**Ashesi University College**

# Online Article Management System

## User- Side (Front-End)

**Joshua Tabu Shehe**

**2010**

**Applied Project**

## Declaration

I hereby declare that this dissertation is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:………………………………………………………………………

Candidate's Name:……………………………………………………

Date:…………………………..

The second half of the page should contain the following statement:

*I hereby declare that the preparation and presentation of the thesis were supervised in accordance with the guidelines on supervision of thesis laid down by Ashesi University College.*

Supervisor's Signature:……………………………………………………………………..

Supervisor's Name:……………………………………………….

Date:…………………………………..

## Acknowledgment

I would like to express my appreciation to my project supervisor, Mr. Kwadwo Gyamfi Osafo-Maafo for assisting me in working on my project. His advice was a great inspiration. I would also like to express my appreciation to Mr. Kevin Harter, the Eldoret Writers staff who kept communicating and updating me with all the significant things that allowed me to make the application that would satisfy their needs.

## Abstract

This is an applied group project done by Joshua Tabu Shehe and Ernest Musyoki Museve. This report looks at the processes by which the web-based article management system called WriteWell is developed and implemented for the user-side (front-end). This application will allow Eldoret Writers Centre writers to manage their own articles in an efficient way than what they are using currently. This report will look at the background information about Eldoret Writers Centre based in Eldoret which will assist in looking at the methodologies, system design, implementation and testing involved. Lastly, we will look at the experiences gained while working on this project.

## Contents

## 1.0.0 Introduction

The emergence of the internet has seen businesses and organizations shifting their activities to the internet. This shift has occurred because the internet is removing barriers allowing people from all over the world to communicate using websites like Facebook and Twitter.

Eldoret Writers Centre is trying to use the same concept of bringing people together in trying to bring writers from different places in the country together and help them publish their articles for half the cost of the article when sold to media houses such as newspapers and magazines.

This project will look at some methods that Eldoret Writers Centre tried to implement and see whether the article management system solves their problem in a more efficient way.

## 1.1.0 Aim

The aim of this project is to develop an article management system for the user-side (font-end) using web-technologies which would allow Eldoret Writers Centre writers to manage their articles efficiently.

## 1.2.0 Background Information

Eldoret Writers Center connects writers to media houses such as newspapers, magazines and even other websites with the aim of publishing the writer's articles to the mentioned media houses. The organization has been doing this for the past 5 years. The main aim of the organization is to bring up new writers or journalists and help them acquire writing skills.

Many media houses are interested in what upcoming writers and journalists have to offer, so the organization has decided to capitalize on this by selling articles to media houses and paying writers a substantial amount of money.

Since many families eventually move away from Eldoret, the organization started losing writers who were not willing to commute to submit articles. This raised serious issues that the organization faced. These are:

- The organization writers are college students, with busy schedules who are only available during the holidays to write articles.
- Increased demand from media houses wanting quality of writing.
- The organization is located in the city, but many college students seem to be attending school outside the city which makes travelling to the city form the college very difficult and time consuming, hence the organization wants to have writers submit articles without coming to their office in the city.

### 1.2.1 The problem

Looking at the background information, the organization is facing major challenges, because there is no easy way for writers living outside the city to submit articles without travelling unless they email articles. One partial solution was to build many branches to enable communication with the writers who lived outside the city but this requires considerable amounts of money.

### 1.2.2 How to solve the problem

The solution designed and implemented in this project to solve the organization’s problem is to implement a web-based application to act as an intermediary between the writers and the organization. This web-application would minimize the need for branches while enabling writers to submit articles. To do this successfully, the web-application must be able to perform features useful for writers and their editors. These features are discussed in the functional requirements.

### 1.2.3 Challenge

The problem is extremely challenging because it means creating a new system to replace the old system which also means it should work perfectly or the organization would lose their writer’s confidence. Also, the difficulty comes in looking at the number of technologies that would be needed to create and implement the system. In addition, since data is sensitive security would be very significant. In order to implement the system, the following should be considered:

- A database application for storing the organization data.
- A web-server to connect to the database and help in hosting the web-application.
- User interfaces that would be used by the writer or visitor.
- A way to link the web-application to a payment system used widely in the country.

### 1.2.4 Choice of project

The choice to do this project was mainly because it incorporates all the website technologies we have been using in the web technology and e-commerce courses such as PHP, MySQL, AJAX, XML and JavaScript.

### 1.3.0 Similar applications

There are several similar applications like the one being proposed to be created and implemented that have been done which illustrate good user-side (front-end) interface design and features that we will review. These are Triond and Xomba. We will look at each.

### 1.3.1 Triond and Xomba

Triond is a web site that collects articles from different writers and publishes them to other media houses or websites. The more articles a writer writes the more money earned. However, visitors to the website cannot access the website unless they are registered. Registered users can write, edit and submit articles. Also, registered users can view their payment details and see what other users have written.

On the other hand, Xomba uses advertisement generated by the article to pay the writers. This means without an Ad sense account, a user cannot become a member. A visitor to the website can view articles written, see writers who are online, write, edit and submit articles. Similarly, a registered user can view his payment details.

### 1.3.2 Difference between these applications and WriteWell

Triond has many nice features that make it stand out, however it does not allow visitors access to the website unless they are registered. WriteWell and Xomba front-end allow visitors limited access to the website where a visitor can view latest articles, article category but not write an article for submission unless registered.

In addition to Triond, registered writers are paid per article they write but WriteWell does not have a payment feature. Similar to Triond, Xomba also pays writers for the articles they write but only if the article generates advertisement. Therefore, the idea was to take some of the best features from Triond and Xomba and come up with a nice WriteWell web-application which would solve the organizations problem.

### 1.4.0 Approaches

There are other web-applications that could solve the organizations problems. Some of the approaches were already in use and therefore the problems were already well documented making evaluation easy. These approaches are discussed shortly.

### 1.4.1 Email approach

This approach allows writers to send their articles using any email account such as Yahoo or Gmail as attachment. After the organization reviews the article, the article is sent back to the writer as a draft via email for corrections for final submission. This approach makes work very difficult for the writer because the writer has to keep on sending the article back and forth for corrections or final submission which is very cumbersome and annoying. This approach was discarded.

### 1.4.2 Building branches in other towns

Building branches in other towns seem to be a very good approach because writers would be able to register with any nearest branch around where they live. The problem materializes when a writer's information from one branch is not synchronized with all the branches all over the country, which means information for the same writer in one branch before moving to another branch would not be available hence the writer would have to register again. This approach was also discarded.

### 1.4.3 Chosen Approach

The two approaches clearly seem to create more problems than solutions. Building more branches throughout the country would solve the problem however the organization would have to spend a lot of money to make this possible. With available affordable internet, it would be easier for the organization get a web-application that would manage their articles. We will discuss why we chose to use a web-based application approach.

**Web-based application approach**

This approach lets writers write articles using the web-application where everything is done on the website. The things that a writer can do are listed below.

- For an article, a writer can
    - Write an article by choosing a topic of interest
    - Edit article contents for errors before submission.
    - Upload a picture to the article or upload an article for submission.
    - Submit an article after writing and editing.
    - View the number of articles written.
    - View article draft, pending articles, declined and approved articles.
    - View latest articles.
    - View the members who are online.
    - Search for articles using a search field box provided.
    - View and edit personal details.

