**SHMUBlog System detailed design**

|  |  |
| --- | --- |
| **Project Name**: | SHMUBlog System |
| **Team Name**: | Group 1 |
| **Team members:** | Zeng Kete |
|  | Liu Tong |
|  | Li Pengfei |
|  | Liu Zhefeng |

|  |  |  |  |
| --- | --- | --- | --- |
| **Draftier** | Whole team | **Date** | 06/04/2019 |
| **Assessor** |  | **Date** |  |
| **ratifier** |  | **Date** |  |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Remarks** |
| 19/03/2019 | 1.0 | Whole team | First deaft |
| 06/04/2019 | 2.0 | Whole team | overhaul |

Content

[1. Introduction 1](#_Toc5543082)

[1.1. Document purpose 1](#_Toc5543083)

[1.2. Document scope 1](#_Toc5543084)

[1.3. The reader 1](#_Toc5543085)

[2. Overview 1](#_Toc5543086)

[2.1. System goals 1](#_Toc5543087)

[2.2. System environment 2](#_Toc5543088)

[2.2.1. Develop environment 2](#_Toc5543089)

[2.2.2. Runtine environment 2](#_Toc5543090)

[3. System structure design 3](#_Toc5543091)

[3.1. System architecture 3](#_Toc5543092)

[3.2. Function structure design 3](#_Toc5543093)

[3.3. Module and Interface Description 4](#_Toc5543094)

[3.3.1. Authentication module 4](#_Toc5543095)

[3.3.2. User module 4](#_Toc5543096)

[3.3.3. Post module 6](#_Toc5543097)

[3.3.4. Administrator module 8](#_Toc5543098)

[4. System main interface design 9](#_Toc5543099)

[5. Major function description 11](#_Toc5543100)

[6. Database design 12](#_Toc5543101)

[6.1. Design basis 12](#_Toc5543102)

[6.2. Database types and features 13](#_Toc5543103)

[6.3. E-R diagram 13](#_Toc5543104)

[6.4. Physical structure design 14](#_Toc5543105)

[7. System error handing design 14](#_Toc5543106)

[7.1. Error massage 14](#_Toc5543107)

[7.2. Remedial measures 15](#_Toc5543108)

[8. Conclusion 16](#_Toc5543109)

[9. References 16](#_Toc5543110)

1. Introduction
   1. Document purpose

The purpose of this document is based on the system module design considerations, including system structure, system module and interface, expected interface, main functions, database design, etc., to provide a basis for later programming and system maintenance.

* 1. Document scope

This document mainly includes system structure design, module division, expected function and interface, database design and error handling design.

* 1. The reader

The intended authors of this document are system designers, system developers, system testers, and project reviewers.

1. Overview
   1. System goals

The purpose of this system is to write a log, express their feeling about everything,share learning experience and communicate.

Users can register in this system to apply for their own blog. Once users have applied for their own blogs, they can post their feelings and experiences on their blogs. When they post their articles, visitors can comment on the content of the logs published by users. Users can publish articles, pictures and messages on the blog to communicate with others.

* 1. System environment
     1. Develop environment

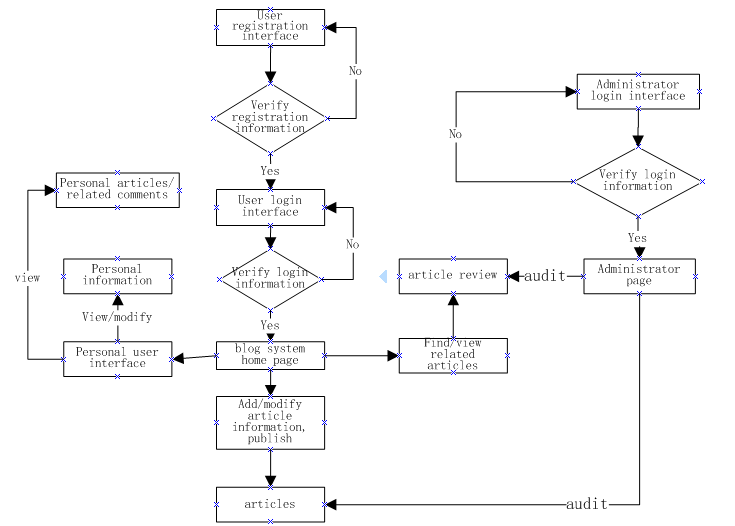
|  |  |  |
| --- | --- | --- |
| Environment | Name | Version |
| Operating system | Windows | Windows10 |
| Database | SQLite | 3.26.0 |
| Programming language | python | 3.7 |
| IDE | Pycharm | 2019.1 |

