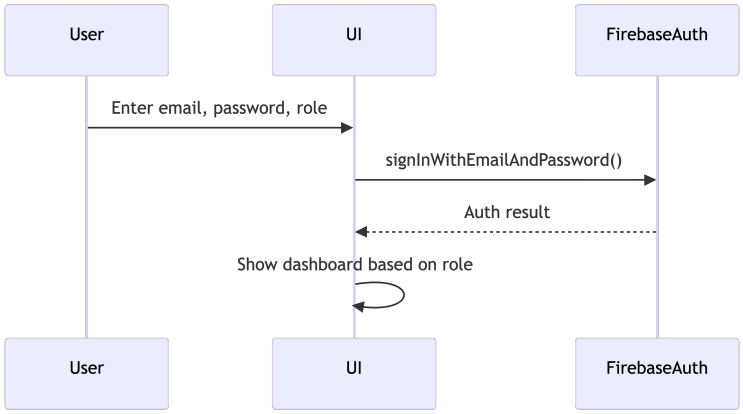
Low Level Design (LLD)

Modules

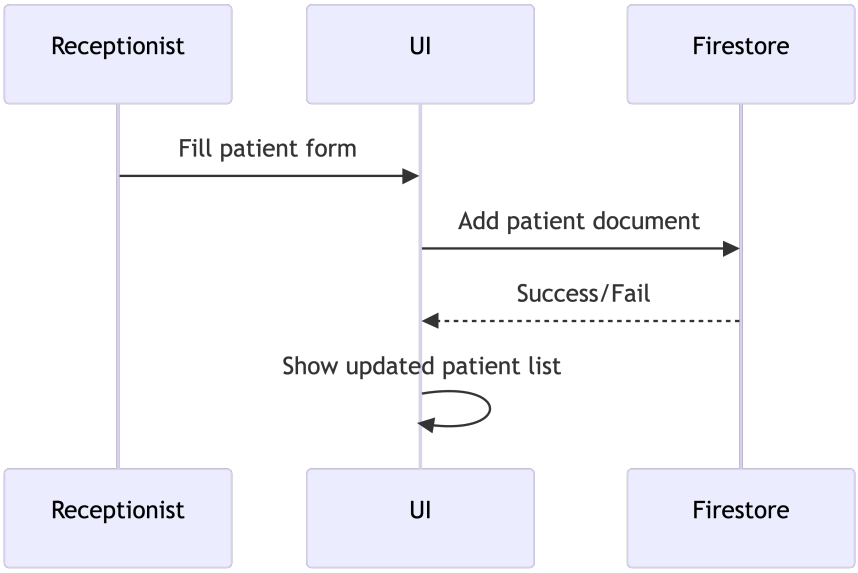
- Authentication (Doctor, Receptionist)
- Token Management
- Patient Management
- Prescription Management
- Billing
- Logging

