|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IIT-536 final project** |  |  | **version** | V1.0 | **secret level** |  |
| **project name** | Blog project -Requirements specification | | | | |
| **project type** |  | | | | |

**Blog project -Requirements specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ed wrote：** | Kun liu | **date** | 2021-11-10 |
| **Check：** |  | **date：** |  |
| **Audit：** |  | **date：** |  |
| **Approved：** |  | **date：** |  |

**change description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **date** | **version** | **change** | **Change Description** | **author** |
| 2021-11-10 | V1.0.1 |  | initial | Kun liu |
| 2021-11-22 | V1.0.2 |  | fixed problems from requirement review | Kun liu |

[Blog project -Requirements specification 1](#_Toc89206483)

[1. Preface 1](#_Toc89206484)

[1.1. Project Overview 1](#_Toc89206485)

[1.2. Project Objectives 1](#_Toc89206486)

[1.3. Member role 1](#_Toc89206487)

[2. Demand summary 1](#_Toc89206488)

[2.1. Overall function description 1](#_Toc89206489)

[2.2. Functional structure picture 1](#_Toc89206490)

[2.2.1. Functional structure diagram of blog project 1](#_Toc89206491)

[3. BLOG\_001 User Management 2](#_Toc89206492)

[3.1. BLOG\_001\_01 Registration 2](#_Toc89206493)

[3.1.1. Service scenario 2](#_Toc89206494)

[3.1.2. Execution conditions 2](#_Toc89206495)

[3.1.3. Flow Description 2](#_Toc89206496)

[3.2. BLOG\_001\_02 Account password login 3](#_Toc89206497)

[3.2.1. Service scenario 3](#_Toc89206498)

[3.2.2. Execution conditions 3](#_Toc89206499)

[3.2.3. Flow Description 3](#_Toc89206500)

[3.3. BLOG\_001\_04 modify password 3](#_Toc89206501)

[3.3.1. Service scenario 3](#_Toc89206502)

[3.3.2. Execution conditions 4](#_Toc89206503)

[3.3.3. Flow Description 4](#_Toc89206504)

[3.4. BLOG\_001\_05 Log out 4](#_Toc89206505)

[3.4.1. Service scenario 4](#_Toc89206506)

[3.4.2. Execution conditions 5](#_Toc89206507)

[3.4.3. Flow Description 5](#_Toc89206508)

[4. BLOG\_002 article classification 5](#_Toc89206509)

[4.1. BLOG\_002\_01 Classification management list 5](#_Toc89206510)

[4.1.1. Service scenario 5](#_Toc89206511)

[4.1.2. Execution conditions 5](#_Toc89206512)

[4.1.3. Flow Description 5](#_Toc89206513)

[4.2. BLOG\_002\_02 new category 6](#_Toc89206514)

[4.2.1. Service scenario 6](#_Toc89206515)

[4.2.2. Execution conditions 6](#_Toc89206516)

[4.2.3. Flow Description 6](#_Toc89206517)

[4.3. BLG\_002\_03 edit category 6](#_Toc89206518)

[4.3.1. Service scenario 6](#_Toc89206519)

[4.3.2. Execution conditions 7](#_Toc89206520)

[4.3.3. Flow Description 7](#_Toc89206521)

[4.4. BLOG\_002\_03 delete category 7](#_Toc89206522)

[4.4.1. Service scenario 7](#_Toc89206523)

[4.4.2. Execution conditions 7](#_Toc89206524)

[4.4.3. Flow Description 7](#_Toc89206525)

[5. BLOG\_003 article management 8](#_Toc89206526)

[5.1. BLOG\_003\_01 list of articles 8](#_Toc89206527)

[5.1.1. Service scenario 8](#_Toc89206528)

[5.1.2. Execution conditions 8](#_Toc89206529)

[5.1.3. Flow Description 8](#_Toc89206530)

[5.2. BLOG\_003\_02 new article 8](#_Toc89206531)

[5.2.1. Service scenario 8](#_Toc89206532)

[5.2.2. Execution conditions 9](#_Toc89206533)

[5.2.3. Flow Description 9](#_Toc89206534)

[5.3. BLOG\_003\_03 article modification 9](#_Toc89206535)

[5.3.1. Service scenario 9](#_Toc89206536)

[5.3.2. Execution conditions 9](#_Toc89206537)