- A writer can write comments that could be useful to the organization.

We can clearly see from the above list of the things a writer can do that this approach is one of the best. However, there are some circumstances that would make accessing the web-application efficient. These circumstances are described below.

- The member (writer) should be able to get access to a computer with a fast internet connection so that they can access the web-application. In addition a fast internet connection would be used because the writers may have pictures to upload or even large articles to submit as attachment.
- Writers should try as much as possible to read onscreen instructions to be able to effectively use all the features that were listed above. For instance, a feature such as searching could be very useful because a writer can search for a particular article he wants to write.
- If there are any problems the writers should as much as possible try to voice their concerns by writing to the administrator using the contact us feature provided in the web-application. This may help the administrator to rectify mistakes or enhance the look of the web-application.

### 1.5.0 Methodology

Let us look at the structures and methods that were used to create the WriteWell web-application for Eldoret Writers Centre.

### 1.5.1 Requirements

In order to find the organization's requirements their daily operation of the organization had to be monitored, but in this case one of the staff provide very useful information that helped in knowing how they manage their articles and carry out operations in their organization. We will look at functional and non-functional requirements.

### 1.5.2 Functional requirements

The functional requirements were derived from the organization, below are the functional requirements that were considered while developing the web-application:

- Writers shall be able to register to become members.
  - Writers shall be able to authenticate themselves by logging on using their username and password.
  - Writers shall be able to write, edit and submit articles.
  - Writers shall be able to view latest articles and RSS feed.
  - Writers shall be able to view their account and edit their details.
  - Writers shall be able to view their article statuses which are draft, pending, declined and approved.
  - Writers shall be able to logout from the web-application.
  - Writers shall be able to keep track on the number of articles they have written.
- Writers shall be able to view other writer (members) in the organization, view part of their data and contact them without the need to use outside sources.
- No writer shall be able to interfere with any other writer’s information stored in the database.
- Writers shall be able to search for articles using keywords, date or article category.

### 1.5.3 Non-functional requirements

The non-functional requirements were also derived from the organization. Below are the requirements.

1. **Security:** since security is essential, writers should be able to log on and off, using their passwords and usernames.

2. **Graphical User Interface:** the user interface should be well designed and maintained to make the writers experience memorable. This would entail the use of HTML and css.

### 1.5.4 Application structure

The diagram below is used to show the data flow in the web- application. The client access the database using a web browser via the web server.

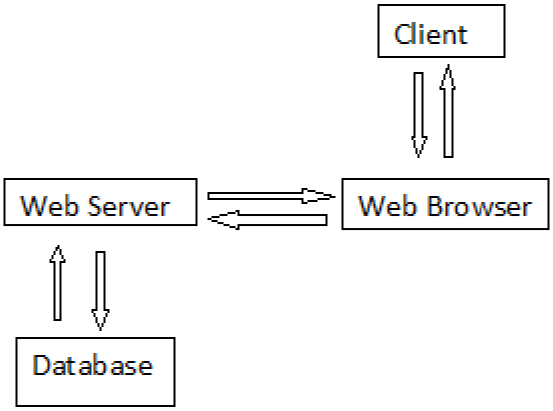


Diagram 1: Diagram showing the architecture of the application

The diagram below shows the structure of the application and the different functions that the application has.

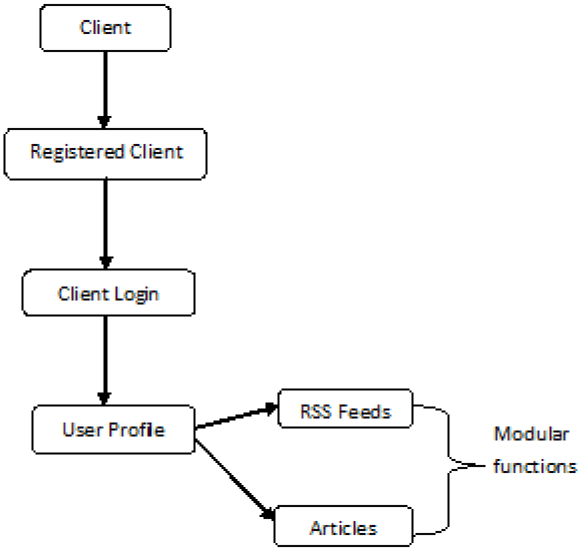


Diagram 2: Diagram showing structure of the web-application

Looking at the diagram, a client registers then logs on to the web-application. Then the user can start to use the modular functions such as RSS feeds to view latest articles.

### 1.6.0 Design

To design the web-application, list of requirements are listed below:

- **Xampp:** this is open source software that has MySQL and apache server which can run PHP scripts. Since MySQL and apache are integrated together, it makes installation of one software rather than installing MySQL and apache separately.
- **Paint:** paint was used for designing the web-application, pictures and many other design related things.
- **Notepad++:** this software was used as an editor to edit the scripts coded. The advantage of this software is that it shows all the languages needed to make a web-application.
- **Visio:** this software was used to design database tables and structures helped in the designing and connect the tables where necessary.
- **JQuiery script:** jQuery is a fast and concise JavaScript library that emphasizes interaction between JavaScript and HTML.
- **Computers:** preferably a computer with specification such as 1GH and memory space of above 100GB.

### 1.6.1 Database design and structure

Before the database was designed some process of normalization was carried out to ensure redundancy was eliminated. Lists of all tables used to design the database are shown below.

## 1. Database tables used

| | |
|---|---|
| Users table | This table is used to store data of the writers of the organization. When a writer logs in this is the table that is queried and also when a new writer signs up his data is stored on this table. |
| Articles table | The articles table in used to store data on all the articles. An article id is what is used to link between an article and a writer of the organization. |
| Categories table | This table contains all the article categories that the articles in the system are divided into. An article category is linked to an article using the category Id which is designated in the article table. |
| Comments table | When a person comments on an article, the data which includes his IP address, email etc. is store in this table. |
| Pages table | This table is used to store data on setting of particular pages. The information on this table will mostly be viewed on the page of which data is stored. |
| Ratings table | Each article that is posted on the websites front page is allowed to be rated. The ratings are stored in this table. |
| Active table | This table is used to store the active writers, active pages active sessions etc. |
| Article-view table | This table information article views, who viewed the article and when. |
| Media table | This table is used store information on the different media stored in the database. |
| Sections table | The web site contains different sections like top, bottom etc. this sections are stored in this table. |
| Options table | This table stores the different options that a writer wants to include in the web site. |

Table1: Table showing all the tables used to store information in the database

### 1.1 User application interaction

A guest (visitor) can view articles, however, to become a member (writer) the guest has to register. A member is then asked for a username and password to log on. A session is started to identify only a particular user and to ensure security. Then, a writer (member) can then write and submit articles after a few validations are conduct to ensure only proper data is submitted as an article. Below is a use case diagram that depicts the actions explained.

