INDRAPRASTHA INSTITUTE of
INFORMATION TECHNOLOGY DELHI

Mobile Computing End-term Project Report

TITLE:

HealthApp

SUBMITTED TO :-

MR. VINAYAK NAIK

SUBMITTED BY:-

AVNI MALHAN (MT15012, avni15012@iiitd.ac.in, high)

KAPISH MALIK (MT15026, kapish15026@iiitd.ac.in, high)

KARISHMA TIRTHANI (MT15027, karishma15027@iiitd.ac.in, high)

MEETIKA ANAND (MT15033, meetika15033@iiitd.ac.in, high)

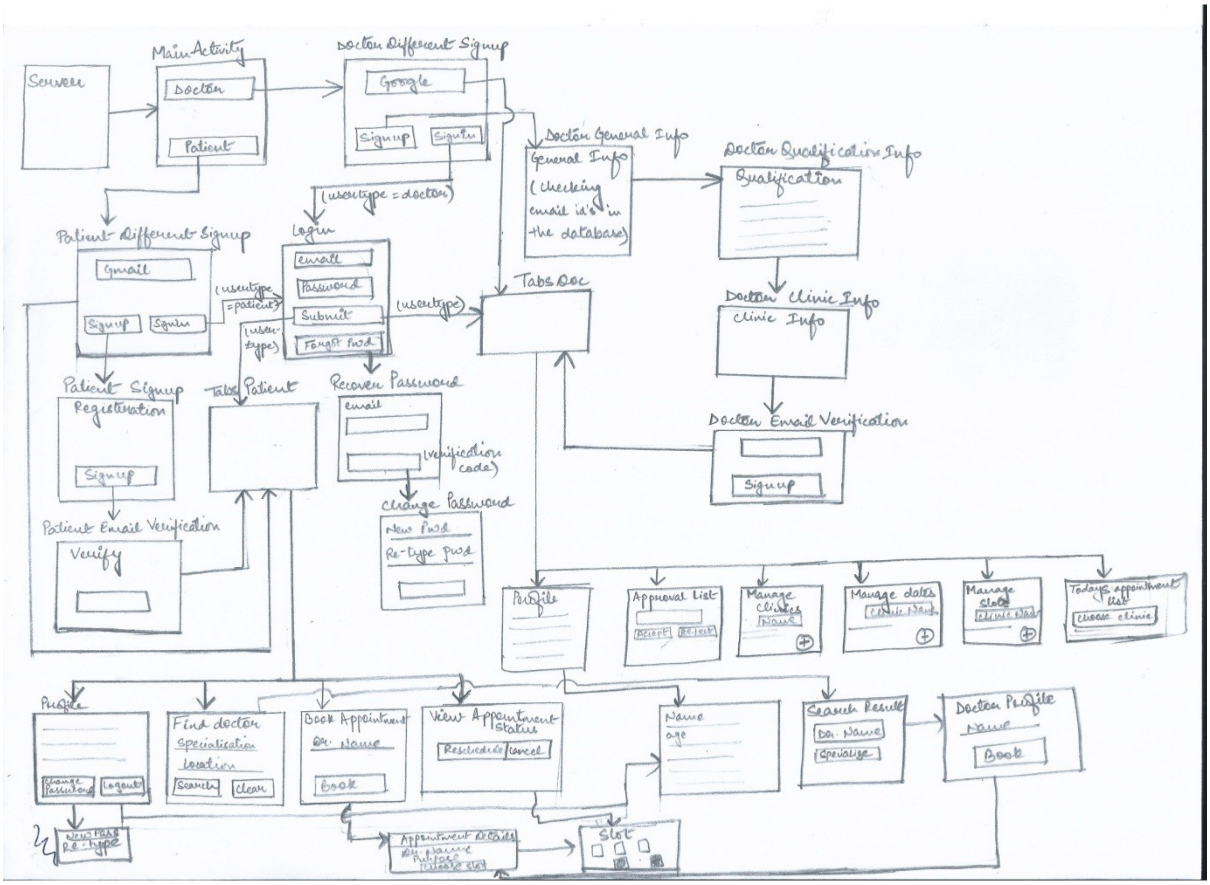
NEETI ARORA (MT15039, neeti15039@iiitd.ac.in, high)

