Software Requirements Specification

Web Publishing System

Sorn

Contents

1.0. Introduction

1.1. Purpose

The purpose of the document is to imitate the operation mechanism of blog.It will explain the purpose and features of the system,the interface of the system.what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for the authors and the readers.they can publish their articles to the system.they can public their opinions to others' articles ,too.

1.2. Scope of Project

This software system will be a Web Publishing System for people of whatever they are interested in.This system will be designed for the people who want to public or read the articles in everything they like.This system is designed to allow an author to publish articles to a public website.

More specifically, The software will facilitate communication between authors and readers via platform. The readers can comment on the articles which have been published. If they are lucky enough ,they might be answered by the author they like.

1.3. Glossary

Term	Definition
Article	The document that is tracked by the system; it is a narrative that is planned to be posted to

	the public website.
User	A person who is the primary user of the application, the registered user of the application, can view, publish articles, publish comments and modify personal information
Database	Collection of all the information monitored by this system.
Session	Divide articles into eight sessions based on the content of the article
Comment	User comments on the published article.
Register	A person can register an account to become a user and the information of new user will add to Database.
Review	A written recommendation about the appropriateness of an article for publication; may include suggestions for improvement.
Manager	A person that examines an article and has the ability to recommend approval of the article for publication or to request that changes be made in the article. Manager is an advanced user with review rights.
Profile	A user center where users can view and modify personal basic information, and can view the number of followers and the number of people being followed
Software	A document that completely describes all of

Requirements Specification	the functions of a proposed system and the constraints under which it must operate. For example, this document.

