https://dailei5.github.io/index.html

Weather Vault ®

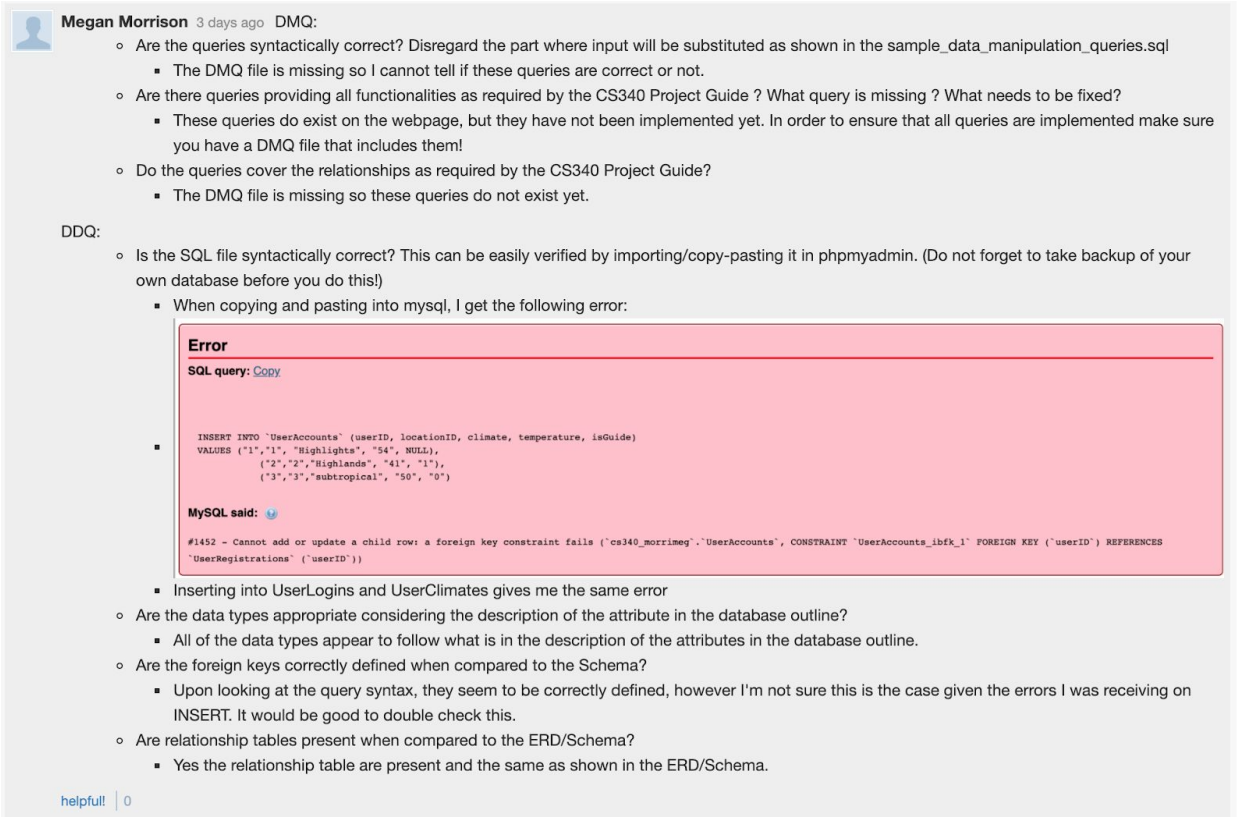
Project Step 5

CS340
Group106
Developers
Dov Sherman
Lei Dai

Fixes based on Feedback from Previous Steps:
Further to the feedback we received, we have:

- included everything we've written from our previous steps (project/database outline, feedback you received, ERD/schema, etc.).
- We now have submitted two query files: one for queries that generate our database, and one for queries that deal with user input

Feedback by peer reviewers from Step 4:



Megan Morrison 3 days ago DMQ:

- Are the queries syntactically correct? Disregard the part where input will be substituted as shown in the sample_data_manipulation_queries.sql
  - The DMQ file is missing so I cannot tell if these queries are correct or not.
- Are there queries providing all functionalities as required by the CS340 Project Guide ? What query is missing ? What needs to be fixed?
  - These queries do exist on the webpage, but they have not been implemented yet. In order to ensure that all queries are implemented make sure you have a DMQ file that includes them!
- Do the queries cover the relationships as required by the CS340 Project Guide?
  - The DMQ file is missing so these queries do not exist yet.

