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

Weather Vault ®

Project Step 3 Final Version

CS340
Group106
Developers
Dov Sherman

1. Feedback by the peer reviewer

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. Implemented a relationship with ability to add NULL values in the table.
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

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*.

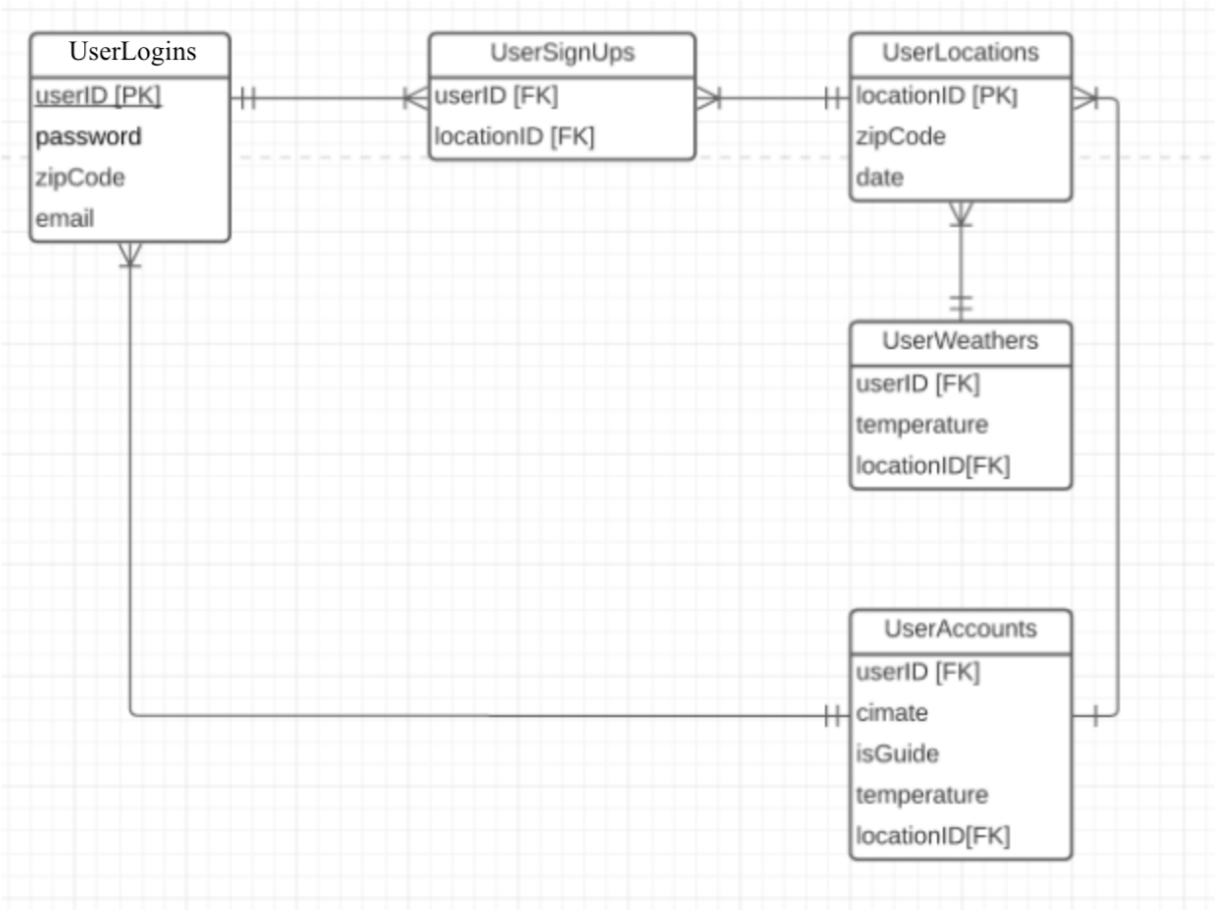
userID: FK, INT, NOT NULL
temperature: DECIMAL
humidity: DECIMAL
climate: VARCHAR
locationID: INT, not NULL, *FOREIGN_KEY*
Relationship: A **1:M** relationship between WeatherVaults and LocationVaults