1.4. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2. Overall Description

2.1 System Environment

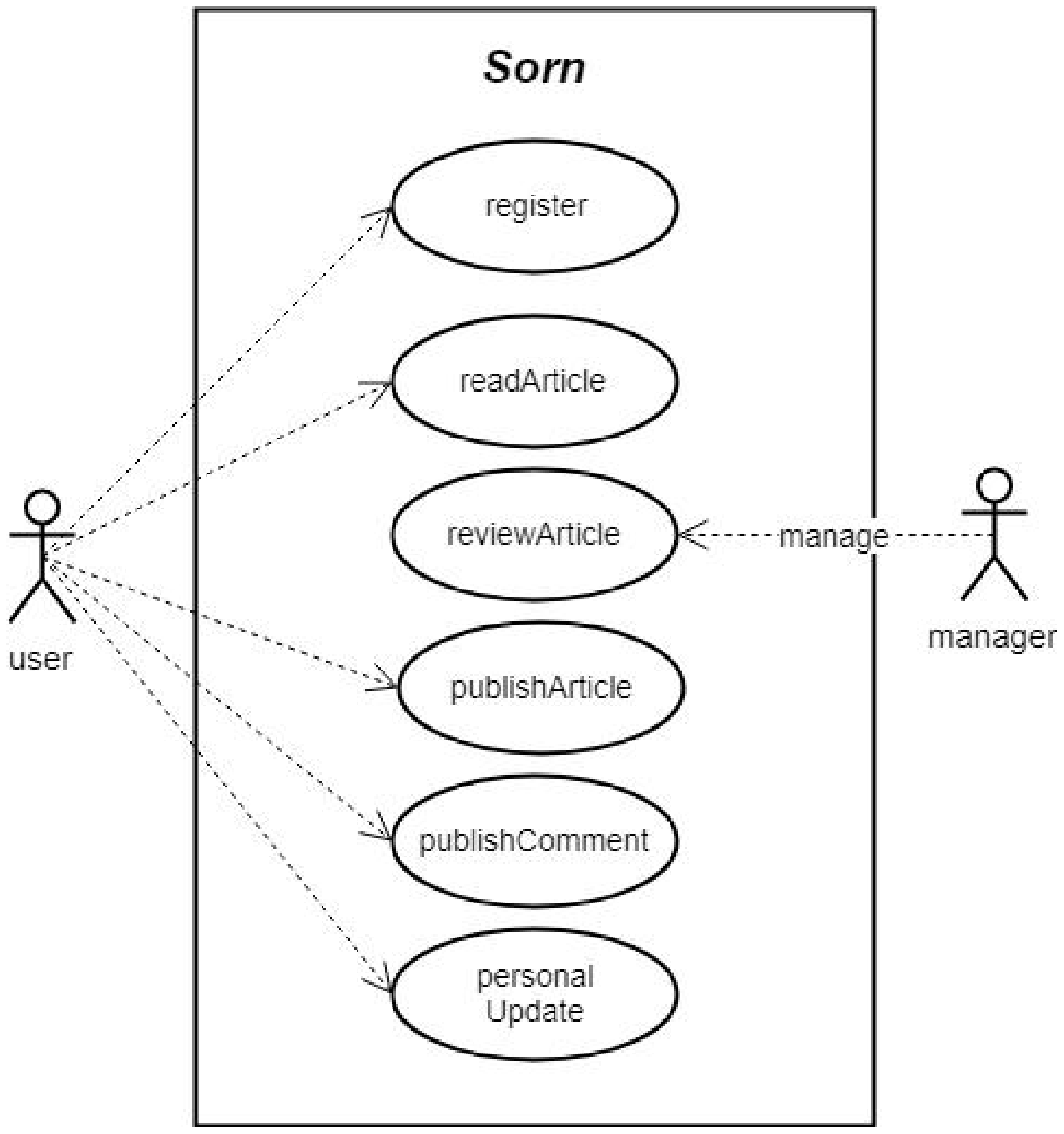
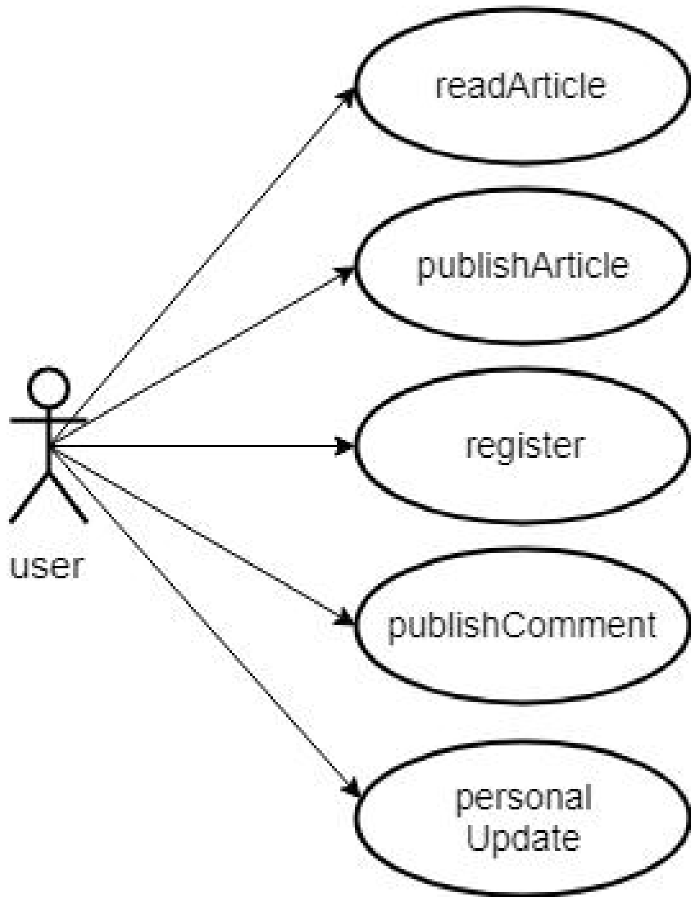


Figure 1 — System Environment

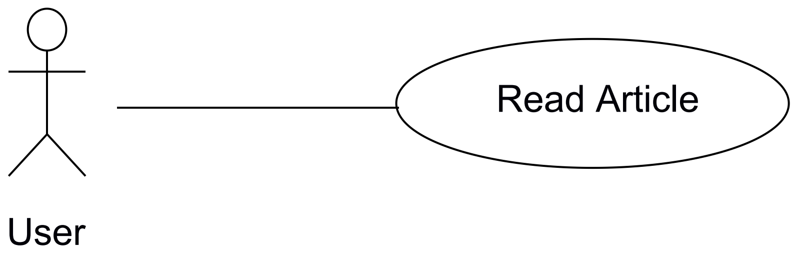
2.2 Functional Requirements Specification

2.2.1 User Use Case



Use case: Read Article

Diagram:



Brief Description：

User selects the desired session and read articles they want.