Sequence Diagrams

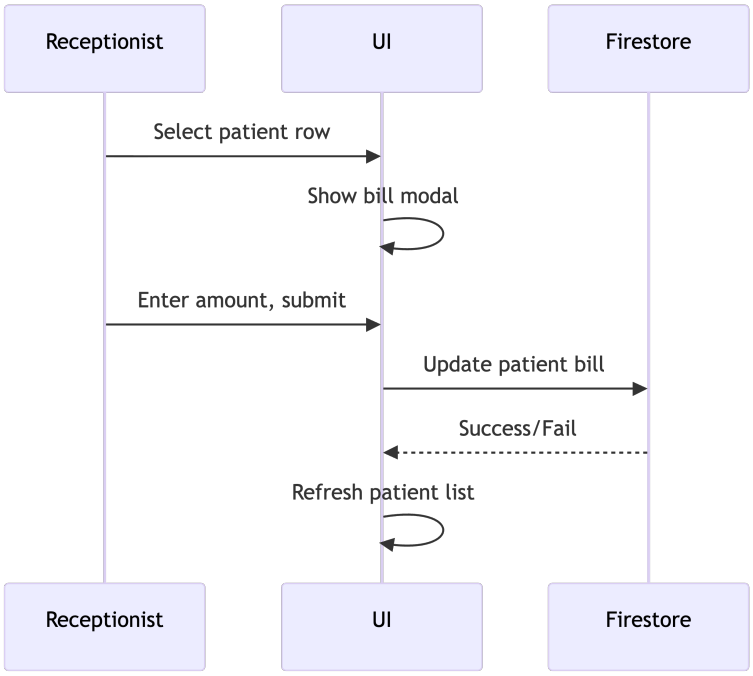
1. Login Flow



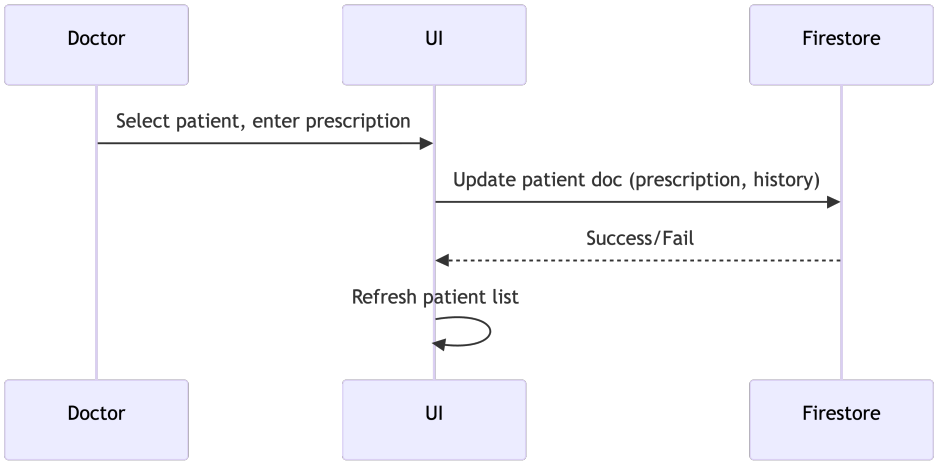
2. Add Patient (Receptionist)



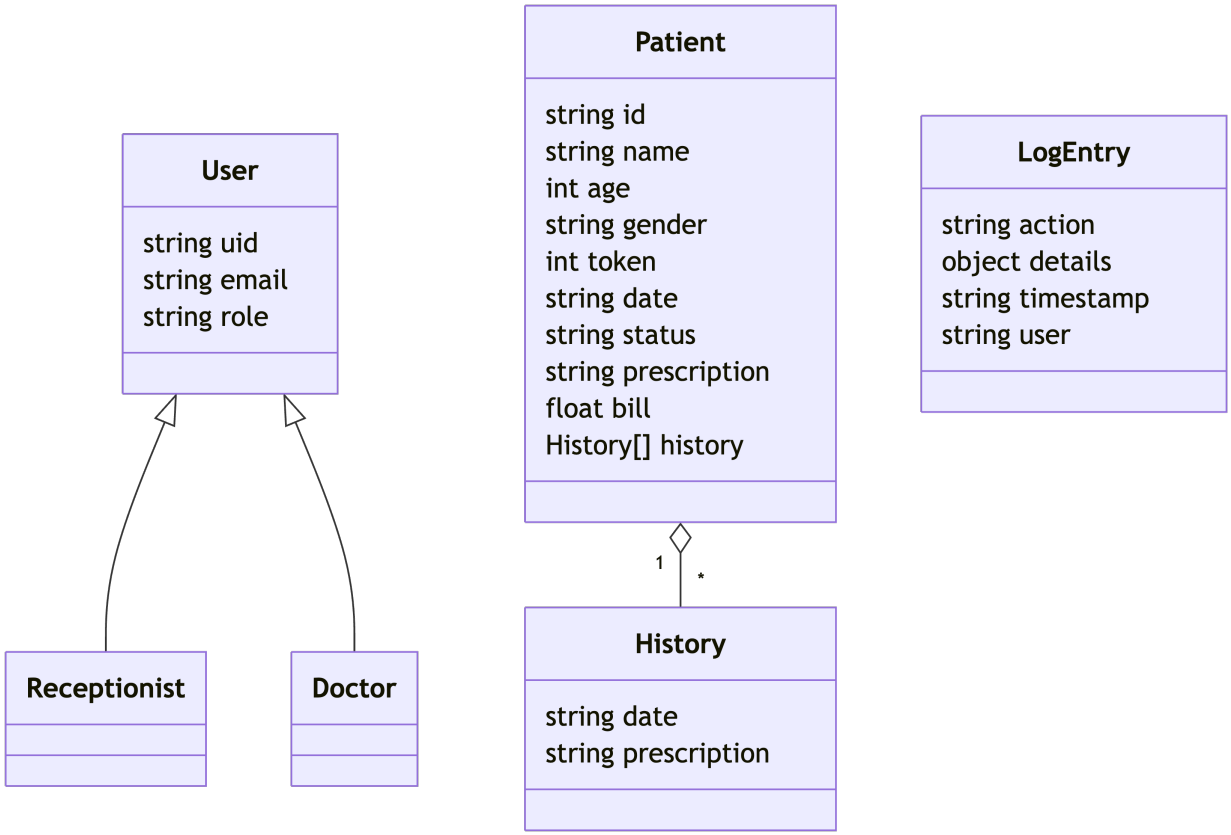
3. Generate Bill (Receptionist)



