URL: http://flip2.engr.oregonstate.edu:7777/

Project Step 5 Draft

Team Name: The Leafettes, Group 9

Team Members: Tristan Harville, Lindsay Hawkins

a) Feedback by the peer reviewer part 4

1. **DMQ:**
Are the queries syntactically correct?

Yes, the queries are syntactically correct.

Are there queries providing all functionalities as required by the CS340 Project Guide ? What query is missing ? What needs to be fixed?

Yes.

Queries are missing for manipulating your M:M relationships between PlantsOwned and UserExpert.

There is a question mark on line 78 "SELECT tagName, tagDescription FROM Experts WHERE expertID = ?"

Do the queries cover the relationships as required by the CS340 Project Guide?

Yes, apart from the PlantsOwned and UserExpert M:M relationships. Otherwise, everything looks great!

DDQ:

Is the SQL file syntactically correct? This can be easily verified by importing/copy-pasting it in phpmyadmin. (Do not forget to take backup of your own database before you do this!)

Yes, I had no trouble importing.

Are the data types appropriate considering the description of the attribute in the database outline?

Everything looks great aside from the tagDescription attribute within the Experts table isn't consistent across the outline (text type) and the DDQ file (varchar type).

Are the foreign keys correctly defined when compared to the Schema?

Yes, everything is consistent.

Are relationship tables present when compared to the ERD/Schema?

Yes, both M:M relationship tables (PlantsOwned and UserExpert) are included.

2. **Data Manipulation Queries**

Are the queries syntactically correct? Disregard the part where input will be substituted as shown in the sample_data_manipulation_queries.sql

The queries are syntactically correct.

Are there queries providing all functionalities as required by the CS340 Project Guide? What query is missing? What needs to be fixed?

Yes, and you appear to be ahead of the curve for this step - there are SELECTs and INSERTs for every one of the main tables, and you have an UPDATE and DELETE function for one table. The only missing pieces are queries for the two intersection tables, PlantsOwned and UserExpert.

Do the queries cover the relationships as required by the CS340 Project Guide?

They do, aside from the the two aforementioned tables.

DDQ File

Is the SQL file syntactically correct?

Yep, I imported the file without issue.

Are the data types appropriate considering the description of the attribute in the database outline?

There was only one issue as far as I could tell. In the Experts table, the tagDescription in the outline is listed as text, but is a varchar(255) in the database dump.

Are the foreign keys correctly defined when compared to the Schema?

Yes, the foreign keys are correctly defined.

Are relationship tables present when compared to the ERD/Schema?

Both PlantsOwned and UserExpert are in the dump and have correctly defined FKs.

3. **Are the queries syntactically correct? Disregard the part where input will be substituted as shown in the sample_data_manipulation_queries.sql**

Yes.

Are there queries providing all functionalities as required by the CS340 Project Guide ? What query is missing ? What needs to be fixed?

Line 11 in DMQ_step4_draft.sql is a bit confusing to me, because I am not sure how the site will function. Based on the query, a user has to select a plant type and enter a common name to do a filtered search. The other option for the user is to only enter in the science name.

It seems like the user should be able to select a plant type and enter a common name to do a filtered search, which is there. However, he/she cannot also select a plant type and enter a science name, since the where type = part of the query is attached to only the commonName = part. Though, this may be intentional and part of the website design in which case it is fine the way it is.

This is more for the next step, but what happens when a user doesn't filter by a type and just leaves it as all? Wouldn't a different query be needed that doesn't have a where type = part? Or, is it possible to do a query for a type that equals "All"?

Line 78 has a ? instead of a semicolon and saying where the expertID will come from. I'm assuming this means the group is unsure of where the id will come from currently.

Do the queries cover the relationships as required by the CS340 Project Guide?

It's hard to interpret the rubric since entity/table is used interchangeably. So, I don't know if select/insert into queries are needed for the UserExpert and PlantsOwned tables. (We added those queries for our composite entities to be safe.) If no, then yes all relationships are covered. If yes, then those ones are missing.