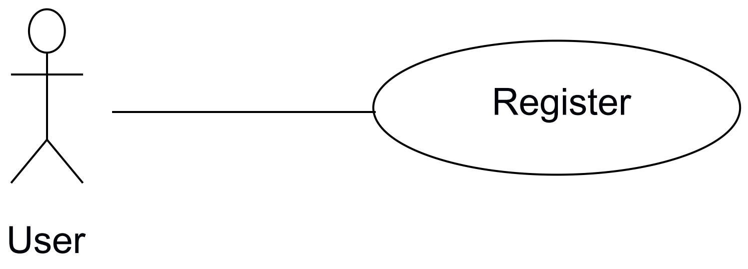
Initial Step-By-Step Description：

Before this use case can be initiated, the User has already accessed the network and login this application.

1. The User chooses how to search the Web site. The choices are by Author, by Category, and by Keyword.
2. If you choose to browse by article session, the system will display all article sessions.
3. The User finds the article classification that he is interested in.
4. The system displays all the articles in the session.
5. The User selects an article.
6. The system displays the Abstract for the article.
7. The User selects to return to the article list or to the previous list.

Use case: Register

Diagram:



Brief Description:

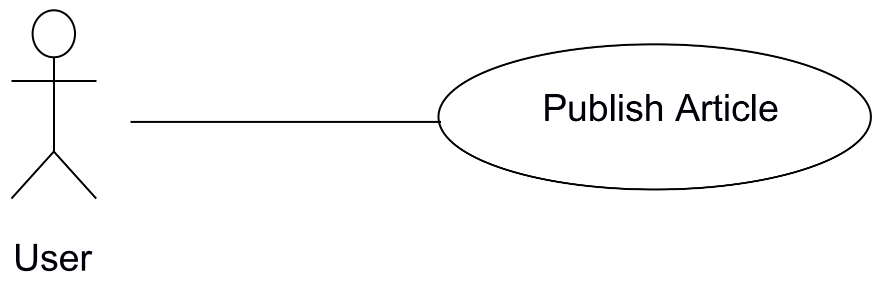
User accessed the network and click the sign up, then enter their Username, Password and Nickname, and then click the “SUBMIT” button to complete the register.

Initial Step-By-Step Description:

1. The user has accessed the register screen.
2. The system accesses the users database.
3. The user enter UserName, NickName and Password.
4. The user determine UserName, NickName and Password
5. The user click “confirm register”
6. The system checks that required fields are not blank.
7. The system updates users database
8. The new user information is entered into the users database.

Use case: Publish Article

Diagram:



Brief Description:

User select a session to publish article with picture(s).

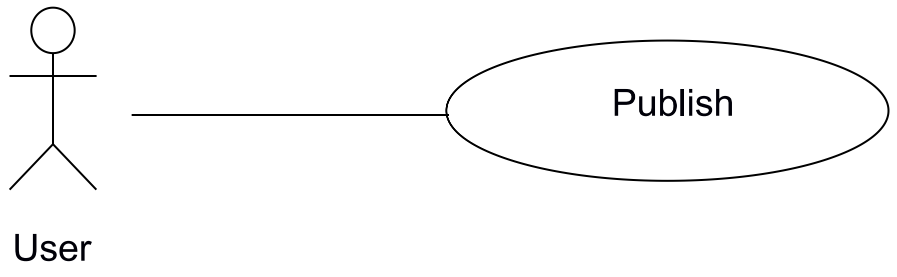
Initial Step-By-Step Description:

1. The system creates and presents an alphabetical list of the active articles that are flagged as having their copyright form returned.

2. The User click the “write” button to enter the writing page.
3. The User select the session to publish article.
4. The User add the title, content and picture(s) of the article.
5. The User click the hook icon to submit the article.
6. The system accesses the Online Database and transfers the article and its accompanying information to the Online database.
7. The article is moved to the active article database.

Use case: Publish Comment

Diagram:



Brief Description:

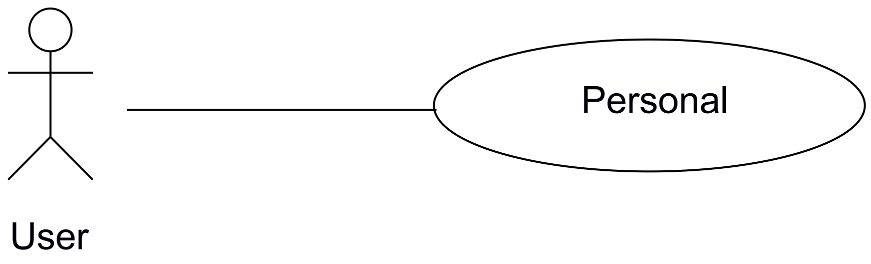
User can publish their comments in any articles.