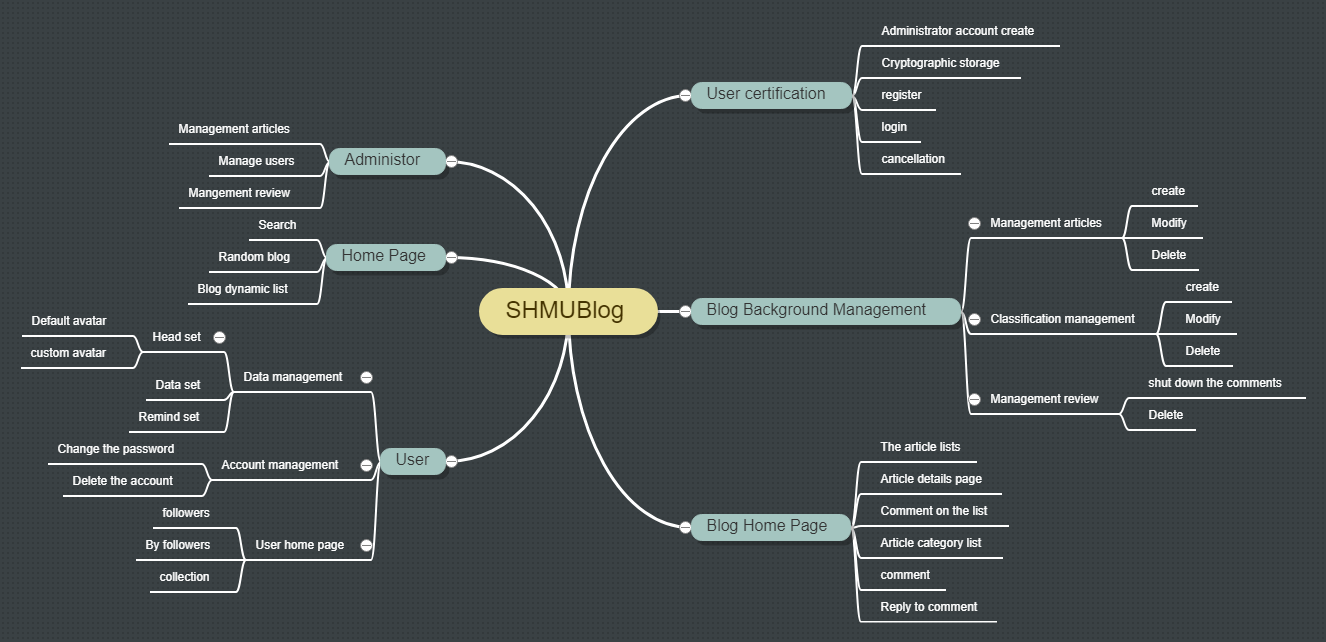
* + 1. Runtine environment

System running conditions:

|  |  |  |
| --- | --- | --- |
| Environment | Name | configuration |
| Operating system | Windows/Mac/Linux | None |
| Hardware | CPU,Hard disk etc. | Minimum |

1. System structure design
   1. System architecture



* 1. Function structure design
  2. Module and Interface Description
     1. Authentication module

1. Module description

Realize account management function.