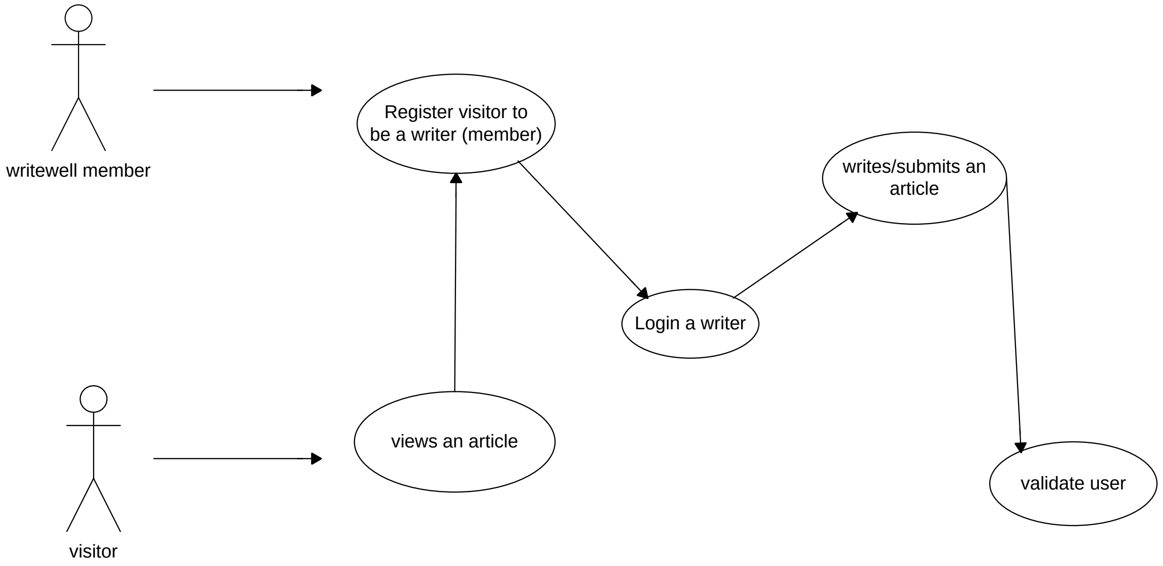


Diagram 3: Use case diagram showing the interaction between users of WriteWell and the System

## 1.2 Relationship between database tables

There are many relationships between the database tables. For example, the articles table is linked to the users' tables. The writer information is stored in the users table while the article's is stored in the articles table. Primary key of one table was used as a secondary key in the table to link them together. This is shown in the database diagram in appendix 1 and also UML diagram in appendix 2.

## 1.3 UML diagram

Appendix 2 shows a UML diagram that shows interaction between users and data in the application. For example, a writer can write more than one article, but an article can only be written by one writer. This brought about multiplicity constraints in between the tables where two tables interacting have a primary key of one table posted on the other table (parent to child) so that the two tables can be linked. This helped to ensure that the tables were linked together so that while coding it would be easier.

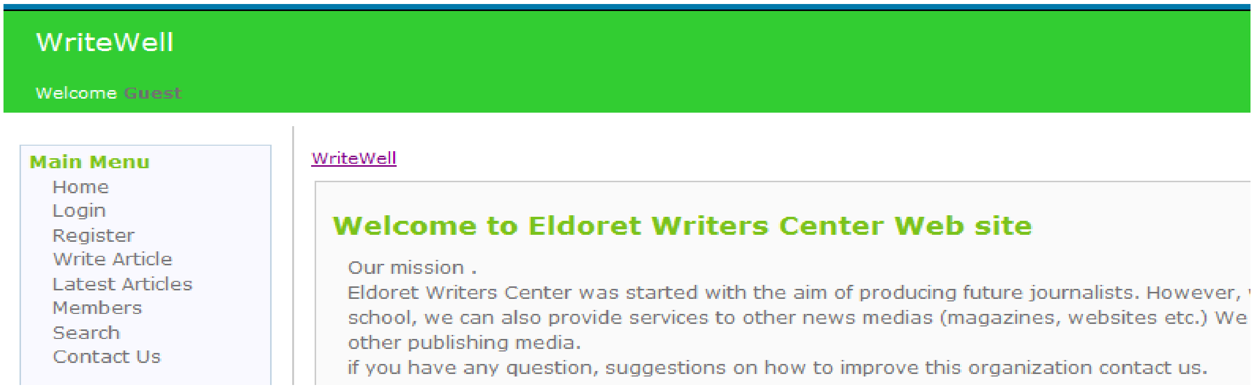
### 1.6.2 Web-application interface design (user-side)

The interface design on the user-side is very important because this is where the writer or visitor interacts with the different features provided. Therefore, it should at least be attractive and very easy to use to allow for a memorable browsing time.

### 1. User-side (Front-end)

The front-end is where visitors or writers of the organization use to access the web-application. Below are a few screen shots that show how the front-end is designed.

### 1.1 Homepage



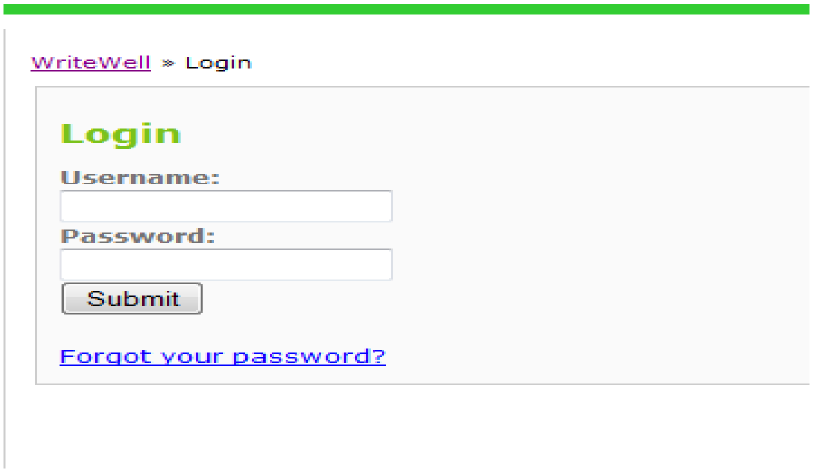
Welcome to Eldoret Writers Center Web site

Our mission .

Eldoret Writers Center was started with the aim of producing future journalists. However, '
school, we can also provide services to other news medias (magazines, websites etc.) We
other publishing media.
if you have any question, suggestions on how to improve this organization contact us.

The above image represents the application home page. Whenever one visits the site this is what will appear first. As you can see there is a main menu on the right hand side which has some of the functional requirements features mentioned earlier. That main menu acts as a link to other pages on the web site. There is also a welcoming message for visitors and registered writers when the web-application is accessed.