Is the SQL file syntactically correct? This can be easily verified by importing/copy-pasting it in phpmyadmin. (Do not forget to take backup of your own database before you do this!)

Yes, I was able to import it with no issues.

Are the data types appropriate considering the description of the attribute in the database outline?

Yes. One small note, tagDescription in Experts in the actual sql file should be text instead of varchar, or in the outline it should be varchar instead of text.

Are the foreign keys correctly defined when compared to the Schema?

Yes.

Are relationship tables present when compared to the ERD/Schema?

Yes, PlantsOwned and UserExpert are both present and correct.

4. Data Definition Queries Is the SQL file syntactically correct? Yes, the SQL file is syntactically correct as it validates in phpmyadmin. Are the data types appropriate considering the description of the attribute in the database outline? Most attributes are correct; I will list the tables and any concerning attributes below: Users.picture is defined as varbinary(8000), I may be incorrect about this, but I believe varbinary(max) allows for storage of files larger than ~8 KiB, which would be considered small for a picture. Plants.maxSize is defined as a varchar(). For data integrity purposes, I wonder if a DECIMAL or NUMERIC data type would be more appropriate, as that is the type of data this column will store. I would provide an example, but I'm not sure what kind of precision they required. I have the same concerns for Plants.picture that I do for Users.picture. Guides.description, is there a reason to limit the text to 255 chars? I've never seen this syntax before, and based on the sample data provided, it seems as if it's not a description of a guide but actual instructions. I would remove this limit. I have the same concerns for Care.humidity that I do for Plants.maxSize. Are the foreign keys correctly defined when compared to the Schema? Yes, all foreign keys are correctly defined compared to the schema, including the appropriate participation for each relationship. Are relationship tables present when compared to the ERD/Schema? Yes, all relationship tables are present compared to the ERD/Schema; however, both PlantsOwned and UserExpert do not specify a primary key. Data Manipulation Queries Group 9 - Project Step 4 Draft Review 2 Are the queries syntactically correct? Most queries are correct, and I believe only "(CREATE/INSERT) Insert a new plant into the plants table using the form" does not insert the careID, which would likely cause a SQL error. However, I did notice the following semantic inconsistencies: The search filter for plants may not return what this team is expecting, as AND takes precedence over OR in a WHERE clause. So the grouping would actually be: (type = :plant_type_selected_from_radio_buttons AND commonName = :name_entered_in_search_bar) OR scienceName = :name_entered_in_search_bar; I don't see a way to add or update a User's picture, as the column is not specified in the INSERT or UPDATE I don't see a way to add or update a Plant's careID, as the column is not specified in the INSERT or UPDATE Are there queries providing all functionalities as required by the CS340 Project Guide ? What query is missing ? What needs to be fixed? For the non-intersection tables, this team has met the requirements, but there is some missing functionality in these statements. I do

not see a way to INSERT into UserExperts or PlantsOwned, nor do I see how to SELECT from these tables. These missing operations are a problem as a) every table should have an INSERT or be involved in a select and b) It should be possible to add and remove things from at least one many-to-many relationship. There is also an unfulfilled requirement DELETE requirement for a many-to-many relationship. Adding or removing from either of these relationships would be as simple as inserting or deleting the appropriate userID and plant/expertID. Do the queries cover the relationships as required by the CS340 Project Guide? The queries do cover the relationships as required in the project guide, as there are two many-to-many relationships and two one-to-many relationships. The relationships are not utilized in any JOINs, which might provide more meaningful data in some of the pages.

b) Actions based on feedback part 4

For this iteration we made sure we had all of our queries for the CRUD functionality on all tables including the intersection tables. We fixed the forms to make sure data types were consistent. We finished the query that ended with a question mark.

c) Feedback by the peer reviewer part 3

1. Overall, I think this is a great idea for a website and the name is great. I could have used information like this before I killed some plants by mistake! Best of luck!

   Does the UI utilize a SELECT for every table in the schema?

   I don’t think there are SELECT for every table in the schema, for example, no SELECT for the PlantsOwned table

   Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

   I do not see a search or filter option in the website. This could be a great function for the plants or guide page, as users may want to search for a specific plant they are trying to grow.

