**USE CASE STUDY REPORT**

**Group No.**: Group 8

**Student Names**: Jinhao Lu & Joshua Ji

**Executive Summary:**

The objective of this use case study focused on understanding how online discussion platforms can affect people’s lives through monitoring different user and page activities from a database-level perspective. We accomplished this goal through implementing a simplified database model that would sufficiently represent the core functions of an online discussion platform such as Reddit.

For designing the logical and conceptual model phase of our database. We used the MySQL workbench to design schemas and drew EER and UML diagrams to better represent and showcase the relations between the different entities for the relational database that we implemented. We decided not to implement every single field of an online discussion platform database and excluded fields that do not affect the fundamental usability of such a database. By approaching the project this way, we can have a better focus on our specific problem while not having to focus so much on maintaining the database itself.

The database can be connected and accessed by Python/R for data manipulation and visualization. Data in the database can be extracted by connecting Python/R as a file. The sample queries in the implementation of relation model MySQL and NoSQL part allow the user or the administrator to retrieve the specific topic on the online discussion platform. The database allows and supports the implementation of fundamental functions of an online discussion platform. To improve this database, the index is needed to increase the performance of the database when a huge amount of data is stored in the database.

We learned that there are a lot of similarities between online discussion platforms and relational databases. Information can be retrieved quite easily through a few search requirements. This has greatly inspired information to be spread around by people and could potentially trigger a wide range of social phenomena.