2. Function

Responsible for account registration, user login and user logout.

3. Detailed design

This module is mainly composed of the following three functions, each of which implements specific functions.

**Login**: To achieve the user login function, if the login is successful, it will automatically jump to the user home page interface. This function will call the index function in the main module to achieve the main interface user sorting display.

**Register**: Realize the user registration function, and user registration information into the database.

**Logout**: Realize user account cancellation function, from user identity to tourist identity.

4. Logical process

Logout

Login

Register

* + 1. User module

1. Module description

Realize the user personal management function module.

2. Function

Realize the user name index, follow the function and its display, personal information modification, account deletion and related blog, comment management and other functions.

3. Detailed design

**Index**: Index all users by user name to provide user lookup.

**Follow**: Follow other bloggers.

**Unfollow**: Remove attention from other bloggers.

**Show\_followers, show\_following, show\_collections**: Displays the user's followers, followers, and favorite posts.

**Edit\_profile**: Users modify personal information and input it into the database.

**Change\_avatar**: The user modifies the profile picture and enters it into database.

**Upload\_avatar**: Upload user profile picture and enters it into database.

**Change\_password**: Change account password and enters it into database.

**Privacy\_setting**: Sets whether a personal collection is public and modifies the values associated with the form.

**Delete\_account**: Delete personal accounts and delete relevant account information from the database.

**Manage\_posts, manage\_comments**: Users view, modify, and delete posts and comments about themselves.

4. Logical process

users

Edit\_profile

follow

Change\_avatar

Change\_passward

Privacy\_setting

Unfollow

Show\_ following

Show\_ followers

Manage\_ comments

Manage\_posts

Delete\_account

* + 1. Post module

1. Module description

Realize the series functions of blog.

2. Function

Realize blog post release, edit, delete, report, display, collection, cancel collection and comment published, status setting, reply, report, delete function.

3. Detailed design

**New\_post**: Publish new articles and enter them into the database.

**Edit\_post**: edit articles.

**Delete\_post**: delete articles and enter them into the database.

**Report\_post**: Report articles and make the number of reported articles plus one.

**Show\_post**: Show the article

**New\_comment**: Post new comments below the article.

**Set\_comment**: Set the comment status of the article.

**Reply\_comment**: Respond to comments and comment on other people's comments.

**Delete\_comment**: Delete the comment and the corresponding content from the database.

**Report\_comment**: report comments.

**Collect, uncollect**: Collection blog articles, cancel the collection blog articles.

**Delete\_post\_category**: Remove categories from blog posts.

4. Logical process

new\_post

edit\_post

collect

report\_post

show\_post

uncollect

delete\_post

new\_comment

delete\_comment

report\_comment

set\_comment

* + 1. Administrator module

1. Module description

The implementation of the administrator management function.

2. Function

Realize the administrator of the blog, classification, comment view and delete functions as well as the user view and ban functions.