[5.3.3. Flow Description 9](#_Toc89206538)

[5.4. BLOG\_003\_04 article deleted 10](#_Toc89206539)

[5.4.1. Service scenario 10](#_Toc89206540)

[5.4.2. Execution conditions 10](#_Toc89206541)

[5.4.3. Flow Description 10](#_Toc89206542)

[6. Test considerations 10](#_Toc89206543)

[7. Non-functional requirements 10](#_Toc89206544)

[7.1. Interface requirements 10](#_Toc89206545)

[7.2. time requirement 10](#_Toc89206546)

[7.3. Performance requirements 11](#_Toc89206547)

[7.4. WebPortal browser compatible 11](#_Toc89206548)

[7.5. WebPortal needs to support resolution 11](#_Toc89206549)

# Preface

## Project Overview

IIT-536 final project

## Project Objectives

## Member role

|  |  |  |
| --- | --- | --- |
| Serial number | name | role |
| 1 | Yingchuan Luo | Unit Test |
| 2 | Kun Liu | Business Analyst |
| 3 | Zhipin Liu | System Test |
| 4 | yanling Jiang | Accept Test |
| 5 | yiling Li | Test Manager |

# Demand summary

## Overall function description

## Functional structure picture

### Functional structure diagram of blog project

|  |  |
| --- | --- |
| Level 1 | Level 2 |
| User Management | register |
| Account password login |
| Change Password |
| Sign out |
| Article Category | Category List |
| New Category |
| Edit Category |
| Delete Category |
| Article Management | Article added |
| Article delete |
| Article revision |
| Article list |

# BLOG\_001 User Management

## BLOG\_001\_01 Registration

### Service scenario

The user opens the homepage of the blog system and registers as a new customer of the blog system.

### Execution conditions

Network communication is normal.

### Flow Description

1. The user enters the blog address to access the login page and clicks "register" to jump to the registration page.
2. The username, email, password, and confirmation password must be entered on the page. The verification rules are as follows.
3. Username: The default is empty, required, and the maximum length is 30 characters, which is a combination of numbers and letters.
4. Email: The default is empty, required, within 100 characters in length, need to verify the email format [xxx@xxx.xx](mailto:xxx@xxx.xx)
5. Password: The password length is between 6-18 characters, case sensitive
6. Confirm password: must be consistent with the entered password
7. Click register to verify as follows:
   1. If the email format is wrong, the prompt: "The email is wrong”.
   2. If the username already exists, it will prompt "Username has been used”.
   3. If the mailbox already exists, it prompts "The mailbox has been registered”.
   4. After completing the above four items, click the register button to verify whether the username, email, and password meet the verification rules. If they do not, it will prompt: "Username or password is wrong, please try again”.
   5. if the entries are correct, then Jump to the blog homepage

## BLOG\_001\_02 Account password login

### Service scenario

The user opens the blog homepage and logs in to the blog system through the account password.

### Execution conditions

1. Network communication is normal.
2. The user has successfully registered in the blog system

### Flow Description

1. The user enters the blog system address to access the login page.
2. Login with email+ password by default.
3. Enter email and password, all items need to be verified, the rules are as follows:

a) email: The default is empty, required, up to 100 digits; if it is empty, it will prompt "Please enter email ".

b) Password: The default is blank, required, and the length is 8-16 digits; if it is blank, it will prompt "Please enter the password"; the default password is \*\* display.

4.Click "Login" to verify whether the email and password are completely correct. If they are correct, log in and transfer to the blog management page. If it fails, it will prompt: " Login failed, please confirm whether the email or password is correct "

## BLOG\_001\_04 modify password

### Service scenario

The user opens the homepage of the blog system, logs in to the blog system through the login page, and modifies the personal password.