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!)
  - When copying and pasting into mysql, I get the following error:
  - **Error**
    SQL query: Copy

```
INSERT INTO `UserAccounts` (userID, locationID, climate, temperature, isGuide)
VALUES ("1","1", "Highlights", "54", NULL),
          ("2","2","Highlands", "41", "1"),
          ("3","3","subtropical", "50", "0")
```

    MySQL said:

```
#1452 - Cannot add or update a child row: a foreign key constraint fails (`cs340_morrimeg`.`UserAccounts`, CONSTRAINT `UserAccounts_ibfk_1` FOREIGN KEY (`userID`) REFERENCES `UserRegistrations` (`userID`))
```

  - Inserting into UserLogins and UserClimates gives me the same error
- Are the data types appropriate considering the description of the attribute in the database outline?
  - All of the data types appear to follow what is in the description of the attributes in the database outline.
- Are the foreign keys correctly defined when compared to the Schema?
  - Upon looking at the query syntax, they seem to be correctly defined, however I'm not sure this is the case given the errors I was receiving on INSERT. It would be good to double check this.
- Are relationship tables present when compared to the ERD/Schema?
  - Yes the relationship table are present and the same as shown in the ERD/Schema.

helpful! | 0

Virginia Link 2 days ago

- DMQ:
  - Are the queries syntactically correct? Disregard the part where input will be substituted as shown in the sample_data_manipulation_queries.sql
    - I didn't see a file with DMQs, so no
  - Are there queries providing all functionalities as required by the CS340 Project Guide ? What query is missing ? What needs to be fixed?
    - (Same as above)
  - Do the queries cover the relationships as required by the CS340 Project Guide?
    - (Same as above)
- 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!)
    - It appears you have some errors when inserting the dummy data as described in Megan's feedback. This has to do with the foreign keys between tables so I would look at disabling and then enabling the foreign key checks with the command "SET FOREIGN_KEY_CHECKS = 0;" at the begining and then "SET FOREIGN_KEY_CHECKS = 1;" at the end, or something similar to allow your to insert data.
  - Are the data types appropriate considering the description of the attribute in the database outline?
    - It looks like the data types you used are appropriate for the data and match your description.
  - Are the foreign keys correctly defined when compared to the Schema?
    - It seems you have all foreign keys set up correctly
  - Are relationship tables present when compared to the ERD/Schema?
    - I might just be missing it but it looks like you don't have any M:M relationships, so I don't see any relationship tables.

helpful! | 0

Jose Candelario 2 days ago

Data Manipulation
o Are the queries syntactically correct?

The selects seems to be correct.

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

There need some updated and deletes.

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

The seem okay.

DDQ File
o 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!)

There seems to be some syntax errors.

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

The the data types are okay. The only issue is the tinyint for the boolen on the UserAccount needs to be set to a default.

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

There are some syntax error in creating the foreign keys.

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

Yes, it does appear to be present

helpful! | 0

Actions made based on peer review feedback (Step 4):

We have fixed the sql error.

Current DB, ERD, and Schema

UserRegistrations: stores each user's personal information.

userID: INT, auto_increment, unique, not NULL, PRIMARY_KEY
lastName: VARCHAR
firstName: VARCHAR
password: VARCHAR, not NULL
email: VARCHAR, not NULL
`zipCode` INT(11) NOT NULL,
Relationship: A **1:M** relationship between **UserSignUps** and **UserLogins**

UserLogins: An intersection table linking **UserSignUps** and **UserAccounts**

`userID` INT(11) NOT NULL ,
`password` VARCHAR(255) NOT NULL,
`locationID` INT(11),
`lastName` VARCHAR(255) NOT NULL,
`firstName` VARCHAR(255) NOT NULL,
FOREIGN KEY (`userID`)
FOREIGN KEY (`locationID`)
Relationship: A **M:1** relationship between **UserLogins** and **UserSignUps**
Relationship: A **M:1** relationship between **UserLogins** and UserAccounts

UserAccounts: Contains all the Users as well as their corresponding climate information.

userID: INT, not NULL, FOREIGN_KEY
locationID: INT, not NULL, *FOREIGN_KEY*
climate: VARCHAR
temperature: DECIMAL
isGuide: TINYINT
Relationship: A **1:M** relationship between **UserAccounts** and **UserLogins** with userID and locationID as FKs