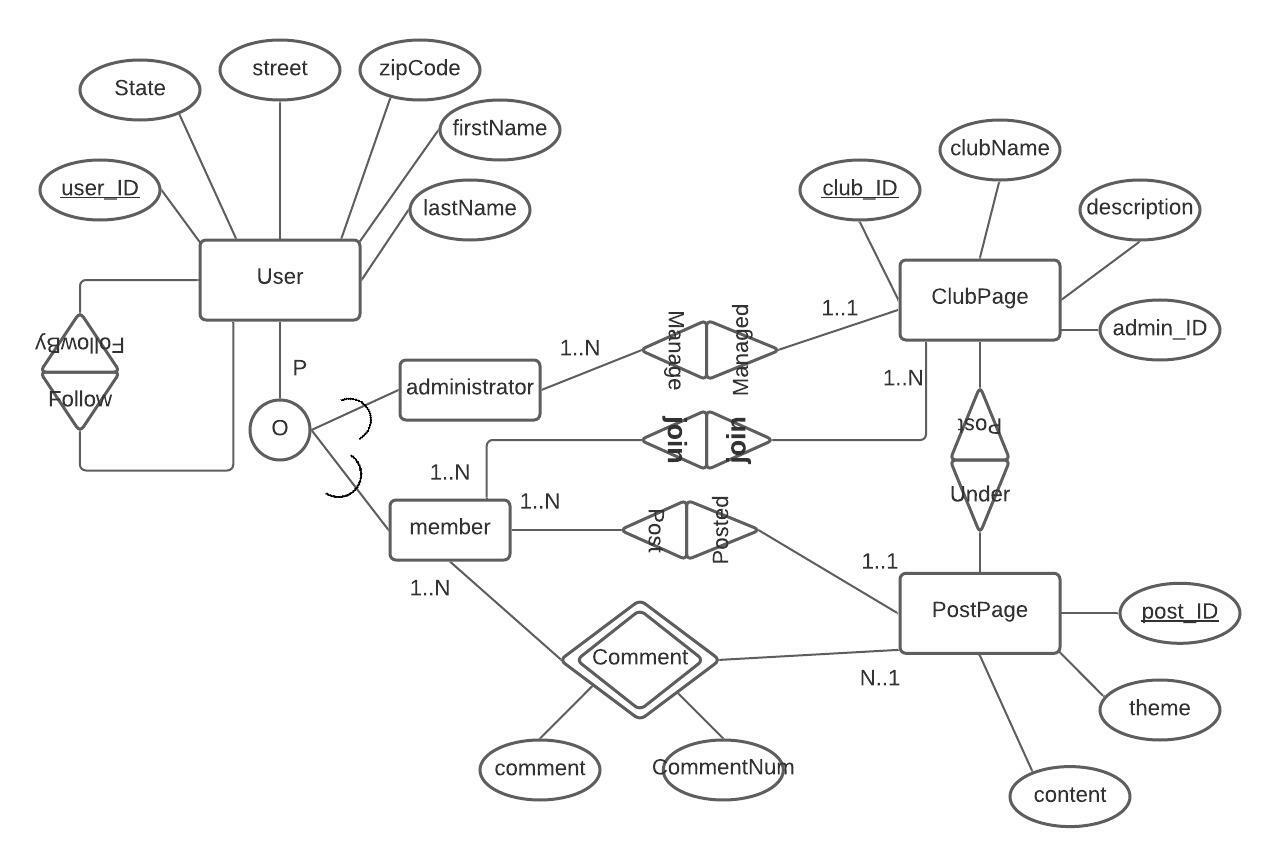
**I. Introduction**

Instagram, Twitter, and Facebook are the mainstream online social networking platforms. The phenomenon of dogecoin and Gamestop stock price rising suddenly and sharply showed that online discussion board platforms like Reddit are still influential in the online community.The goal of this case study is to design a database that implements the fundamental function of an online discussion board platform.

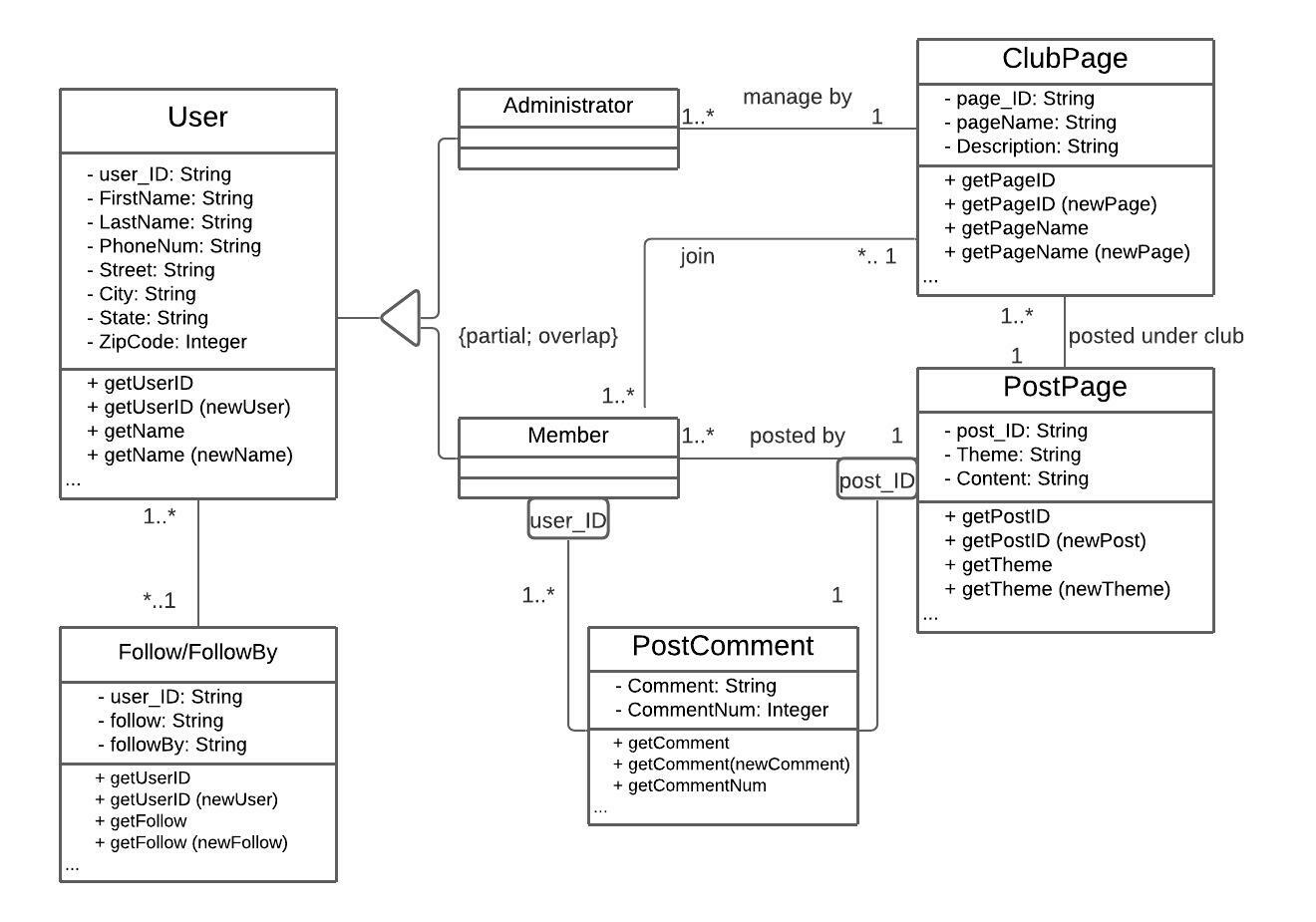
An online discussion board platform requires a relational database to store user’s information and the post’s information document, Follow, FollowBy, and Member. Once the user creates a new account on the online discussion board platform, the database will generate a unique user\_ID and store the user’s basic personal information into the user table in the database. If the user is interested in some clubs and joins the clubs, the member table will record that the user joins the clubs by storing the user id and the club id (page\_ID). The ClubPage table includes the identity of the club (page\_ID) on the platform and the basic setting of the club page like the name of the club and the short description for this club. The PostPage table includes the id, theme, and content for each post under a particular club. The PostComment table doesn’t contain a unique ID for each comment the user makes. The comment record can be tracked by the post\_ID and the user\_ID. Each comment has a comment number under a particular post. The Follow/FollowBy tables allow users to retrieve the followers or retrieve the following list.

