**Synopsis Notes**

While designing our database and writing our tables, we wanted to ensure that our system was robust. This is where testing comes in. We noticed flaws in our design choices while going through the testing requirements. This document is to address those flaws so that in the next iteration of implementation, we fix these flaws to enhance our system and make it more robust.

The Child-Parent relationship is a many-to-many relationship, requiring a ‘junction table’ to associate a child to many guardians, and to associate a guardian to many children.

The Child table does not allow a child to be created without first having a guardian. This is because we cannot insert a record into the Child table without a valid guardian id due to the FK constraint being enforced by the database. One thing to note is that it is still possible to insert the child into the patients table without a guardian. The same goes for an adult patient, uninsured patient, and insured patient as well. This is something that may potentially lead to unwanted behaviors. To circumvent this, a possible design change consideration would be to separate out child and adult patients into their own tables and enforce FK constraints directly on those tables.

A patient over the age of 18 cannot have a parent or guardian. This is because there is nothing in the database tying an adult patient to a guardian. A guardian may be able to be created successfully and an adult patient may be able to be created successfully separately. However, the two do not have any dependency on each other which means an adult patient will never be permitted to have an associated guardian.

Our database does not allow one child patient to have multiple guardians. The way that the database has been designed prevents us from doing so because the child table’s PK is patient id and therefore a single patient can only exist in the table once. This is why the system allows us to insert the record (2,1) which assigns our first guardian to the child, but not the record (2,2). This is because the primary key constraint is violated. One possibility to solve this problem would be to create an intervening table in order to capture the many to many relationships between guardians and children. Another option could be to potentially make guardian id a multivalued attribute.

The database allows a patient to be uninsured and have their credit card information collected to be charged for the visit and any supplies used during the visit. This is because we were able to insert an adult patient into the database, insert a new credit card into the credit card table, and tie that new adult patient to the newly inserted credit card record through a FK constraint which allowed us to indicate that the patient is uninsured. Since their credit card information has been collected before they can be registered as an uninsured patient, they may successfully be charged for the visit and any supplies used during the visit. On a related note, we are not able to put an uninsured patient into the system without first collecting their credit card information. We know this because there is a FK constraint on the uninsured table on the credit card number field. This constraint states that we cannot register an uninsured patient in the system if we do not have a credit card to tie them to so they can be charged for any visits.

We realized that our system could not differentiate between different visits. We need to create a Visit table in our database that would keep track of the individual visits, having foreign keys to different relations. VisitId will be the primary key. The table will store the date and time of the visit and the table will also have foreign keys referencing the patient who is visiting, the intake clerk, the service provider, and the nurse who assisted during the visit.

The Visit table will be associated with the ‘Logs’, ‘Orders’, ‘Primary Contact’, and ‘Completes’ relationships. The association with these relationships will enable us to differentiate between visits when it comes to examining when diagnoses were logged, when treatments were ordered, which guardian acted as the primary contact, and when an initial assessment was taken.

We would also add foreign key references of the specific visit id to tables like the Logs, Completes, Collects, and Orders table to differentiate between specific visits.

Our system allows us to track an insured patient's insurance information. It also retains that information for future visits - unless the patient themselves chooses to update their insurance information.

In order to check for if a patient can be seen by only one service provider per visit, we would need to be able to differentiate a specific visit. Since only one service provider is associated per visit this would satisfy the testing requirement. As of right now, we are associating only one service provider per diagnosis which is how I tested. We also cannot keep track of if a patient can be seen by different service providers per visit until we have our visit table in place. As of right now, we are associating one service provider per diagnosis which shows that a patient cannot have more than one service provider logging a diagnosis for them in a particular visit. And a patient can have multiple diagnoses per visit. A visit does not **need** to have a diagnosis.

It was initially assumed that additional tests/procedures would be identified as a separate visit. However, the current version of the system does not have a way of differentiating between specific visits for a patient. In the next iteration of system development, we would implement the concept of a Visit that has an attribute Date to help differentiate between visits. At this time, there would not really be a way to test this requirement, since the relation being tested does not exist in the system.