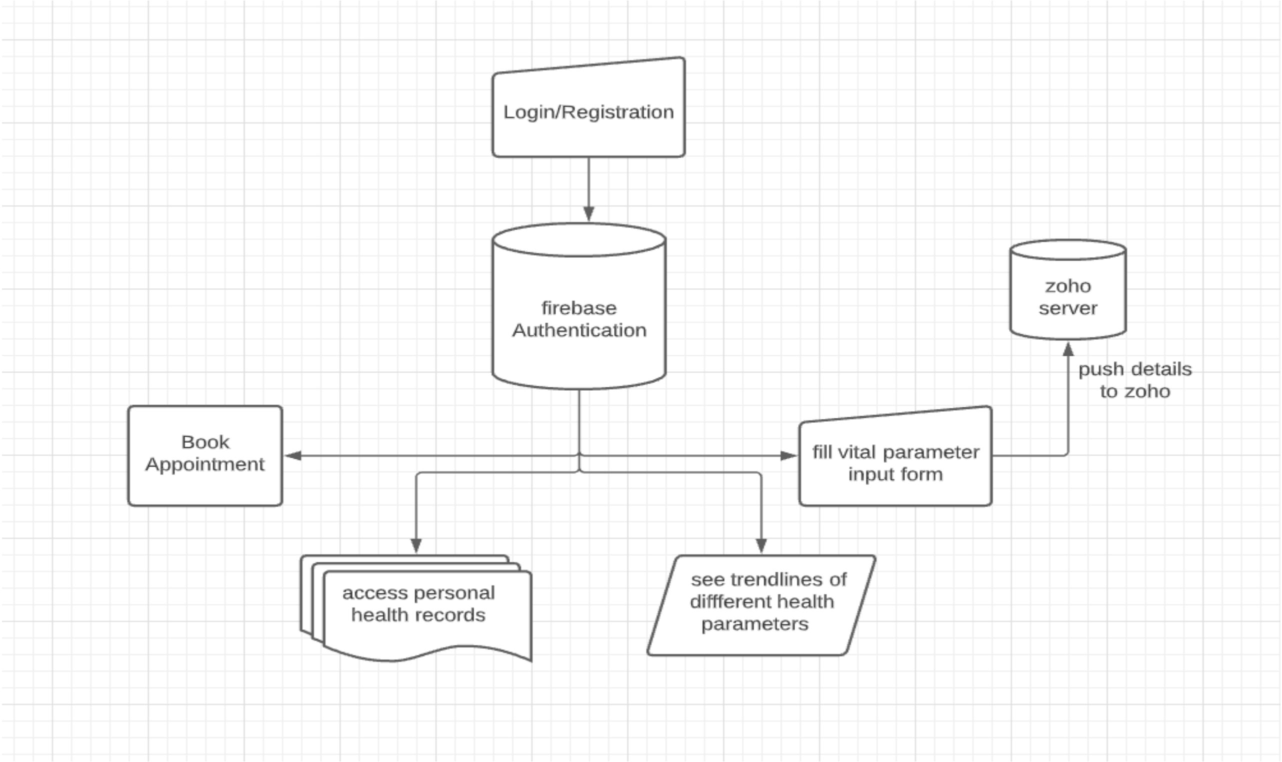
Product Design

Team 40, Remote pregnancy monitor, Ananya, Bhargavi, Kiran, Rohan

Design Overview

Architectural design

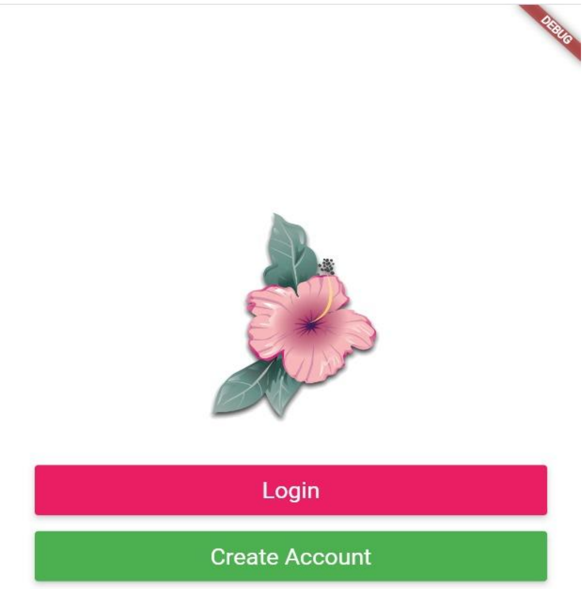
We’re building a mobile application using flutter and firebase. Flutter is a tool that allows us to build cross-platform applications with one programming language and a single codebase. Firebase helps us in authentication, storing users' details, etc. we would also interact with the Zoho server because we need to fetch trendlines and display them to respective users.