**II. Conceptual Data Modeling**

1. **EER Diagram**



**2. UML Diagram**



**III. Mapping Conceptual Model to Relational Model**

**Primary Key - Underlined Foreign Key - *Italicized***

**USER** (user\_ID, FirstName, LastName, PhoneNum, Street, City, State, ZipCode)

**CLUBPAGE** (page\_ID, *admin\_ID*, PageName, Description)

FOREIGN KEY admin\_ID refers to user\_ID in USER; NULL NOT ALLOWED

**POSTPAGE** (post\_ID, *user\_ID*, *club\_ID*, Theme, Content)

FOREIGN KEY user\_ID refers to user\_ID in USER; NULL NOT ALLOWED

FOREIGN KEY club\_ID refers to club\_ID in CLUBPAGE; NULL NOT ALLOWED

**POSTCOMMENT** (*user\_ID, post\_ID*, Comment, CommentNUM, ComOnComNum)

FOREIGN KEY user\_ID refers to user\_ID in USER; NULL NOT ALLOWED

FOREIGN KEY post\_ID refers to post\_ID in POSTPAGE; NULL NOT ALLOWED

**FOLLOW** (user\_ID, Follow) user\_ID NULL NOT ALLOWED

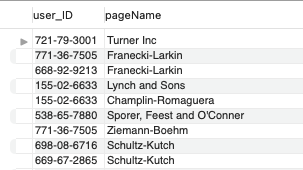
**FOLLOWBY** (user\_ID, FollowBy) user\_ID NULL NOT ALLOWED

**MEMBER** (*user\_ID, page\_ID*, level)

FOREIGN KEY user\_ID refers to user\_ID in USER; NULL NOT ALLOWED

FOREIGN KEY page\_ID refers to page\_ID in CLUBPAGE; NULL NOT ALLOWED

**IV. Implementation of Relation Model via MySQL and NoSQL**

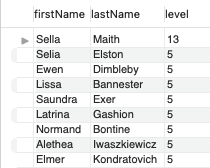
**MySQL Implementation:**

**Query 1: Find the club name that the users join.**

select u.user\_ID, c.pageName

from user as u, clubPage as c, member as m

where u.user\_ID = m.user\_ID and m.page\_ID = c.page\_ID;