## 1.2 Login page


WriteWell » Login

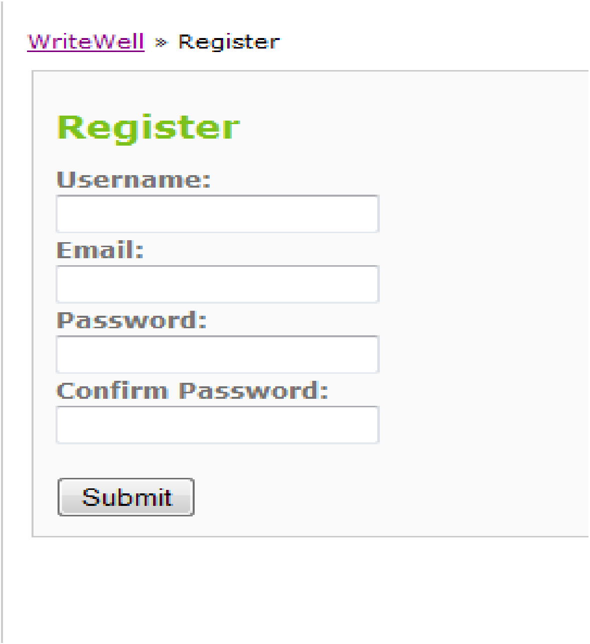
Login

Username:

Password:

This is the interface that will enable a writer of Eldoret Writers Centre to login to the Web site. The forgotten password link placed at the bottom of the screen is very important since it enables the member of the organization to easily move to the page that enables him/her to change the password. There is no better place to put this link than here. Whenever a person enters his/her username and password, a validation check has to take place before he/she is granted access. If a user is not allowed into the system then he/she will not be able to pass this stage.

## 1.3 Registration page



WriteWell » Register

Register

Username:

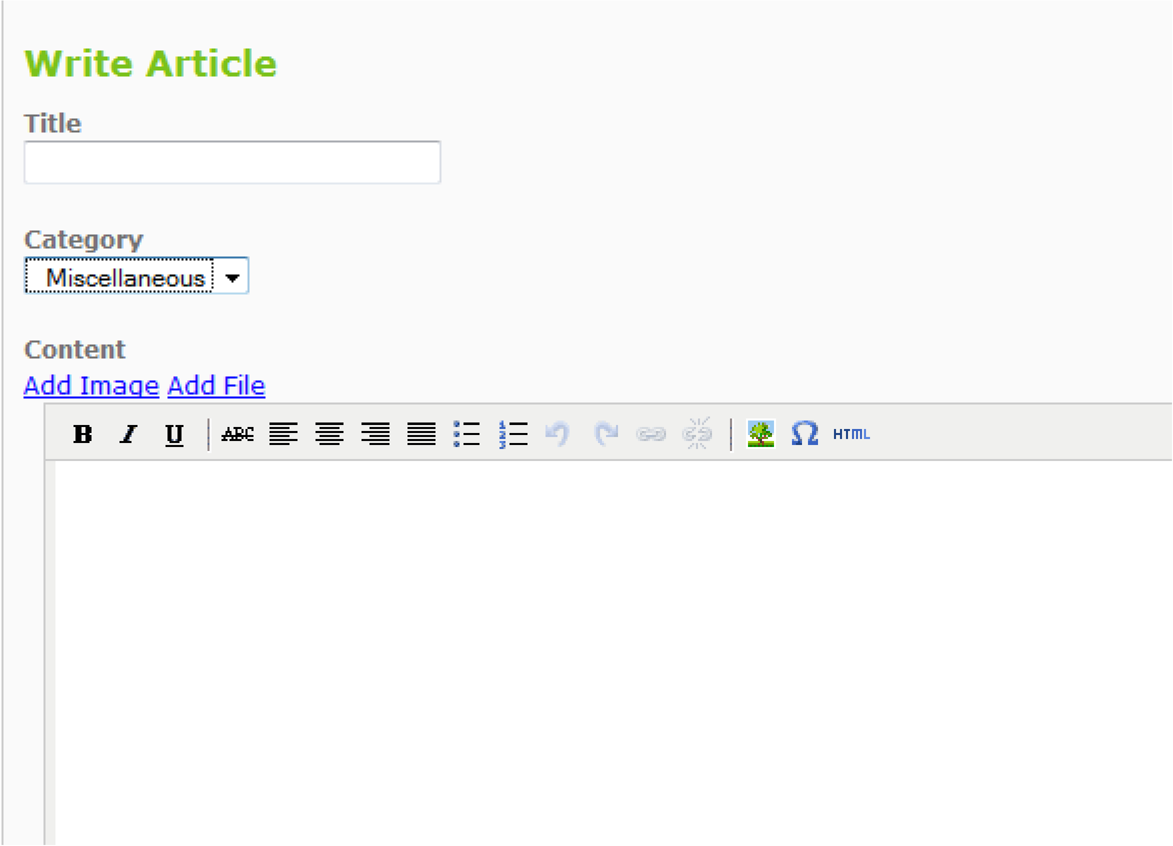
Email:

Password:

Confirm Password:

All unregistered writers will need to fill this form which is submitted to the database before they are granted membership to Eldoret Writers Centre. There is a code created to validate all the information passed through this form to make sure that no invalid data is stored in the database. Invalid data does not include invalid email addresses or wrong name format only. The validation also prevents a person from entering a code that can later destroy the application.

## 1.4 Writing or submitting an article



Write Article

Title

Category

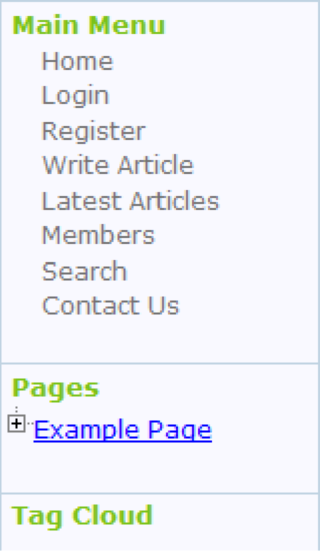
Miscellaneous

Content

When writers want to submit or write an article, there has to be a form of text editor that will enable them to do that. There was an option of using some of the text editors provide by Google or any other web site but the decision was to create a new one. This was easy to integrate with the application. Above is a screen shot of the text editor that appears when one wants to write an article. Notice that this includes the category in which the article will be based on, the option of adding an image or a file and the title of the article. This means that a writer who wishes to do all these things will find all the required features on the same page.

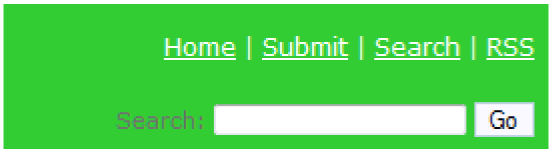
## 1.5 Navigation

A navigation menu is a very important feature in this web site. This is because all the content of a web site cannot be displayed in one page. Moreover, the placement of the navigation is important. In this case the navigation menu was placed on the left hand side. This is because it would be the first thing to be shown when the site is accessed. Below is the navigation menu that was included in the web site. The links in the main menu have been mentioned when looking at functional requirements.



