

<https://github.com/dailei5/dailei5.github.io>

## Weather Vault ®

### Project Step 3 Draft Version: Design HTML Interface

CS340  
Group106  
Developers  
Dov Sherman  
Lei Dai

**Note: We have already made all the recommended changes to this file during the Project Step 2 Final.**

#### **1. Changes based on our design decisions (if any):**

- a. LIST CHANGE MADE AND REASON TO WHY THE CHANGE WAS MADE
- b. Renames entities
  - i. New naming scheme is easier to understand.
- c. User's first and last names are now optional. User's email, password and zipCode are now mandatory.
  - i. First and last names are not needed since the user supplies their email address. They can add their name after they create their account.

#### **2. Feedback by the peer reviewer**

Include verbatim the review that your Draft submission received in the Group Review. If you did not receive any reviews, mention that. You would include feedback that both members of the Project Group received.

**Does the overview describe what problem is to be solved by a website with DB back end?**

Josh McKerracher: *The overview looks great! The problem being solved by this site is clearly explained.*

I Chun Jurng: *Yes the website and DB helps people determine which cities has their preferred climates.*

Fiona Zhu: *Yes the website and DB helps people determine which cities has their preferred climates.*

Darian: Yes, it is said that the DB is designed for people moving to another city in the US and concerned about change of climate.

**Does the overview list specific facts?**

Josh McKerracher: Yes, the overview does list specific facts. It says, for example, how many users it currently has, along with how some of the features work.

I Chun Jurng: Yes the overview gives a good view on the website and what the DB will achieve Very interesting concept, I think if the authors expand a bit on what a WeatherVault guide does would be good.

Fiona Zhu: Yes, it lists that there are 25k active users and is expected to grow to 1M.

Darian: Yes, it lists some specific facts such as the number of users currently in the website and the number it is going to achieve in the future.

**Are at least four entities described and does each one represent a single idea to be stored as a list?**

Josh McKerracher: Yes, there's four entities, with each representing one distinct idea.

I Chun Jurng: Yes, exactly 4 entities

Fiona Zhu: Yes, there are four entities.

Darian: Yes, there are at least four entities described and each one to be stored as a list.

**Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?**

Josh McKerracher: The outline does a great job of describing each entity's purpose, along with including all the details for each entity's attribute and relationship(s).

I Chun Jurng: Yes, outline describes entities and lists attributes. Division of work is clearly stated

Fiona Zhu: Yes each entity describes its purpose alone with the attribute datatypes and constraints if any. Relationships are described between entities. Work is divided accordingly for each member of the team.

Darian: Yes, each entity was described and the purpose is clear. Relationships between entities are also described. Work load are divided in the team accordingly.

**Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?**

Josh McKerracher:

The use of capitalization in entity names vs. camel case for entity attributes is a small inconsistency.

I'd also say that I'm a little confused about the M:M relationships in the schema. For example, for the LocationVault entity, the database outline says, "Relationship: A M:M between many Users to many zipCodes." But in the schema, it doesn't seem like that M:M relationship is there.

I Chun Jurng:

The 1:M relationship between User and UserVault makes sense, however, it might be helpful to have a Primary Key for UserVault. Instead of having UserVault contain the Foreign Key of userID, the authors can then have a UserVaultID as a Foreign Key under the User Entity. That way, many users can share a UserVault and one user can only have one UserVault.

There were a few mentions about an entity having a relationship with an attribute (ie, WeatherVault, an entity, and zipCode, an attribute) I'm not sure if this is possible.

Overview and entity/attribute naming is consistent but entities are not plural. Capitalization is used for naming.

Fiona Zhu:

1:M are correctly formulated. There is at least one M:M relationship.

Yes, there is consistency with entity and attributes. Entities are plural and attributes are singular. Camel case is correctly used.

Darian:

*1:M relationships are correctly formulated and there are at least one M:M relationships mentioned.*

*Yes, there are consistency in naming, entities are all plural, attributes are singular and capitalization is consistent.*

### 3. Actions based on the feedback

Based on the above feedback, we have:

- added a new entity “Users\_LocationVaults” as an intersection table between Users and LocationVaults entities.
- removed zipCode relationship from all entities.
- rearranged ERD and Schema to reflect these changes.

### 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

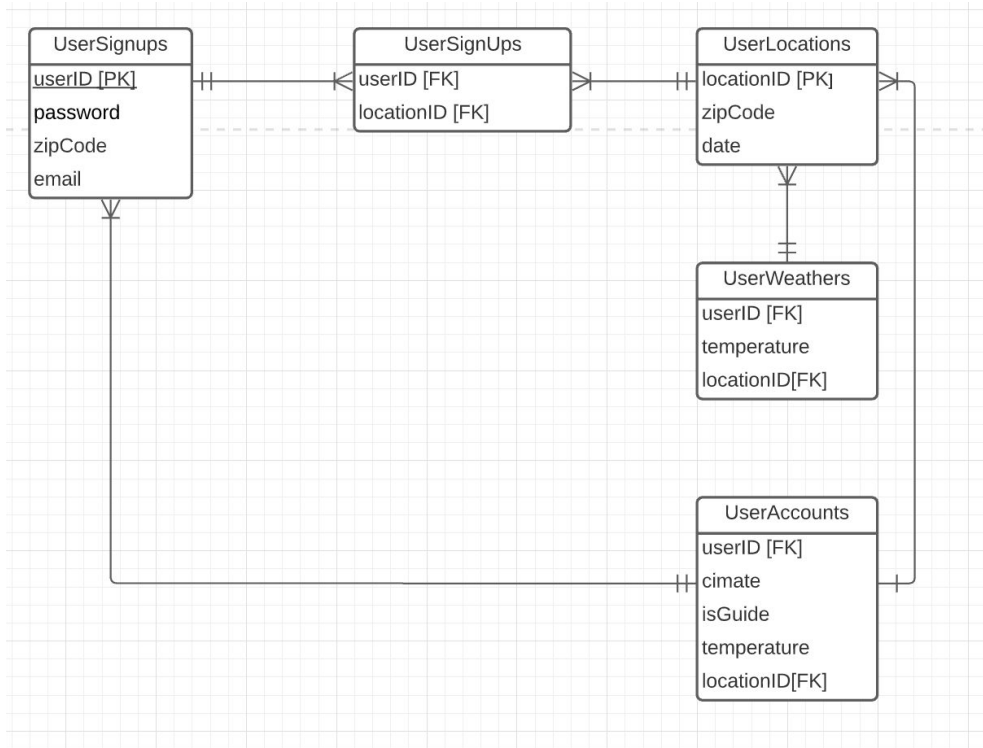
userID: FK, INT

dateID: DATE, NOT NULL

zipCode: VARCHAR not NULL

**Relationship:** A **1:M** relationship with Users\_LocationVaults and WeatherVaults

ERD



## Schema