**Query 2: Find the member's id, name, and level of the club "Reichert and Sons".**

select u.firstName, u.lastName, m.level

from user as u, member as m

where u.user\_ID = m.user\_ID and m.page\_ID

= (select page\_ID from clubPage

where pageName = "Reichert and Sons");



**Query 3: Find the name and city come from of a club administrator who has more than 3 followers.**

select distinct u.firstName, u.lastName, u.city

from user as u, clubPage as c

where u.user\_ID = c.admin\_ID and 3 < (select count(u.user\_ID) from user as u, follow as f

where u.user\_ID = f.follow);

**Query 4: Find out the club that contains the post that title is "Aga Khan University"**

**and return the number of members of the club.**



select c.pageName, count(m.user\_ID)

from clubPage as c, member as m, postPage as p

where p.title ="Aga Khan University" and p.page\_ID = m.page\_ID and p.page\_ID = c.page\_ID group by c.page\_ID;

**Query 5: Find out the user's followers and the user's name who reply "LoL" under the post "Assumption University of Thailand".**

select u.firstName, u.lastName, fb.followBy

from user as u, followBy as fb, postPage as p, postComment as pc

where pc.Comment = "LoL" and p.title = "Assumption University of Thailand" and u.user\_ID = pc.user\_ID and pc.post\_ID = p.post\_ID;

**NoSQL implementation:**

We implemented the NoSQL database via MongoDB atlas, and after connecting the Mongodb server to our local laptop via command line language, we were able to run NoSQL queries on terminal.

We started the NoSQL implementation by import data with command line tool: “mongoimport”

*mongoimport --host cluster0-shard-00-01.zk3eq.mongodb.net:27017 --db FinalProject --type json --file ~/Desktop/IE6700\_Final\_Project/Data/member.json --jsonArray --authenticationDatabase admin --ssl --username yixuaj --password tBu8twZAUdlBmYo9*

Then we connected onto our MongoDB server through command line tool: “mongosh”

*mongosh "mongodb+srv://cluster0.zk3eq.mongodb.net/FinalProject" --username yixuaj*