### Execution conditions

1. Network communication is normal.
2. The user successfully logs in to the blog system.

### Flow Description

1. The user successfully logs in to the blog system through email and password.

2. Click on the avatar in the upper right corner and click "Modify Password" to modify the personal password.

3. Enter the old password, the new password, and confirm the new password. All items need to be verified. The rules are as follows:

a) old password: password length is 8-16 digits, required; if it is empty, prompt: "please enter old password”.

b) new password: The default is empty, 8-16 digits, required, and it needs to be a mixture of numbers, letters (uppercase or lowercase), and special characters. Special characters are currently only supported: @#$%^&\*-+= ; When the password is empty, the prompt: "The new password cannot be empty"; when the user enters the password in the wrong format, the prompt: "The password length is 8-16 digits, and it must be a combination of numbers, letters and characters";

c) Confirm new password: the default is empty, 8-16 digits, required, it needs to be a mixture of numbers, letters (uppercase or lowercase), and special characters. Special characters are currently only supported: !@#$%^&\*-+ = ; Is empty prompt: "Confirm that the new password cannot be empty"; when the user enters the password format is incorrect, prompt: "password length is 8-16 digits, it must be a combination of numbers, letters and characters"; if it is inconsistent with the new password, prompt : "The two password entries are inconsistent";

d) After filling in the above items, click "OK" to check whether the old password is entered correctly. If it is wrong, it will prompt: "The old password is incorrectly entered", otherwise update the current user password. After the user password is modified, you need to log out and log in again; click " "Cancel" to give up the modification.

## BLOG\_001\_05 Log out

### Service scenario

The user opens the blog system, logs in to the blog system through the login page, and then logs out.

### Execution conditions

1.     Network communication is normal.

2.     The user successfully logs in to the blog system.

### Flow Description

1. The user successfully logs in to the blog system through the account and password.
2. Click on the avatar in the upper right corner, and click "Logout" to log out.
3. The user clicks "Logout", the user successfully logs out and returns to the login page.

# BLOG\_002 article category

## BLOG\_002\_01 category management list

### Service scenario

The account logged in to the blog system can view the blog classification in the classification management and maintain the classification in the database.

### Execution conditions

1. The user successfully logged in to the blog system.

2. Network communication is normal.

### Flow Description

1. The user enters the category management page.

2. Display fields include: category name, category description, creation time.

3. The account can be displayed in pages according to the category name (fuzzy matching) query category list;

## BLOG\_002\_02 new category

### Service scenario

The account logged into the blog system can view the blog classification in the classification management and maintain the classification in the database

### **Execution conditions**

1. The user successfully logged in to the blog system.

2. Network communication is normal.

### Flow Description

1. The user logs into the blog system to enter the category management page.

2. The newly added classification input items are as follows:

Category Name:

* 1. The default is blank, required, and the maximum length is 50 characters. It is necessary to verify whether it contains characters other than English.
  2. When it is empty, the prompt below the field: "The category name cannot be empty”.
  3. If it is not empty, further check whether the characters meet the requirements. If the characters do not meet the requirements, a prompt "Classification name can only be composed of English" is displayed below the field.
  4. Category description: optional, up to 200 characters.
  5. save:
     1. Click Save to verify the required items on the page. When the required items are not filled in, the prompt "xxx cannot be empty" is displayed at the bottom of the field.
     2. Check whether the category name is duplicated, and judge whether the input category name is the same, if it prompts "the category already exists”.
  6. If the verification is passed, the new addition is successful, close the pop-up window and refresh the query result list area according to the previous query conditions.

## BLG\_002\_03 edit category

### Service scenario

The account logged into the blog system can view the blog classification in the classification management and maintain the classification in the database

### Execution conditions

1. The user successfully logged in to the blog system.
2. Network communication is normal.

### Flow Description

1. The user logs into the blog system to enter the category management page.

2. Edit category.

3. Click the "Edit" button in an operation row, and the page will display the classified content.

a)      The category name and category profile show the corresponding attribute value of the selected row category, and the constraints of each field are the same as newly added.

b)      save:

Click Save to verify the required items on the page. When the required items are not filled in, the prompt "xxx cannot be empty" is displayed at the bottom of the field.

Check whether the category name is duplicated, and judge whether the input category name is the same as other category names, if it prompts: "A category already exists [category name]”.

If the verification is passed, the editing is successful, close the pop-up window and refresh the query result list area according to the previous query conditions.

## BLOG\_002\_03 delete category

### Service scenario

The account logged into the blog system can view the blog classification in the classification management and maintain the classification in the database

### Execution conditions

1. The user successfully logged in to the blog system.

2. Network communication is normal.

### Flow Description

1. The user logs into the blog system to enter the category management page.

2. Click delete, the dialog prompts: "Are you sure to delete the category [category name]?”.

3. Confirm, and then check whether there are any articles in the subject under this subcategory. If there is a prompt: "An article already exists in this category, are you sure to delete it?", the user will delete the category and articles under the category after the second confirmation, close the pop-up window and refresh Page, the user will abandon the operation if canceled.

