Group Organization Plan

(Group 16A – Nikola, Ethan, Callum, Kristian and Tare)

**Ethan**:

In sprint 1, Ethan created the Patients and the Bookings databases, together some accompanying ease-of-life methods for the rest of the files to refer to. He also took on the functionality and the checks for the Registration page, in the form of RegistrationFunc which uses the check file to confirm the legitimacy of the data and then sends it to the DB and RegistrationCheck which tests the data to prevent exceptions and future errors due to false information, illegal datatypes and general data inconsistencies.

**Nikola:**

In sprint 1, Nikola did the LoginPage and the GeneralPage GUIs, together with the appropriate functionality (LoginFunc) and check (LoginCheck) files to go along with them. He also added more checks to RegistrationCheck. Nick also spent time enforcing coherency between the different pages, including consistent font, font size, colour theme, spacing, and window size. Finally, for sprint 1, Nikola did two code quality reviews which included adding author comments, enforcing variable declarations to be in the beginning of the code, removing unnecessary auto-generated comments and adding useful ones, imposing basic indenting and whitespace practices, and implementing proper project structure.

**Callum:**

Callum took the role of the Tester for all the sprints. All in all, he committed 8 different test files in the Test package in our git repo. He used both junit tests and Mockito to make sure that our interface doesn’t break in any corner cases. Callum wrote the three test specification documents by himself, where he provided more details on the different tests he conducted and justified his decisions.

**Kristian:**

In sprint 1, Kristian created the registration UI and making sure that it inserts new information into the database, I will be doing test cases for making sure that no duplicate information is inserted and correct format is also followed with proper error messages

I will also make a general UI to allow the patient to select whether to login or register and make the corresponding buttons to open the correct UI.

**Tare:**

- more test cases

**Database Design:**

The database is a simple one table design, where the table holds the details of registered patients. The details included a patientID that auto increments as more details are added (first patient to register is 1, second is 2, etc). Other details are First Name, Surname, Age, Gender, Doctor Chosen, Phone Number, Details. The only fields that allow null values are Phone Number and Details, since there is a chance that the patient doesn't have access to a phone and would need to communicate with the doctors in a different way. No details may be needed as well. The database is initialised when the app is launched, and won't be reset unless General Page is directly opened again.