**User Roles**

|  |  |
| --- | --- |
| **User Role** | **Description** |
| Patient | The patient user is the person who the hospital/doctor serves and is the one who gets prescribed. |
| Doctor | Doctors are the doctors which can serve 1 or multiple patients, and are the only ones who prescribe treatments or alter the purpose or side-effects tables. They can also write details to the patient regarding their treatments. Doctors can be patients. |
| Hospital | Hospital staff are all under the hospital user, and can view all the tables. All doctors are Hospital users, and Hospital users can be patients. |
| Suppliers | Suppliers are entities, companies or otherwise, that supply the treatment if the treatment is one to be supplied (something like “tylenol”, but not “strength training” or “cardio”). They are required to write the purpose and side-effects of a treatment if they insert one. |

**User Story Descriptions**

|  |  |
| --- | --- |
| **Story ID** | **Story description** |
| Patient | As a patient, I am able to access the treatments table, the details table, and the supplier table. The treatments table allows me to see the treatment they have been prescribed, to see possible side-effects of the treatment, and see why I am are taking a specific treatment. The details table allows me to see what details the doctor put for me regarding my treatment, and the supplier table allows me to see from what supplier the treatment comes from, if it does come from a supplier. |
| Doctor | As a doctor, I am able to access the hospital table, the treatments table, the details table, and the company table. The treatments table allows me to see the treatments that can be prescribed, to see possible side-effects of treatments, and see the purpose table shows why to prescribe a specific treatment. The details table allows me to see what details I put for my patient(s) regarding the treatment, and the supplier table allows me to see from what supplier the treatment comes from, if it does come from a supplier. I can prescribe a certain treatment to a patient, I can change the purpose of a treatment or the side-effects based on new information, and I can write details for my patient(s) regarding specific treatments. |
| Hospital | As a hospital user, I am able to access the hospital table, the treatments table, the details table, and the company table. The treatments table allows me to see the treatments that can be prescribed, to see possible side-effects of treatments, and why to prescribe a specific treatment. The details table allows me to see the details given to patient(s) regarding the treatment, and the supplier table allows me to see from what supplier the treatment comes from, if it does come from a supplier. I can add patients to the patients table, or hospital staff to the hospital table. |
| Supplier | As a supplier, I am able to access the treatments table, and the company table. The treatments table allows me to see the treatments that I have made, to see and list the possible side-effects of treatments, and why one should prescribe a specific treatment. The supplier table allows me to show that I am the supplier of specific treatments. I can insert a treatment into the treatments table, which links the treatment to me, and I have to link the treatments to its specific purpose and side-effects in their respective tables. |

Conceptual Design:

**Entities**:

**Primary Key**, Foreign Key

Cardinality - Attribute Max(Attribute Min)

* Patient
  + **patientId**, firstName, lastName, illness, doctorId, treatmentId
    - 1-1(1), M-1(1), M-1(1), M-M(0), 1-1(1), 1-M(0)
* Doctor
  + **doctorId**, firstName, lastName, hospitalId
    - 1-1(1), M-1(1), M-1(1), 1-1(1)
* Hospital
  + **hospitalId**, hospitalName, address, phoneNumber
    - 1-1(1), M-1(1), M-1(1), M-1(1)
* Supplier
  + **supplierId**, companyName, address, phoneNumber
    - 1-1(1), M-1(1), M-1(1), M-1(1), 1-1(0)
* Treatments
  + **treatmentId**, treatmentName, purpose, sideEffects, supplierId
    - 1-1(1), M-1(1), M-M(1), M-M(0), 1-1(1)
* Details
  + **detailsNumber**, details, doctorId, patientId, treatmentId
    - 1-1(1), M-M(1), 1-1(1), 1-1(1), 1-M(0)

**Main Known Relationships**

**Relationships:**

Cardinality - (Relationship Max-Relationship Min)

* Patient has a Doctor
  + (M-M), (0-0)
* Doctor has a Hospital
  + (M-M), (0-0)
* Treatment has a Supplier
  + (M-M), (0-0)
* Treatments are given to Patients
  + (M-M), (0-0)
* Doctor prescribes Treatments
  + (M-M), (0-0)
* Doctor adds details to treatments for a patient
  + (M-M), (0-0)
* Supplier adds Treatments, Purpose, and Side Effects
  + (M-M), (0-0)
* Doctor alters Treatments, Purpose, and Side Effects
  + (M-M), (0-0)

Logical Design:

**Primary Key**, Foreign Key

Cardinality - Attribute Max(Attribute Min)

* Patient
  + **patientId**
  + firstName
  + lastName
  + illness
  + doctorId
  + treatmentId

Normal Form: BCNF

Justification: firstName, lastName, and illness are dependent on the patientId which is the super-key. DoctorId and treatmentId are dependent on patientId but their attributes are separate, and their information is not displayed in this table.

