

Range Finder - Interim Report 1

Team Members

Benjamin Hargett - bharget - C56059977

Jacob Hurd - jhurd - C19777168

Web Portal Description

Our project, Range Finder, is a web portal that is designed to allow users to search a database of mountain peaks in North America as well as track the peaks that they have climbed. This involves a user registering for an account on our site which will then allow them to query the database based off of mountain name, country, state, elevation, latitude and longitude. Once logged in the user will be able to mark a mountain as “visited” as well as leave reviews and ratings about their trip. Users will also be able to search for other user accounts and follow users to see information about their “friends” trips.

There will be an administrative component where a user with admin permissions can login, edit, update, backup, or delete information in the database. Passwords will be encrypted to ensure the integrity and safety of the user’s information.

Database Population

We have already gathered all of the information that we plan to use in our “production” dataset. Currently it is contained in a table named “mountains” and consists of over 64,000 mountains across North America. This data was gathered from openstreetmap.org. We downloaded their map data for North America in XML format and then used their open-source tool, *osmosis*, to scan the XML for any datapoint that was labeled as a mountain peak. We then developed our own ruby script to parse this XML data into a JSON file. The script took the name, elevation, latitude, and longitude from the XML data entry and reverse-geocoded the coordinates to determine the country and state that the mountain is in. We then developed another ruby script to use this newly created JSON file to create a “mountains” table in our database and populate it with the data.

Assumptions about Data

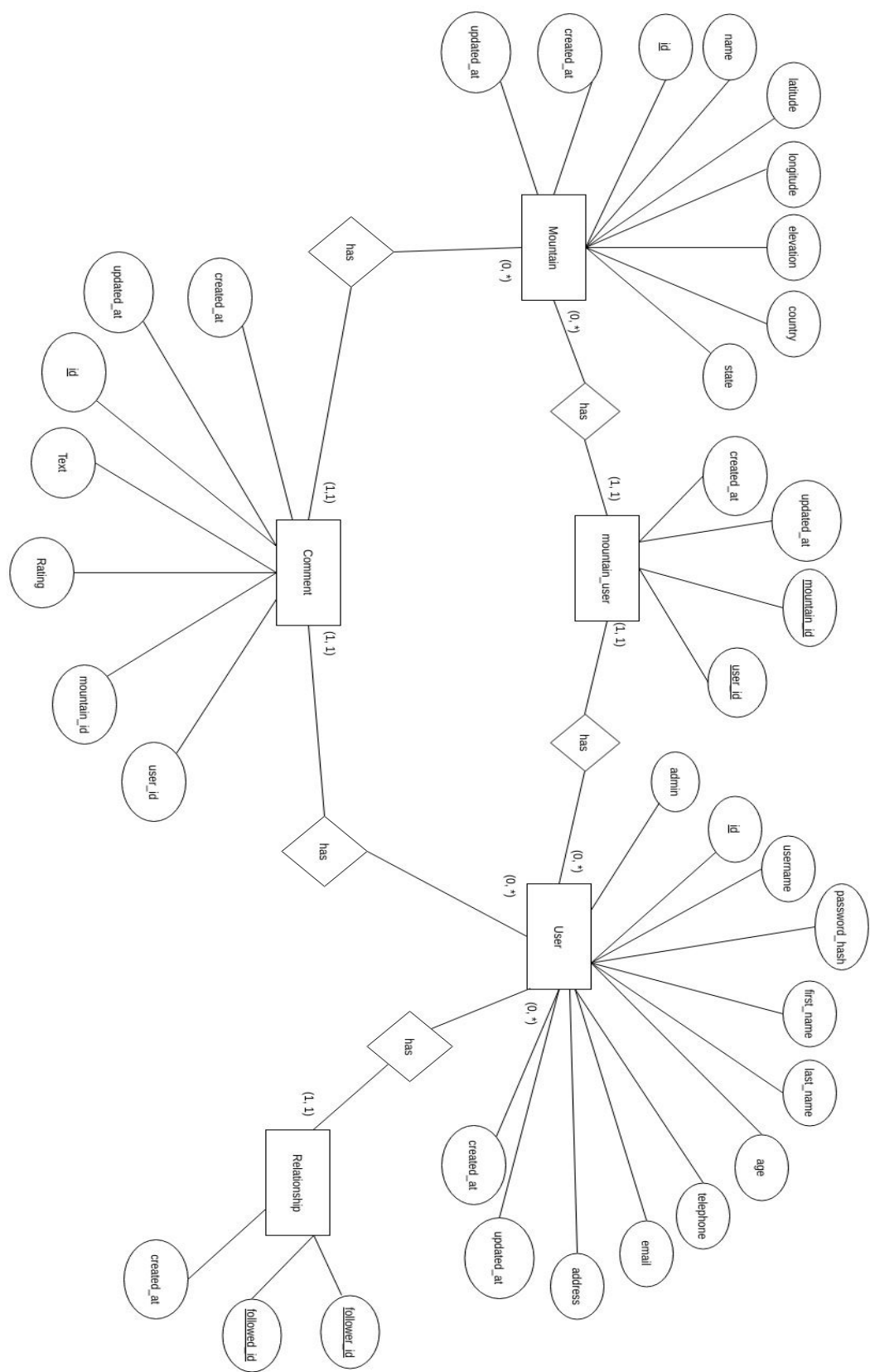
The dataset that we are using to populate our database is rather large so there are a few implicit assumptions that are made to allow users to accurately use the information from our site. We are assuming that the dataset is an accurate representation of mountains in North America. We are assuming that there are no duplicates in our dataset and we are assuming that no fields in the data are null. We have taken steps to try to eliminate any inconsistencies in our dataset, such as removing duplicates, but since the dataset is so large and it comes from an open-source organization with thousands of contributors, it is difficult to easily check the entire set for all assumptions made above.