GROUP ID :- 35

3) Objective:

It is a health-app with two types of profile. A user can sign-up either as a doctor (authenticity is taken care by the medical council number, which is unique for any doctor) or as a patient, where he can search for a particular type of doctor and book an appointment. Accepting or rejecting the appointment is in the hands of the doctor.

5) Architectural Diagram:



6) Hardware and Software Pre-requisites:

Hardware:

- Laptop
- Android phone

Software:

- Android Studio
- Wamp Server (initial testing of the database was done with wamp)
- Server (server has been created on freeoda.com by the name- mchealthapp)

Additional libraries:

1) com.github.amlcurran.showcaseview:library:5.3.0

2) `com.astuetz:pagerslidingtabstrip:1.0.1 -  for TabView`

3) `com.android.support:design:22.2.1 - for floating action button`

4) `libs/activation.jar`

5) `libs/additionnal.jar`

6) `libs/mail.jar`

Minimum version supported: API level 15

7) Link to Source code and executable:

Source Code :

https://drive.google.com/a/iiitd.ac.in/file/d/0B2YY0IGHpodWM3YtN080NkZ4Tms/view?usp=sharing

APK :

https://drive.google.com/a/iiitd.ac.in/file/d/0B2YY0IGHpodWazVTSU1lSkxyWmM/view?usp=sharing

PHP Files :

http://mchealthapp.freeoda.com/android/

8) Summary of the Mid-Sem Demo:

- Patients are able to create their accounts and linking with the database has been started.
- UI has been created for the entire app.
- Current location is being fetched through GPS.
- UI for the patient approval page is ready, but linking to the database and the patient is yet to be done.
- Doctors are able to create their account on the app and update their qualifications, specialization and appointment timings.

Our Database is as follows:

- Doctor(email, name, gender, age, year_of_exp, medical_council_no, qualificationid, password)
- Qualification(qualificationid, UGdegree, PGdegree, speciality)
- ClinicDetails(clinicid, clinic_address, clinic_contactno,fee)
- Slots(slotid, clinicid, start_time, end_time)
- WorkingDates(doctor_email, clinicid, date)
- Appointment(patient_email, doctor_email, appointmentid, clinicid, start_time, end_time, appointment_status)
- Patient(patient_email, name, contact_no, password, address, age, gender)
- ClinicRDoc(doctor_email, clinicid)

Additional Work:

- User verification by sending an OTP via mail
- We had many screens (approx 70)
- Connection to the server was done.
- New Password setting if user forgets his/her password by sending OTP via mail.

Comments received:

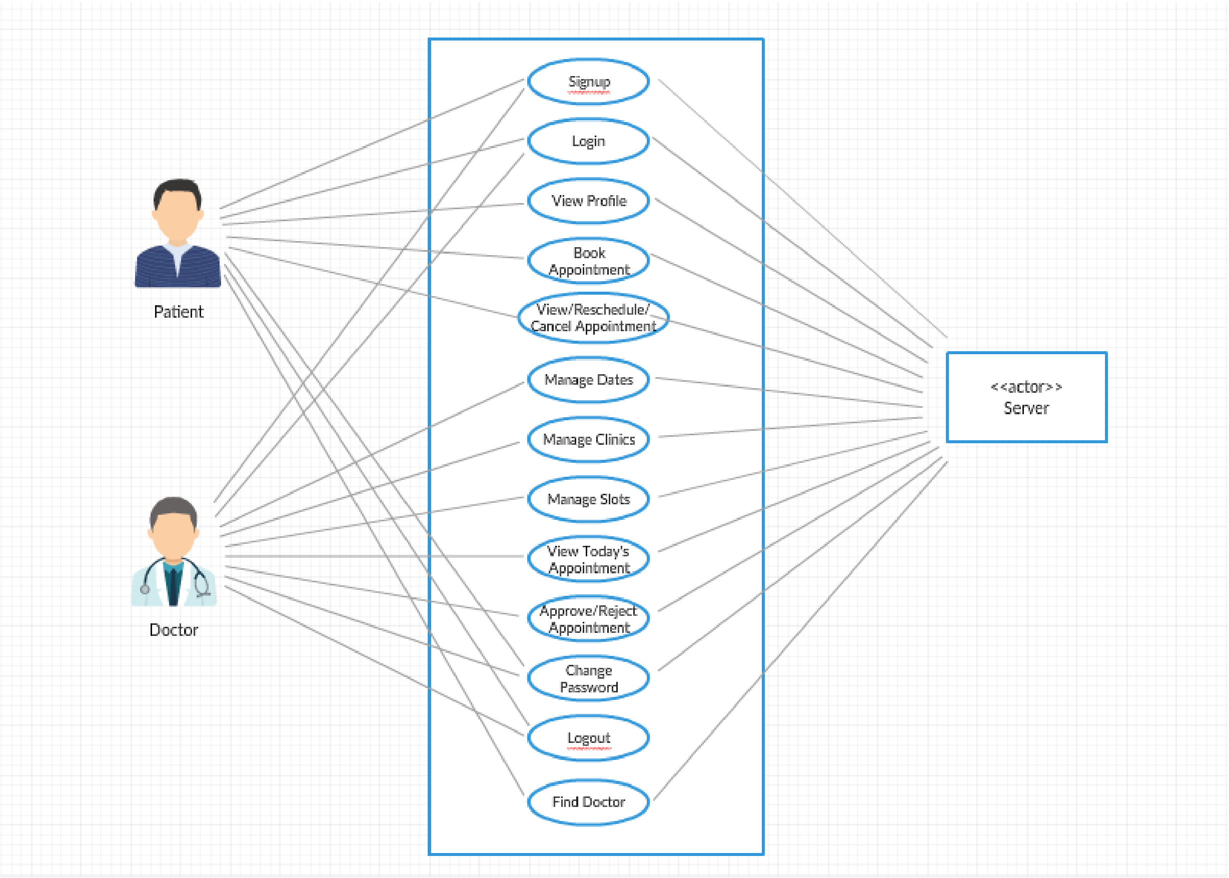
1. Password encryption is missing - done in this phase.
2. Timestamp ordering can be done .
3. Add rating to doctors if possible.

9) <u>Progress after the midsem demo:</u>

9.1) Login/Authentication:

- Login is done either through the general signup functionality or through Google sign in.
- Authentication of doctor is done by his/her medical council number, which is unique for each doctor.
- While creating accounts for both, patient as well as the doctor, a 4-digit verification code will be send via mail, which the user has to enter in the app. Only then will he be able to successfully create an account.
- Session is maintained using Shared Preferences. User needs to logout before logging through other account.

9.2) Data Processing and reporting of results via use-case:



- **Signup**
  - **Input :** User (patient/doctor) provides his/her personal and professional details.
  - **Processing Logic :** Checks whether the user already exists in the database or not. If not, User (patient/doctor) record added to database.
  - **Output :** User's (patient/doctor) HealthApp account is created.

- **Login**
  - **Input :** User (patient/doctor) provides his/her email id and password.
  - **Processing Logic :** Email Id and password authenticated with database.
  - **Output :** User (patient/doctor) can access his/her account.

- **View Profile**
  - **Input :** page directly opens whenever the doctor/patient logs in.
  - **Processing Logic :** basic details like name, email, gender, date of birth and years of experience are fetched from the database.
  - **Output :** all the above details are fetched using json and displayed on the page.