   Does the UI implement an INSERT for every table in the schema?

   Yes, each table has an INSERT option on the website.

   Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?

   I am not sure if each new plant insertion its tied to a specific user. If that is the case and you can only add a plant is you are a registered user, then yes this its correct.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?

Assuming that plants can only be added by registered users, then yes this is in the website.

Is there at least one UPDATE for any one entity?

Since there is no data in the website yet, I do not know if there is an update option for an entity. I am willing to bet that it is an option.

Is at least one relationship NULLable?

Yes, each plant does have to have a care guide for it.

Do you have any other suggestions for the team to help with their HTML UI?

I think the idea of this website is great since many of us have probably bought quarantine plants. I would suggest some styling (CSS) updates to make it more visual appealing and maybe adding images of each type of plant so if people don't know what they have, they could reference those images.

2. **Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.**

   Yes, each page and table will utilize a SELECT query for displaying data.

   Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

   I do not see any search/filter functionality.

   Does the UI implement an INSERT for every table in the schema?

   Yes, every table can insert an individual value.

   Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?

The inserts do include M:M relationships and some of them do use FK attributes, but not everything uses FK attributes and doesn't need to according to the project schema.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?

Yes at least one, for every table not only the M:M relationships.

Is there at least one UPDATE for any one entity?

There is no current functionality to edit/update an existing field. This can be easily added.

Is at least one relationship NULLable?

There are many nullable relationships.

Do you have any other suggestions for the team to help with their HTML UI?

I think all you guys have to add is a search/filter, and also the ability to edit fields that are already present, and I think the project will be good to go then!

3. Hi, that is the most clever name I have seen out of all the projects, props to whoever came up with that.

   Does the UI utilize a SELECT for every table in the schema? **In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.**

   I do not see a UI display for every table in the schema. The tables that I see in the schema but do not see a UI display for are the following: Plants Owned, User Experts, and Care. If these displays are hidden behind a function it may be a good idea to make them more accessible so that the users don't have to dig through the pages to find them.

   Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

   I do not see any search/filter functionality with a dynamically populated list of properties.

***Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table.**

There is an INSERT implementation for the tables that are on the page which are Users, Plants, Guides, and Experts, however no pages therefore no INSERT implementation for Plants Owned, User Experts, and Care.

***Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).**

The intersection tables do not have a UI implementation therefore I am unable to see how the FK attributes will interact with the tables in the schema. I recommend to create the html pages for the intersection table to show this interaction. It describes the interactions in the schema, you will just have to implement a UI for it.

***Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.**

I see an explanation for the plan to incorporate a DELETE function but do not see a DELETE function that can remove items from the database in the UI. However, this will not be too difficult to implement. I will add a link that describes how to use javascript to show this function in the UI: https://stackoverflow.com/questions/5933157/how-to-remove-an-html-element-using-javascript. You can add a function like "DELETE ROW" as a separate column for each row.

***Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?**

Similar to the DELETE function, I see a section that describes the intent to add an UPDATE function for Users but I do not see any function on the UI to update their profile information. I added a link that might help with this function.

https://www.developerdrive.com/allowing-users-to-edit-text-content-with-html5/.

Is at least one relationship NULLable? **In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.**

Yes, there is one relationship that is NULLable between a plant and a care plan. Having a care is optional for a plant.

Do you have any other suggestions for the team to help with their HTML UI?

I would recommend having the function of a page be displayed from load, there are some functions that are hidden by clicks and actions. However, this will be hard for users to navigate through as they don't know how to reach those parts. I recommend having everything shown from the moment that page is loaded.

Great concept for the project, good luck!

4. ***Does the UI utilize a SELECT for every table in the schema?*** **In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.**

   I'm not certain I am seeing a specific SELECT for experts, however they are involved with the site. I could be missing where the SELECT would be used though.

   Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

   I am not seeing search/filter functionality.

   Does the UI implement an INSERT for every table in the schema? **In other words, there should be UI input fields that correspond to each table and attribute in that table.**

   Each table can be inserted in to.

   Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? **In other words if there is a M:M relationship between Orders and Products,**

INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

I believe that users will be tied to experts and plants by using the login functionality, so I think the answer to this question is yes.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? **In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.**

Yes, users are able to delete plants that they own, which is a M:M relationship.

Is there at least one UPDATE for any one entity? **In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?**

Users are able to update their profile information.

Is at least one relationship NULLable? **In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.**

Yes, a plant can exist without a care plan.

Do you have any other suggestions for the team to help with their HTML UI?

Be sure you have a SELECT for experts, as well as a search and filter option for an entity. Some dummy data may help with navigating the site, but its not a huge deal. Other than that, seems like you are on the right track!

d) Actions based on the feedback part 3

The biggest change we made from the feedback for part 3 was adding a search and filter feature to the Plants page. We had previously missed this requirement, so we did not show any sort of intention to add a search feature. We also did not have selects for every table in the database, only for entity tables. We have since added selects for all entities and the intersection tables in the admin section of our website. One reviewer recommended adding CSS styling and Pictures to the website, we did not add them for this iteration but will be adding them for the final project. We also added an admin page that has delete and update functionalities for all tables in the database. We did not originally realize that the website was not supposed to be user facing, so we still have all of our user facing pages because we did not have enough time to change the website to be solely admin facing before the deadline. The admin page is a draft of how we intend for the features of the website to look going forward. We will be implementing insert functionality to this page eventually.

e) Feedback by the peer reviewer part 2

Hello Lindsay and Tristan! This is a very neat and unique project idea. Overall, I don't have much feedback as it looks like your group has everything pretty much in line for this project. I did notice a few areas of opportunity. In your outline, under the Care entity, the Humidity attribute is still listed as decimal instead of varchar, don't forget to update that as you will now be using "high, medium, and low" instead of a percentage. Also under the Care entity, I had a hard time understanding the Water attribute. Is the Water attribute an int to indicate something like (int) ounces of water per day? Maybe consider adding a small description to better understand the Water attribute. Lastly, on your outline, you have the relationship between Users and Experts listed as a M:M, but on the ERD and Schema it reflects a 1:M, consider updating either the ERD/Schema or Outline so that those match. Great job so far, best of luck the rest of the way!

Your overview directly states the problem and how a website with a database backend can solve it. The overview is very thorough, and left me with no questions. Specific data was mentioned for the motivation of the website and clearly communicated the expected size of the database. There are at least four entities and the proper number and type of relationships that are needed. All entities include their purpose and attributes with data types. The outline clearly states what each team member is suppose to implement. All 1:M relationships are implemented correctly and there is at least one M:M relationship. All entities and attributes stick to a standard naming convention. I really have no questions or corrections to suggest about your project. Your outline looks great and is very clear and easy to follow.

Nice work! The problems that need to be solved by a website with DB back end is clearly indicated in the documents. The overview also talks about the research results from the interview that average users have at least 12 planets. At least four entity details are clearly listed in the draft version. For every entity, the ideas are clearly indicated and listed. The attributes data types and constraints are also clearly indicated for each entity. The M:M relationship intersection table has also been declared in the documents. Just a friendly reminder that we just need to implement one M:M relationship this course, so for the people assigned for each implementation may need to adjust. Choose one M:M relationship and maybe convert the other one into 1:M. There are totally 2 1:M relationships, and 2 M:M relationships. The PK and FK are clearly labeled for each entity and the names consist of the entities ER diagram and schema. However, I believe the table in the schema also needs the relationship between each table. That is the arrow pointing from the PK to FK for each table. Your schema are still showing the ER diagram relationship, the table interaction will be more sufficient for the schema part. Although, decent work and hope you guys good luck on the final draft!