*--<password>*

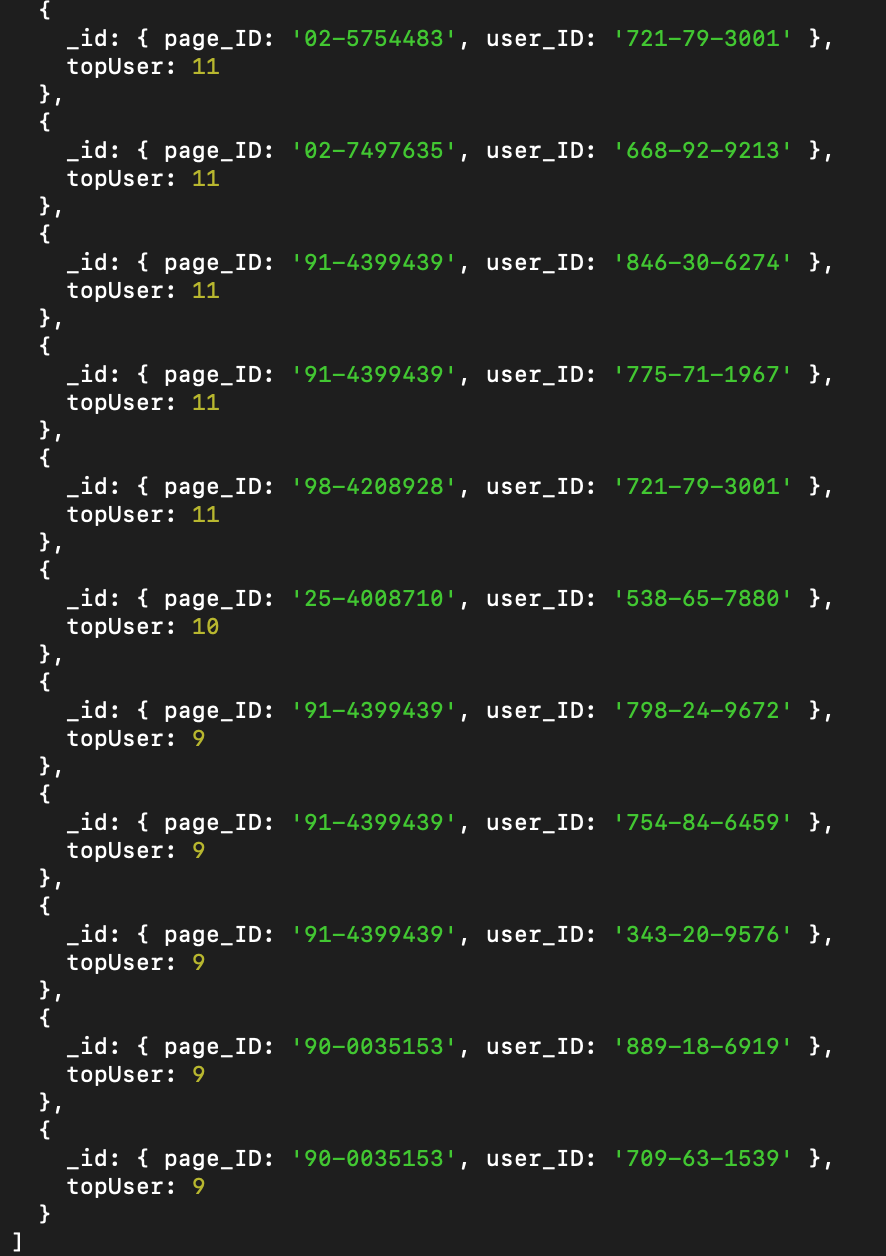
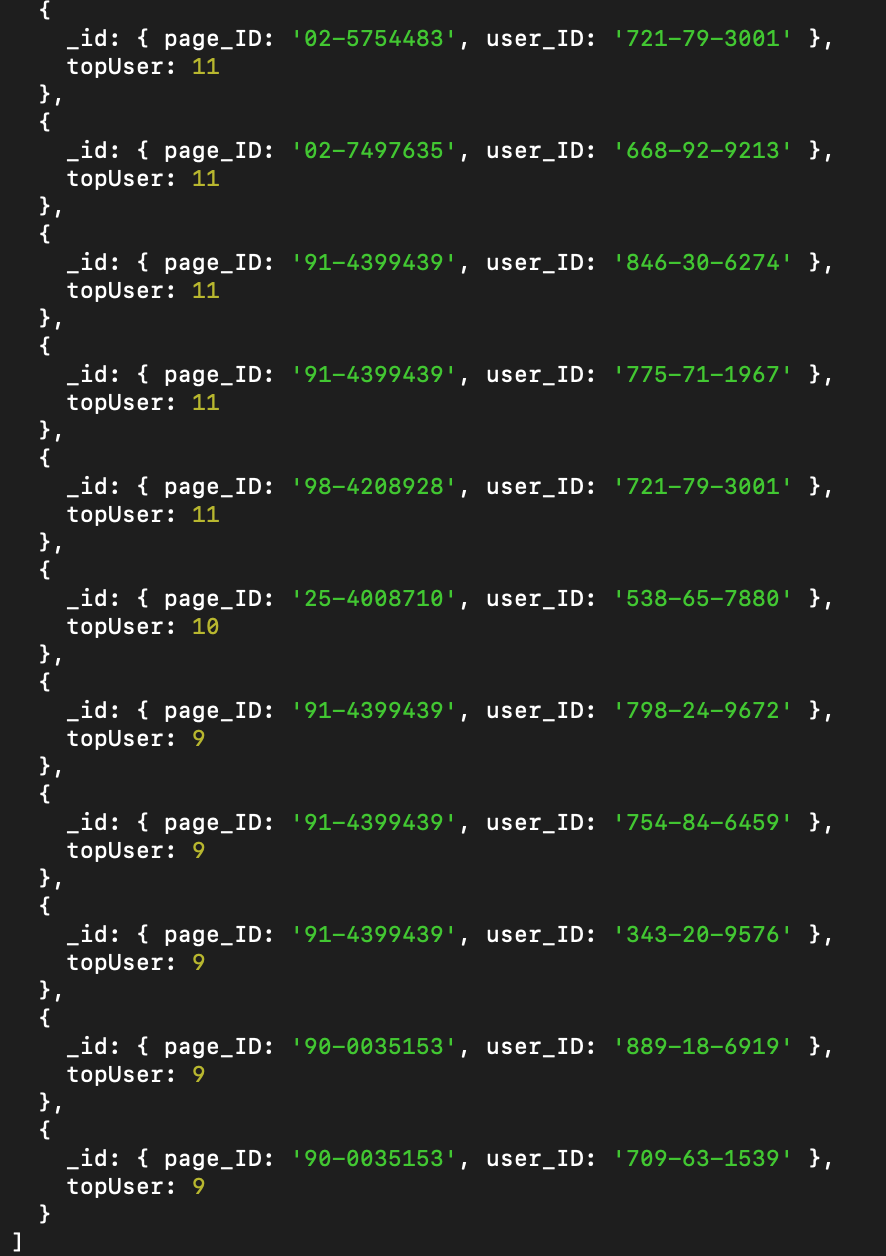
**Query 1: Final the top user from each page**