3. Detailed design

**Index**: The order in which posts, categories, comments, and users are displayed when viewed by an administrator.

**Manage\_post**: Check out the blog post and related information.

**Delete\_post:** Delete the blog post and remove the relevant information from the database.

**Manage\_user:** Check out users and related information.

**Block\_user, unblock\_user:** Blocked users, unblocked users, and update the database information in a timely manner.

**Manage\_comment:** Check out comments and related information.

**Delete\_comment:** Delete comments and remove the relevant information from the database.

**Manage\_category:** check out category and related information.

**Delete\_category:** Delete category and remove the relevant information from the database.

4. Logical process

administrator

Manage\_ category

Manage\_user

Manage\_article

Manage\_ comment

Delete\_ comment

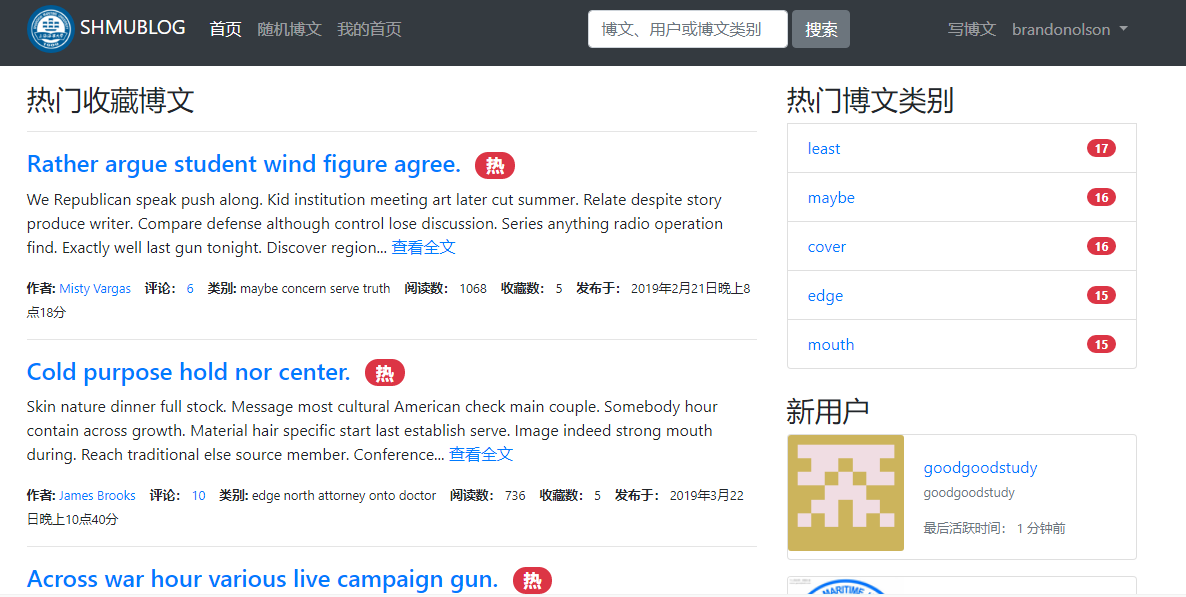
Delete\_ category

Unblock\_user

Block\_ user

Delete\_article

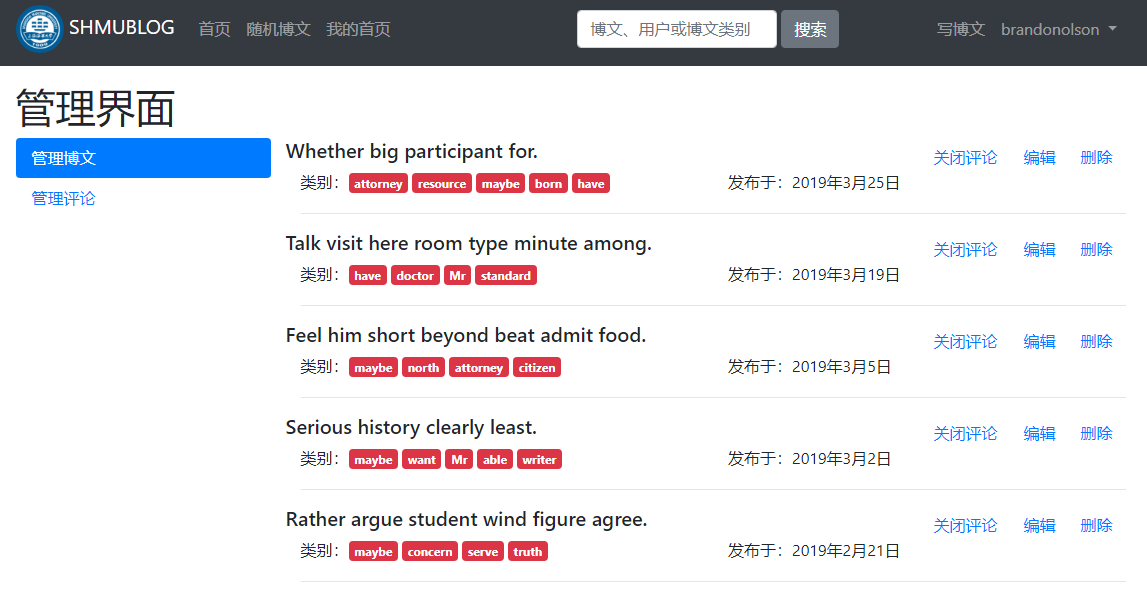
1. System main interface design
2. Home page



