience, specialization, area, hospital, and fees.

Admin:

- An Admin is also a specialized user, with userId as the primary and foreign key.
- Admins are responsible for system-level oversight (e.g., managing users, pharmacies, or orders).
- No additional attributes are shown in the EERD, but system privileges are implied by their role.

Appointment:

- Each appointment is uniquely identified by appointmentId.
- Linked to a userId (patient), doctorId, and scheduleId via foreign keys.
- Stores: bookingDate, appointmentDate, appointmentDay, appointmentTime, shiftName, status, amount, paymentMethod, and paymentStatus.
- A user can book multiple appointments, and a doctor can handle multiple appointments.

Schedule:

- Each schedule is uniquely identified by scheduleId.
- Belongs to one doctor (linked via doctorId FK).
- Defines the doctor's availability using: dayOfWeek, shiftName, startTime, totalSlots, and bookedSlots.
- A doctor can have multiple schedules (1:N relationship).

Medicine:

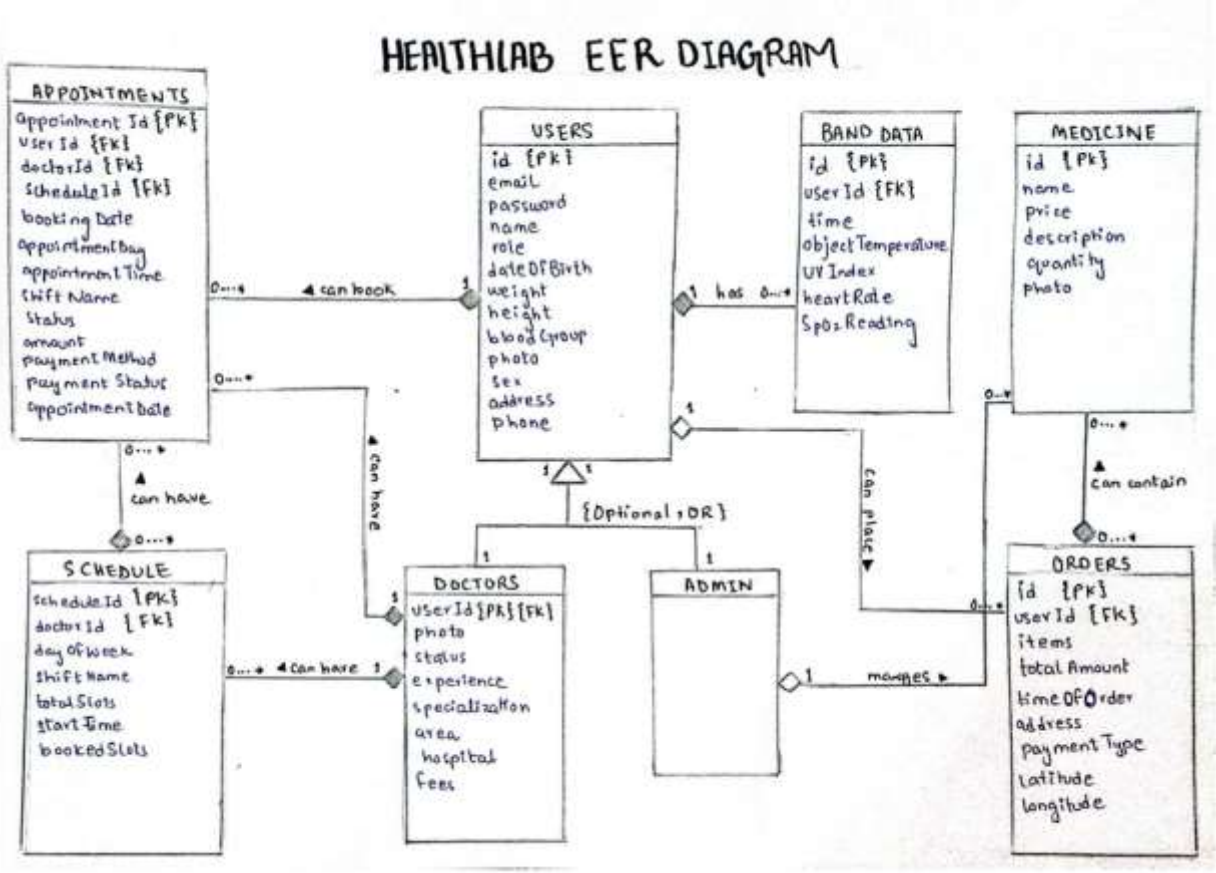
- Each medicine is uniquely identified by id.