- **Book Appointment**
  - **Input :** Patient provides doctor’s name, his/her name, clinic name and purpose. User selects appointment date and time from the doctor’s available dates and time slots.
  - **Processing Logic :** Appointment record added to database with status “Sent for approval”.
  - **Output :** Appointment is sent for approval by the doctor.

- **View/Reschedule/Cancel Appointment**
  - **Input :** Patient views the appointment which indicates the appointment schedule and it’s status. If patient wants to reschedule, he/she selects the new appointment date and slot.
  - **Processing Logic :** If appointment rescheduled, corresponding appointment record updated in the database. If appointment cancelled, corresponding appointment record deleted from the database.
  - **Output :** Patient’s Appointment is rescheduled or cancelled as per patient request.

- **Manage Dates**
  - **Input :** Doctor selects the clinic name and can view and add his/her available dates corresponding to the selected clinic.
  - **Processing Logic :** Date records are added to the database corresponding to the selected clinic.
  - **Output :** The dates list is updated and updated list is displayed.

- **Manage Clinics**
  - **Input :** Doctor can view and add his/her available clinics by providing the clinic name, address, contact no and fees.
  - **Processing Logic :** Clinic record added to the database corresponding to the doctor.
  - **Output :** The clinics list is updated and updated list is displayed.

- **Manage Slots**
  - **Input :** Doctor selects the clinic name and can view and add his/her available time slots corresponding to the selected clinic.
  - **Processing Logic :** Time slot records are added to the database corresponding to the selected clinic.
  - **Output :** The time slots list is updated and updated list is displayed.

- **View Today's Appointment**
  - **Input :** Doctor can view all appointments scheduled for today.
  - **Processing Logic :** All the appointments corresponding to the doctor having appointment date as today's date are fetched from the database.
  - **Output :** All the today's appointments scheduled for the corresponding doctor are displayed.

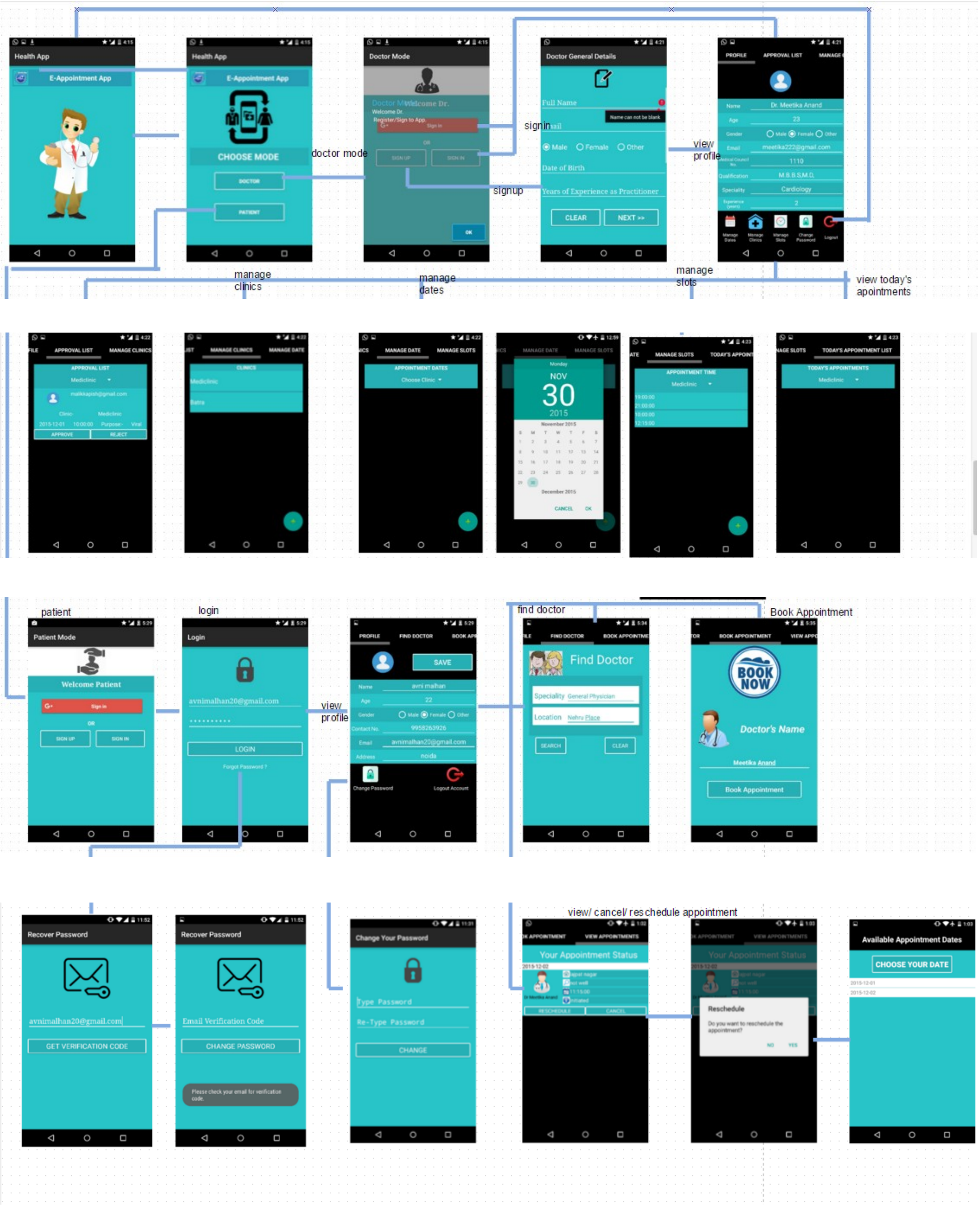
- **Approve/Reject Appointment**
  - **Input :** Doctor can view the approval list and approve/reject the appointment.
  - **Processing Logic :** The corresponding appointment record status will be updated in the database.
  - **Output:** On Approval/Reject, Mail is sent to patient about the status and in Appointment table, appointment status code is updated. (Appointment Status Code field 0- initiated, 1- Approved, 2- Rejected, 3- Done).