Does the overview describe what problem is to be solved by a website with DB back end? -Overview was extremely detailed and explained what the DB was solving 2. Does the overview list specific facts? -Lots of specific facts were mentioned about demographic and consumers. 3. Are at least four entities described and does each one represent a single idea to be stored as a list? -Yes, 7 entities were described 4. Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? - Each entity includes detailed description about the entity, lists the attributes, datatypes, and relationship is included. 5. Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)? -Outline clearly describes implementation of entities 6. Are 1:M relationships correctly formulated? Is there at least one M:M relationship? - 1:M relationships are clearly described and 2 M:M relationships exist 7. Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming? - Naming convention is mostly correct. I was just wondering if Care should be Cares just to follow the plural entity naming convention but then the name of the entity doesn't really make sense anymore. So not too sure about that one.

f) Actions based on the feedback part 2

Based on the feedback from the peer reviews we changed the error where the humidity attribute was still listed as a decimal instead of a varchar in the database outline section. We also changed the water entity from int to varchar as we felt telling users how frequently to water ("once a week") would overall be more helpful than how much to water. We updated the schema and ERD to reflect the M:M relationship between Users and Experts. We did not make any changes to the relationships as suggested by one reviewer because although only one M:M relationship is required, our understanding is that we can have more than one, so we will keep it as such until told otherwise by the grader. Likewise, we also are not changing the format of the schema at this time as this reviewer suggested because we formatted it after the professor's response to question #149 on Ed discussions. Finally, we also decided not to change the name of the Care entity to Cares as one reviewer suggested as the grader did not tell us to change it on our original submission and we believe Cares does not make sense.

Upgrades to the Draft version

We did not make any changes to the file based on our own changed design decisions.

g) Fixes based on Feedback from Step 1:

In Step 1 of our project, our grader recommended that we change one of the relationships within our project. The relationship that was recommended to be changed was a recursive M:M relationship with a single entity (Users). The idea behind this relationship was to create a social media aspect to our application where users could follow other users. This would be similar to Twitter or Instagram. It was recommended we remove the relationship as it would be difficult to implement.

Our group decided to heed the advice and change this relationship. To replace the relationship, our project now contains a Users entity and an Experts entity. Users will be allowed to have a M:M relationship with the Experts entity. The way this will work is a user will be allowed to be an expert in multiple fields, and these fields may have multiple users who identify as an expert in that particular field.

If a user is an expert in a field, then they will have a special tag that will identify them as an expert, which other users will be able to observe.

Another simple change our group decided to implement is changing the humidity from a decimal type to a varchar type. This is because our group decided to use “high, medium, and low” for a humidity specification instead of a humidity percentage. We feel that this will help Users better understand what is needed for a plant in terms of humidity.

Finally, our group decided to add another relationship to this project. In order to allow our Users to view the profile of another user, we have decided to implement a relationship between Users and Guides. This will allow Users to view the profile of a member who has written Guides. The reasoning behind this is so a user can view other Guides that someone has written, if they have created more. This feature will also be the only user-to-user interaction in our application as of now.

h) Project Outline and Database Outline - Updated Version:

Team Member Names: Lindsay Hawkins and Tristan Harville

Project Title: Plant Parenting Web App

Problem to Be Solved: This web application will be based on an app that was designed in part by team member Lindsay Hawkins in CS 352 Introduction to Usability Engineering. The app prototype can be accessed through the following link: Plant Parenting Prototype. During the user research stage of development, the CS 352 team found that 83% of interviewees had to consult multiple resources to find reliable care information about their houseplants and 50% of interviewees stated that the information they obtained from different sources often conflicted [1]. The Plant Parenting web app aims to solve the problem of individuals with houseplants having to search multiple sources that may conflict by providing a consolidated and reliable source for care information specific to the plants each user owns. The application will accomplish this by allowing User entities to record Plant entities that they own, which will be associated with Care and Guide entities. The user research found that on average users had 11.8 plants, so the database should support users having at least 12 [1]. Two interviewees did have more than 12 plants, so allowing only 12 plants would likely not be sufficient on a large scale [1]. The maximum number of plants owned by a single user found in the research was 29[1]. There are approximately 33 million U.S. households with one or more houseplants, so the database should support as many User entities as possible [2]. There are also thousands of varieties of houseplants and endless options for guides, so the Plant and Guide entities should also support as many entries as possible.

