Goal-e: Database Design

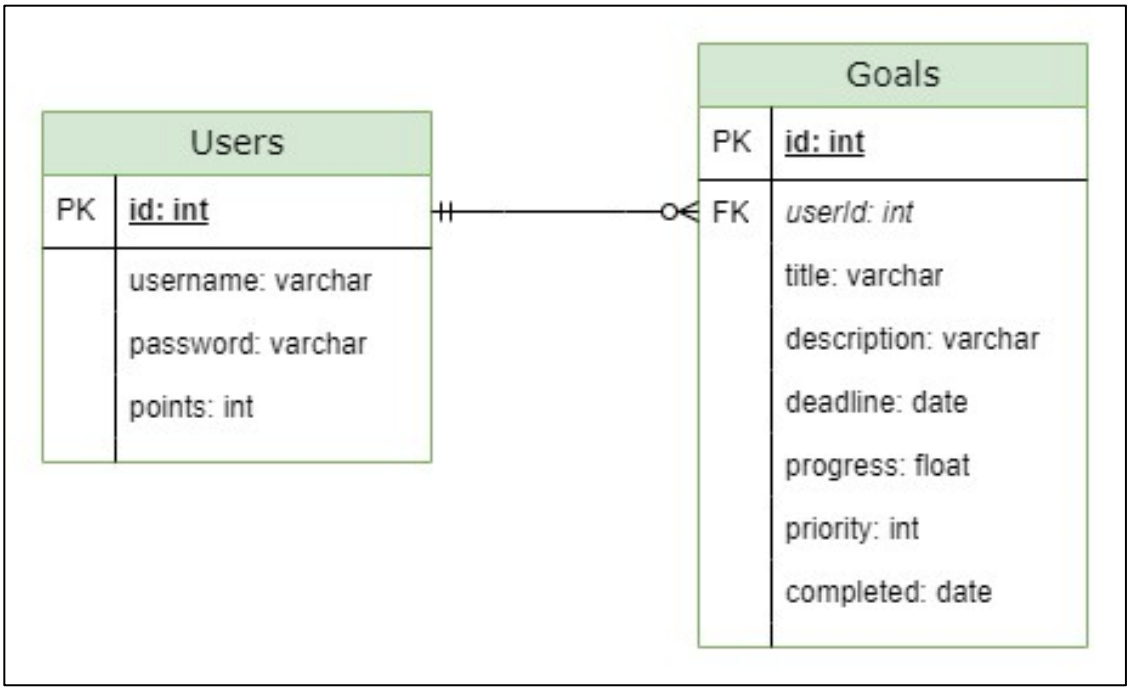
Maryville University

Fall 2023

Database Choice

I plan on using a relational database for my capstone project. Specifically, I plan on using PostgreSQL for my deployed application and SQLite while developing the application. I plan on using a relational database because they are reliable, and I can store the data in a normalized form. Also, users will have a one-to-many relationship with goals, in which a user can have any number of associated goals. The main database queries will involve this relationship, and it is straightforward to implement this, in a performance friendly way, using a relational database. I plan on using PostgreSQL for production because it is open source and scalable. I plan on using SQLite for development because it is simpler to set up on my computer.

Entity-Relationship Diagram



Users

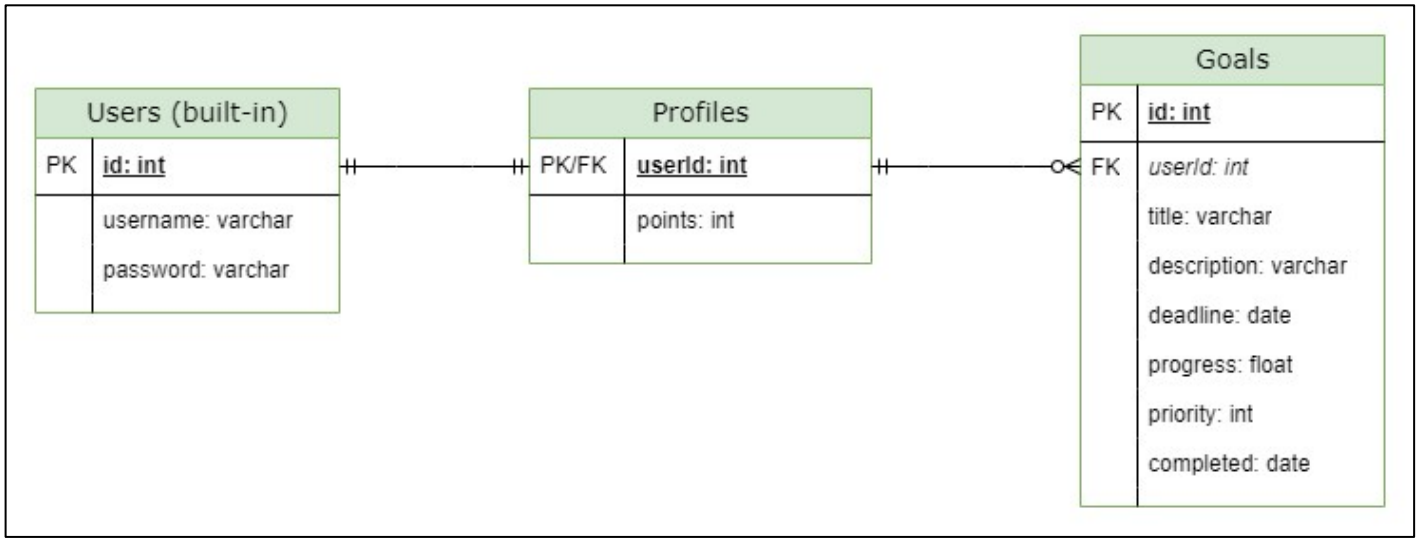
The users table will be used for storing and retrieving information about user accounts. Each user row entry will contain a unique id, username, password, and the number of points the user has earned. It is important to make sure the password is stored securely, such as by salting and hashing it before storage. This table could be expanded to include information such as email, a bio, date joined, and theme preference.

Goals

The goals table will be used for storing and retrieving information about goals. Each goal row entry will contain a unique id, as well as the id of the user that created the goal. The user id will be a foreign key that references the id column in the users table. This foreign key will be used in queries to get, update, and create the goals of a specific user. Each goal entry will also contain a title, description, deadline date, current progress, the priority of the goal, and the date completed. I plan on storing goal progress as a floating-point number between 0 and 100. For priority, I plan on giving users the option of ‘low’, ‘medium’, and ‘high’. I plan on storing this information as integers in the database, with 1 corresponding to ‘low’, 2 to ‘medium’, and 3 to ‘high’. The completed date will be a nullable column because a goal does not start out completed. This column can also be used to check whether a goal has been completed.

Implementation

I plan on defining and creating the above database schema using Django's ORM. Django has a built-in user class that is mapped to the application's database. The recommended way to expand this class is to create another class with a foreign key reference to Django's built-in user class. My plan is to use Django's user class for authentication and storing information such as username and password. I plan on expanding the built-in class, so that points can be stored, using the recommended strategy. As a result, my implemented entity-relationship diagram would look like the following.



Stretch Goal: User Settings / Dark Mode

For the additional feature of an account settings page with a dark mode, I would add a theme column to the users table. I would store integer values in the themes column, with a default value of 1 corresponding to ‘light’ and 2 corresponding to ‘dark’. I would store this information as integers, as opposed to Boolean values, to make it easier to expand with more themes. If I have enough time, I would like to implement this feature or implement it after the course is complete. Below is what my entity-relationship diagram will look like if this feature is implemented.