This application, PlantJournal, is intended to be used to keep track of plants and their growth needs, along with basic information about each plant. For each plant, the user can add what light, soil, humidity, water, and fertilizer it needs, the location the plant is in, whether the plant is toxic, non-toxic, or toxic to certain categories (such as only toxic to dogs), and what kinds of flowers, if any, the plant produces. Multiple notes can be attached to each plant, and these are intended to keep track of anything the user would like to write down regarding the plant. Finally, the user can add generic growth types, such as “new leaves” or “flower buds,” and these can be connected to any plant through growth instances. Growth instances are sorted by date, so the same plant can have multiple instances of new leaves, for example, and this sorting mechanism will make it easy to see patterns such as what time of year each plant grows new leaves, starts getting flower buds, and so forth.

This application was designed with two different user communities in mind. Firstly, for personal use—one person tracking their own plants, like their houseplants or garden plants. The second intended use is for a larger environment with multiple people taking care of plants, perhaps a horticulture department at a botanic garden.

I have identified four different groups and five different usernames with varying permissions. The groups are SuperUser, pj_admin, pj_user, and pj_guest. The corresponding usernames are sysadmin, admin, user, guest, and associate. For an individual user of this application, they would receive sysadmin (SuperUser) privileges. For a group setting/work environment, the sysadmin (SuperUser) would be in charge of the entire application, and would either represent the IT department or the whoever is in charge of the horticulture department. Admin (pj_admin) privileges would be given to team leaders, and user (pj_user) privileges would be given to team members. Guest (pj_guest) privileges are intended for anyone who needs to view the information in the catalog, but not change it. Associate privileges are for other people who could have access to the server, but are not part of the horticulture team, such as staff members in the administration department, who may use a different application on the same server.

Currently, sysadmin and admin have the same privileges—the ability to view, add, edit, and delete any information. I made them two separate categories in case I wanted to add more models to the application down the road, so I would have the option to give the admin different permissions than the sysadmin. As the application currently exists, however, the sysadmin and admin have identical privileges.

Users (the team members) can view everything, but only have additional permissions to add and change Plants, Notes, and Growth Instances. This decision was made because once information is added to the other models (such as Light, Water, etc.), additional types will very rarely need to be added. Most of the work for a large team would include working with the Plant objects, adding new Notes, and taking stock of new Growth Instances. Deletion privileges are also reserved for admin/sysadmin, to avoid errors, as rarely will a Plant, Note, or Growth Instance need to be deleted once it has been created.

Guests only have view privileges for everything, and associates can only see the About page.

For each of the five user IDs—sysadmin, admin, user, guest, and associate—the password is "(secret)."

To summarize the permissions and groups:

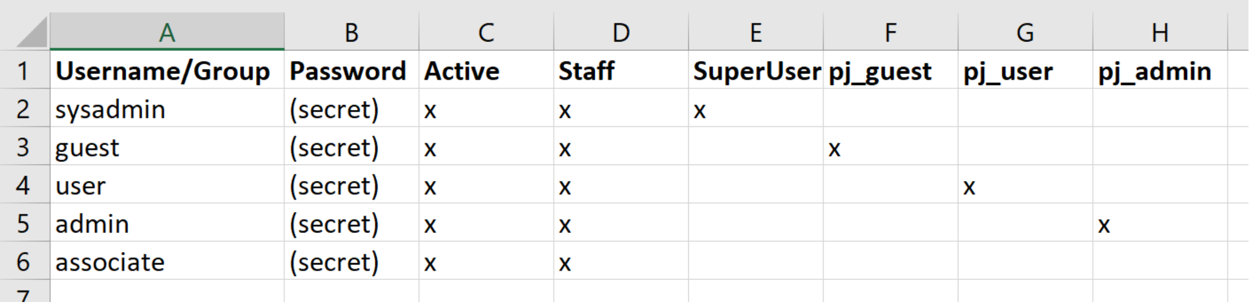
The sysadmin and admin should be able to do everything (view, add, edit, and delete all twelve models).

The user should be able to view everything, and additionally should be able to add and edit the models Plant, Note, and Growth Instances. The user should not be able to delete anything.

The guest should only be able to view everything.

The associate should only be able to see the About page.

Here is a screenshot outlining the username and group scheme.