DATABASE OUTLINE

- **Users**: Entity which carries details about each user in the database. Users will be allowed to put their planting zone and a profile picture of themselves to personalize their account. Other information this entity contains is basic user information that will be displayed on their account in order for other users to identify them (first name, last name, and email).
    - userID: int, auto-increment, not NULL, PK
    - first: varchar, not NULL
    - last: varchar, not NULL
    - email: varchar, not NULL
    - location: varchar
    - picture: varbinary
    - **Relationship:** a M:M relationship with Plants will be implemented using the userID as a FK inside of the PlantsOwned Relation. Users will be allowed to "own" plants and have their experience tailored to them based on the plants that they own. The plants owned will also be viewable by other users.
    - **Relationship:** a M:M relationship with Experts will be implemented in which Users can be Experts in a field or fields. This will be implemented using the UserExperts table with userID as an FK in the table. A user may be an expert in multiple things (house plants, trees, etc.). Multiple Users will also be allowed to hold the same tag (i.e. multiple users can be an expert in the same thing).
    - **Relationship**: a 1:M relationship with guides will be implemented between Users and Guides. This will be implanted using the userIDas a FK inside of Guides. This will have partial participation with Guides, since a user does not have to author a guide. This relationship will allow a user to be the author of a guide.

- **Plants**: Entity which carries details about each plant in the database. All that will be required is the plant's common name. Plants can have much more detail though. The science name will be used to give the scientific name of a plant. Type describes the category this plant belongs to (i.e. succulent, foliage, flowering, etc.). The color attribute will be used to specify the flower's primary color. Only one color will be accepted per plant. Variegated will be used to describe if the plant has additional coloring. The petSafe attribute will tell if the plant is toxic to animals (If it is toxic to one household animal, it will be considered toxic). maxSize describes the maximum height of a plant. This will be stored precisely. Optionally, one picture of the plant can be added.
    - plantID: int, auto-increment, not NULL, PK
    - commonName: varchar, not NULL
    - scienceName: varchar
    - type: varchar
    - color: varchar
    - variegated: bool
    - petSafe: bool
    - maxSize: varchar
    - picture: varbinary
    - careID: int, FK
    - **Relationship**: a 1:M relationship with guides will be implemented between Plants and Guides. This will be implanted using the plantID as a FK inside of Guides. This relationship will display guides related to a specific plant to users.

  - **Relationship**: a M:M relationship between Plants and Users will be implanted using the plantID as a FK inside of the PlantsOwned Relation. Users will be allowed to “own” plants and have their experience tailored to them based on the plants that they own. The plants owned will also be viewable by other users.
  - **Relationship**: a M:1 relationship will be implemented between Plants and Care. The careID will be used as a FK in Plants. Each plant will have general care guidelines which will be linked by this relationship.

- <u>**Experts**</u>**:** Entity which contains different types of “expert” tags that a user can have. A verified user will receive an expert tag. The attribute “tagName” will refer to the type of expert a user can be. The tagDescription refers to the description of the tag that a user will hold.

  - expertID: int, auto-increment, not NULL, PK
  - tagName: varchar, not NULL
  - tagDescription: text, not NULL
  - **Relationship:** a M:M relationship with Users will be implemented in which a User can be Experts in many different fields. This will be implemented using the UserExperts table with expertID as an FK in the table. This relationship will give a tag to the Users allowing other Users to know if someone is an Expert in a particular area.

- <u>**Guides**</u>: Entity that contains guides about plants, techniques, or anything else that is associated with plants. A guide can optionally be directly linked to a specific plant if it applies directly to that plant (e.g. a care guide for that specific plant). Guides will not be able to be created unless they at least have a title and a description. The description of a guide can be the whole guide itself. Videos will be links to other video hosting platforms and will be used to embed the video into the guide. There will be a limit of one video per guide.
  - guideID: int, auto-increment, not NULL, PK
  - title: varchar, not NULL
  - video: varchar
  - description: text, not NULL
  - plantID: int, FK
  - userID: int, not NULL, FK
  - **Relationship**: a M:1 relationship between Guides and Plants will be implemented. This will have partial participation with Plants, as a guide does not have to be assigned to a specific plant and can be considered “general”. If a guide is to be used for a specific plant, then the plantID will be used as a FK in Guides.

