## **Database Technology Selection**

DogPro will utilize MongoDB, a document-oriented database, in conjunction with Mongoose, an object data modeling library for MongoDB and Node.js. MongoDB's capabilities align well with the application's requirements, as it supports independent document storage for related data and accommodates one-to-many relationships with finite boundaries.

DogPro's primary focus is to manage user profiles, canine profiles, training schedules, and training materials. These elements typically have one-to-many relationships (one user with multiple dogs, one dog with multiple training tasks, etc.). MongoDB's flexibility in storing such data structures is a significant advantage. Plus, its schema-less nature permits fluid data adaptation as the application expands.

Mongoose will facilitate the creation of schemas for our models and manage data relationships. This will help maintain data integrity and provide a user-friendly interface for executing database operations.

## **Data Structures**

Our MongoDB database will primarily house the following document collections:

1. Users: This collection retains user data, encompassing login information, contact details, and an embedded document of their dogs. Each user document is structured as follows using JSON:

```
"_id": ObjectId(),

"name": String,

"email": String,

"password": String,

"dogProfiles": [

{

    "dogName": String,

    "dogBreed": String,

    "dogAge": Number,

    "dogWeight": Number,

    "feedingSchedule": {

        "breakfast": String,

        "lunch": String,

        "dinner": String

},

"pottySchedule": {
```

## How Data Structures are Employed in DogPro

In our DogPro application:

- Users: This collection is pivotal for handling user authentication and establishing relationships between users and their dogs. The embedded 'dogProfiles' field contains a list of dog profile documents owned by the user, which include all relevant details about each dog.

Note: The 'Dogs' and 'Training Schedules' collections have been encapsulated in the Users collection under 'dogProfiles' field. This design choice streamlines the data access process, as all relevant user and dog data can be fetched in a single query.

- Training Resources: This separate collection is a repository for various training resources, such as articles, videos, and recommended trainers.

```
{
  "_id": ObjectId(),
  "type": "Article",
  "title": "Training your dog to fetch",
  "url": "https://example.com/article"
}
```

While the Training Resources collection doesn't directly link to other collections, it provides valuable supplementary resources to support the training process.

Note: No change