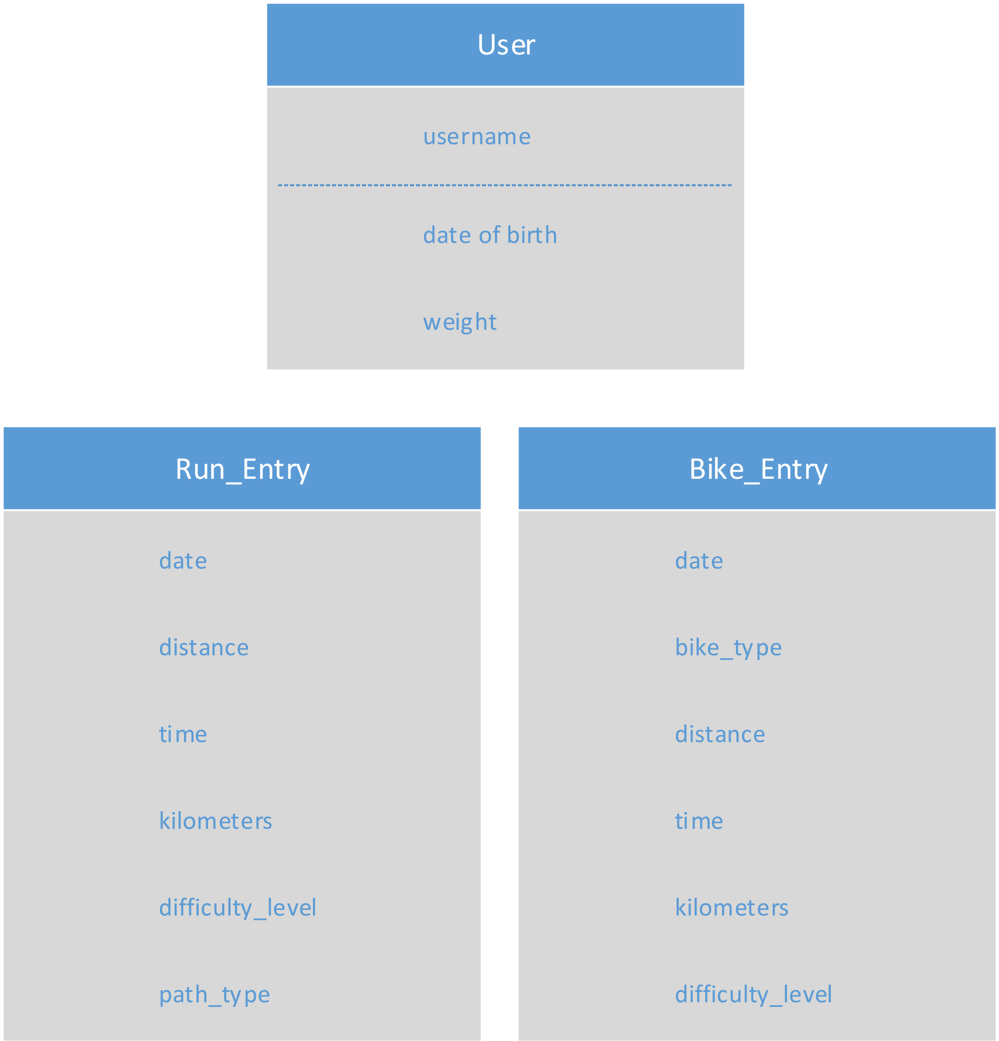
Holly Harmon
CS 419
Wolford
OSUID: 932456676

Assignment 3: Part 1

I am going to stick with my application from Assignment 2, which let the user keep track of bike rides they went on. The two main types of things being stored are individual users and the ride entries they have entered into the system. I'll be expanding on my previous assignment to include unique users, so that more than one person can use and store data (right now when viewing the website it shows all entries for one single user, so whoever visits the website updates/views the same info as everyone else. Very helpful if I'm the only one ever using the site, not very helpful otherwise). Each user would have a user-name/ID and optionally their age and weight. Each user has any number of ride entries they can track. Ride entries have a date, distance value, whether the distance is miles or kilometers, time (in minutes), type of bike, and the difficulty level.

Since this is essentially what I made for the previous assignment (just not in API form) I am also, probably, going to add another type of workout entry that users can track. I both bike and run, so the most helpful information to me would be bike rides and runs that I go on. So, the other type of thing to store would be runs, which would have distance, time, and an optional difficulty or location (maybe type of path like trail, sidewalk, track, etc).

For the database I am planning to store the data in entity groups with individual users as the root of each group. A user will be the parent of a number of bike entities and run entities that correspond to them. A user will only be viewing/updating data within their group and should not run into 1 write per second issues, since only one user be able to add one entry to their own database at a time. Consistency should also not be an issue since users will not be using queries that span entity groups. The database is also very flexible if ever it was clear some new properties were needed or I wanted to update the API to handle more types of workout entries or a generic all-purpose entry.



I looked Azure Table Storage as an additional non-relational database. It creates *tables* that contain *entities*. I think the structure of my database would remain relatively the same, but each user would have their own table, which would contain bike or run entries as entities.