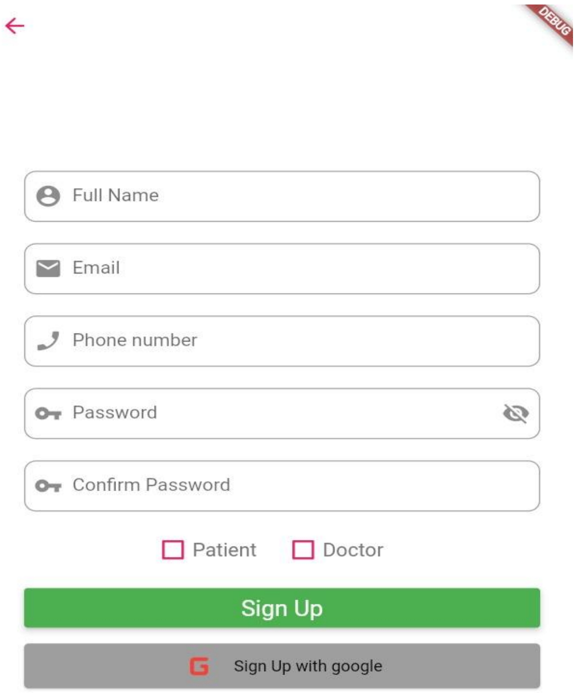

Login/Registration
firebase Authentication
zoho server
push details to zoho
Book Appointment
fill vital parameter input form
access personal health records
see trendlines of diffferent health parameters

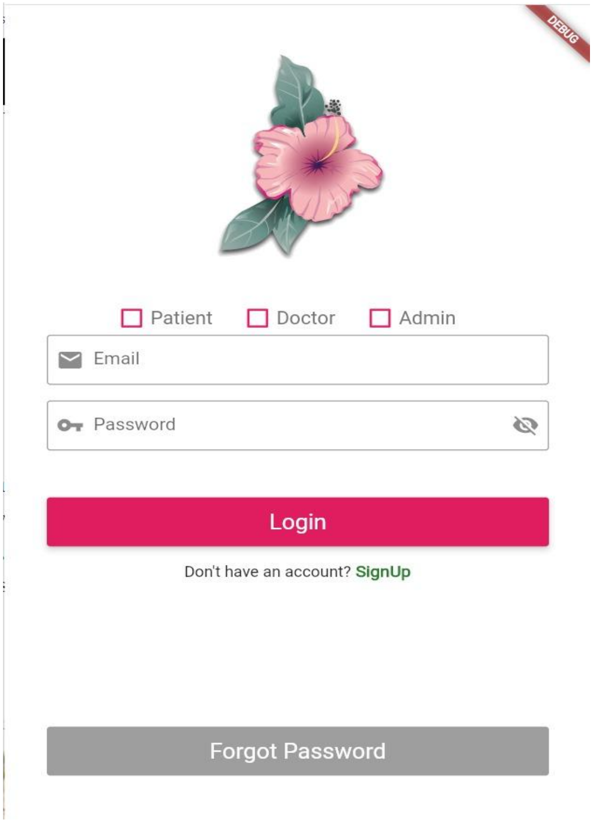
https://lucid.app/lucidchart/bf8dbe53-bb50-4a2d-9b08-3e31a018b145/edit?invitationId=inv_92a2f773-2676-4b62-86a2-d011e543139f