1. User home page



3. User management page



4. User data set page



5. Administrator management page



1. Major function description

1. Articles management: The administrator can view all the articles in the article library. The articles are sorted according to the number of reports. The administrator can delete the articles. The users can view, modify and delete their own articles.

2. Users management: The administrator can view the user name, email address, profile, registration time, and last active time of all users, and can ban and unban users. Once the user is banned, it will be equivalent to the identity of tourists.

3. Comment management: The administrator can view all comments, commenters, related blog posts, the number and time of being reported. The comments are sorted by the number of being reported. The administrator can delete the related comments. Users can view and delete comments related to their own, and can choose to open or close their own article comments function.

4. Classification management: Administrators can view and delete any categories. Users can create, modify and delete categories.

5. User data management: Users can change their profile information, profile picture and password, and set whether their favorite list is public.

6. User authentication: Realize user registration login and account logout function.

7. Comment: Users can comment on the articles and the comments.

8. Collection and attention: Users can collect their favorite articles and follow their favorite bloggers, and can cancel the collection and cancel the following.

1. Database design
   1. Design basis

This blog system is aimed at the minority, the frequency and traffic of data access are not high, so there is no special demand for the maximum data storage, data growth and storage time.

* 1. Database types and features

The system uses the relational data database to store data. The relational database has the characteristics of data centralized control, data independence, less data redundancy and data structure, which is suitable for the less data volume of the system.