Initial Step-By-Step Description:

1. The system accesses the comment database and file database.
2. The user is in the article view page.
3. The user click the “publish comment”.
4. The user input comment and can insert a picture.
5. The user click the “confirm publish”
6. The system updates the comment database and file database.
7. The new comment will be added to this article and both users and readers can see it.
8. The publishing comment is completed.

Use case: Personal Update

Diagram:



Brief Description:

User update their personal basic information and avatar in the profile.

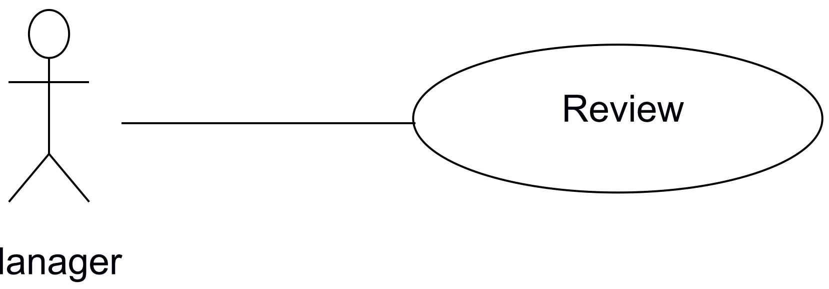
Initial Step-By-Step Description

1. The user enter to the profile.
2. The user can view personal information including NickName, UserName, Password and etc.
3. The user selects information which he wants to modify.
4. The user updates the information and submits the modification.
5. The system checks that required fields are not blank.
6. The system updates the information of user.
7. The personal update is completed.

2.2.2 Manager Use Case

Use case: Review Article

Diagram:



Brief Description:

Manager review the article decides if the article can pass the review

Initial Step-By-Step Description:

1. After reading the article, Manager decide whether it can be published.
2. If the article can be published, the Manager chooses to pass the review.
3. The article was published successfully, and the reader can see the article in the interface.

2.3 User Characteristics

Managers should have basic Internet knowledge and a certain literary literacy to remove low-quality articles.

Users should have Internet knowledge and be able to use search engines. Users should also have basic literacy and be able to review articles in the comment area. In addition, users need to be able to use the button that publishes the article and learn how to upload an image for the article.

The detailed look of these pages is discussed in section 3.2 below.

2.4 Non-Functional Requirements

Users can connect to applications on different platforms: Android, ios and pc. The speed of the Users connection depends on the quality of the network and not the characteristics of the application.

The article manager will run on the manager’s PC and will contain an article database. MySQL is already installed on this computer and is a Windows/Mac operating system.

3. Requirements Specification

3.1 External Interface Requirements

An external link is the database management system to confirm whether the user has permission to review articles. The user (manager) can review the article only after confirming that he has the permission.

The database management system must run correctly and process data from the server in a timely manner: including the publish of new article, the publish of new comments, and changes to user information.

The Publish Article use case sends the picture file and the article content to the Database. The Publish Comment use case sends the comment to the Database.

3.2 Functional Requirements

The Logical Structure of the Data is contained in Section 3.3.1.

3.2.1 Read Article

Use Case Name	Read Article
Trigger	The Reader(include User) selects to transfer an approved article to the Online Journal.
Precondition	Readers enter the article interface.
Basic Path	1. The User chooses how to search the Web site. The choices are by Author, by Category, and by Keyword.

	2. If you choose to browse by article session, the system will display all article sessions. 3. The User finds the article classification that he is interested in. 4. The system displays all the articles in the session. 5. The User selects an article. 6. The system displays the Abstract for the article. 7. The User selects to return to the article list or to the previous list.
Alternative Paths	In step 2, if the Reader chooses to browse by push article, the system will display the current popular article. Return to step 5.
Postcondition	The selected article is displayed to the user interface.
Exception Paths	The Reader may abandon the read at any time.

3.2.2 Publish Article

Use Case Name	Publish Article
Trigger	The User selects to transfer an approved article to the Online Journal.
Precondition	The User has accessed the Article Manager main screen.
Basic Path	1. The system creates and presents an

	alphabetical list of the active articles that are flagged as having their copyright form returned. 2. The User click the “write” button to enter the writing page. 3. The User select the session to publish article. 4. The User add the title, content and picture(s) of the article. 5. The User click the hook icon to submit the article. 6. The system accesses the Online Database and transfers the article and its accompanying information to the Online database. 7. The article is moved to the active article database.
Alternative Paths	None.
Postcondition	The article is properly transferred.
Exception Paths	The User may abandon the operation at any time.