4. Add/Edit Prescription (Doctor)



Class Diagrams



Detailed Logic

Authentication

- Uses Firebase Auth (email/password)
- On login/register, user role is checked from Firestore users collection
- UI updates based on role (doctor or receptionist)

Token Management

- Each patient gets a unique token for the day
- Token = max(token) for today + 1

Patient Management

- Receptionist adds patient (name, age, gender)

- Patient is stored in Firestore with token, date, status, and empty history
- Patient list is shown in a table, updated in real time after adding.

Prescription Management

- Doctor can check a patient and add a prescription
- Doctor can edit prescription for any patient
- Each prescription is appended to the patient's history array

Billing

- Receptionist selects a patient row to open bill modal
- Enters amount, which updates the patient's bill field in Firestore
- Bill info is shown in the patient list

Logging

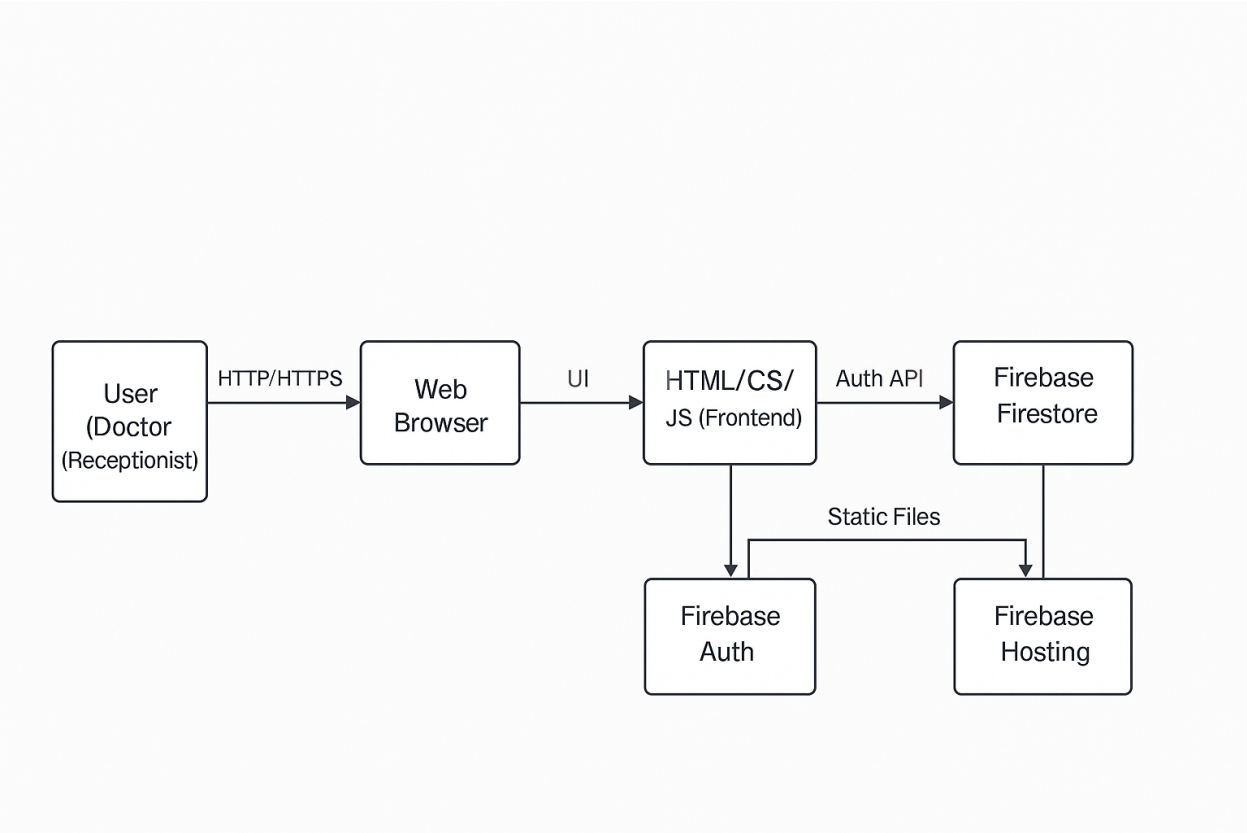
- All actions (login, add patient, bill, prescription, etc.) are logged to Firestore logs collection
- Log includes action, details, timestamp, and user