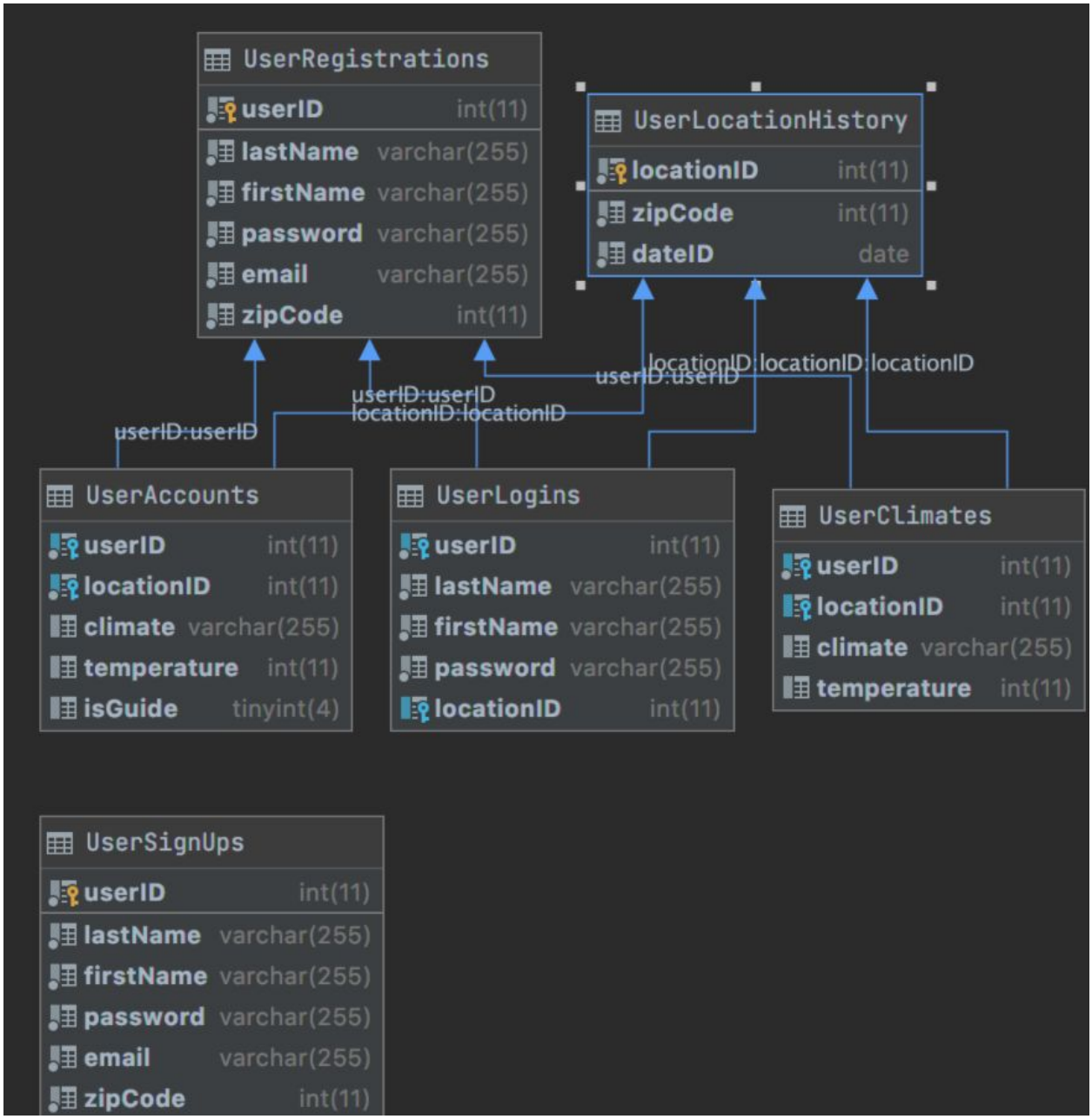
UserClimates: Stores the weather information associated with each *User*.

userID: FK, INT, NOT NULL
temperature: DECIMAL
climate: VARCHAR
locationID: INT, not NULL, *FOREIGN_KEY*
Relationship: A **1:M** relationship between WeatherVaults and LocationVaults