## 1.6 Search

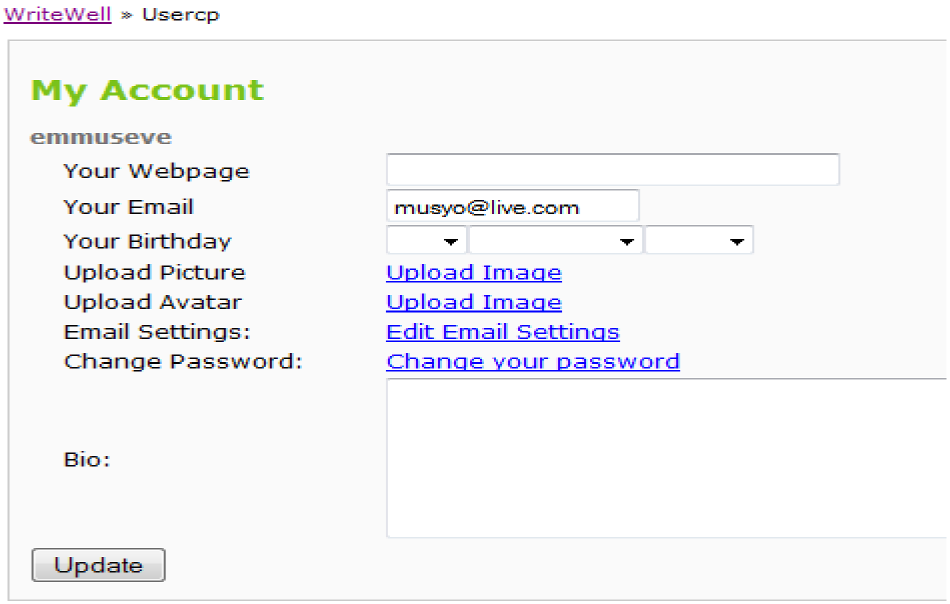
Searching for a particular article has been made easier. In the header menu of the website a search box is visible on the top right corner. However, as you browse through the web site you will notice that the search box appears in every page. This is because the header is the same for all the pages. There is no need for a user to find a specific page in order to perform an article search. Also notice included links which take users to the home page, submission page and RSS on top of the search box. The RSS link takes a writer to the latest submitted articles.



## 1.7 My Account

After registering, the data that has been provided is not adequate to grant full membership of the organization. After registration takes place the writer is required to go to his account and fill in the form that is provided. This form looks like in the diagram below.

By clicking on update, this data is combined with the data that a writer provided earlier in the registration form and is all saved in the same table. There are other web sites that require that when one registers should enter all the data at once, however, this application provides flexibility. A user may not need to fill in particular information at the time of registration thus making the registration process easier for such users.



WriteWell » Usercp

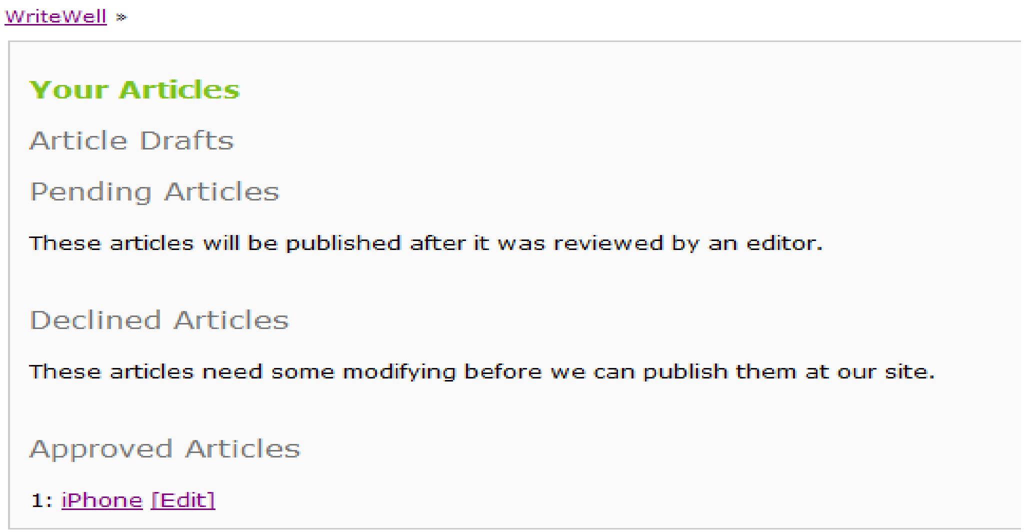
My Account

emmuseve

| | |
|---|---|
| Your Webpage | |
| Your Email | musyo@live.com |
| Your Birthday | |
| Upload Picture | Upload Image |
| Upload Avatar | Upload Image |
| Email Settings: | Edit Email Settings |
| Change Password: | Change your password |
| Bio: | |

## 1.8 My Articles

A registered writer (member) views the details of his articles which could be an article draft (saved but not submitted), pending, declined or approved articles. Here, the writer would know how well he is performing. For instance, if he sees many declined articles then he would know something is wrong with his articles. On the positive not, if he sees many approved articles then he would know that he is performing very well.



Your Articles

Article Drafts

Pending Articles

These articles will be published after it was reviewed by an editor.

Declined Articles

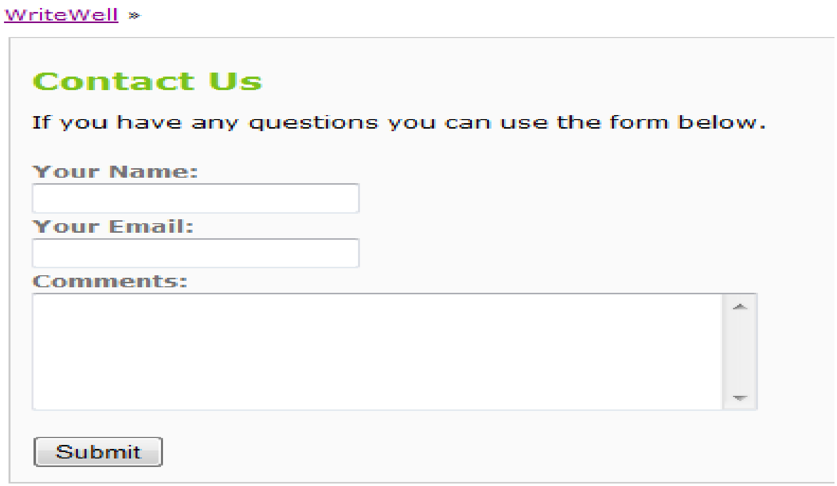
These articles need some modifying before we can publish them at our site.

Approved Articles

1: iPhone [Edit]

## 1.9 Contact Us

A visitor or a registered writer can contact the administrator sharing concerns about articles written or just making a comment about how nice the web-application interface looks.



Contact Us

If you have any questions you can use the form below.

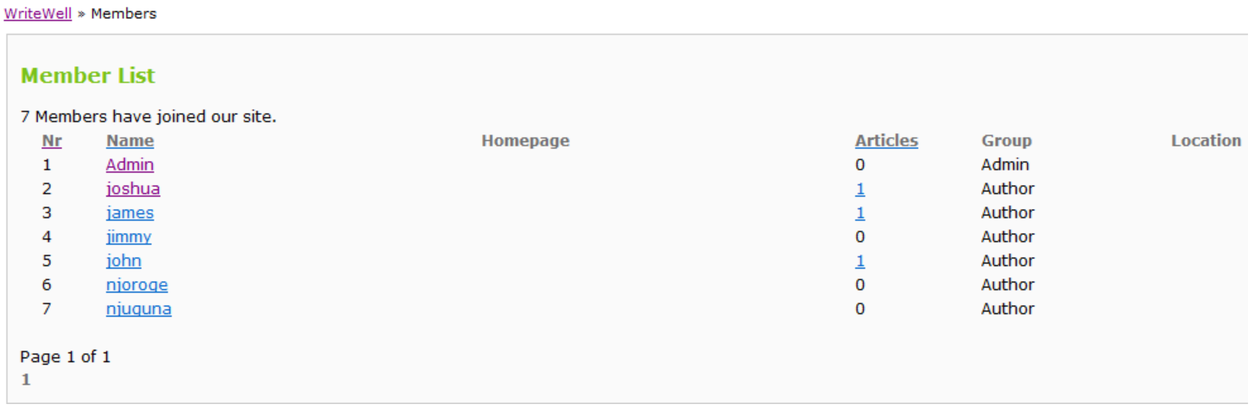
Your Name:

Your Email:

Comments:

## 1.10 Members List

Members list can be viewed by the visitor and registered writers mainly to attract more writers to the organization. A visitor can be attracted to the website if he sees his friend living near him as members. Also, the more writers the organization has the more visitors would be attracted to join.