System Architecture

Overview

- Frontend: HTML, CSS, JS
- Backend: Firebase (Auth, Firestore)
- Hosting: Firebase Hosting

Architecture Diagram



1. **User Interaction**
    - The **Doctor or Receptionist** accesses the system via a **Web Browser**.
    - They initiate actions like login, adding patients, viewing dashboards, etc.
2. **Frontend Rendering**
    - The browser loads the **HTML/CSS/JS frontend** hosted on **Firebase Hosting**.
    - This UI provides forms, tables, and dashboards for user interaction.

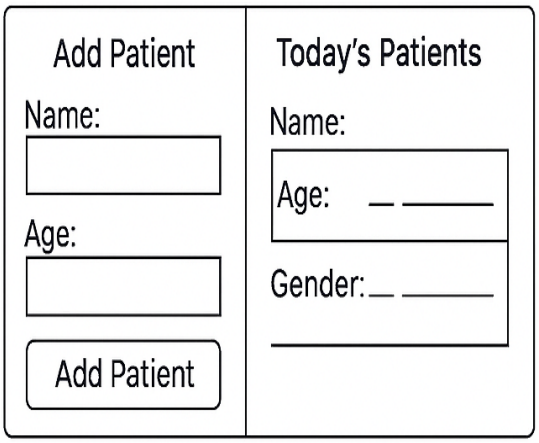
3. **Authentication**

- When a user logs in, the frontend calls **Firebase Auth** using `signInWithEmailAndPassword()` or similar methods.
- Firebase Auth verifies credentials and returns an authentication token.

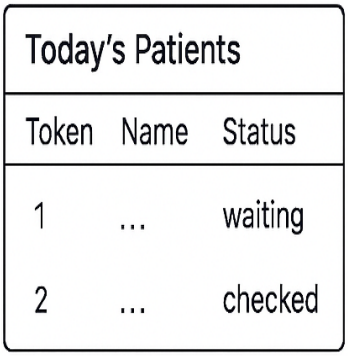
4. **Data Operations**
    - Once authenticated, the frontend interacts with **Firebase Firestore** to:
        - Add or update patient records
        - Fetch today's patient list
        - Update prescriptions or billing info
5. **Hosting & Static Files**
    - The frontend files (HTML, CSS, JS) are served from **Firebase Hosting**.
    - This ensures fast and secure delivery of the UI components.
6. **Backend Communication**
    - **Firebase Auth** and **Firestore** may communicate internally to validate access rules and user roles.
    - This ensures that only authorized users can perform specific actions (e.g., only doctors can edit prescriptions).

Wireframes

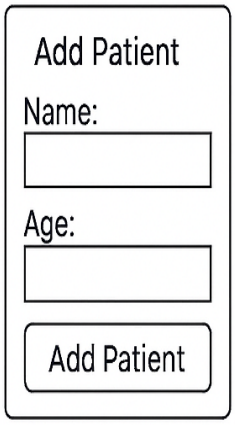
Receptionist Dashboard (Desktop)



Today's Patients



Mobile Layout (Recentionist/Doctor)



Mobile Layout (Receptionist/ Doctor)