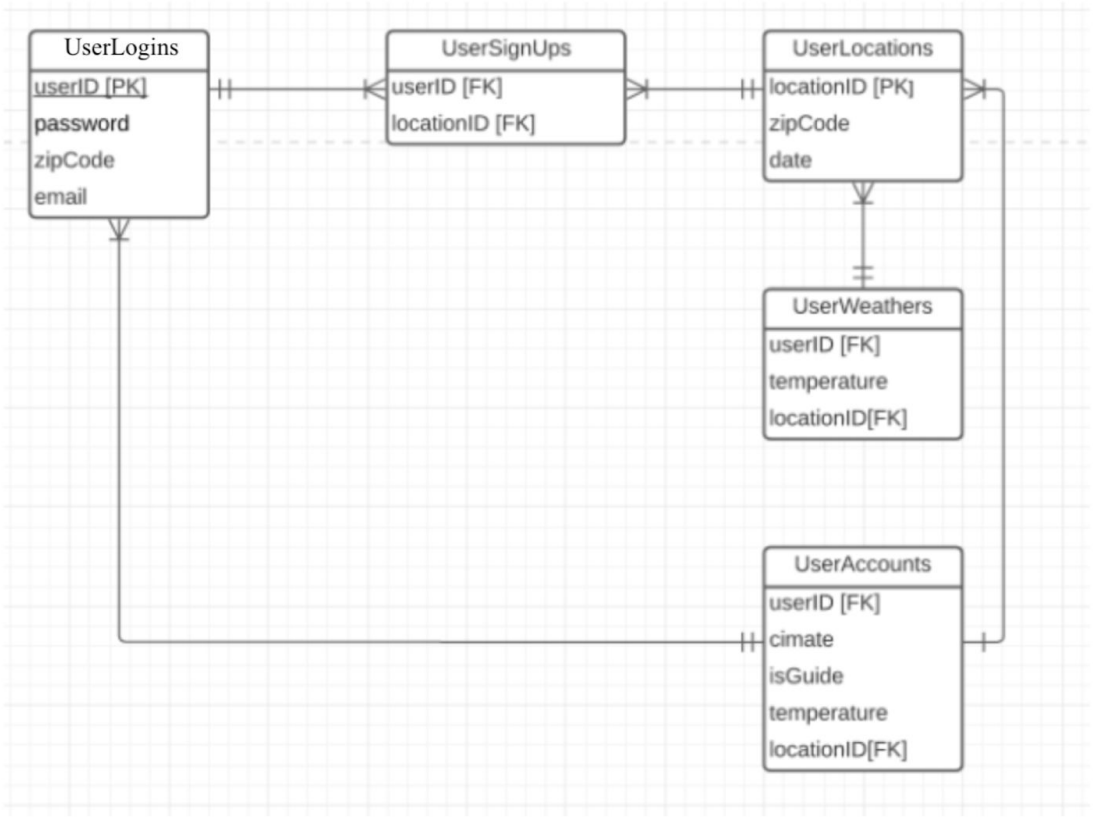
UserLocationHistory: Stores the history of all *Users'* locations.

locationID: INT not NULL *PRIMARY_KEY*
dateID: DATE, NOT NULL
zipCode: VARCHAR not NULL

Current Schema



ERD


UserLogins
userID [PK]
password
zipCode
email
UserSignUps
userID [FK]
locationID [FK]
UserLocations
locationID [PK]
zipCode
date
UserWeathers
userID [FK]
temperature
locationID[FK]
UserAccounts
userID [FK]
cimate
isGuide
temperature
locationID[FK]

Project Step 3 feedback:

1. 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. There are pages that display data for the UserSignUps, UserLocations, UserAcounts, and UserWeathers.
● All but the homepage have data from each table.
● Yes, they all seem to be present.
● Yes, every table is present with a select.
2. Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
● I do not see a search/filter anywhere on the website.
● There is not a search function.
● Maybe it will appear after a user adds multiple locations, but as of now I can't find any option to sort or otherwise reorganize any SELECTs.
● No, there is no search function to be found.
3. 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.
● I see an insert for signing up/in and adding a location.
● I only see the "Add Location" and "Sign Up" as INSERT. There should probably be an "Add Weather" as INSERT for the UserWeather entity.
● The Weather page looks like the only one without the option to add items.
● No, insert is only shown on the locations page and the sign up page.

4. Does each INSERT also add the corresponding FK attributes, including at least one M: M relationship? In other words, if there is an 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).
● No. The M: M relationship is not clearly explained in the proposal, the only INSERT I see is "Sign Up" and "Add Location". From the schema, there is an M: M relationship between UserSignups and UserLocations, and their intersection table is UserSignUps. The FK attribute is userID, which seems to be the email on the website. I think the explanation and implementation need to be more clear.
● The intersection tables for M: M relationships have almost the same names as one of the entities, which can be very confusing to read.
● I don't see a M:M relationship at present, but can see where it will be added in the future.
● No, the location INSERT only adds a string, but the sign-up INSERT adds FK attributes.
5. Is there at least one DELETE and does at least one DELETE remove things from an 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.
● No. I don't see DELETE anywhere on the website.
● There is not 1 delete.

