Jack Fitzpatrick, Kyle Harvey, Samuel Appiah-Kubi, Kalito Luna
CS 316
10/11/18

Project WILD Maps Application: Milestone 1

Introduction

Project WILD (PWild) is a pre-orientation program at Duke University. Every year, before orientation week, PWild leads incoming freshmen on a two week backpacking trip in Pisgah National Forest. In addition to this August trip, PWild also leads trips in October and March, for fall and spring break. This program is entirely student led, so often it is difficult to maintain institutional knowledge, when the maximum staff retention is three years. Additionally, PWild has undergone many changes in recent years, so past knowledge is more important than ever. We are seeking to offer a way for staff to acquire knowledge and advice from past trips, as well as leave their own, in order to aid staff training and route planning.

The PWild Maps Application will be a map-based web application that serves to consolidate staff advice and knowledge for future use. It is based off an old map application built by a previous PWild staff member, which offers a functional (but woefully unoptimized and basic) method of leaving comments on certain areas of a map (located here: http://people.duke.edu/~abn7/pmaps/tiling%20base%20map%20fusion/PMAPS_v6.html). This older application is used frequently by current staff members, so reworking it entirely will be incredibly beneficial for the organization.

Datasets

Fortunately, due to the older version of the application, a dataset already exists. It has been attached to this submission, in the form of a csv file. Additional data will be acquired as current staff members continue to leave comments. Currently, the data is stored on a public Google sheet document, so translating this database to SQL will make it more modular, and able to stand alone. Additionally, making this application available offline could allow staff to take it to Pisgah (potentially on mobile), so they can look at the advice while on a trail.

We are assuming that any person can leave advice, and this is not restricted to PWild staff members. We want to allow alumni to continue leaving comments without an issue. Additionally, we are assuming that PWild will continue working in the same section of Pisgah.

This is a relatively safe assumption, since the same area has been used for over 45 years. We want to limit comments to general, place, and route, and assume that no other comments can be left. We assume new routes are left by staffers, but new “places” (points on the map) are not, since they are just a specific longitude and latitude. Finally, we are assuming a limited number of comment types, such as a view, campsite, water source, etc. These assumptions and constraints are reflected in the ER diagram below.

Entity-Relationship Diagram

The (in-progress) ER diagram of the database is shown below in Figure 1. Filled arrows represent a many to one or zero relationship, while non-filled arrows are many to exactly one. We are still working out how exactly routes will be implemented, but as of now, they are considered as separate locations. This will not always be the case, as technically routes are collections of places, but this is another issue. For now, leaving comments on route entities is sufficient. Keys are fairly self explanatory, with the exception of the person entity. We decided to use emails as keys, since assigning separate IDs to each person seemed unnecessary, and emails must be unique between people.

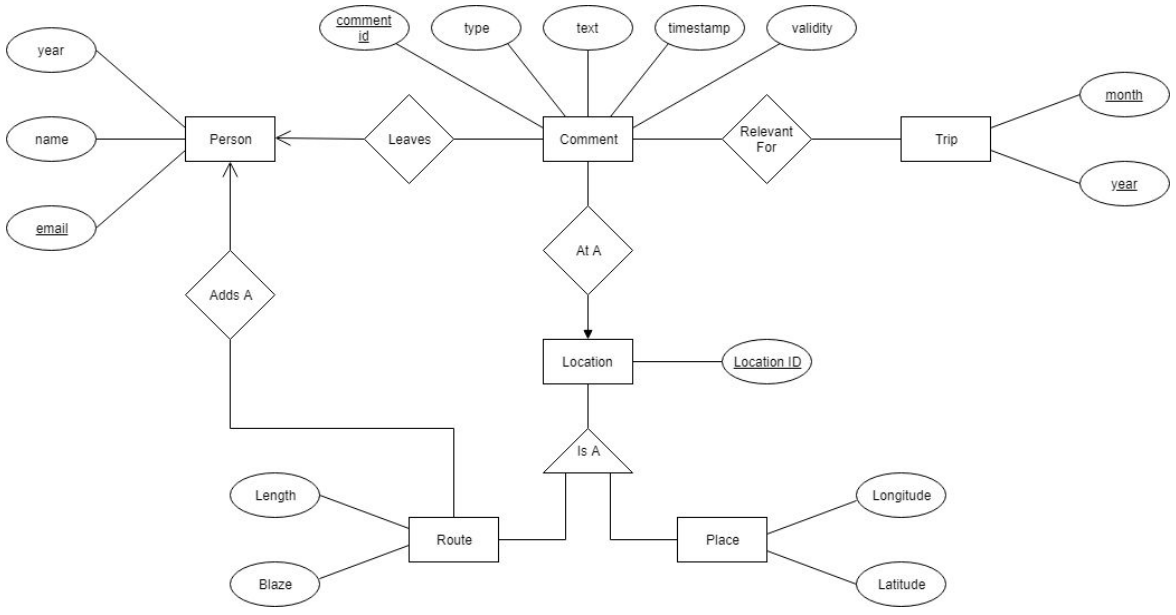

Figure 1: The ER Diagram of the Pmaps Database

Database Design

The tables for this design are listed below. Keys are underlined, and repeated attribute names are references between tables. For example, “CID” is listed in both Comment (where it is a key) and RelevantFor, so “CID” is a foreign key for Comment in RelevantFor.

Person (email, name, year)

Comment (cid, text, type, timestamp, validity, email, lid)

RelevantFor (cid, month, year)

Place (lid, longitude, latitude)

Route (lid, length, blaze, email)

Web Interface

We are planning to use Amazon Web Services to create a web-application. The app will host our interactive map. We are currently employing a simple wordpress web page with an Amazon EC2 instance. As we are still brainstorming, we may add to the features of the app but currently, we are planning to have a page for the interactive map, a page to submit new advice, a page to contact the developers, and an information page. Advice will be accessible by either clicking icons on the map (corresponding to the type of advice), or by filtering by various criteria such as relevant trips, areas, contributors, or types. We are also going to provide printable functionality, where all advice in an area will be printed to plain text. With the addition of new features we may switch to a different web-application format. Currently the webpage can be accessed via the following IP Address: 18.217.198.34.

Sample SQL

We have not yet finished creating the SQL database and sample queries in time for this milestone. This will be the next step in our development, alongside a more fleshed out web interface design. Most everything in the database other than keys will be nullable, and triggers will be put in place to populate timestamp attributes. Constraints mainly involve ensuring users don’t input invalid data, such as places not on the map, invalid months or years, invalid route lengths, and the like.

The bulk of the SQL queries will be used with filtering advice, a rather large and important part of this application. These filters can be applied by a user, much like shopping filters (price range, color, etc), and the comments will be displayed both on the interactive map,

as well as on a sidebar. For all filtered queries, they will involve joining Comment with the relevant other tables, and selecting the desired tuples.