Database Design (Updated)

Update: After reviewing the feedback on my database design, I have decided, for the time being, to ultimately stick with my original plan to use a document-based data system, which means there will not be many changes made to this report. In terms of stretch features, I plan on potentially adding certain key statistics to the Eye Profile system, like counters that track how often an individual uses the active & rest timers. This paper will include a model showcasing an updated JSON profile object with these stretch features implemented.

My web application will be storing data for use with the Eye Profile system, which allows individuals to persist certain articles of information relating to their eye health. Considering the specific nature of the data being stored through this system, the details provided by users for their Eye Profiles will be managed using a document-based database like MongoDB. The choice of using this particular style of database stems from a multitude of reasons. Firstly, the information being collected by my app won't necessarily have dependencies with data in other collections, and, as such, the profile objects my application will utilize won't interact with outside entities. This means that my app won't require many of the features prominent in relational databases. Furthermore, since my project will be incorporating JavaScript coding, MongoDB will give me the added benefit of avoiding context switching between JS and SQL. Lastly, as the primary developer for this application, my decision to employ a document-based database is partially rooted in my familiarity with such data storing facilities.

My project's database will maintain information provided by users in the form of Eye Profile JSON objects, and these objects will contain the following fields: a unique ID number, a username (string value), a password (string value), a first name (string value) , a last name (string value), medical contact details (string value), health conditions (string value), and an

eyeglass/contacts prescription (string value). For my app's stretch features, the profile objects will include two additional fields: an active timer count (num value) and a rest timer count (num value). These data structures will be used to provide a simple printout to users that they can access anytime, anywhere via a simple login confirmation.

Note: Any specifications/features listed here are subject to change.

Figure 1: Profile object example

Figure 2: Profile object example (updated with stretch features implemented)