* 1. E-R diagram

Users

focus

inform

massage

set

Blog home page

publish

comments

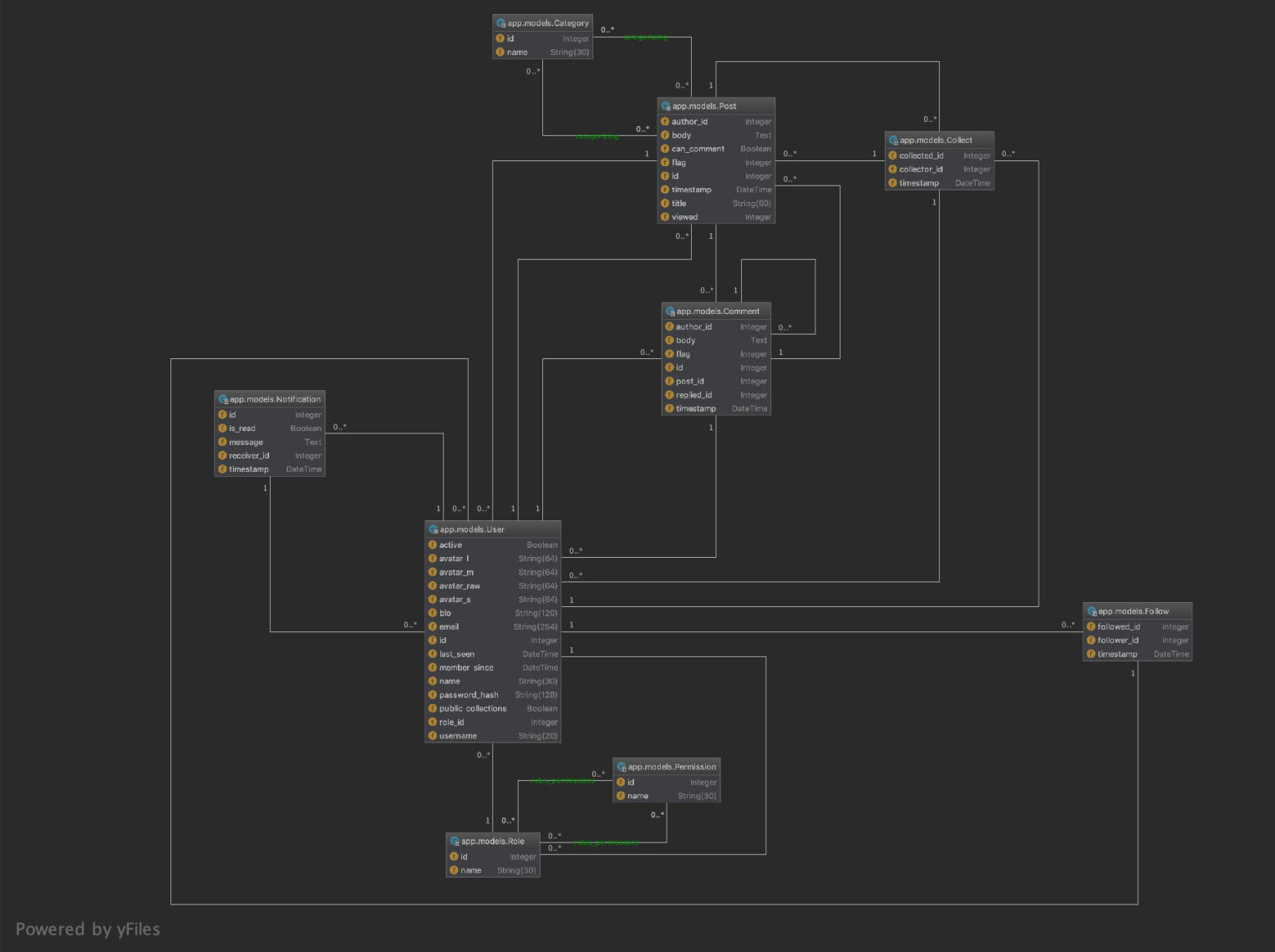
belong

essay

publish

collect

like

* 1. Physical structure design

1. System error handing design
   1. Error massage

Chart the form of the system output information and the corresponding processing method for each possible error situation.

Because the input information does not conform to the specification, it is called soft error;

Due to hardware errors (such as network transmission timeout, hardware error, etc.), it is called a hard error;

Due to some key operations, the validation mechanism should be provided;

For data, test documents, to provide the corresponding privacy Settings.

|  |  |  |
| --- | --- | --- |
| Error type | Problem | Reason |
| Database error | Connection | Timeout |
| Interput |
| Database | Data transfor error |
| Database overflow |
| System partial customization error | Input error | The user entered an invalid character |
| Program error | There is an unknown BUG in the program |

* 1. Remedial measures

Indicate the workarounds that may be taken after the failure, mainly including:

For soft errors, the system maintains the current page and returns the prompt "invalid input, please re-enter"

For hard errors, the system will locate the error step and output the corresponding error code, requiring the user to re-run.

Backup technology a backup technology intended to be used, such as periodically recording disk information, for the creation and start-up of copies enabled when raw data is lost.

Recovery the recovery restart technique that will be used in the startup technology description to enable the software to recover execution from the point of failure or a way for software to run from scratch.

1. Conclusion

As a team, we feel that this project has great potential to very useful to our user.It will enable them to more effectively communicate and learn each other, and thus making a large impact in their life. This project consists of two main actors: User and Administrator. And the Flask framework , Jinjia2 makes it easy to design and implement.In the end ,we will improve this system all the way.

1. References

<http://naotu.baidu.com/file/>

<https://translate.google.cn/>

<http://www.doc88.com/> <https://wenku.baidu.com/view/e93da44abf23482fb4daa58da0116c175f0e1ea4.html>

https://wenku.baidu.com/view/98b0095177232f60ddcca1df.html