Subin Silwal

Prescreminder – Database Design

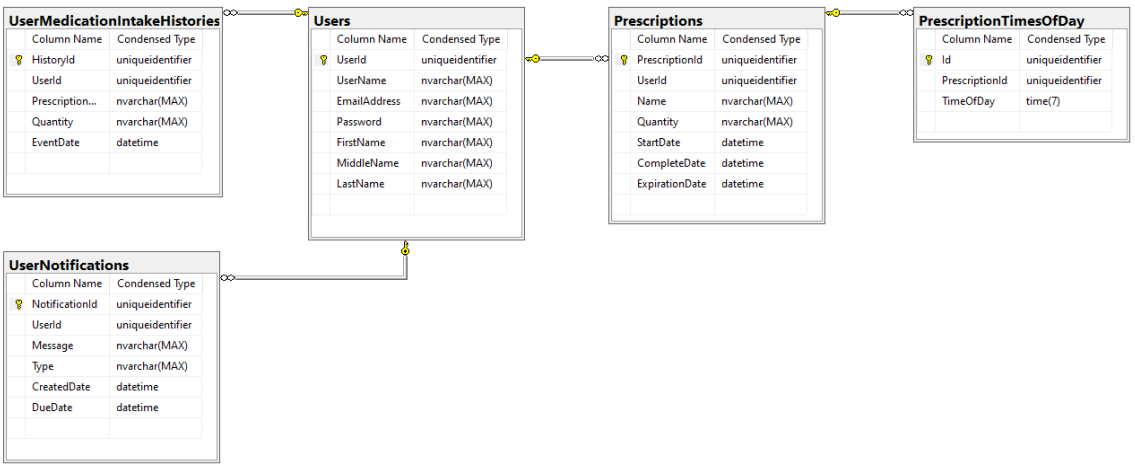
Database Technology Choice

For Prescreminder, I have chosen Microsoft SQL Database for data storage. Prescreminder will have a lot of normalized data. Thinking about the future of Prescreminder, beyond the MVP, I feel more of this is to come. Also, this is the database I am most familiar with, so I will avoid the learning curve for new technology.

Table Structures and Data Types

Below is the Entity-Relationship diagram for the tables that I would be using for Prescreminder. This is the diagram that has been generated from SQL Server Management Studio after creating these tables with required primary keys and foreign keys. The diagram shows every table, its relationships to other tables, column names, and their types.

I chose uniqueIdentifier as a type for all kinds of Ids. I have chosen nvarchar (MAX) for all the string types for simplicity.



Individual Tables Deep Dive

Table	Design and Purpose
Users	This is the primary table for the application. Users table will consist of user information such as name, email address, password, etc. Each user must have a unique email address, username, and userId.

Prescriptions	Users will have a one-to-many relationship with Prescriptions as a user can have multiple prescriptions. Each prescription will have a unique PrescriptionId and UserId is a Foreign Key in this context.
PrescriptionTimesOfDay	Prescriptions will have a one-to-many relationship with this table since a prescription might have to be taken multiple times of a day. Each record here will also have a unique Id and PrescriptionId is a Foreign Key in this table.
UserMedicationIntakeHistories	This table persists each medication intake information. Users table will have a one-to-many relationship to this table as a user will have a lot of histories associated with them. UserId is a Foreign Key in this table.
UserNotifications	This table will persist any notification that needs to be delivered to the user. For example, a soon to be expiring prescription. The reason why I choose to persist this is I want users to take some actions to make it go away vs. showing it just one time as users might miss it somehow. Users table has one-to-many relationship with this table. And UserId is a Foreign Key.

Deployment and User Accounts

As of now, I don't think I am going to create any user accounts. For MVP and simplicity, I think I will host SQL Server together with the application server, so nothing but the application server can access it. I believe the application server would be able to access the SQL server via windows authentication, eliminating the need to create a user account. I might figure out it is a bad idea later, but I want to see how it goes, and if I require some user accounts, I will create them later.