db.member.aggregate([

{$group:{\_id:{"page\_ID":"$page\_ID", "user\_ID":"$user\_ID"}, topUser:{$max:"$level"}}},

{$sort:{topUser:-1}}

]);



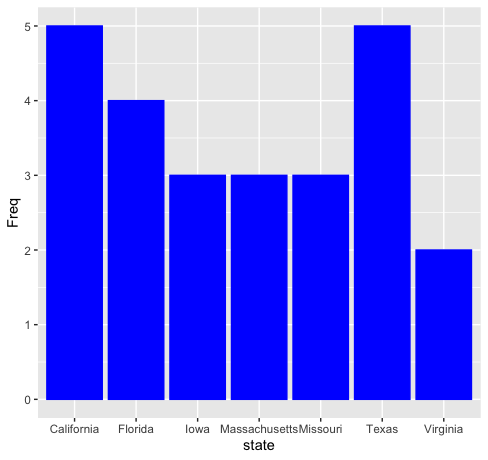
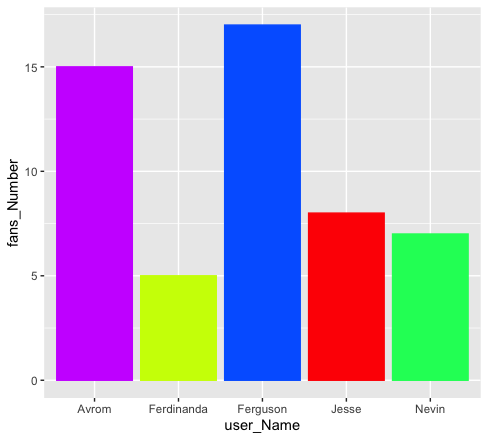
**Query 2: Find all users who live in California**



db.user.find({state:"California"});

**V. Database Access via Python/R**

The database can be accessed by R with the package “RMySQL” and the visualization of analyzed data is shown below.



