Implementation of TakeCare App

Front-End

The list of implemented Interfaces Include

- **Dashboard:** Each user type has a main interface upon login, catered to its specific functionalities
    - **Patient Dashboard:** It contains Prescription List, QR tab (for viewing QR)**,** reports Tab, and a button for scanning QR.
    - **Doctor Dashboard:** It prompts to make prescription.
    - **Chemist Dashboard:** It immediately shows QR code for patient scanning
    - **Pathologist Dashboard:** It contains a queue for patients so the reports can be given and a button to scan patient QR for adding Patient in queue.
- **Patient:** Dedicated interfaces for patient include
    - **View Prescription:** It includes a detailed description of the prescription, as well as the facility to delete, set or reset alarms for the appropriate prescription timings
    - **Buy Prescription:** It has the facility for scanning chemists QR to buy a specific prescription.
    - **View Report:** This interface provides the link to online uploaded report by the pathologist website.
    - **Scan Prescription:** It provides facility to scan prescription QR and save it.
- **Doctor:** Dedicated interfaces for patient include
    - **Make Prescription View:** It includes a nicely guided interface to create a prescription and generate the Prescription's UNIQUE QR, so patient can scan it.
- **Chemist:** Dedicated interfaces for patient include
    - **QR view:** For scanning by patient.
- **Pathologist:** Dedicated interfaces for pathologist include
    - **Scan QR:** for scanning incoming patient QR to add them to queue.
    - **Give Report View:** For giving a link to the report to patient in the queue.
- **Registration:** the registration process follows a 3-Phase interface design:
    - **Phase I:** User registration, gathering general user information
    - **Phase II:** User Type, at this stage, user tells its type from the above mentioned 4 user classes.
    - **Phase III:** This contains forms gathering information specific to each user type
- **Sign in:** This follows a regular implementation of sign in design asking username and password.

Back-End

Client-Side (Phone) Back-End

- **Form Validations:** All form fields filled by every user are validated on the client side through REGEX utility of Java.
- **QR code scanner:** All the QR code scanning is built locally into the app with the help of Google Vision API, requiring Android 5.0 or higher with 1 GHz processer. Camera permissions are not handled left to vendor specific implementations in the device itself.
- **QR code generation:** the generation of QR code is handled with the open source “zxing” library adhering to industry standard QR code scanning
- **Server Connection:** The app connects to database server through java.net interface
- **Alarms:** Patient dose alarms are implemented natively via Android Broadcast receiver, and are offline and local to each device.

Server-Side Back-End

- The server requires installation of Apache-2.6, PHP 5.7, and MySQL for functioning (PROCEDURE DESCRIBED IN README FILE)
- All client side requests are handled by PHP, which acts as an interface between client and the database on the server.
- For this purpose, a custom API has been written completely in PHP for the server side implementation called “database.api” in server code.
- The provides implemented functions for getting, setting and deleting elements in the database.
- Hence the maximum number of simultaneous users is purely limited server request handling capacity and not the API itself.
- The communication happens via standard http GET requests under SSL encryption.

GUI

The front end GUI interfaces of the TakeCare App are implemented using the official Android API (level 21) requiring android version greater than or equal to 5.0. Interfaces are built adhering to Android design paradigms having being designed in the “Layout Resources” of the API in XML format.

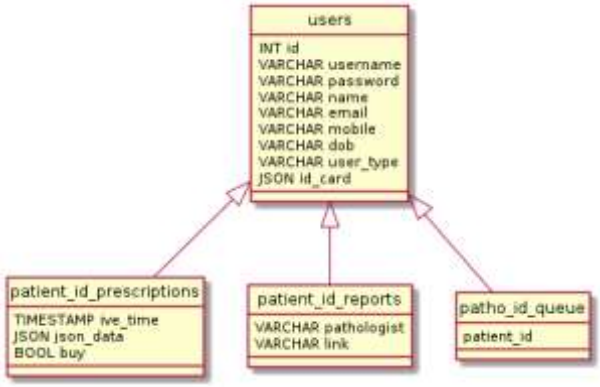
Only native interface elements were used to ensure wider compatibility. These elements include

- Android TextView (Textboxes)
- Android EditText (Input fields)
- Android Button
- Android Spinner (Dropdown menus)
- Android SurfaceView (For Mapping camera input)
- Android ImageView (For displaying generated QR)

All front end interfaces are implemented by above guideline.

Database

The app operates on a single MySQL database called “TakeCareDB”. It has a central table called “users” which contains all the information gathered in Registration Phase I. Each patient has two tables, one for prescription and one for reports. Pathologist has a table for patient Queue.



Structure of database

The database info is systematically converted into class diagram described in Design document on client-side.

Networking

All requests are made through java.net.HTTPurlConnection class and hence server domain using javascript or any other redirection WILL NOT WORK.