**Project Report**

**Report of *HAHAHAHA forum* Project**

Group 15:

黄心桐2030026064马思嘉2030026105

马逸舒2030026107宁嘉翊2030026110

1. **Project Description:**

(1) Theme: HAHAHAHA Forum

(2) Purpose: Provide a place for people to share their happiness

(3) Functions:

a. registration

b. Log in

c. Post

d. Delete the Post

e. Modify the Post

f. Comment

g. Nested Comment

h. Notice

i. History List

j. Jump List in Notice and History

k. Automatic Matching of Users’ Avatars (Based on the initial letter of user’s name)

l. Browsing Page of Post (Paging, hotlist, classification, details)

m. Searching by Keywords

1. **Data Description and Data Collection**

Downloaded HappyDB dataset which contains 100536 data and 5000 data crawled from Twitter.

1. **Data Processing and Data Analysis**
2. Data Analysis and Data Processing: Regular expression + Text2emotion + LSTM Network
3. Visualization: Pyecharts
4. Database: Multiple entities that satisfy a paradigm design
5. **Latest Version of ER Diagram:**

图示

描述已自动生成

1. **Assumptions we made:**

(1) Login users have their own unique basic information.

(2) Users can release posts; a post can only be released by one user.

(3) A user with privilege can manage many columns, and a column can be managed by one user.

(4) A post can only belong to one module.

(5) A comment can only follow one blog or one comment, and a blog or a comment can contain many comments.

(6) A notice only has one release user and one send user. A user can release and send many notices.

1. **Functional Dependencies**

LoginUser (id→password,

id→last\_login,

id→is\_superuser,

id→first\_name,

id→last\_name,

id→email,

id→is\_staff,

id→is\_active,

id→date\_joined,

id→username,

id→privilege,

username→id,

username→password,

username→last\_login,

username→is\_superuser,

username→first\_name,

username→last\_name,

username→email,

username→is\_staff,

username→is\_active,

username→date\_joined,

username→privilege)

Basic\_Information (id→age,

id→gender,

id→country,

id→uid\_id,

uid\_id→id,

uid\_id→age,

uid\_id→gender,

uid\_id→country)

Column (id→name,

id→description,

id→post\_number,

id→created\_at,

id→updated\_at,

id→manager\_id)

Comment (id→content,

id→created\_at,

id→updated\_at,

id→author\_id,

id→post\_id,

id→comment\_parent\_id)

Notice (id→object\_id,

id→status,

id→type,

id→integer,

id→created\_at,

id→updated\_at,

id→content\_type\_id,

id→receiver\_id,

id→sender\_id)

Post (id→title,

id→content,

id→photo,

id→created\_at,

id→updated\_at,

id→comment\_num,

id→author\_id,

id→column\_id)

1. **SQL Codes and Explanations:**

--

-- Create model LoginUser

--

CREATE TABLE LoginUser IF NOT EXIST(

id integer AUTO\_INCREMENT NOT NULL PRIMARY KEY,

password varchar(128) NOT NULL,

last\_login datetime(6) NULL,

is\_superuser bool NOT NULL,

first\_name varchar(30) NOT NULL,

last\_name varchar(150) NOT NULL,

email varchar(254) NOT NULL,

is\_staff bool NOT NULL,

is\_active bool NOT NULL,

date\_joined datetime(6) NOT NULL,

username varchar(25) NOT NULL UNIQUE,

privilege varchar(200) NOT NULL

);

--

-- Create model Basic\_Information

--

CREATE TABLE forum\_basic\_information (

id integer AUTO\_INCREMENT NOT NULL PRIMARY KEY,

age integer NOT NULL,

gender varchar(2) NOT NULL,

country varchar(50) NOT NULL,

uid\_id integer NOT NULL UNIQUE,

FOREIGN KEY (uid\_id) REFERENCES class(LoginUser)

);

--

-- Create model Column

--

CREATE TABLE Column (

id integer AUTO\_INCREMENT NOT NULL PRIMARY KEY,

name varchar(30) NOT NULL,

description longtext NOT NULL,

post\_number integer NOT NULL,

created\_at datetime(6) NOT NULL,

updated\_at datetime(6) NULL,

manager\_id integer NOT NULL,

FOREIGN KEY (manager\_id) REFERENCES class (LoginUser)

);