- **Find Doctor**
  - **Input :** Patient can find the doctor of a particular speciality in the nearby location provided by him/her.
  - **Processing Logic :** GPS is used to fetch the doctor list in the nearby location provided by patient.
  - **Output :** Doctor's list is displayed based on the location and speciality.

- **Change Password**
  - **Input :** User (patient/doctor) provides his/her new password.
  - **Processing Logic :** User (patient/doctor) record updated in the database.
  - **Output :** User's account password is updated.

- **Logout**
  - **Input :** User (patient/doctor) wishes to exit from the system.
  - **Processing Logic :** The current session expires.
  - **Output :** User (patient/doctor) exits from his/her account and reaches the home screen.

9.3) Screenshot for the wireframes:



**All the screens could not be placed into the wireframe as diagram would become too complicated. We have included all the main screens as directed by Sir.

Login Information required for checking the app - users already registered on the app.

Patient Credentials:

Email: malikkapish@gmail.com   Password: kapish1234

Doctor Credentials:

Email: meetika222@gmail.com   Password: meetika

9.4) Additional Work:

- Guiding tour of the entire app on first time using it.
- Online server made, so as to practically implement the app.

  Server URL: http://mchealthapp.freeoda.com/
  Database Access: http://mchealthapp.freeoda.com/pma/
  Credentials: UserName: 1046389   password : kapish1234

- Made complete Calendar View through grid layout.
- Session is maintained using Shared Preferences. User needs to logout before logging through other account.
- Sending email to doctor for approval when patient book the appointment and also, to patient when doctor approves or reject his/her appointment.

10) Unfinished Task:

- Could not implement queue-status of the patients.

Note:

- Please switch on your internet and GPS while running the app.
- Sometimes, Server may be down as it is free one. Please wait and try to use the app after sometime.