System interfaces

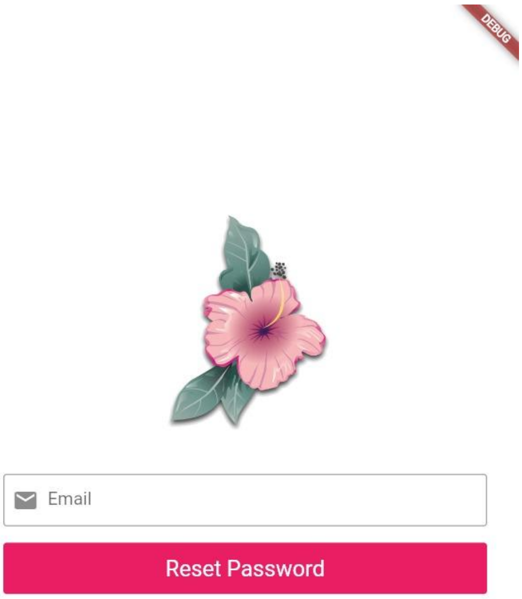
User Interface

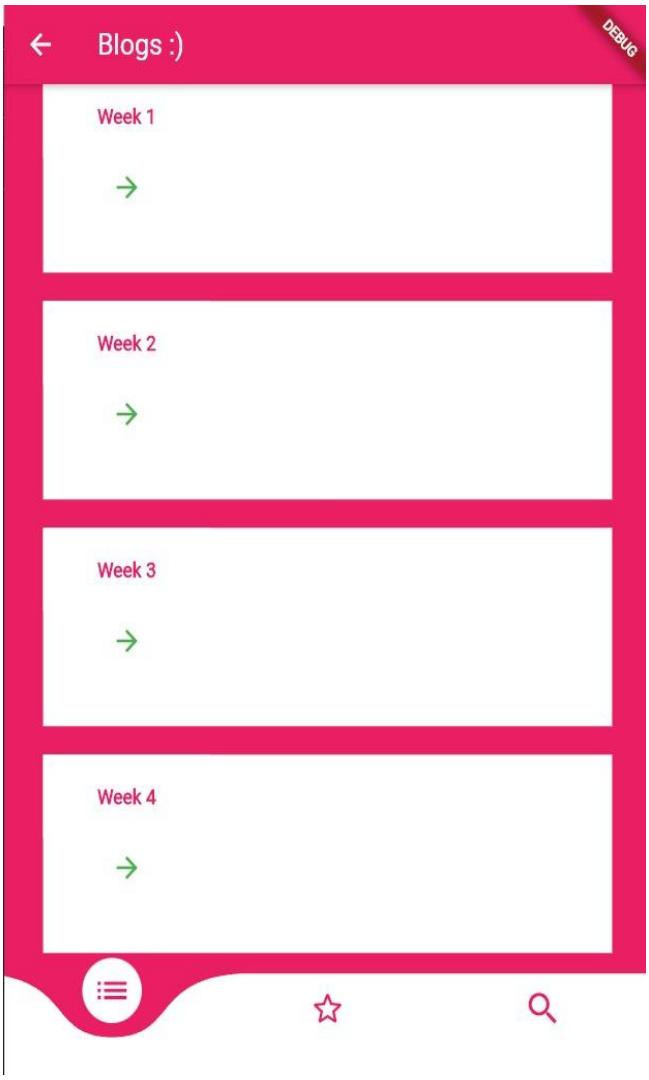






Don't have an account? SignUp





Blogs :)

Week 1

Week 2

Week 3

Week 4

Model

https://lucid.app/lucidchart/334cd6f1-474e-48a3-87a6-4e1db2bf1c2d/edit?invitationId=inv_8132e995-8c62-42ff-af31-30a6bc94b754