3.2.3 Review Article

Use Case Name	Review Article
Trigger	After the User edits the article for submission.
Precondition	The Editor has accessed the Article Manager main screen and the article is already in the database.

Basic Path	1. After reading the article, Manager decide whether it can be published. 2. If the article can be published, the Manager chooses to pass the review. 3. The article was published successfully, and the reader can see the article in the interface.
Alternative Paths	In step 2, if the article can't be published, User will be notified that the user has not passed the review, and the user can modify the article and submit it again.
Postcondition	Article successfully published or published failed.
Exception Paths	The Manager may abandon the operation at any time.

3.2.4 Register

Use Case Name	Register
Trigger	The Editor selects to add a new reviewer to the database.
Precondition	The reader has accessed the application register screen.
Basic Path	1. The user has accessed the register screen. 2. The system accesses the users database. 3. The user enter UserName, NickName and Password. 4. The user determine UserName, NickName and

	Password 5. The user click “confirm register” 6. The system checks that required fields are not blank. 7. The system updates users database 8. The new user information is entered into the users database.
Alternative Paths	In step 6, if any required field is blank, the reader is instructed to add an entry. No validation for correctness is made.
Postcondition	The new user’s information has been added to the database.
Exception Paths	The Server may disconnect from the network at any time.
Other	The user information includes userName, userID, userPassword,userAvatar and etc.

3.2.5 Publish Comment

Use Case Name	Publish Comment
Trigger	The user selects to publish comment on an article.
Precondition	The user has accessed the article main screen.
Basic Path	1. The system accesses the comment database and file database. 2. The user is in the article view page. 3. The user click the “publish comment”.

	4. The user input comment and can insert a picture. 5. The user click the “confirm publish” 6. The system updates the comment database and file database. 7. The new comment will be added to this article and both users and readers can see it. 8. The publishing comment is completed.
Alternative Paths	None.
Postcondition	The comment is published on the article .
Exception Paths	The user may disconnect to network and missing the unposted comment.

3.2.6 Update Person

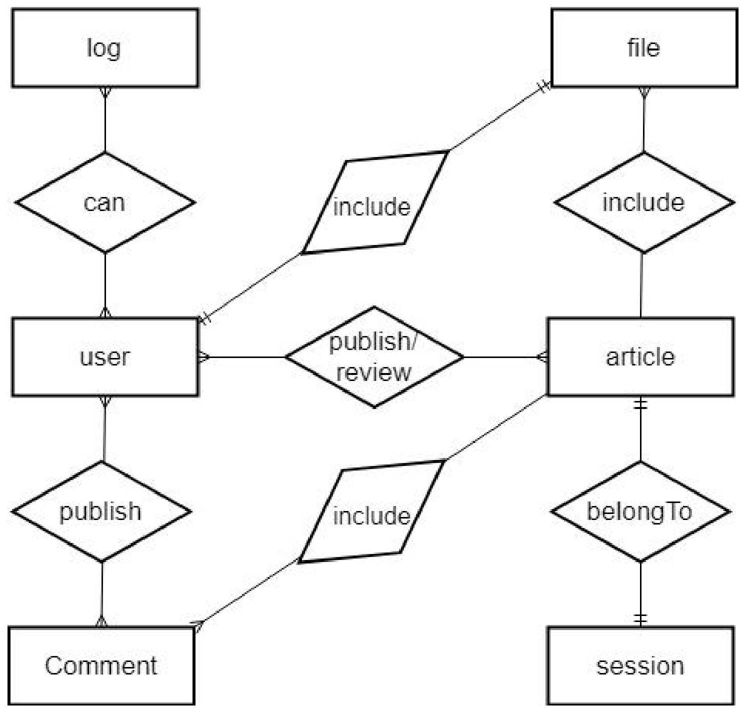
Use Case Name	Personal Update//个人信息修改
Trigger	The user selects to update information and the user is already in the database.
Precondition	The user has login and accessed to the userCenter.
Basic Path	1. The user enter to the profile.//用户进入个人中心 2. The user can view personal information including NickName, UserName, Password and etc.//用户浏览个人信息 3. The user selects information which he wants to modify.

