P1 Task 1: Report

Raymond Zhu

923008555

Project 1 Task 1 took me approximately 5 ½ hours to complete.

MongoDB will be a suitable database for our application compared to a RDBMS since it is a document based database which allows it to hold different kinds of documents. This means that, a specific schema is not required, and each person’s information can be dynamic. Like a RDBMS, using MongoDB will not require the use of complicated join syntaxes along with the ability to use nested queries, etc. Additionally, information and storage retrieval will be a lot faster than the traditional RDBMS systems. Consequently, because MongoDB allows for flexible schemas, less complicated queries, and provides scalability, it is a clear choice to use MongoDB as our database for AggieFit.

Some fixes I think that could make the documents better formatted is to group attributes into nested documents. Fields such, weight, height, and age can be grouped together in a document such as “info”. Similarly, activityDuration, gymDuration, and stepCount could be grouped together into another document such as “activity”. Although a MongoDB database is flexible for different schemas and attribute types, it is still better to keep field types consistent. For example, activityGoal had attributes of integers and strings. For a user to interpret the data, the user will not fully understand what 60 means compared to 60 min. Is 60, 60 seconds, minutes, or hours?

In my opinion, it is not a good idea to place our data all on one server. This can potentially be catastrophic if we were to experience power loss, hardware/network failure, cyber-attacks, etc. To simply rely on just one server, we can lose important data given that any of the cases listed above were to occur. Although there may be solutions to saving/backing-up our database with snapshots, etc., there would not be any resources available to help us validate the recovered data.

Our data should be spread across multiple servers. Our servers should all have a copy of the entire database but have a focus on a subset of the data. A user will be directed to a certain server given the information the user is accessing. Changes made on a specific server will from time to time communicate with other servers to ensure every party is in-sync with one another. To ensure changes are atomic, or in our case, able to keep the ACID properties of a database, servers that are in direct communication during queries, etc. will be updated instantaneously.