AJ Henmy

SWDV 691

July 19, 2020

TheFriendZone

DATA MODEL

TheFriendZone's layout consists of users with personal contact details that will remain hidden but used to filter searches for other users and posts on the app.

The following is my beginning schema of how I wish to use a non-relational database to tie things together.

I will **embed** necessary objects that are related into documents. while **referencing** for objects that need to be accessed alone, like message board posts and chat messages.

These referenced objects will also included embedded objects like comments. Or in the case of discussion objects, other posts will be r**everse referenced**.

User Account
{ id: <objectid1> name: "James Charles", username:"newtotown, age: "18-22" ContactDetails: [{location: "32356" phone: "1234567890" email:"ashjay@yahoo.com }]

Profile
{ id: <objectid2> userid :<objectid1> message: "Hi, I'm new in town, looking for some friends" interests: [fitness, coding, movies] {

Profile (counselor)
{ id: <objectid5> userid :<objectid26> usertype: "Counselor" message: "Hi, I believe mental health should be our number one priority... link: joycejones.thefriendzone.com] {

Counselor Discussions
{ id: <objectid100> userid :<objectid26> Posts: [post_entries: (objectid3), (objectid13), (objectid6)] }

Post

{ id: <objectid3>
userid :<objectid1>
Subject: "I need a friend"
message: "lorem ipum unum.
lorem ipum unum.lorem ipum
unum"
{
 	
    comment_tid: 315
    user:'COMMENT_BY',
    message: TEXT,
    dateCreated: DATE_TIME,
 	
   },

Discussions
{ id: <objectid20> userid :<objectid1> Posts: [post_entries: (objectid3), (objectid13), (objectid6)] }

Zonebox (notifications)
{ boxid: <objectid100> userid :<objectid1> Notifications: [alerts : (objectid13), (comment_id 315), (objectid60)] }

Zonebox (Messages)
{ boxid: <objectid100> userid :<objectid1> (objectid12), Message Subject: [Message Content : Date Time (objectid13), (Message Subject: [Message Content : Date Time] }

Why MongoDB?

I chose to use MongoDB to process data and hold data for **TheFriendZone** app because it is fast and has a flexible schema design that scales.

Using **MongoDB** is appropriate because it won't require the use of joins since they can be expensive when querying information. Also, I can create several documents with embedded documents as well as reference documents on their own. I won't be limited to specific table schema and have to make columns for every type of field. The flexibility is what ultimately drew me to this option.

*I DECIDED TO STICK WITH MY PREVIOUSLY DESIGNED MODEL AS IT WAS APPROVED BY INSTRUCTOR/PEER REVIEWERS