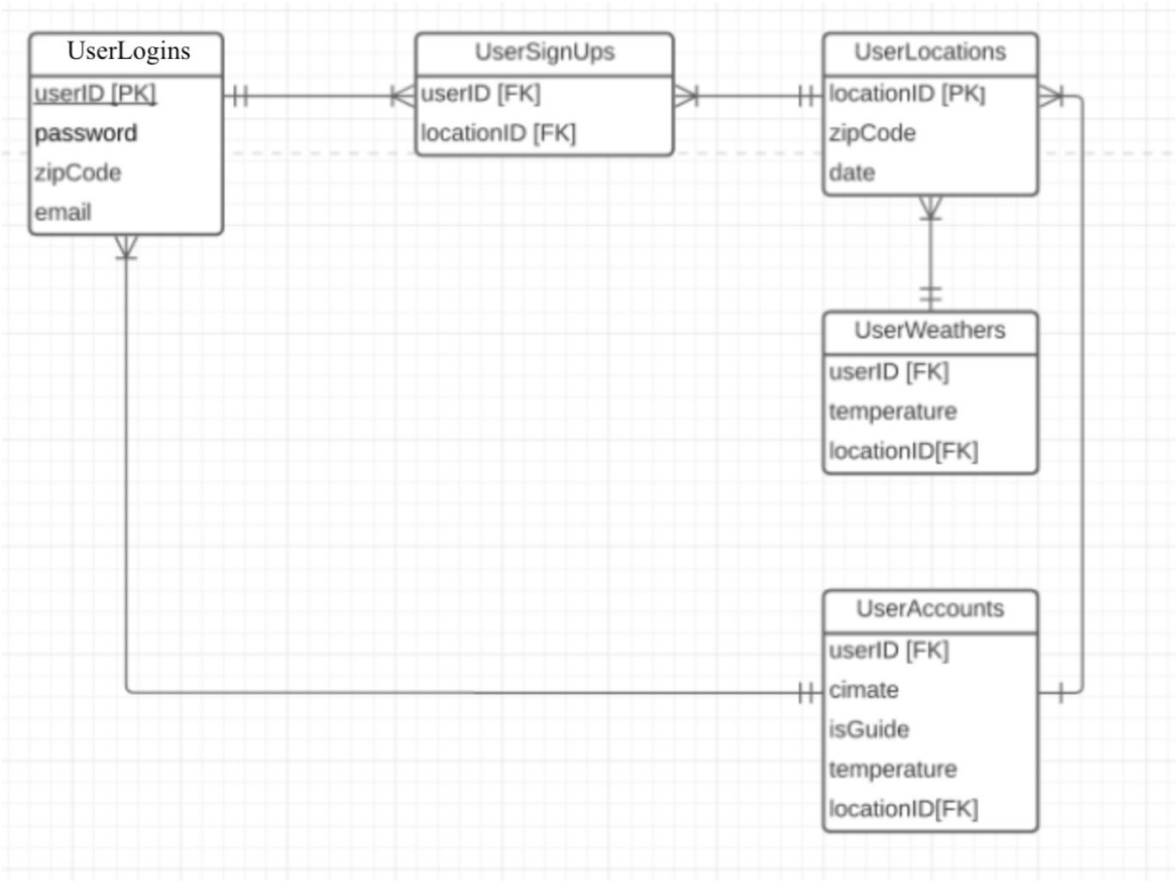
● There is no option to delete anything, as far as I can tell.
● No, the location INSERT only adds a string, but the sign-up INSERT adds FK attributes.
6. Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?
● No. There is not an UPDATE for any entity.
● I don't see an update.
● No, there is no place to update any of the entitities. ● No, there is no update for any entities
7. 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.
● I don't see any NULLAable relationships.
● There are no NULLable relationships.
● No, there are no NULLable relationships.

● No, there aren't any.
8. Do you have any other suggestions for the team to help with their HTML UI?
● The horizontal navigation bar is not available on the "Sign Up" and "Sign In" page, which can be inconvenient for the user to navigate the website.
The title for each page should be unique.
● It's difficult to move around the website, and the "weather" page and "my page" are redundant.
● Nothing specific. Fixing the above problems would greatly help their HTML UI.
2. Actions based on the feedback
List briefly the actions that you chose to take based on the above feedback. If you decided not to act on a specific suggestion, you need to describe in detail your reasoning.
1. Added 2 pages: EditUserT and EditWeatherT, which allow the user to edit their user and weather information. Users can now delete their information, and delete their account. We also now have a search function!
2. ImplementedarelationshipwithabilitytoaddNULLvaluesinthetable.
3. Added horizontal navigation bar on the "Sign Up" and "Sign In" page
4. Added a unique title for each page.
3. Upgrades to the Draft version
We made a few visual tweaks to the index.html home page, most notably, the redesigned top menu bar. Each button has its own little icon.