- Stores data including name, price, description, quantity, and photo.
- Medicines can be ordered through the Orders module and linked to multiple orders (M:N relationship).

Orders:

- Each order is uniquely identified by id.
- Linked to a user (patient) via userId (FK).
- Stores: items, totalAmount, timeOfOrder, address, paymentType, latitude, and longitude.
- A user can place multiple orders, each containing multiple medicines (M:N relationship with Medicine).

2.3.2. EERD:



ASSUMPTIONS:

Generalization (User):

- The entity User acts as a general superclass for two specialized entities: Doctor and Admin.
- The generalization is:
- Disjoint: A user can be either a Doctor or an Admin, not both at the same time.
- Partial: Not all users need to be doctors or admins — some users may simply be Patients (default role).
- Attributes unique to each subclass (e.g., specialization and fees for Doctors) are stored in the respective specialized entities and not in the base User table.

Band Data Relationship:

- The relationship between User and Band Data is 1:N (one-to-many), with total participation on the Band Data side.
- Each Band Data record must belong to exactly one user.
- A user may have zero or more associated Band Data entries (e.g., from a wearable device).
- This relationship enables continuous health tracking per user.

Appointment Booking:

- A User (in the role of a Patient) can book multiple Appointments (0:N).
- Each Appointment must be associated with one user and one doctor.
- Additionally, each appointment must be linked to an available Schedule.
- The participation of Appointment in all three roles (User, Doctor, Schedule) is total, meaning an appointment cannot exist without being associated with all three.

Schedule Assignment:

- Each Doctor can have multiple Schedules (1:N).
- Each schedule is associated with one and only one doctor.
- Participation of Doctor in the Schedule relationship is total, implying every schedule must be owned by a doctor.
- A Schedule includes metadata such as day, time, and slot availability

Order Placement:

- A User (typically a Patient) can place multiple Orders (0:N).
- Each Order must belong to one user, indicating total participation from Order to User.
- Orders track delivery information, payment method, and timestamp.

Medicine Inclusion in Orders:

- The relationship between Orders and Medicine is many-to-many (M:N).
- A single order can include multiple medicines, and a single medicine can appear in multiple orders.

Admin Supervision:

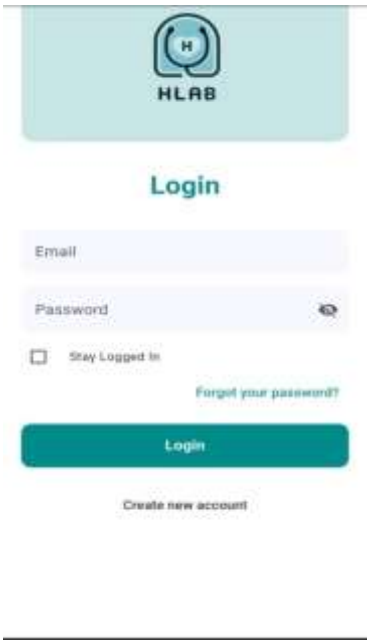
- An Admin is a specialized type of user who can manage system operations.
- The EERD shows that the Admin entity is directly linked to the User entity with a 1:1 specialization relationship.

2.4. USER INTERFACES:

The mobile application is designed with distinct interfaces for three types of users: **Patients**, **Doctors**, and **Admins**. Each user type is provided with relevant screens to ensure a seamless and efficient experience.

Splash Screen:

Login Screen:



2.4.1. **Patient Screens**:

Patients can access multiple features through a user-friendly interface:

- **Health Monitoring Screen**

Displays real-time readings of heart rate, SpO_2, body temperature, room temperature, and UV index fetched from the ESP32 device.

- **Available Doctors Screen**

Shows a list of all registered doctors available for consultation, including their specialization and availability.

- **Profile Screen**

Allows the patient to view and edit personal information such as name, email, phone number, and address.

- **Pharmacy Screen**

Lists available medicines for purchase. Patients can add medicines to their cart and place orders directly through the app.

Medicine Details

Medicine Catalog

- **Order Placement Screen**

Cart Screen

Checkout Screen

Order Receipt

- **Purchase History and Details**

Purchase History

Purchase Details

- **Charts Screen**

Visualizes the patient's health data (like heart rate, temperature, etc.) using line or bar charts for easy tracking over time.