Sample Data

Mountains								
id	name	state	country	latitude	longitude	elevation	created_at	update_at
1	Kelly's Mountain	Nova Scotia	Canada	46.2519258	-60.5281258	240	09/19/17	09/24/17
2	Eliza Rock	Washington	United States of America	48.6437357	-122.5793553	0	09/19/17	09/24/17
3	Mount Whitney	California	United States of America	33.1088014	-117.1551691	527	09/19/17	09/24/17
4	Juniper Butte	California	United States of America	41.7914044	-121.4509942	1287	09/19/17	09/24/17
5	Wildcat Peak	California	United States of America	37.9192307	-122.2621766	370	09/19/17	09/24/17

Users											
id	username	password_hash	first_name	last_name	age	telephone	email	address	created_at	updated_at	admin
1	jhurd	\$2y\$10\$nOwe.BTD.YCQPC...	Jacob	Hurd	20	123-123-1234	jake@email.com	Somewhere	09/19/17	09/24/17	true

ER Diagram



Database Schema



Web Portal Interface Description

The user's interaction with the portal interface begins with a login screen. The home page of the site is a prompt for the user to enter their username and password to login to Range Finder. If they do not have an account, there is an option to register for an account. If the user is already logged in, then instead of displaying a login prompt, a dashboard will be displayed with pertinent information for that user such as their most recent climbs as well as recent climbs by users that they are following.

Once logged in, they will be able to view all of the mountains that they have marked as climbed. They will also have the option to perform basic database queries using the name of a mountain. They can select the option to do an advanced search which will give them the ability to specify an elevation range, state, country, and coordinates. A list of results will be displayed and the user can select a mountain which will bring up a page with the pertinent information regarding that mountain. There will be an option to mark the mountain as climbed as well as an option to leave a comment. If the user chooses to leave a comment, then they will be prompted to enter text for the body of their comment as well as give their climb a numerical rating.

From the logged-in user's dashboard, they will have the ability to view the list of people that they follow as well as the list of people that follow them. They will also have the option to search the database of users by entering their first name, last name, or username. A list of results will be displayed and the logged-in user can select a user they wish to view. A page will display that user's publicly viewable information such as their name and username as well as their recent climbs. The logged-in user will then be able to follow or unfollow that user.

If an administrative user logs in to Range Finder, they will be taken to a dashboard very similar to the normal user dashboard. They will be able to search the database for mountains similar to a normal user, but they will have an additional option to edit or delete that mountain's information when viewing the mountain's page. They go through a similar process with users; they are able to search for users but with the additional option to edit/delete them. If editing a mountain or user, they will be prompted with a form where they can input the updated information. The administrative user will also have the ability to delete comments on the selected mountain. In addition to the features listed above, administrators will have the ability to go to a management page from the dashboard where can download a backup of the database.

Completed Tasks

So far we have made really good progress on our project as a whole. We have determined a clear scope/context of our web portal, developed and implemented plans to populate our database using real datasets, and made sure to include a data model that is friendly to both queries and updates of the data. We have also developed an ER Diagram and Relational Schema for our database that has further defined the direction of our project as a whole. With those in place, there is no confusion in how the data will work together and in how their relationships will be integrated into our web portal application. In designing those diagrams, we made it a priority to follow good database design principles and to reduce redundancies throughout the system. Once we laid out a plan for our web portal based on these diagrams, we began implementing the design. We have now populated a database with the real “production” dataset and have begun to incorporate it into our web portal. Starting with a static page to lay out our user interface design, we quickly then moved to designing dynamic pages with embedded PHP that allows a user to register and login to our web portal. In addition to all of that we have also designed some SQL queries to allow users to search and view data based on the relations listed in the ER Diagram and Relational Schema.

Future Plans

Week (from current date)	Task - Benjamin	Task - Jacob
1	Finish designing sql queries	Create page to dynamically serve search queries for mountains
2	Design Mountain information pages	Create page to dynamically serve search queries for users
3	Design User information page	Implement relationships between users as well as between mountains and users
4	Add functionality to web portal to mark mountains as climbed as well as follow users	Implement the ability for users to leave comments on mountains and start user dashboard
5	Finish adding information to user dashboard	Finish adding information to user dashboard
6	Add administrative functionality	Add administrative functionality
7	Polishing user interface and testing	Polishing user interface and testing