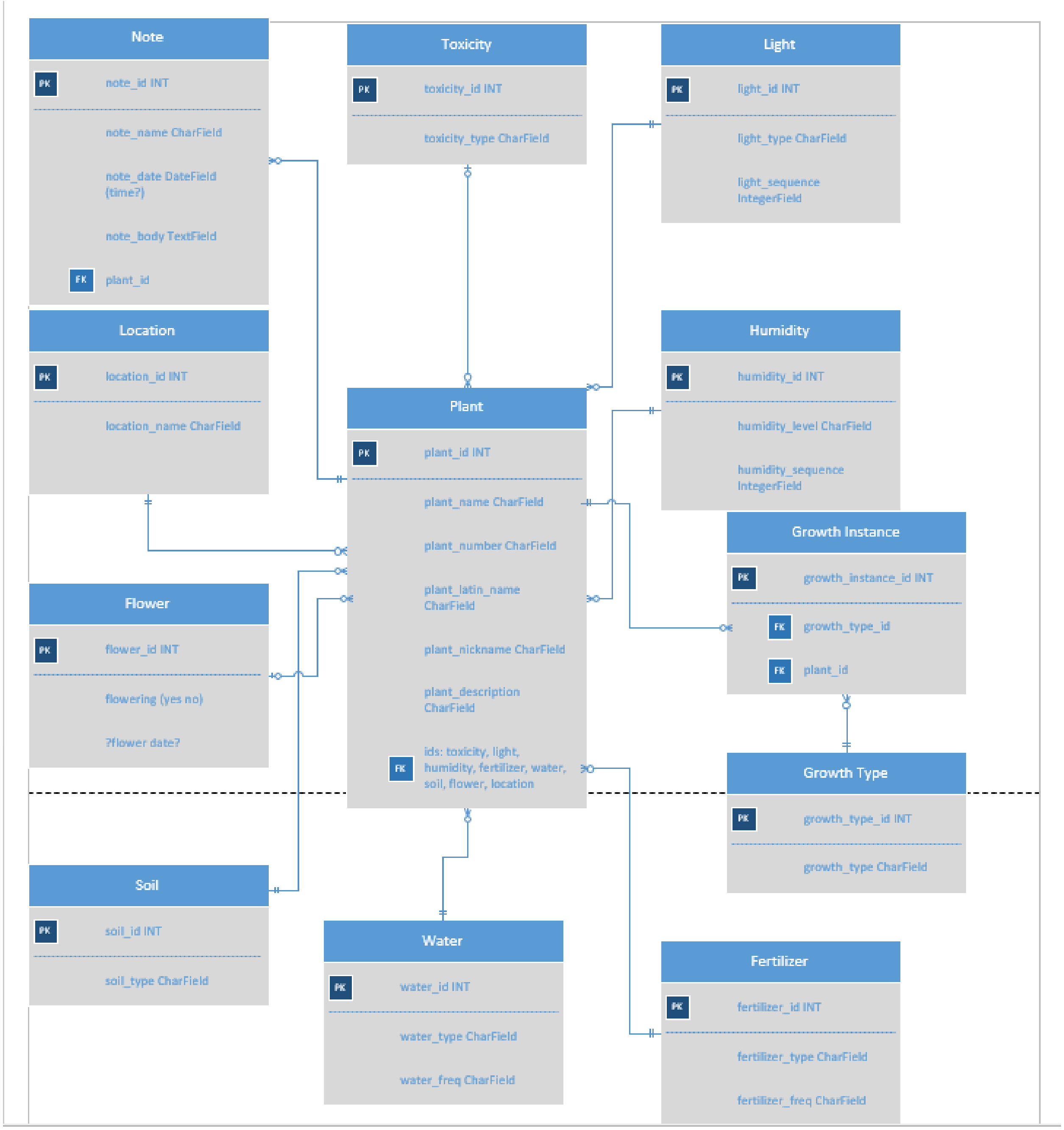
	A	B	C	D	E	F	G	H
1	**Username/Group**	**Password**	**Active**	**Staff**	**SuperUser**	**pj_guest**	**pj_user**	**pj_admin**
2	sysadmin	(secret)	x	x	x			
3	guest	(secret)	x	x		x		
4	user	(secret)	x	x			x	
5	admin	(secret)	x	x				x
6	associate	(secret)	x	x				
7								

I have left the test data that was migrated in for the Plant and the Note models to allow for pagination testing. These migrations should be able to be backed out if you would like to remove them after testing pagination, and are migrations 0009 and 0010. I have added a plant, ".Saguaro cactus 1 (Baby sag)" at the top of the Plant list, to avoid having to search through all the pagination while the test data exists. Currently, this plant has all of the fields filled in, and when editing it, these fields can be removed to test to see what it looks like when data is missing. I have created the Plant model so that the only required field is the Plant Name (or, a unique combination of Plant Name, Plant Number, and Plant Nickname). This means that when a new plant is created, only the name needs to be entered first, and all of the rest of the data can be added at a later time.

I also added a Note at the top of the Note List (as they are sorted newest to oldest by date) that was not part of the migrations for pagination testing. This note is connected to the .Saguaro plant object, and can be used for testing as well.

The first eight models (Light, Soil, Humidity, Water, Fertilizer, Location, Toxicity, and Flower) are all used exclusively by the Plant model. The Note model requires a Plant object. Growth Instance is a model that connects the many-to-many relationship between Plants and Growth Types.

For clarity, I am attaching a screenshot of my database model.



When creating new Watering Schedules and Fertilizer Schedules, only a frequency must be added—the type is optional for both. For any model that includes a required date field, at least these two entry formats are accepted: YYYY-mm-dd and mm/dd/YYYY. I added default values of 00/00/0000 to all the date fields to help users understand what format the dates should be in. No matter what format the dates are entered, they always display and sort as YYYY-mm-dd on the list pages.

In the static files folder, my style.css file shows an error in PyCharm, on line 60 (the url link for the logo). I am not sure why this error is appearing, because the logo shows up as desired. Additionally, I can make the error message go away by adding "/plantjournal" to the beginning of the path name, but this causes the logo to disappear from the webpage. Even though I am using my own image for the logo, I tested it with the logo Pinkham provided and that we used for EZ University, and the same behavior occurs with that logo.

I believe that is all the information that is necessary to test this application. I am pleased with how this application turned out, and I plan on continuing to improve and refine it even more. I would like to figure out a way to add pagination to some of the detail pages, since for example, under each plant detail page, it lists all of the notes associated with the plant. When adding note test data with migrations, I attached all the notes to a single plant, and this list of 150 notes is rather lengthy without pagination. I tried a few things to get this kind of pagination to work, and while I have not yet figured it out, that is one of the aspects I plan on working on in the future. I would also like to test this application with a few people and get some feedback on it to see how the database design works out. At some point I would like to try and implement a "history" feature that keeps track of all the changes for each plant—for example, it would provide a list of all the Locations that a plant has been in.