Justification for Database Design Choices

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

The Database as of the end of Sprint 2:

In the database there are 5 tables: Patients, Bookings, Doctor, Doctorcred. Beginning with patients this table includes relevant information regarding the patients which are stored in the hospital's records. This includes their first name, surname, date of birth etc. Each entry in this table is uniquely identified via the primary key patientID which is an integer value. This should stop overlap as a patient should not appear multiple times in this table. The foreign key doctorID references a patients associated doctor, who they are assigned to.

The booking table contains information regarding a booking of an appointment that can be made by patient. Example information here includes doctorid, patientid, date, time etc. Here a specific booking is uniquely identified via bookingid which similarly to the patientid is of type integer. The foreign keys here, doctorid refers to the doctor who would take the appointment and the patientid refers to the patient attending. Here prescription and notes are also apparent so the doctor can refer to any prescribed medication or notes made.

The doctor table is very similar to that of the patient one containing relevant information in regards to a doctor like phone num and the like. Again the primary key doctorid is used to uniquely identifier each entry.

The messages table is necessary as it stored confirmation messages of when a change is made to the database system. So for instance a new message will be generated and stored here when either a booking entry is added, deleted or updated. Information relevant to these messages are stored here like what it says and the date of when it was made. Doctorid is present here as the messages are sent to/regard changes made by a specific doctor.

Doctorcred is a table that stores doctors login credentials i.e. name and password. Contained within is also doctorid as these credentials are specific to each doctor entry. As with all the tables there is an id that uniquely identifies it as a primary key.

How each table is used for a user story/ feature:
As mentioned, prior, the doctorcred table is used specifically to stored the details necessary for a user/doctor to sign into there account. By having each doctor have a separate account information relevant to them can more easily be shown. As well as this account helps to keep information which should perhaps be private stay that way.

A doctors can view their specific bookings by utilising the bookings table. Where the doctorid in bookings matches the id of the doctor currently signed in the correct bookings will be displayed.

When changes are made to the bookings in the system, i.e. an entry is changed, this will change the information stored within the booking table.

As stated prior a doctor can view visit details and prescriptions as this information is stored in the bookings table also.

The date attribute in booking is also what allows the doctor to search for these appointments by a specific month and year.

Changes from Sprint 1:
The two tables doctor_bookings and patient_bookings were merged into one as we deemed it unnecessary to have both tables as they essentially represented the same thing. Similarly the extra bookings table which only contained the unique id was removed in place of this new bookings table that represented all of these things.
The prescriptions and notes attributes were also added to this table so this extra relevant information was present somewhere.
The messages table was added as to meet the confirmation message requirement mentioned in the brief.