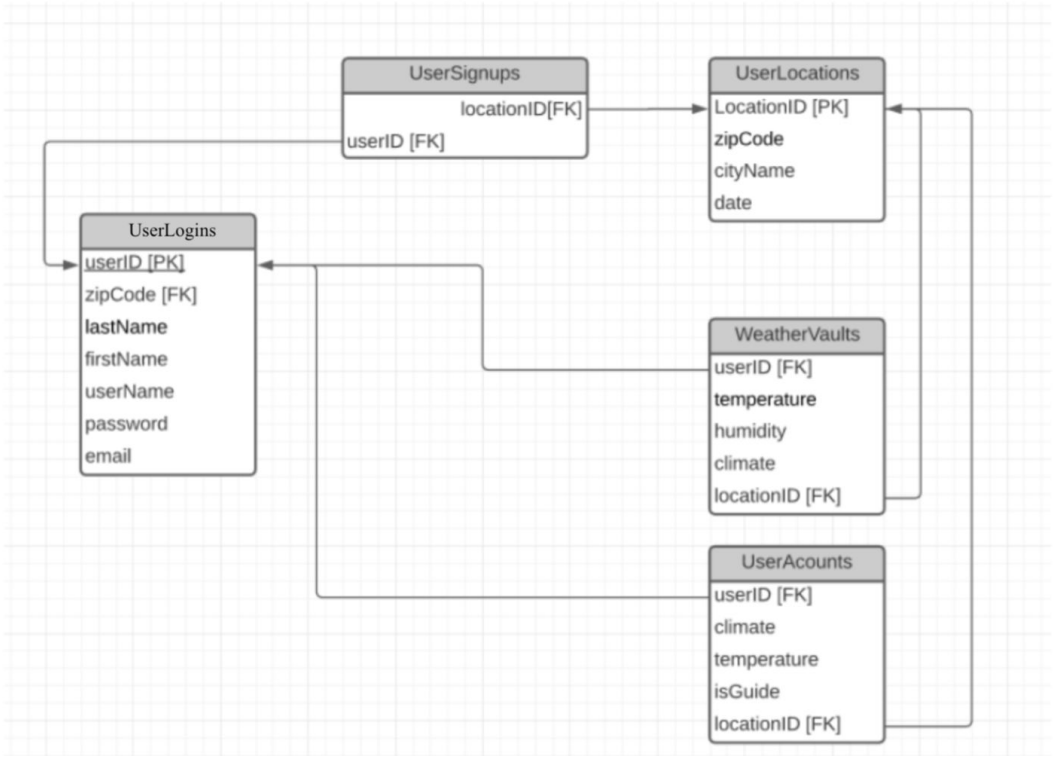
4. Upgrades to the Draft version (step 3)
UserSignUps: a user which stores their location
userID: INT, auto_increment, unique, not NULL, PRIMARY_KEY
lastName: VARCHAR
firstName: VARCHAR
password: VARCHAR, not NULL
email: VARCHAR, not NULL
zipCode: VARCHAR, not NULL
Relationship: A 1:M relationship between UserLocation and UserAccounts
UserLocations: An intersection table linking Users and LocationVaults

locationID: INT, not NULL,FOREIGN_KEY
userID: INT, not NULL, FOREIGN_KEY
Relationship: A 1:M relationship with Users and 1:M relationship with LocationVaults
UserAccounts: Contains all the User as well as their corresponding climate information. userID: INT, not NULL, FOREIGN_KEY
locationID: INT, not NULL, FOREIGN_KEY
climate: VARCHAR
temperature: DECIMAL
isGuide: TINYINT
Relationship: A 1:M relationship between Uservault and User with userID and locationID as FKs
UserWeathers: Stores the weather information associated with each User. userID: FK, INT, NOT NULL
temperature: DECIMAL
humidity: DECIMAL
climate: VARCHAR
locationID: INT, not NULL, FOREIGN_KEY
Relationship: A 1:M relationship between WeatherVaults and LocationVaults
UserLocations: Stores the history of all Users' locations. locationID: INT not NULL PRIMARY_KEY
userID: FK, INT
dateID: DATE, NOT NULL
zipCode: VARCHAR not NULL
Relationship: A 1:M relationship with Users_LocationVaults and WeatherVaults

ERD


UserLogins
userID [PK]
password
zipCode
email
UserSignUps
userID [FK]
locationID [FK]
UserLocations
locationID [PK]
zipCode
date
UserWeathers
userID [FK]
temperature
locationID[FK]
UserAccounts
userID [FK]
cimate
isGuide
temperature
locationID[FK]

SCHEMA



Are there any suggestions for improvement for your team and what are your goals moving forward?

We have received feedback about a few of our pages being “difficult to navigate through”. We have taken note of this, but we are not concerned. We know what we are doing and have a specific design in mind that will make our webpage very simple to navigate.

For now, our website has been prepared in a way that we hope will make it easy for us to communicate with our database backend.

Going forward, we will continue to simplify our website’s layout, but we must not lose focus on our key objective, which is the seamless transfer and display of information between our website and our soon to be database.