4. Otherwise, the deletion is successful, close the pop-up window and refresh the page.

5. Cancel, then abandon this operation.

# BLOG\_003 article management

## BLOG\_003\_01 list of articles

### Service scenario

The user opens the blog system to view the list of articles.

### Execution conditions

1. Network communication is normal.

2. Successfully logged in to the blog system

### Flow Description

1. The user logs into the blog system to enter the article management list page.

2. The list of all articles on the display platform is arranged in reverse order by default by creation time.

3. All articles of the current user can be queried based on classification, title (fuzzy matching), and creation time

4. Display fields include: article title, article introduction, creation time (format: YYYY-MM-DD HH:mm:ss)

## BLOG\_003\_02 new article

### Service scenario

The user opens the blog system and adds a new article.

### Execution conditions

1. Network communication is normal.

2. User successfully logged in

### Flow Description

1.The user logs in to the blog system to enter the article management list page

2.  Click the "Create Article" button, and the Create Article page appears. The page display items are as follows:

Category: Please select by default, required, list the categories in the category management, and single-select them according to their category names.

Article title: The default is empty, the maximum length is 50 characters, and it is required; when it is empty, the prompt "Article title cannot be empty" will be displayed below the field.

File introduction: The default is empty, the longest is 100 words, and it is optional.

Article content: required, after selecting a category, you need to fill in the content of the article according to different categories

3.  Click the "Save" button to save the article and jump to the article list

## BLOG\_003\_03 article modification

### Service scenario

The user opens the blog system and modifies the article.

### Execution conditions

1. Network communication is normal.

2. User successfully logged in

### Flow Description

The user logs into the blog system to enter the article management list page.

Classification is not editable, and other constraints are the same as creating articles.

【save】:

Click the "Save" button to close all pop-up windows and refresh the query result list area according to the previous query conditions.

Editing the article does not change its status.

## BLOG\_003\_04 article deleted

### Service scenario

The user opens the blog system and deletes the article

### Execution conditions

1. Network communication is normal.

2. User successfully logged in

### Flow Description

The user logs into the blog system to enter the article management list page.

Click "Delete", a pop-up window prompts: "It cannot be restored after deletion. Are you sure you want to delete this article?”.

After the second confirmation, delete the article

# Test considerations

1 .  Full-process testing based on business scenarios

# Non-functional requirements

## Interface requirements

Entry check to prevent SQL injection.

## time requirement

1.From the user's point of view, the response given by the PC within 3 seconds is considered "very attractive" by the user, and the response given within 5 seconds is considered "relatively good"; the response given by the APP within 2 seconds is considered by the user It is considered "very attractive", and a response within 3 seconds is considered "relatively good". Therefore, the present internet terminal PC in response to the user needs ≤ 5 seconds, APP for an end user response ≤ 3 seconds.

2.When the user does some operations with a long processing time, a prompt message can be given to remind the user. When the amount of returned data is too large and the response time is too long, it can provide partial responses, such as paging data extraction, etc., to reduce the waiting time of the operator.

## Performance requirements

1. The entire platform can work continuously for 7×24 hours. The available time in every 1000 hours of system operation is at least not less than 999 hours, and the time between failures should be greater than 1000 hours.
2. The platform must ensure the consistency, completeness, and accuracy of the data to reach 99.99%.
3. Check the legality of manually entered data and data from different interfaces to ensure the validity and standardization of the data and can perform error prompting and error correction processing on the wrong data.
4. There must be a data storage backup to ensure a smooth transition of the application platform when the platform is replaced.
5. The main performance indicators of this system are as follows:

* Stability index
* The platform guarantees 7×24 hours of uninterrupted operation.
* Information Service Response Index
* Platform query response time ≤ 3 seconds.
* Platform statistics response time ≤ 5 seconds.
* The platform can support 3000 concurrent users.

## WebPortal browser compatible

|  |  |  |
| --- | --- | --- |
| **NO.** | **browser name /name** | **remark** |
| 1 | Google Chrome、Edge、safari | blog system |

## WebPortal needs to support resolution

|  |  |  |
| --- | --- | --- |
| **NO.** | **resolution** | **remark** |
| 1 | 1024\*768 or more | blog system |