Justification for Database Design Choices

Group 8B: Doctor Interface
Doctor -> Bookings -> Patients

In the database design we see five tables by the names of ‘patients’, ‘patients_bookings’, ‘bookings’, ‘doctor_bookings’, ‘doctorcred’ and ‘doctor’. All tables include an ID primary key which is used to uniquely identify each column in the tables. The patient and doctor tables include details about each person’s name, gender, address etc. We have also included a doctor credentials table to ensure the that doctors(users) can access their accounts with their usernames and passwords. The booking table has a one-to-many relationship to the ‘doctor_bookings’ and ‘patient_bookings’ table as well as the ‘doctor’ and ‘patient’ entities. This is to ensure there is a relationship between the three tables doctor, patient and bookings. We also include a relationship between ‘doctorcred’ and the ‘doctor’ table so that both doctors table data remains connected. As we will be needing doctors to view their patients’ bookings, we have added the doctor id in the patient’s table. This allows each doctor to view each patient which is allocated to them. We have also added doctor id and details to the patients_booking table in order to make a connection between the two tables.

The reason for the decision to make these particular choices is because the doctor interface requires the functionalities to login and out, view the patients details and bookings so in order to perform these functionalities, we would need a table of the patients details and bookings to connect to the doctor table. I was able to do this by creating tables and then creating relationships within those tables using primary and foreign keys in order to link the patient’s bookings and the doctor bookings.