Ethan - I will be creating all of the background code that communicates with the database, as well as code all code that moves data between java files and the app itself. This includes the validation used by the registration interface to ensure the details the patient provides are valid for the database to use. I will also create all SQL statements that are used by the application to communicate with the database. This includes the statements that create both the database and tables in General Page, as well as statements that send and retrieve required data.

**Nikola:**

- Login page GUI - username and password fields and a functional log-in button (send info to LoginCheck.java)

- General Page GUI - a 'Hello' message and two buttons, the left button sends back to the login page, and the right button sends to the register page

- Adding more checks for the registration checks (both in the Registration\_gui.java and in the Registration.java)

- Code quality control - This includes adding author comments at the top of the different files and monitoring what everyone has done, enforcing variable declarations to be in the beginning of the code, removing unnecessary auto-generated comments and adding useful ones (to every relatively complex and important method and code segments), imposing basic indenting and whitespace practices, and implementing proper project structure (e.g. having a separate folder for the functional files and a separate folder for the GUIs)

- Coherency between the different pages of our app/website - consistent font (including font size), colour theme (white and a specific shade of blue), spacing, and same sized windows

**Callum:**

Test cases (learn how to do it and then do it)

- for registration I will be testing inputs against desired outcomes, will check that details are tested correctly against registration creating test cases with JUNIT 5. Age will be checked to see if they accept the correct age: Cannot be under 18 and over 120. Phone number follows correct format and only one is accepted. First name and surname must also not exceed 15 characters. Lastly, details should not exceed 100 characters.

- for log in Will

**Kristian:**

I will be creating 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.