WriteWell » Members

Member List

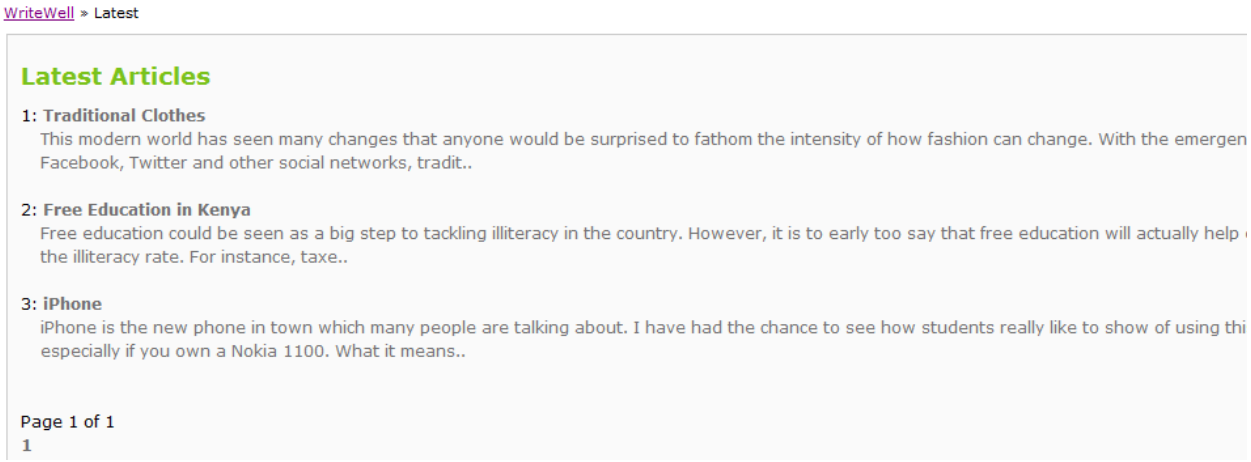
7 Members have joined our site.

| Nr | Name | Homepage | Articles | Group | Location |
|---|---|---|---|---|---|
| 1 | Admin | | 0 | Admin | |
| 2 | joshua | | 1 | Author | |
| 3 | james | | 1 | Author | |
| 4 | jimmy | | 0 | Author | |
| 5 | john | | 1 | Author | |
| 6 | njoroge | | 0 | Author | |
| 7 | njuguna | | 0 | Author | |

Page 1 of 1
1

## 1.11 Latest Articles

Latest articles are displayed on the home page so that the visitors or registered writers could easily see them. Similarly, by clicking the link "latest articles" a visitor or registered would see the latest articles.



WriteWell » Latest

Latest Articles

1: Traditional Clothes
This modern world has seen many changes that anyone would be surprised to fathom the intensity of how fashion can change. With the emergen
Facebook, Twitter and other social networks, tradit..

2: Free Education in Kenya
Free education could be seen as a big step to tackling illiteracy in the country. However, it is to early too say that free education will actually help
the illiteracy rate. For instance, taxe..

3: iPhone
iPhone is the new phone in town which many people are talking about. I have had the chance to see how students really like to show of using thi
especially if you own a Nokia 1100. What it means..

Page 1 of 1
1

### 1.7.0 Application Implementation

Special PHP classes and functions were coded to allow users to be able to write, edit and submit articles online. In addition, JQuery was used to for submission of pictures and articles as files. The picture submission worked only for certain pictures. Also, some PHP codes took care of displaying article status that is pending, draft, declined and approved.

Apart from that, there are other members or users of the application who would like to print articles from the system to review them as hard copy. For this to happen, there had to be a way in which an article could be printed in a PDF format. This also required a specific PHP code to work effectively. All files that a member of the system uploads to the website are stored in the root folder where the web site has been installed.

When a member directly writes an article using the web site text editor the article is directly stored in the database. There is no feasible way that could be implemented to store files in the database itself. Moreover, this type of file storage works perfectly thus there is no need to acquire another way of storing these files.

### 1.7.1 Testing

The testing manly focused on testing the user-side (front-end) functional requirements features. These functional requirements have been mentioned earlier in the functional requirements section. However, we will look at whether the functional requirements were satisfied.

The test performance done was based on a certain criteria. In some places a scale had to be devised to measure satisfaction. In this section results that were obtained from the test are displayed below.

| **Feature** | **Response** |
|---|---|
| Is it possible to register a large number of writers successfully? | Yes |
| Is it possible to validate the information provided by the writer while registering to become a member? | Yes |
| Is it possible for a large number of members to login? | Yes |
| Can a member log off successfully from the system? | Yes |
| Is it possible for more than one member to view the same article? | Yes |
| Can more than one member write and submit articles to the application at the same time? | Yes |
| Is it possible for two members (writers) to search for the same article? | Yes |
| Is it possible for members to see information of other member in the system successfully? | Yes |
| Can members view information on the articles they have submitted successfully? | Yes |
| Does the system automatically update the information? For instance, the number of articles written by writers without problems? | Yes |
| Is it possible to view article payment details? | No |

Table 2: Table showing our response to some tests done to the application

Apart from these results, other results were obtained on how the basic features (functional requirements) of the web-application would work on a very busy environment. A scale of measurement from 1 to 4 where 1 was not satisfied, 2 least satisfied, 3 satisfied and 4 very satisfied. These results shown below were obtained from some people that were hired to act as members of the web-application.

| **Feature** | **Writer's Response (1=not satisfied, 2=least satisfied, 3=satisfied, 4=very satisfied)** |
|---|---|
| Registering to the website | 4 |
| Information validation while registering | 4 |
| Login to the website application | 4 |
| Logging off from the website application | 4 |
| View latest articles | 4 |
| Write and save articles | 4 |
| Submit articles | 4 |
| Search articles | 4 |
| View other members | 4 |
| View article categories | 4 |
| View articles written | 4 |
| View article status | 4 |
| View article payment details | 1 |

Table 2: Table showing results obtained from other people concerning the features of our application

According to the results obtained from table 1, you can see that twelve of the features in the application worked very well when the application was used by more than one member. This implies that the application is working as intended, doing most of the tasks that a member (writer) would want to accomplish. However, we got a response of "No" for being able to view article payment. This means that writers wanted the possibility of viewing article payment possible. On the other hand, looking at table 2, we can see that the people that worked on testing the application were very satisfied by the web-application's features, because according to them the features were all working to their satisfaction except the payment system which was not working (not implemented).