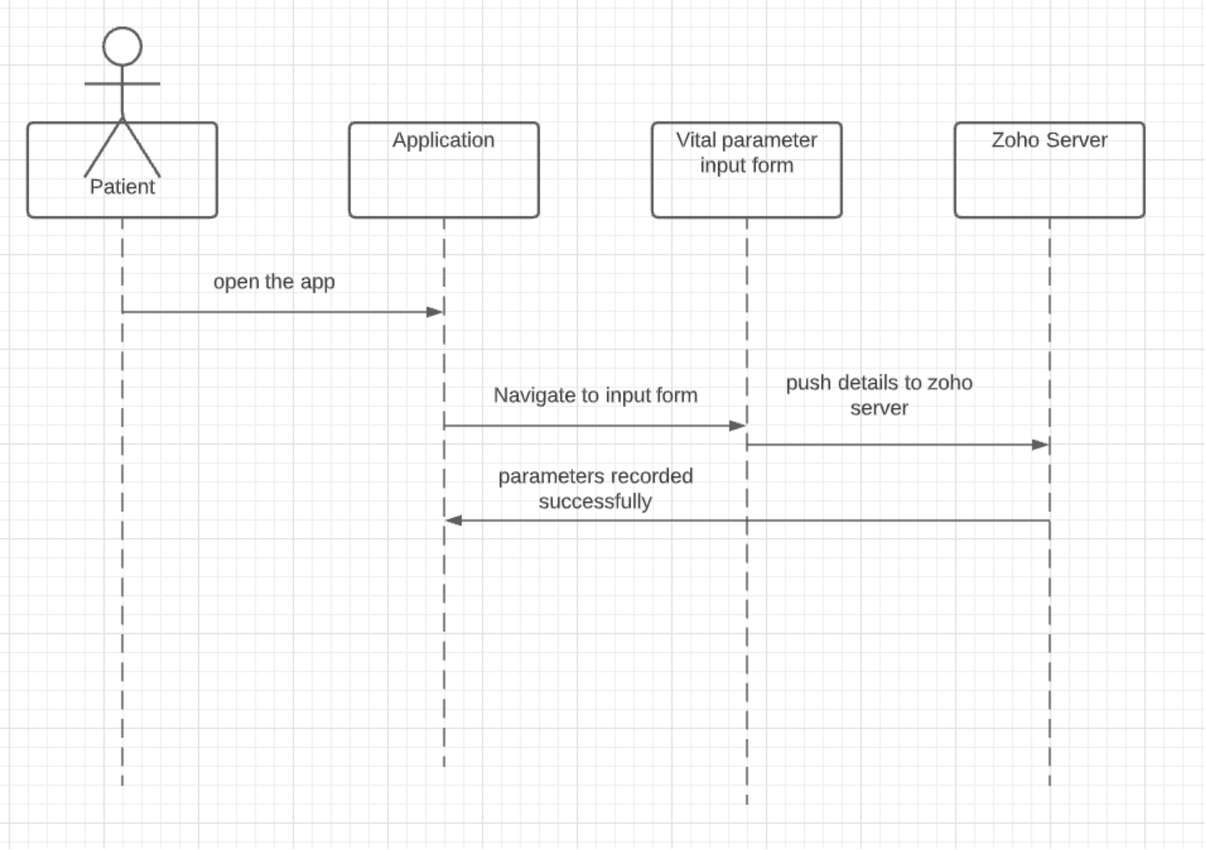
Doctor	Doctor State ● The class is responsible for storing doctor’s details (email ID and the password) **Class behaviour** ● The class, in real world, has the following methods: ○ **login():** authenticates a doctor’s login
Patient	Patient State The class is responsible for storing patient’s details (email ID and the password) **Class behaviour** ● The class, in real world, has the following methods: ○ **login():** authenticates a patient’s login
User	**Class behaviour** ● The class, in real world, has the following methods: ○ **register():** user registers as patient or doctor

.

Sequence Diagram(s)

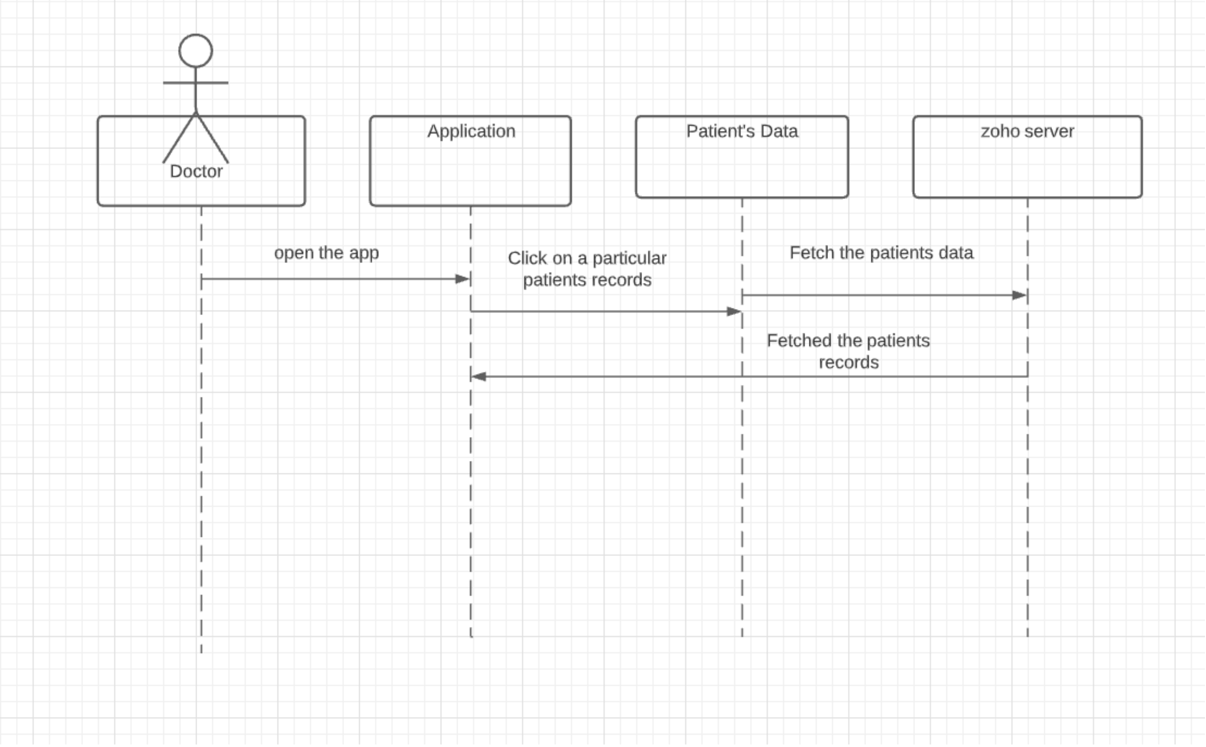
USE CASE 1: Patient logs into application and fills vital input form

