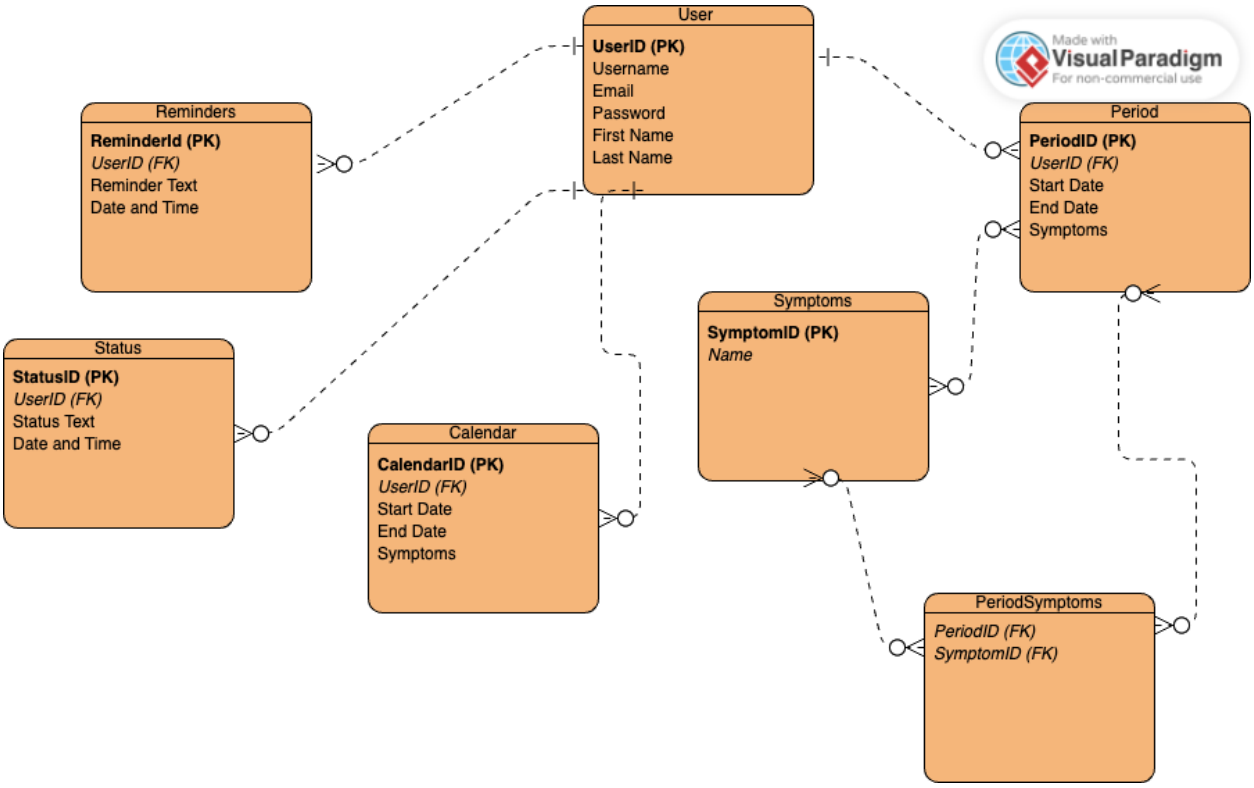
Kaycee Williams

Design Database

Blame Eve Web Application

I will be using PostgreSQL to persist the data in my Web Application. Initially I thought I would use MongoDB because I was a little more familiar with document based databases, but after further consideration, a relational database makes more sense for my application. I chose a relational database because I have several different types of relationships between my collections, and making queries will be more efficient using an SQL database. PostgreSQL is great for complex data structures which is important for modeling the data I will need for my application. PostgreSQL is also great for scalability, which will be essential for when users and all of their information is added to the database. PostgreSQL also supports complex queries, indexing, and full text searching.

ER Diagram



Collections:

- User:
    - Attributes: UserID (Primary Key), Username, Email, Password, First Name, Last Name
- Period:
    - Attributes: PeriodID (Primary Key), Start Date, End Date, Symptoms
- Reminders:
    - Attributes: ReminderID (Primary Key), User ID (Foreign Key), Reminder Text, Date and Time
- Status:
    - Attributes: StatusID (Primary Key), User ID (Foreign Key), Status Text, Date and Time
- Symptoms:
    - Attributes: SymptomID (Primary Key), Name
- Calendar:
    - Attributes: CalendarID (Priamry Key), User ID (Foreign Key), Start Date, End Date, Symptoms

Relationships:

- User-Period (One-to-Many):
    - A user can have multiple period records.
    - A period belongs to one user.
    - Foreign Key: User.UserID (in Period)
- User-Status (One-to-Many):
    - A user can have multiple status updates
    - A status belongs to one user.
    - Foreign Key: User.UserID (in Status)
- User-Reminders (One-to-Many):
    - A user can set multiple reminders.
    - A reminder belongs to one user.
    - Foreign Key: User.UserID (in Reminders)
- User-Calendar (One-to-Many):
    - A user can have multiple calendar events
    - A calendar belongs to one user.
    - Foreign Key: User.UserID (in Calendar)

- Period-Symptoms (Many-to-Many):
    - A period can have multiple associated symptoms.
    - A symptom can be associated with multiple periods.
    - This relationship is represented using a junction table.
    - Junction Table: PeriodSymptoms
        - Attributes: PeriodID (Foreign Key), SymptomID (Foreign Key)