Indexes: Clustered

Columns: patientId

Justification: The only people who need the patient’s info are the patient, the doctor, and the hospital, all of which have the patient Id. There is no need for people who know the name or details to be able to find the patient’s pharmaceutical information.

* Doctor
  + **doctorId**
  + firstName
  + lastName
  + hospitalId

Normal Form: BCNF

Justification: firstName and lastName are dependent on the doctorId which is the super-key. hospitalId is dependent on doctorId but it’s attributes are separate, and their information is not displayed in this table.

Indexes: Clustered

Columns: doctorId

Justification: Similar to the patient, everyone who needs to be in contact with the doctor and know about them already have their Id. There is no need for suppliers or anyone else to be able to find a doctor’s account.

* Hospital
  + **hospitalId**
  + hospitalName
  + address
  + phoneNumber

Normal Form: BCNF

Justification: hospitalName, address, and phoneNumber are dependent on the hospitalId which is the super-key. There are no other foreign key dependencies because hospital is the “one” all of its one-to-many relationships.

Indexes: Clustered

Columns: hospitalId

Justification: hospitalId is the primary key, which plays a decent role in the reason it is the index, but it is also due to the fact that other columns can be very similar or may change over time, which makes hospitalId the best choice.

* Supplier
  + **supplierId**
  + companyName
  + address
  + phoneNumber

Normal Form: BCNF

Justification: companyName, address, and phoneNumber are dependent on the supplierId which is the super-key. There are no other foreign key dependencies because supplier is the “one” all of its one-to-many relationships.

Indexes: Clustered

Columns: supplierId

Justification: The Suppliers table is similar to the Hospital table in that, there is a decent chance that many of the other columns may change their value depending on what happens, however the supplierId will remain the same.

* Treatments
  + **treatmentId**
  + treatmentName
  + purpose
  + sideEffects
  + supplierId

Normal Form: BCNF

Justification: treatmentName, purpose, and sideEffects are dependent on the treatmentId which is the super-key. supplierId is dependent on treatmentId but it’s attributes are separate, and their information is not displayed in this table.

Indexes: clustered

Columns: treatmentId

Justification: Many treatments can have the same name if they are made by the same supplier, and they can have very similar purposes and side effects, which makes many of these suboptimal candidates for indexes. For these reasons, treatmentId is used, aside from the fact that not everyone should be able to find specific treatments/drugs.

* Details
  + **detailsNumber**
  + details
  + doctorId
  + patientId
  + treatmentId

Normal Form: BCNF

Justification: the details column is dependent on the detailsNumber which is the super-key. DoctorId patientId, and treatmentId are dependent on detailsNumber but their attributes are separate, and their information is not displayed in this table.

Indexes:clustered

Column: detailsNumber

Justification: The details regarding a specific and their treatment should not be easy to find and are very specific, therefore they are searched for using the detailsNumber, which is only with the patient and the doctor, which are the only people that need to see the info.

Procedure: GetPatient()

Parameter: IN, specifying name of patient

Receive data of patient based off specific patient traits

Procedure: GetTreatment()

Parameter: IN, specifying name of treatment

Receive data of patient based off specific treatment traits

Function: PatientInHospital()

Parameter: IN, name of doctor

Receive name of hospital based off name off doctor

Function: DoctorInHospital()

Parameter: IN, name of doctor

Receive name of hospital based off name off doctor

Procedure: DisplayDetails()

Parameter: IN, specifying name of detail number

Displays all details based on the detail number presented.

Trigger: BEFORE INSERT on Treatment

Supplier triggers event and adds Treatments, Purpose, and Side Effects

Trigger: BEFORE INSERT on Treatment

Supplier triggers event and adds Treatments, Purpose, and Side Effects

Event: ChangeDetails()

Doctor adds details to treatments for a patient