https://lucid.app/lucidchart/0c68f51a-8b2b-462b-a59b-86d58fc1ee92/edit?invitationId=inv_56f55d80-8ed1-4987-9bb4-efc9d62dd053


Patient
Application
Vital parameter input form
Zoho Server
open the app
Navigate to input form
push details to zoho server
parameters recorded successfully

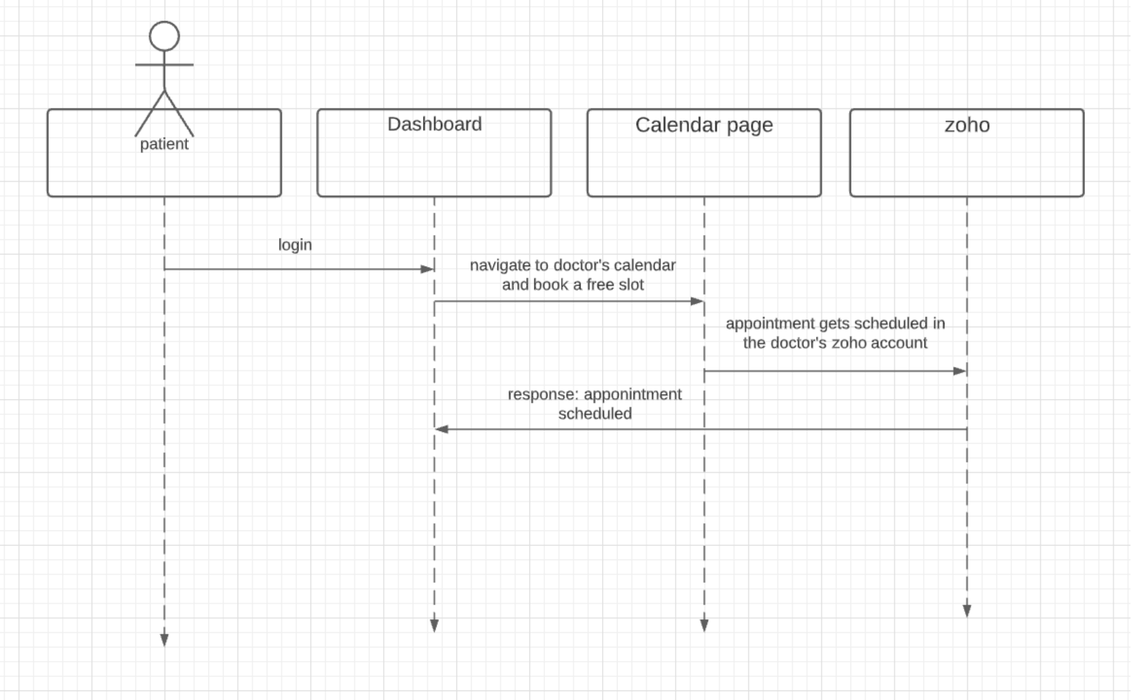
USE CASE 2: Doctor checks trendlines of different health parameters

https://lucid.app/lucidchart/e530fcf1-dbe1-41f9-aec4-4d17ad86d91b/edit?invitationId=inv_c79788c7-1bee-415d-b346-2c218824d44b