### 1.8.0 Final deliverables

After testing the application, we can say that this application fulfils all the functional requirements of Eldoret Writers Centre except the payment system.

All of the core application features are in place, tested and functioning. Members can register with the system, write and submit articles which could be classified as draft, pending, declined or approved. Also, the searching tool enables the members to choose search criteria. Finally, considering that this was also a learning practice addition feature such as RSS was implemented.

### 1.9.0 Challenges

During the development of this application, there are certain challenges that were encountered on the user-side. These challenges are discussed below.

- It was a challenge to add an image to the text editor because the add image function seemed to choose certain images to add. Therefore, the code would have to be revised in order to allow different formats of pictures such as JPEG, GIF and PNG to be added.
- It was also challenging to add files because it seemed to give an error that the file type is invalid. This may imply that the function adding a file only selects certain file like PDF files. Therefore, a detection file format function would have to be added.
- The payment system proved to be difficult to implement. The problem was that the development of the other features of the web-application were already implemented, thus it would have been difficult to implement the payment system code.
- Although it is possible to view a PDF file, it was not easy at all to get this function working. A lot of tutorial had to be read. The reference below shows where help was found to enable the viewing of articles as PDF files.
- In the mobile programming course, using RSS feeds was simple because a mobile phone website was used. However, there were difficulties remembering the structure of RSS but after referring to the class notes it

became clearer how to write PHP RSS script to allow it to be displayed on the web-application.

### 1.10.0 Discussion

The approach to create a web-application for the organization to solve their cumbersome article emailing system seems to have paid off, although the payment system which seemed to be one of the major features was not implemented, most of the features the organization wanted were successfully implemented and working.

However, since the main focus was the user-side (front-end), we can see from the testing that many of the functional requirements features were implemented except the payment system. These functional requirements such as a user shall be able to register, login, logoff, search, contact, view categories, view members, view article which are draft, pending, declined or approved were implemented and worked very well. Another feature that was important was the RSS feed which allowed users to view latest articles in other websites or mobile phones with GPRS internet just by using the RSS link.

Moving on to the design of the website, much design was done to enhance the experience of the writer. If we look at the home page, a general feeling of a well designed and structured web-application materializes. The blue background and the green color were all done using CSS to give good attractive color. The navigation main menu on the right with links to many things a writer can do such as write, viewing latest article etc ensured that the writer (member) had only one place where the entire task could be done. The screen shot of the home page in the web-application interface design shows how the content is displayed in the middle of the web-application. Since, the content only changes when the user clicks on the navigation main menu links, it makes the writer feel like he is working only in one place (content area).

In addition, it would be better if we look at the screen shots shown in web-application interface design section to see actually how the web-application looks like or to make it better, it would be nice to view the web-application to have an opinion.

### 1.11.0 Conclusion

The solution of making the web-application (WriteWell) for the user-side (front-end) paid off because most of the functional requirements features were implemented and worked according to what the organization wanted except, however because of time constraint and the complexity of coding the payment system, it became impossible to finish on time, but in the future a payment system that would simplify the payment of money to articles written by writers would be implemented. Finally, much was learnt from making the front-end for the web-application especially designing and coding using PHP, MySQL, HTML, JavaScript and JQuery.

## 1.12.0 References

Barnamos (2005). PHP skin. Retrieved 10/3/2010 from http://www.daniweb.com/forums/thread24128.html#

Bluephoneix (2004). Writing a Template System in PHP. Retrieved 10/3/2010 from http://www.codewalkers.com/c/a/Display-Tutorials/Writing-a-Template-System-in-PHP/

Fpdf (unknown). FPDF Library. PDF generator. Retrieved 10/3/2010 from http://www.fpdf.org/

Gertjan (2007). Creating a simple login script with php mysql. Retrieved 3/12/2009 from http://www.ineedtutorials.com/code/php/creating-a-simple-login-script-with-php-and-mysql

Noah, Jacques (2006). Creating a Login Script for a PHP / MySQL Blogging System. Retrieved 3/2/2010 from http://www.devshed.com/c/a/MySQL/Creating-a-Login-Script-for-a-PHPMySQL-Blogging-System/

PHP Login Script (2008). PHP Login Script v2.1. Retrieved 2/12/2009 from http://php-login-script.com/

Stryker250 (2006). [PHP] Simple PHP\ MySQL Login. Retrieved 1/1/2010 from http://forums.tizag.com/showthread.php?t=798

Triond (unknown). Triond. Retrieved 21/2/2010 from http://www.triond.com/

W3S (2010). The Largest Web Developer Site on the Net. Full Web Building Tutorials – all free. Retrieved 2/12/2009 from http://www.w3schools.com/

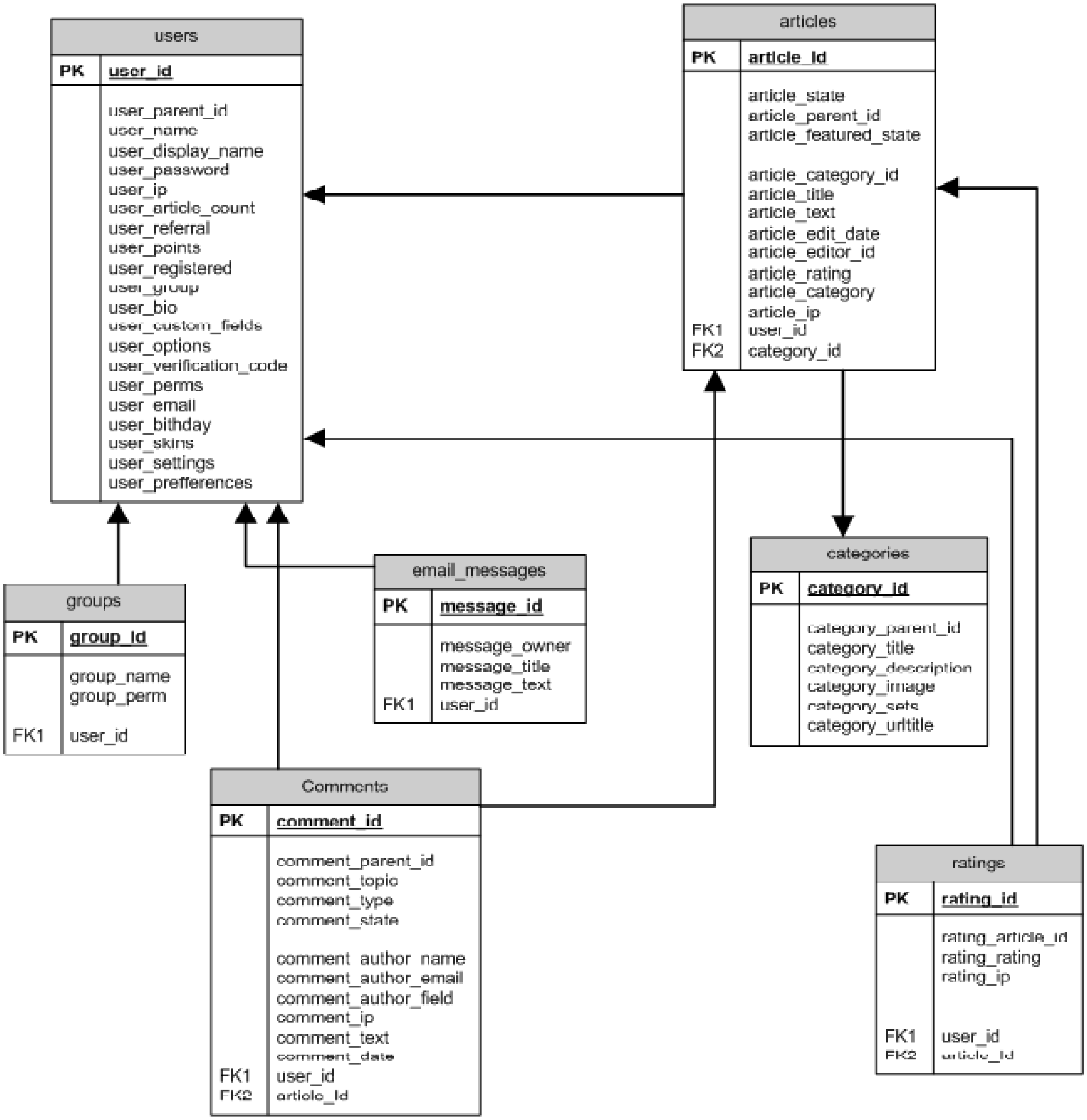
Williams E. Hugh, Lane David (2004). Web Database Applications with PHP and MySQL, Second Edition. Building Effective Database-Driven web sites. Retrieved 2/12/2009 from http://oreilly.com/catalog/9780596005436

