**Functional Specification of Doctor Scheduler**

Summary: This software is expected to allow doctors to post their daily/weekly work schedules and make changes in real time. It will also allow anyone else (e.g. patients) to check a doctor’s schedule and whereabouts from a mobile device.

Although the first target is ‘doctors’, I wanted to extend this app to other users as we go along. There are other use cases, such as trainers, visiting lecturers informing their students – that are feasible once the app is established and the app should be open enough to add people other than doctors.

We will hence think about nomenclature, app name etc that is more generic and represents the utility rather than a specific end user.

This software consists of the following interfaces:

1. Doctor: A mobile application. We will call it Docapp.
2. Patient: A mobile application. We will call it Patapp.
3. Secretary: A browser app for Doctor or his/her secretary. We will call it Secapp.

*Docapp*: This mobile application will be used a doctor. It will let the doctor upload his/her schedule information. It will also let the doctor post his/her whereabouts in real time.

*Patapp:* This mobile application will be used by a patient. It will let the patient search for a doctor and then view the doctor’s most current schedule.

*Secapp*: This browser application will be used either by a doctor or his/her secretary. It will essentially do the same job as the Docapp.

**Specification of Docapp:**

Workflow:

<<Kiran Bhagwat:

As we discussed, we may have to create a database of doctors ourselves (to the extent possible) and give them registration code that they can use to login and modify password/sign in process. Doctor’s database seems to be available on the net and this will be an efficient way to get doctors to sign up without too much work.>>

1. ENTRY SCREEN
   1. Scenario 1: First usage
      1. Sign up screen. Multiple options to sign up/log in:
         1. Email + password (default)
         2. Facebook (requires user to sign in with Facebook account)
         3. Google (requires user to sign in with Google account – this is easily facilitated on an Android platform)
      2. Create profile
         1. Set profile picture (if logged in with Facebook or Google, then automatically set to Facebook or Google profile picture)
         2. Set profile full name
         3. Set email (if default sign-up, automatically set; if alternate sign up is used, required to enter)
         4. Set phone number (i.e. primary work contact)
         5. Set doctor details (degrees, experience, area of expertise, etc.)
   2. Scenario 2: Return usage
      1. Sign in screen. Option to save password
         1. Email + password (default)
         2. Facebook
         3. Google
2. Home screen
   1. Set current status/location
   2. Option to edit weekly schedule (and Send button to send updates to those following you)
   3. Modify today's current schedule (and Send button to send updates to those following you)

**Docapp Screens:**

To be done later (after finalizing the functionality)

**Specification of Patapp:**

Workflow:

1. HOME SCREEN
   1. My Doctors
   2. Add new Doctor
2. MY DOCTORS VIEW
   1. Simple list view of all the selected doctors
   2. Each doctor can be selected/expanded to show "DOCTOR VIEW" for that specific doctor
3. DOCTOR VIEW
   1. Shows current location/status
   2. Full week view
   3. Details view
   4. Contact view – directly links to <doctor>'s work phone, email, etc.
   5. Show all other details from <doctor>'s database file
4. ADD DOCTOR VIEW
   1. Access to public database, search algorithm by name, email, contact #, specialization, location, etc. (I guess we will facilitate search within the app database too so that they can quickly search for doctors in a specific locality or based on specialization etc. This will also help us go towards a paid version in the future where those who pay would show up on top of search just as google does)

**Specification of Secapp:**