- **Doctor Details**

2.4.2. Doctor Screens:

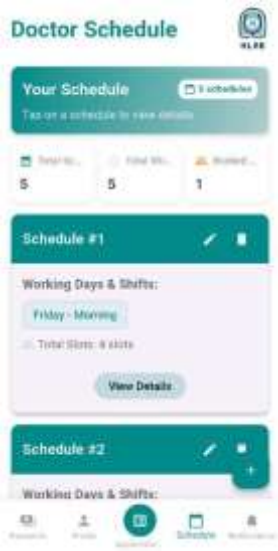
Doctors are provided with tools to manage appointments and monitor interactions with patients:

- **Appointments Screen**

Displays a list of booked and pending appointments. Doctors can accept or reject request

- **Schedule Screen**

Enables doctors to create or update their available time slots for appointments.



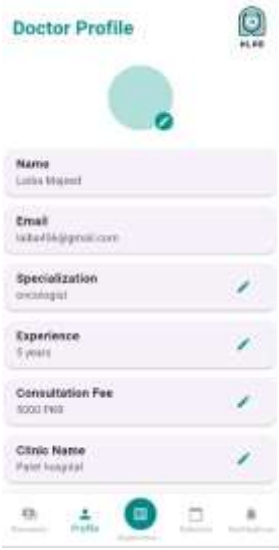
Schedule Screen Schedule Details Add New Schedule

- **Notifications Screen**

Shows updates from the admin and notifications when patients book appointments

- **Profile Screen**

Allows the doctor to edit their professional profile, including specialization, availability, and credentials.



- **Payment Screen**

Displays details of transactions received from patients and payouts to the admin.

2.4.3. Admin Screens:

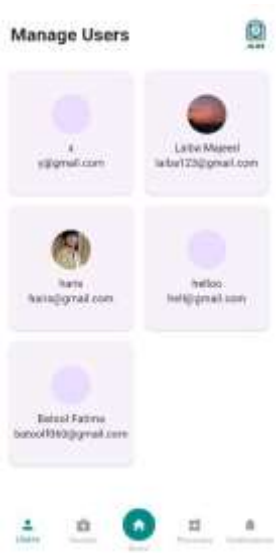
Admins manage the overall system, including users, doctors, and medicine inventory:

- **Dashboard (Home)**

Shows key metrics such as total users, number of doctors, appointment stats, and system notifications.

- **Manage Users Screen**

Lists all registered patients along with their basic details.

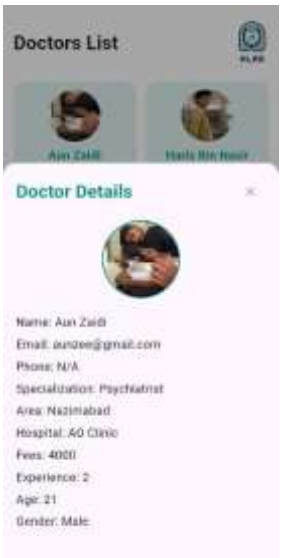


Registered Users List Screen

User Details Screen

- **Doctors Screen**

Displays all registered doctors and allows the admin to verify or remove doctors from the system.

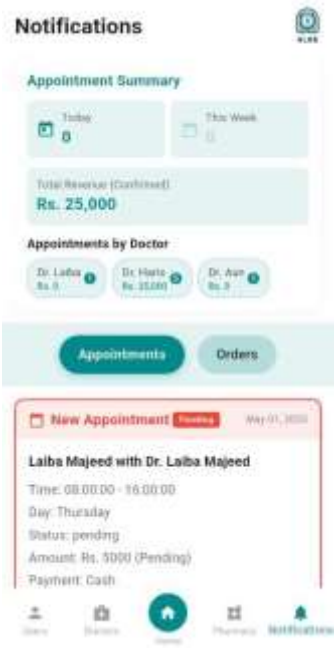


Registered Doctors List Screen

Doctor Details Screen

- **Notifications Screen**

Shows all system-level notifications related to medicine orders and booked appointments.



- **Manage Pharmacy**

Enables the admin to add, update, or remove medicine listings in the app.

Add New Item in Pharmacy

Pharmacy Screen

Item Details Screen