	4. The user updates the information and submits the modification.//用户更新个人信息并提交修改 5. The system checks that required fields are not blank.//系统检查必填项目是否为空 6. The system updates the information of user.//系统更新用户个人信息 7. The personal update is completed.
Alternative Paths	In step 5, if any required field is blank, the user is instructed to add an entry. No validation for correctness is made.//在第五步，如果必填项目为空，要求用户添加字段。没有验证是否正确
Postcondition	The database has been updated. The personal information has been updated.//个人信息已经更新
Exception Paths	If the user is not already in the database, the use case is abandoned. In addition, the Editor may abandon the operation at any time.

3.3 Detailed Non-Functional Requirements

3.3.1 Logical Structure of the Data

The logical structure of the data to be stored in the internal Web Publishing Systerm database is given below.



3.4 Logical Structure of the Web Publishing System Data

The data descriptions of each of these data entities is as follows:

list1: Users

Users				
FieldName	Type	NULL	Remarks	Description
user_id	bigint	no	PK	
userName	varchar(20)	no	unique	
userPwd	varchar(20)	no		
userAvatar	bigint		FK,reference to File(fileID)	default(1)
userDescriptio	varchar(100			

Users				
n	)			
userRegDate	datetime	no		
userIsManager	int(bool)	no		
userNickname	varchar(20)	no		
userAttention	int		default 0	
userFans	int		default 0	

list2: File

File				
FieldName	Type	NULL	Remarks	Description
file_id	bigint	no	PK	
fileUrl	varchar(100)	no		
type	enum	no	enum("video","pic")	the type of file
fileName	varchar(100)	no		
groupName	varchar(100)	no		

list3: Article

Article				
FieldName	Type	NULL	Remarks	Description
articleId	bigint	no	PK	
articleSessionId	bigint	no	FK,reference to Session(session_id)	the ID of sessions,foreign key
articleUserId	bigint	no	FK,reference to Users(user_id)	the ID of user,foreign key
articleTitle	varchar(50)	no		
articleContents	text	no		
articleTime	datetime	no		
articleClickCount	int			
articleIsConsentient	int	no		
articleLastCommentTime	datetime			
articleLastContentsHTML	longtext	no		

list4: Comments

Comments				
FieldName	Type	NULL	Remarks	Description
commentID	bigint	no	PK	
commentAID	bigint	no	FK,reference to Article(articleID)	article which is including the comment
commentUID	bigint	no	FK,reference to Users(usersID)	user who sent the comment
commentContent	text	no		
commentTime	datetime			

list5: Sessions

Sessions				
FieldName	Type	NULL	Remarks	Description
session_id	bigint	no	PK	
sessionName	varchar(20)	no	unique	
session_img_i	bigint	no		

Sessions				
d				

list6: Log

Log				
FieldName	Type	NULL	Remarks	Description
log_id	bigint	no	PK	
user_id	bigint	no	FK,reference to Users(user_id)	
logTime	datetime	no		
logLastIPAddress	varchar(20)	no		

list7: Following

Following				
FieldName	Type	NULL	Remarks	Description
user_id	bigint	no	PK,FK,reference to Users(user_id)	
following_user_id	bigint	no	PK,FK,reference to Users(user_id)	