Project Step 2 feedback:

a) Fixes based on Feedback from Step 1:

1. Our project idea has changed slightly. We have revised our Overview to reflect this.
2. Our new project proposal has addressed the comments made by our marker. Our changes add:
a. how many users use our website on a daily basis
b. how often weather data is updated.
3. Our descriptions concerning the entities in the overview are now more thorough and direct.
4. Entity names are now consistent throughout the project.
5. Creating the intersection table for M:M relationships that contain the FKs of the relationship as well as add PKs for each of your entities

b) Project Outline and Database Outline - Updated Version:

Overview:

The Weather Vault ® website is designed for people moving to another city in the US and concerned about change of climate.

The User enters a desired climate into the Weather Vault ®. The WeatherVault and the UserVault entities are queried, then retrieve and display a list of all matching cities and profiles of certified guides are displayed. After the User selects a guide the User receives the guide's contact information (email, first name, last name).

Becoming a Weather Vault ® guide is a process initiated by a User by selecting the "become a Weather Vault ® guide" option on the website. Upon receipt of a request the information is reviewed and if granted, the "isGuide" attribute is switched to true..

Weather Vault ® currently has 25,000 daily active users and updates its weather and User's locations daily. We have recently announced establishing an IPO and are expecting an increase in our user-base to about 1,000,000 daily users. It is expected that our heads of software design will be rebuilding and reorganizing the database system to handle the projected load.

Entities (tables) we plan to implement on our project:

User UserVault WeatherVault LocationVault

Developer Roles:

Lei Dai: Commander of WeatherVault and LocationVault

Dov Sherman: Head of design for the UserVault entity and its incoming and outgoing relationships.

Database Outline (in words): User: a user which stores their location

userID: INT, auto_increment, unique, not NULL, PRIMARY_KEY lastName: VARCHAR, not NULL

firstName: VARCHAR, not NULL

email: VARCHAR, not NULL

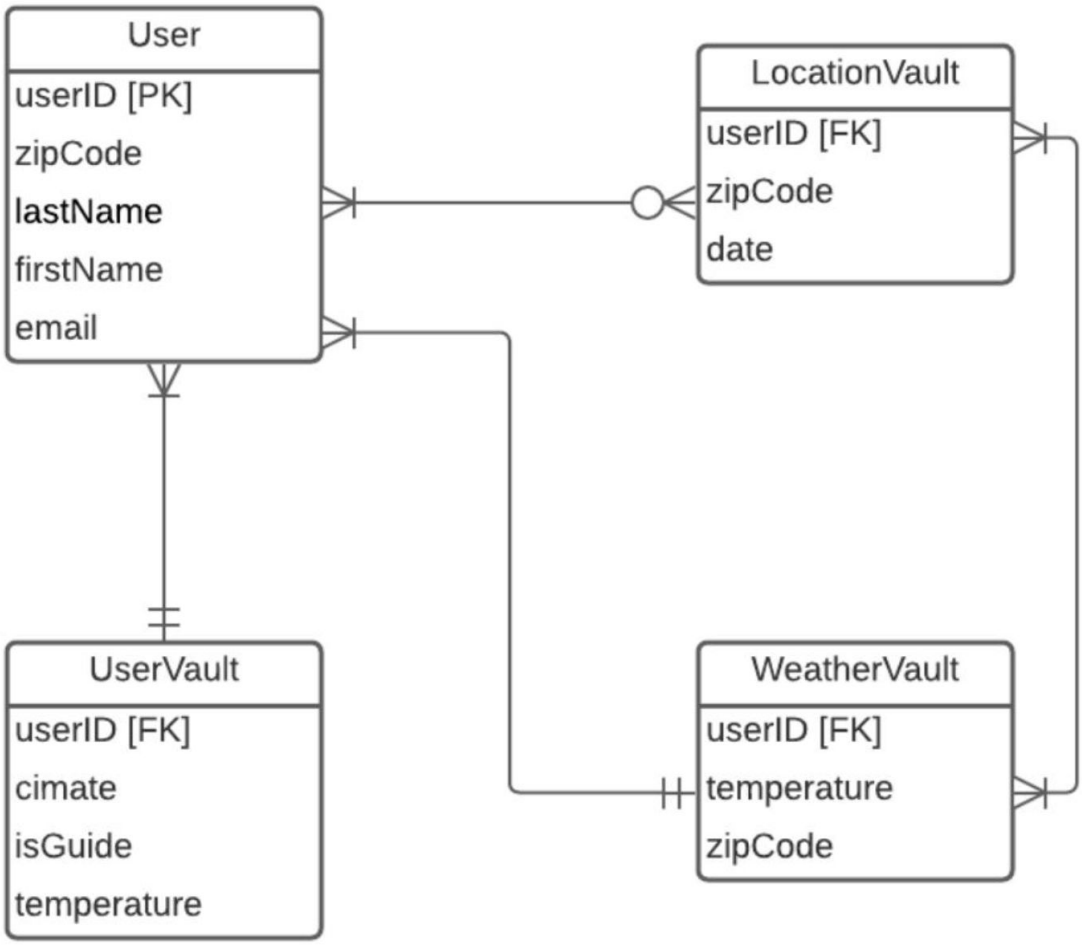
zipCode: VARCHAR, not NULL

Relationship: A 1:M relationship between UserVault and User with userID as a FK inside UserVault.