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

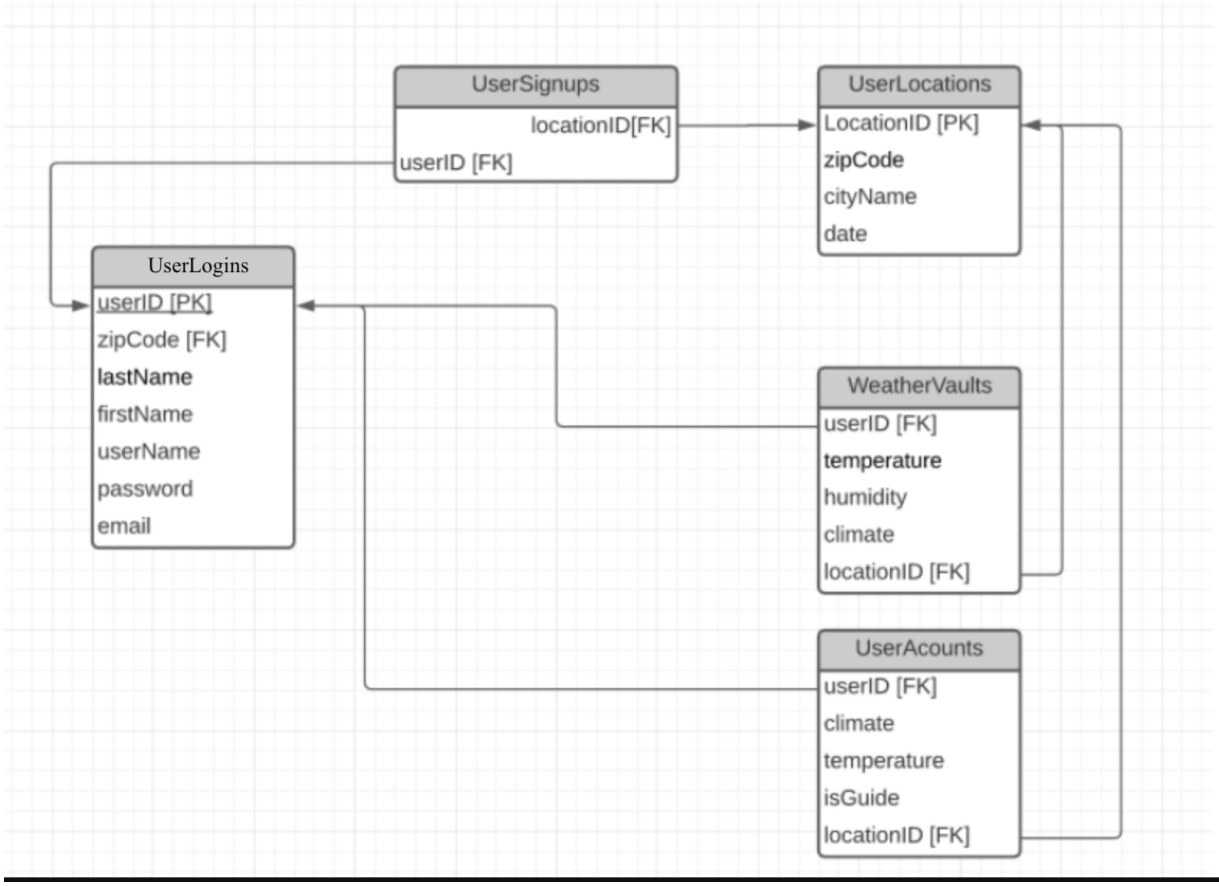
locationID: INT not NULL *PRIMARY_KEY*
userID: FK, INT
dateID: DATE, NOT NULL
zipCode: VARCHAR not NULL
Relationship: A **1:M** relationship with Users_LocationVaults and WeatherVaults

Relationship: A **M:M** between many *LocationVault* and *User*

ERD



SCHEMA


UserSignups
locationID[FK]
userID [FK]
UserLocations
LocationID [PK]
zipCode
cityName
date
UserLogins
userID [PK]
zipCode [FK]
lastName
firstName
userName
password
email
WeatherVaults
userID [FK]
temperature
humidity
climate
locationID [FK]
UserAcounts
userID [FK]
climate
temperature
isGuide
locationID [FK]

5. Team evaluation form

Group number	Group106	
Name of Group TEAM Members:	Dov Sherman, Lei Dai	
SCALE AND COMMENTS	RATING	ADDITIONAL COMMENTS
How Prepared was your team? Research, reading, and assignment complete	4	Well prepared for each project and assignment

How responsive & COMMUNICATIVE were you both as a team? Responded to requests and assignment modifications needed. Initiated and responded appropriately via email, Slack etc.	4	We communicated through google hangouts, as originally planned. Each time's communication is timely and efficient. We have utilized our time zone differences to ensure a 24hr working schedule.
Did both group members Participate equally Contributed best academic ability	4	Yes.
DID YOU BOTH FOLLOW THE initial team CONTRact? **Were both team members both positive and productive?**	4 4	Yes, team members are positive and productive.

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.