  - **Relationship**: a M:1 relationship between Guides and Users will be implemented. All guides will have a user that is the author. The userID from Users will be a FK in this table to keep track of guides and their authors.

- <u>**Care**</u>: Entity that describes different types of care for plants. These entities are a set of general care requirements for a group of plants. The care entity can be used for many plants, as it is often the case that plants share similar requirements for health. The required components to create a care data set will be water, light, and temperature. Humidity will be a decimal in the range 0 to 1, which will be used to make a percentage. Soil and fertilizer will be one per an attribute, and will be used to describe plant nutrition.

    - careID: int, auto-increment, not NULL, PK
    - water: varchar, not NULL
    - light: time, not NULL
    - temperature: int, not NULL
    - humidity: varchar
    - fertilizer: varchar
    - soil: varchar
    - **Relationship**: 1:M relationship between Care and Plants, where a set of general care instructions can be used for multiple plants. This will be implemented using the careID as a FK inside of Plants.

- **<u>PlantsOwned</u>**: a table of all plants that are associated with certain users. This is used to keep track of the relationships created by users and the plants they choose to "own."
    - userID: int, not NULL, FK
    - plantID: int, not NULL, FK

- **<u>UserExpert</u>**: a table of all users who are experts in a certain area of botany. This table is used to track the relationships created between Users and Experts. A user may hold many expert tags and an expert tag may be held by many users.
    - userID: int, not NULL, FK
    - expertID: int, not NULL, FK

IMPLEMENTATION

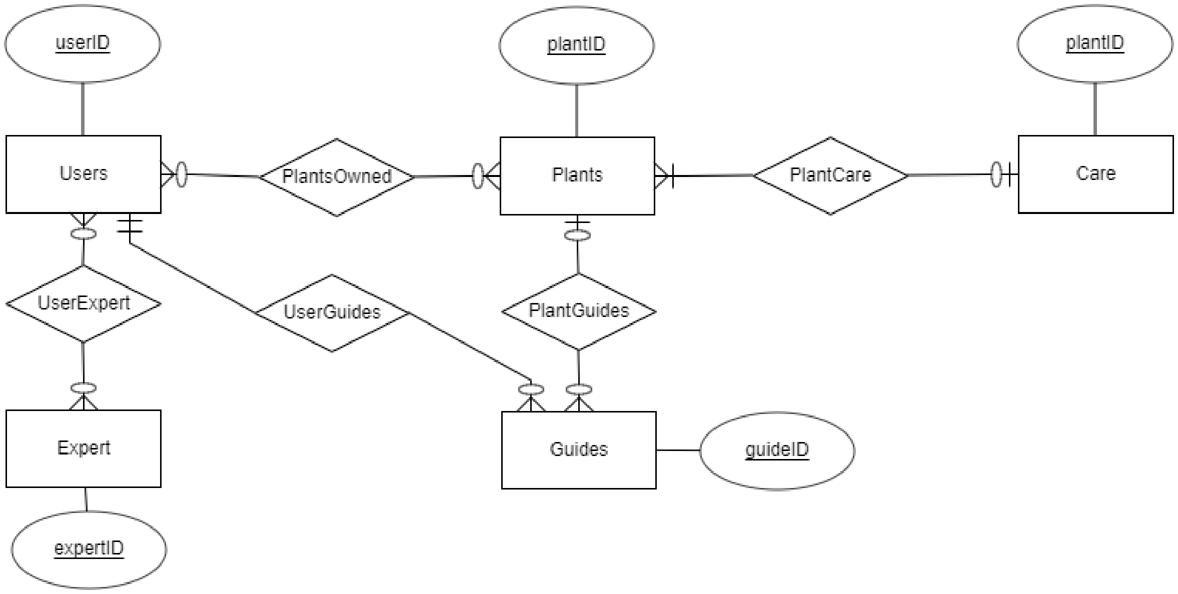
Our group has decided to split the implementation work as follows:

- Lindsay
    - Create the tables for the Plants, Guides, and Care entities
    - Create Plants webpage, which will show all the available plants for viewing
    - Create Care webpages, which will show the care specifications for each plant.
    - Help create the homepage for the website.
- Tristan
    - Create the tables for the Users, Experts, PlantsOwned, and UserExpert entities
    - Create Users webpage, which will display information about a user and guides they have authored
    - Create Guides webpage, which will display Guides that Users have written.
    - Help create the homepage for the website

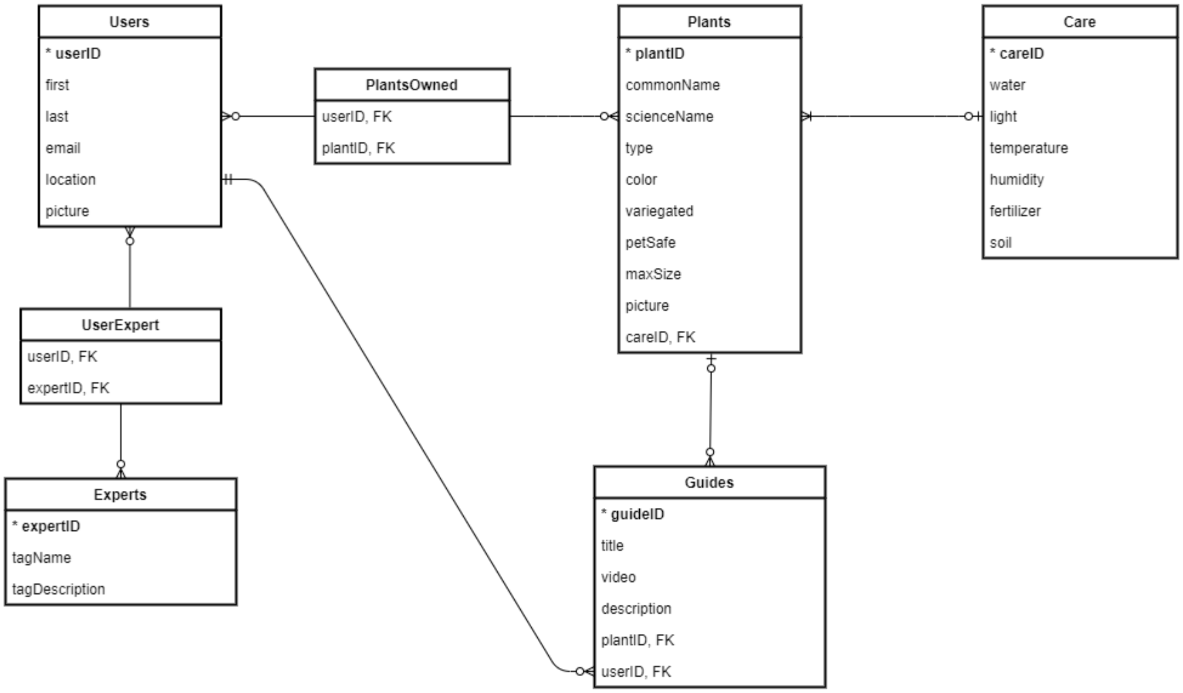
REFERENCES

1. Feth, Chris, et al. 2021. "CS352 Module 4, 2021, Plant Parenting — Fomalhaut".
2. Statista. 2019. "Indoor houseplant gardening participation in the United States from 2010 to 2019". https://www.statista.com/statistics/716535/us-houseplants-participation/ Accessed 08 04 2021.

i) Entity-Relationship Diagram:



j) Schema:


Users
* userID
first
last
email
location
picture
PlantsOwned
userID, FK
plantID, FK
Plants
* plantID
commonName
scienceName
type
color
variegated
petSafe
maxSize
picture
careID, FK
Care
* careID
water
light
temperature
humidity
fertilizer
soil
UserExpert
userID, FK
expertID, FK
Experts
* expertID
tagName
tagDescription
Guides
* guideID
title
video
description
plantID, FK
userID, FK