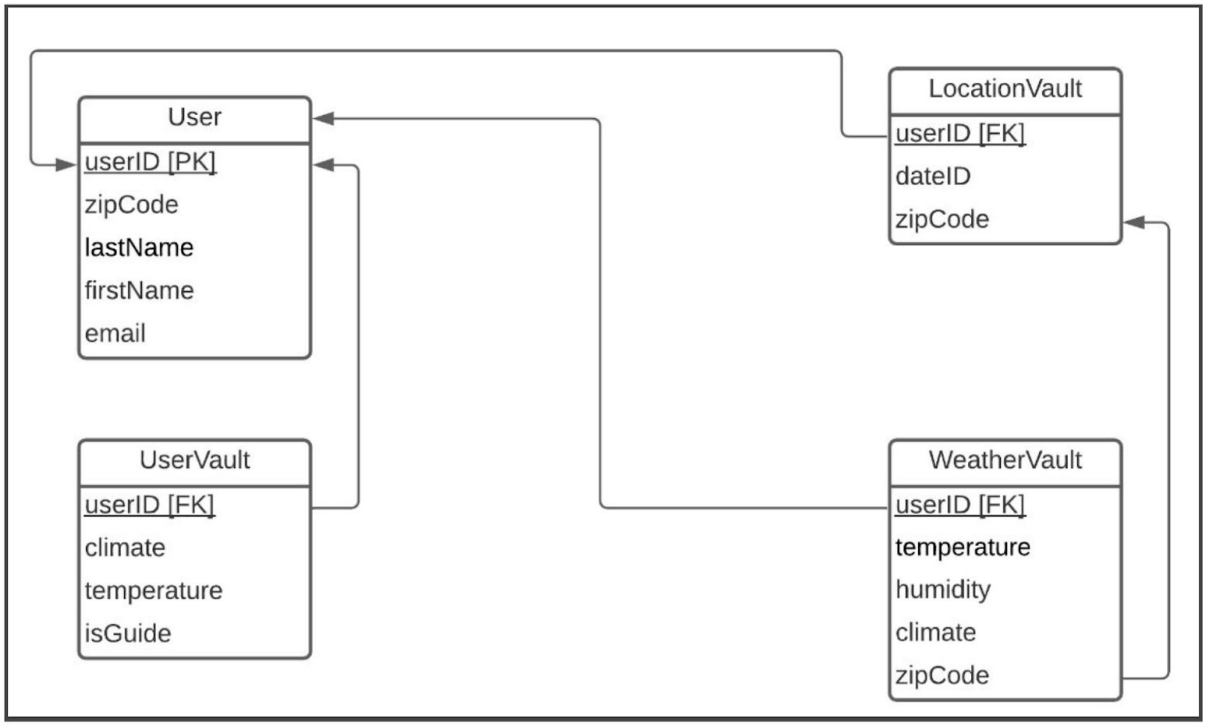
UserVault: Contains all the User as well as their corresponding climate information. userID: INT, not NULL, FOREIGN_KEY

climate: VARCHAR
temperature: DECIMAL
isGuide: TINYINT
Relationship: A 1:M relationship between Uservault and User with userID as FK
WeatherVault: Stores the weather information associated with each User. userID: FK, INT, NOT NULL
temperature: DECIMAL humidity: DECIMAL climate: VARCHAR zipCode: VARCHAR

Relationship: A M:M relationship between Weathervault and User with userID as a FK inside WeatherVault and a relationship between WeatherVault and LocationVault with zipCode.
LocationVault: Stores the history of all Users' locations. userID: FK, INT
dateID: DATE, NOT NULL
zipCode: VARCHAR
Relationship: A M:M between many Users to many zipCodes.
c) Entity-Relationship Diagram:



d) Schema:


User
userID [PK]
zipCode
lastName
firstName
email
LocationVault
userID [FK]
dateID
zipCode
UserVault
userID [FK]
climate
temperature
isGuide
WeatherVault
userID [FK]
temperature
humidity
climate
zipCode