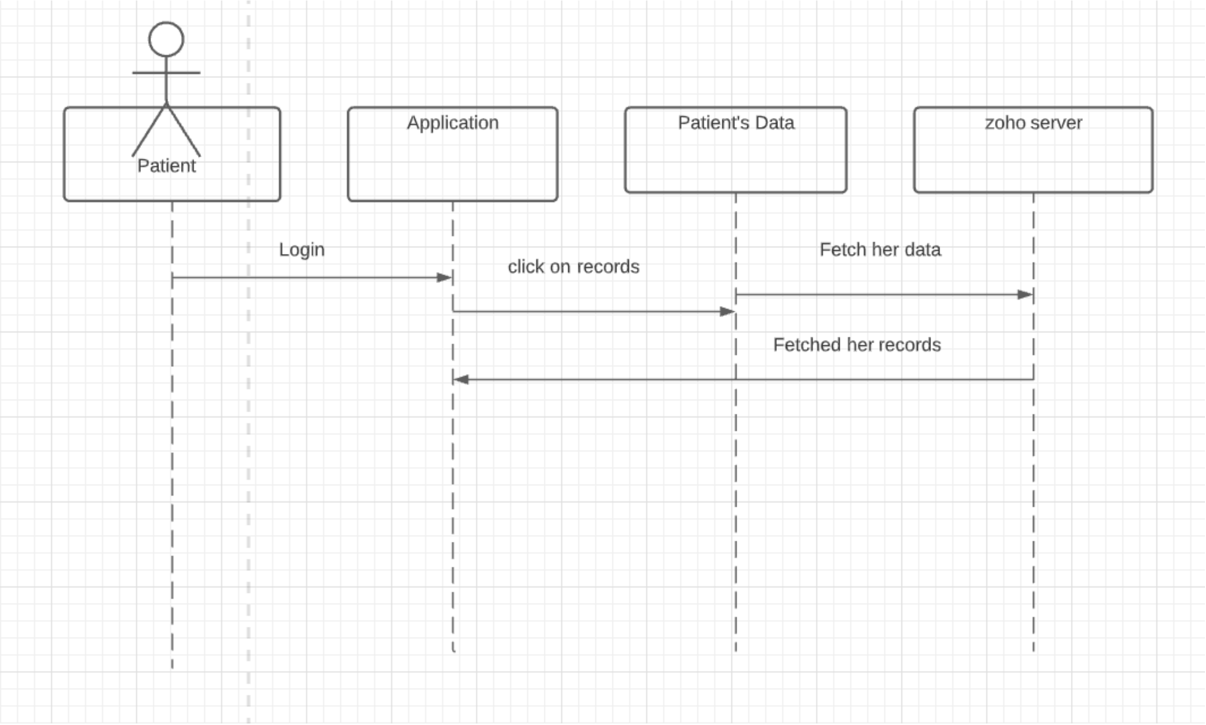
USE CASE 3: patient books an appointment

https://lucid.app/lucidchart/3d21302b-a286-48fa-9476-8353a5dd400a/edit?invitationId=inv_1c44f4fa-dcbc-418c-a73a-0cede717f59a


patient
Dashboard
Calendar page
zoho
login
navigate to doctor's calendar and book a free slot
appointment gets scheduled in the doctor's zoho account
response: apponintment scheduled

USE CASE 4: patient checks her records

https://lucid.app/lucidchart/3e3d07a5-3da1-47fa-9241-51c3b6e9506f/edit?invitationId=inv_404be79c-5a4d-42ba-bbf9-703d18859d9f



Design Rationale

Design Rationale 1:

The original model structure for our application consisted of another role called “admin” which was eventually omitted in our first draft of the model structure. Admin is at the top of the hierarchy of our project. A patient can view only her records, A doctor can view all the patient records assigned to him, admin has the access to all the records of the patients. Introducing the admin role at the start increased the complexity of the application, so the client removed the admin role.

Design Rationale 2:

Teleconsultation at first was a compulsory requirement of the project but later on it has become a low priority requirement because a new requirement chat with the hospital has been introduced.