--

-- Create model Comment

--

CREATE TABLE Comment (

id integer AUTO\_INCREMENT NOT NULL PRIMARY KEY,

content longtext NOT NULL,

created\_at datetime(6) NOT NULL,

updated\_at datetime(6) NOT NULL,

author\_id integer NOT NULL,

post\_id integer NOT NULL,

comment\_parent\_id integer NULL,

FOREIGN KEY (Post\_id) REFERENCES class (Post),

FOREIGN KEY (comment\_parent\_id) REFERENCES class (Comment), --for comment follows comment

FOREIGN KEY (author\_id) REFERENCES class (LoginUser)

);

--

-- Create model Notice

--

CREATE TABLE Notice (

id integer AUTO\_INCREMENT NOT NULL PRIMARY KEY,

object\_id integer UNSIGNED NOT NULL,

status bool NOT NULL,

type integer NOT NULL,

created\_at datetime(6) NOT NULL,

updated\_at datetime(6) NOT NULL,

content\_type\_id integer NOT NULL,

receiver\_id integer NOT NULL,

sender\_id integer NOT NULL,

FOREIGN KEY (receiver\_id) REFERENCES class (LoginUser),

FOREIGN KEY (sender\_id) REFERENCES class (LoginUser)

);

--

-- Create model Post

--

CREATE TABLE Post (

id integer AUTO\_INCREMENT NOT NULL PRIMARY KEY,

title varchar(64) NOT NULL,

content varchar(3000) NOT NULL,

photo varchar(128) NULL,

created\_at date NOT NULL,

updated\_at datetime(6) NULL,

comment\_num integer NOT NULL,

author\_id integer NOT NULL,

column\_id integer NOT NULL，

FOREIGN KEY (author\_id) REFERENCES class (LoginUser),

FOREIGN KEY (column\_id) REFERENCES class (Column)

);

1. **Website Design and feature implementation**
2. Posting Page:

图形用户界面, 应用程序, Teams

描述已自动生成

1. Category Page:

**图形用户界面, 应用程序

描述已自动生成**

1. Posting Page in enjoy\_the\_moment category:

**图形用户界面, 应用程序

描述已自动生成**

1. Visualization Page:

**图形用户界面, 图表

描述已自动生成**

1. Personal Notice Page:

**图形用户界面, 文本, 应用程序

描述已自动生成**

1. Threads Page:

**图形用户界面, 应用程序

描述已自动生成**

1. Searching Page:

**图形用户界面, 应用程序

描述已自动生成**

1. **Difficulties & handling**

(1) Data Collection:

When we collect data, the first question is where to collect the data. We finally decided to gather information on social media and apply the Twitter API. But tweets are truncated when they reach a certain number of words. In order to solve this problem we search the tweepy package, find a get\_full\_text() method to solve this problem.

(2) Data Cleansing:

When we see our cleaned data, one headache is its classification of types of happiness. When using new data, existed package only can judge happiness or not. So, we are faced with the problem of classifying data. Since the text is NLP, this is usually the best time to use a neural network, but we didn't build NLP models before and had to learn from scratch. Finally, we learned the hard way how to build our model in LSTM. The final classification.

(3) Web Design:

In the process of finishing the project, I encountered some problems with website layout design, function realizing and the way of connecting the functions I wrote in views with the templates. I think writing functions are the hardest for me. I searched for some courses about Django on bilibili and CSDN, and then tried to write my own program. I also went to ask teachers for help. All the teachers that I asked were patient and enthusiastic. They helped me to debug my code and it could run successfully. I would like to express my sincere thanks for that.

(4) Visualization

Visualization is not a hard module. But it’s quite important to find an appropriate package, a better tool to show the visualization. I choose to use pyecharts, combined with python and echarts, which turned out to have a very good effect at last. Then, it’s about data cleansing. The data need to be cleaned several times since the dataset is so big and so much interesting data is hidden inside.

(5) Database

After we build up model.py, we need to make migration. However, it is hard for us to connect pycharm with and upload csv file to mysql, phpadmin or sqlite which can automatically download in pycharm on our own. Because our computers have problems in recognizing the right password to the database. After 2 days of struggling in asking help from classmates, TAs, and professors, we finally give up building the local database which is our first choice. Instead, we seek help from our database TA Kitty to reset our phpadmin password which we ourselves can’t deal with. Thank god, it finally works. Additionally, uploading data needs patience, circumspection, and experience. Next time, we may try our best to make clear conversation and fully understand what data is needed so that we can save more time.