Xomba (unknown). Xomba. Retrieved 20/2/2010 from http://www.xomba.com/

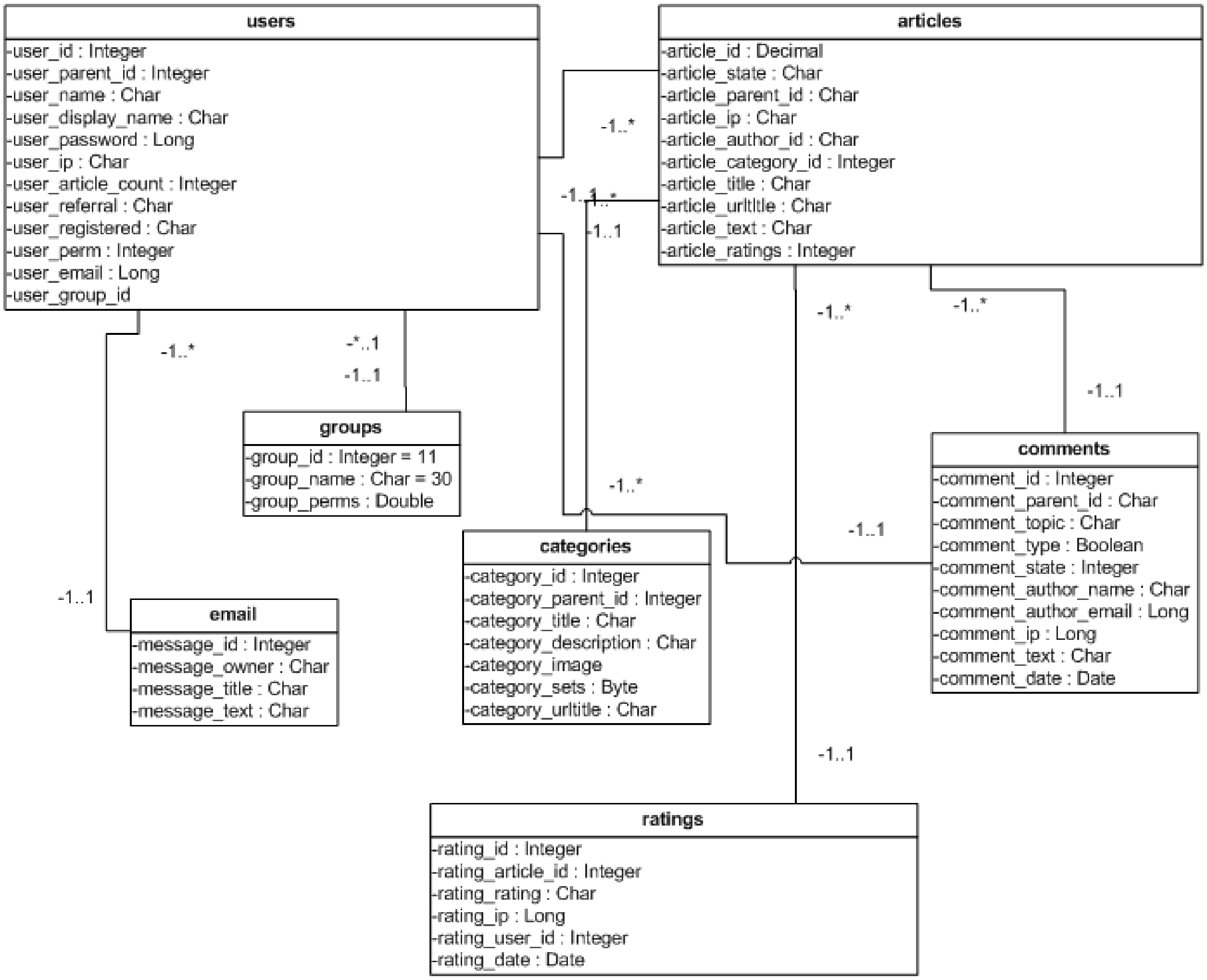
## 1.13.0 Appendix

### Appendix 1: database diagram

Database diagram for WriteWell project



## Appendix 2: UML diagram


users
-user_id : Integer
-user_parent_id : Integer
-user_name : Char
-user_display_name : Char
-user_password : Long
-user_ip : Char
-user_article_count : Integer
-user_referral : Char
-user_registered : Char
-user_perm : Integer
-user_email : Long
-user_group_id
articles
-article_id : Decimal
-article_state : Char
-article_parent_id : Char
-article_ip : Char
-article_author_id : Char
-article_category_id : Integer
-article_title : Char
-article_urltitle : Char
-article_text : Char
-article_ratings : Integer
-1..*
-1..1
1..*
-1..1
-1..*
-*..1
-1..1
-1..*
-1..*
-1..1
groups
-group_id : Integer = 11
-group_name : Char = 30
-group_perms : Double
comments
-comment_id : Integer
-comment_parent_id : Char
-comment_topic : Char
-comment_type : Boolean
-comment_state : Integer
-comment_author_name : Char
-comment_author_email : Long
-comment_ip : Long
-comment_text : Char
-comment_date : Date
-1..*
-1..1
categories
-category_id : Integer
-category_parent_id : Integer
-category_title : Char
-category_description : Char
-category_image
-category_sets : Byte
-category_urltitle : Char
-1..1
email
-message_id : Integer
-message_owner : Char
-message_title : Char
-message_text : Char
-1..1
ratings
-rating_id : Integer
-rating_article_id : Integer
-rating_rating : Char
-rating_ip : Long
-rating_user_id : Integer
-rating_date : Date