**Graph 1: Number of followers the users have Graph 2: Number of users from the states**

**PyMongo:**

PyMongo is a python package that allows us to run NoSQL queries with python code. This is what we did:

*>>import pymongo*

*>>from pymongo import MongoClient*

After calling MongoClient we could connect to our MongoDB server through:

>>*cluster = MongoClient("mongodb+srv://yixuaj:tBu8twZAUdlBmYo9@*[*cluster0.zk3eq.mongodb.net/FinalProject?retryWrites=true&w=majority*](http://cluster0.zk3eq.mongodb.net/FinalProject?retryWrites=true&w=majority)*")*

Where we copy the address of our MongoDB server. Finally, we could retrieve different collections with NoSQL.

**# Query 1: Find users who live in the state California**

*>>state = {"state":"California"}*

*>>results = user.find(state)*

*>>for result in results:*

*>> print(result);*

**# Query 2: find how many users follow the page with page\_ID: 91-4399439**

*>>member = db["member"]*

*>>pageid = {"page\_ID": "91-4399439"}*

*>>cursor = member.find(pageid)*

*>>print(cursor.count())*

**VI. Summary and Recommendation**

The unexpected spike in Gamestop stocks and Dogecoin value in the past year has drawn millions of people’s attention. Trying to understand social phenomena like these is very intriguing. Our use case study focused on understanding how online discussion platforms can affect people’s lives through monitoring different user and page activities from a database-level perspective.

We started off the project thinking we would be able to implement a full-bloom database model that would perfectly represent an online discussion platform such as Reddit. Our first approach was to implement a database model based off of the Chinese reddit site, Baidu Tieba(百度贴吧). But we quickly ran into our first obstacle of not being able to find an API key that would grant us access to the Baidu Tieba database structure. However, we did not want to just give up and completely change up the project. So, we found a way to work our way around this. We took a look at the Reddit database model and did an in-depth study on the usability of the Baidu Tieba. Because we both had previous experiences using both platforms, we quickly came up with a conceptual design of our Database. We had to make some compromises because we realized that implementing the whole Baidu Tieba database would be too ambitious of a goal. This does not mean we wanted to simplify our work, more so we wanted to only present the core features of an online discussion platform that we think would sufficiently deliver the results we are looking for. Designing the EER and UML diagrams was not an easy step, for we spent a lot of time trying to figure out what were the best fields and tables to include to prevent making our database too simple or too trivial. The advantage of us spending so much time trying to simplify the fields for our database is so we could effectively cut to the core of the problem. The shortcomings are obvious as well, for our model does not fully resemble how an online discussion platform’s full capacity of functions. Some recommendations would spend more time studying the usability of discussion platforms and try to gain more user experience so that we could know what the main draw of the general population is when using such a platform.

We ran into our second obstacle when trying to design the database schemas. We realized the different relations between the tables were hard to map out, for there were a lot of many-to-many relationships between our entities. For example, a user could be following multiple pages, and a page could be followed by multiple users. An online discussion platform is so open and loosely connected, it makes designing the logical connections very tricky. We were able to bypass the problem by carefully normalizing our different tables and creating a few join-tables. In our case, we have a table called “members” which consists of the “user id” and “page id” so that we could easily create composite keys with these two fields in other tables. The advantage of this is that our entities have clear relations, but the shortcoming is that we are simplifying the database again. A recommendation to improve this process next time would be creating index tables, it is extremely helpful on large databases to implement index tables. Another plus side for index tables is that it improves the query performance a lot.

After finally setting up the database in the MySQL environment and running a few queries, we were also able to implement the NoSQL database within MongoDB. We wanted to make a few final remarks regarding our project. The goal of our project originated from trying to monitor social phenomena through a database-level perspective was partially accomplished. We were able to run queries that allowed us to have a general idea of why people would pay so much attention to topics originated from online discussion platforms. To some degree, an online discussion platform is like a database by itself. With people inserting data into it almost every second. People could easily retrieve the information they wanted to look for by simply navigating through different topic pages. And because of the accessibility to such an open-sourced database, it is not hard to imagine how much value such a platform possesses. While trying to fully represent such a complex “database” would require a lot more work and a lot more manpower, by sacrificing some features and preserving the core features of an online discussion platform and resembling it on a much smaller scale still provided us with a lot of useful insights.