3.5 Create list:

```
CREATE TABLE `Article` (
`articleID` bigint(20) NOT NULL AUTO_INCREMENT COMMENT
'article_ID',
`articleSID` bigint(20) NOT NULL COMMENT 'article_ ID of sessions',
`articleUID` bigint(20) NOT NULL COMMENT 'article_ ID of users',
`articleTitle` varchar(50) NOT NULL COMMENT 'artcle_Title',
`articleContents` text NOT NULL COMMENT 'article_Contents',
`articleTime` datetime NOT NULL COMMENT 'article_Publish Time',
`articleClickCount` int(11) DEFAULT NULL COMMENT
'article_Click_Count',
`articleConsentient` int(11) NOT NULL COMMENT
'article_publish_Is_Allowed',
`articleLastCommentTime` datetime DEFAULT NULL COMMENT
'article_Lastest_Comment_Time',
PRIMARY KEY (`articleID`),
KEY `articleSID` (`articleSID`),
CONSTRAINT `article_ibfk_1` FOREIGN KEY (`articleSID`)
REFERENCES `Sessions` (`session_id`),
CONSTRAINT `article_ibfk_2` FOREIGN KEY (`articleID`)
REFERENCES `Users` (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `Comments` (
`commentID` bigint(20) NOT NULL AUTO_INCREMENT COMMENT
'comment_ID',
`commentAID` bigint(20) NOT NULL COMMENT 'comment_ ID of
article include the comment',
`commentUID` bigint(20) NOT NULL COMMENT 'comment_ ID of
user sent it',
`commentContent` text NOT NULL COMMENT 'comment_Content',
`commentTIme` datetime DEFAULT NULL COMMENT
'comment_Publish_Time',
PRIMARY KEY (`commentID`),
KEY `commentAID` (`commentAID`),
KEY `commentUID` (`commentUID`),
CONSTRAINT `comments_ibfk_1` FOREIGN KEY (`commentAID`)
REFERENCES `Article` (`articleID`),
CONSTRAINT `comments_ibfk_2` FOREIGN KEY (`commentUID`)
REFERENCES `Users` (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `FIle` (
`file_id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT
'file_ID',
`fileUrl` varchar(100) NOT NULL COMMENT 'file_Url',
`type` enum('pic','viedo') DEFAULT NULL,
`fileName` varchar(100) NOT NULL COMMENT 'file_Name',
`groupName` varchar(100) NOT NULL COMMENT 'group_Name',
PRIMARY KEY (`file_id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT
CHARSET=utf8;

CREATE TABLE `Log` (
`log_id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT
'log_ID',
`user_id` bigint(20) NOT NULL COMMENT 'user_ID',
`logTime` datetime NOT NULL COMMENT 'log_Time',
`logLastIPAdress` varchar(20) NOT NULL COMMENT
'Lastest_IP_Address of user to login',
PRIMARY KEY (`log_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `Sessions` (
`session_id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT
'session_ID',
`sessionName` varchar(20) NOT NULL COMMENT 'session_Name',
PRIMARY KEY (`session_id`),
UNIQUE KEY `sessionName` (`sessionName`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `Users` (
`user_id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT
'user_ID',
`userName` varchar(20) NOT NULL COMMENT 'user_Name',
`userPwd` varchar(50) NOT NULL COMMENT 'user_Password',
`userAvatar` bigint(20) DEFAULT '1' COMMENT 'user_Avatar',
`userDescription` varchar(100) DEFAULT NULL COMMENT
'user_Description',
`userRegDate` datetime NOT NULL COMMENT
'user_Register_Date',
`userIsManager` int(11) NOT NULL COMMENT 'user_IsManager',
`userNickname` varchar(20) NOT NULL COMMENT 'user_nickname',
PRIMARY KEY (`user_id`),
UNIQUE KEY `userName` (`userName`),
KEY `userAvatar` (`userAvatar`),
CONSTRAINT `Users_FIle_file_id_fk` FOREIGN KEY (`userAvatar`)
REFERENCES `FIle` (`file_id`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT
CHARSET=utf8;

trgger:
CREATE TABLE Attention
(
user_id BIGINT COMMENT 'user_id PRIMARY KEY',
attention_user_id BIGINT COMMENT 'attention_user_id PRIMARY
KEY',
CONSTRAINT PRIMARY KEY(user_id,attention_user_id),
CONSTRAINT FOREIGN KEY(user_id) REFERENCES
Users(user_id),
CONSTRAINT FOREIGN KEY(attention_user_id) REFERENCES
Users(user_id)
)
```

```
ALTER TABLE Users ADD COLUMN userAttention int DEFAULT 0;
ALTER TABLE Users ADD COLUMN userFans int DEFAULT 0;

CREATE TRIGGER trigger_addAttention AFTER INSERT ON
Attention FOR EACH ROW
BEGIN
UPDATE Users set userAttention=userAttention+1 WHERE
Users.user_id=(SELECT user_id FROM INSERTED);
UPDATE Users set userFans=userFans+1 WHERE
Users.user_id=(SELECT attention_user_id FROM INSERTED);
END;

CREATE TRIGGER trigger_deleteAttention AFTER DELETE ON
Attention FOR EACH ROW
BEGIN
UPDATE Users set userAttention=userAttention-1 WHERE
Users.user_id=(SELECT user_id FROM INSERTED);
UPDATE Users set userFans=userFans-1 WHERE
Users.user_id=(SELECT attention_user_id FROM INSERTED);
END;
```