### **RouteRanger**

By Brian Kenney and Julia Nguyen

D-532: Applied Database Technologies

Summer 2023

1. How you built your project: Our project was implemented in 3 phases:

*Planning*

**Ideation.** We initially brainstormed on what ideas to cover before agreeing on building a database that revolves around an activity we both enjoy, hiking. We then set up goals and discussed requirements, outlining what we aimed to achieve, the scope, and what would be needed to execute the project.

*Executing*

**Designing.** Once the dataset was obtained, we then decided on what type of database we wanted. Based on the criteria we would like to focus on, we decided to use SQL over NoSQL. We moved on to designing the database schema and Entity Relationship Diagram (ERD). Several changes were made to the ERD during this process as we discussed different outcome scenarios.

For the database design, the first two normalizations were used in our relational database. The first two were used because it maintains data integrity and does not require the raw dataset to be split into too many tables. If there are too many tables present, then there would be a lot of joins which would create more problems than it is worth.

On top of using the first two normalizations in the relationship database, it is critical to maintain data integrity so we needed the normalizations followed by proper constraints. Primary keys, foreign keys, uniqueness, and views were used in the database. The first three constraints were primarily used to follow the first two normalizations, whereas the views were needed because units of measurement were needed.

**Coding.** The rough draft of the database was designed through Jupyter Notebook to see how the database would work, based on the schema that was established during the design phase of the project. Once the architecture of the database was set up, we took this onto Visual Studio to start building the final version.

**Developing.** The project was built by using Visual Studio Code and using multiple Python libraries such as Streamlit, Sqlite3, and Pandas. The libraries were integrated together in order to create a database and a webpage.

*Closure*

**Review.** We tested the web-app, discussed and made adjustments as needed.

**Deploying.** The final interactive database was deployed using Streamlit which allows users to interact directly with the database and functionalities.

**Documentation.** We documented everything as required in this course, including preparing a video demo and final paper. Then we saved everything into GitHub where we both have access.

Both collaborators worked on the code and the logistics such as proposals, design layout, presentation layout, and the final paper.

1. What did you learn as a team (what was not covered in the class)

The first thing we learned as a team is we became acquainted with how to use Streamlit as this is a Python library package we have never used before. There was a slight learning curve to using this library package as our background is working with backend code, not frontend code.

Working with real data from start to finish helped us gain exposure to how database design and management work in real life. We learned how to use Python and SQL interchangeably, such as importing data from csv file to managing front-end user interface, and to ensure user actions are implemented properly in the back end.

Another experience we gained insights into is learning how to effectively collaborate and managed the project from scratch. By setting up clear expectations and having proper communication throughout the process, we were able to successfully complete this project together.

Though this was a summer semester, this was an action-packed group project that allowed us to implement the concepts taught from this course into an applicable database system for all hiking enthusiasts.

1. Functionalities (what options you included and what options you were not able to incorporate from the original planning)

Included:

Following the CRUD system, we integrated the following:

* Search for existing trails
* Add a new trail
* Update an existing trail
* Delete an existing trail

Since we want the users to be able to interact with the database, the four functionalities were integrated into the webpage so that users can update as trails can be created or removed over the course of time for various reasons. As we want this to be a robust database, it is important that the community that uses it makes sure that the data is accurate.

Not Included:

One feature that was not included in the final design is for users to create an account prior to interacting with the data. The purpose of the account creation is to keep track of what changes are made by whom and in order to prevent multiple changes to maintain data integrity. Due to limited resources (i.e. time constrain, headcount…), we decided not to proceed with this setup as it is not necessary for the database and webpage to function properly.

Another feature that was not included in the final layout is the inclusion of visualizations, which could have transformed raw data into meaningful diagrams. However, the lack of resources did not allow us to implement this useful functionality.