**Table of contents**

Table of contents

**Review 1**

* 1. [Introduction](#_Introduction_1)
  2. [Existing Scenario](#_Existing_Scenario)
  3. [Customer Requirement Specification](#_Customer_Requirement_Specification)
  4. [Functional Requirement Specification](#_Functional_Requirement_Specificatio)
  5. [System Requirements](#_System_Requirements)
  6. [Development Software](#_Development_Software)
  7. [Technology](#_Techonology)

**Review 2**

* 1. [Use Case and Sequence Diagram](#_Use_Case_Diagram_2)
  2. [Entity Relationship Diagram (ERD)](#_Entity_Relationship_Diagram_1)

**Review 3**

* 1. [Database Relationship Diagram](#_Database_Relationship_Diagram_3)
  2. [Table Definition](#_Table_Definition_1)
  3. [Other Constraint](#_Other_constrains:_1)
  4. [Site map](#_GUI_Design_1)
  5. [GUI Design](#_GUI_Design_3)
  6. [Report](#_Report_1)

**Task sheet**

[Task sheet](#Tasksheet)

**Monitoring Report**

[Unit Test](#UnitTest)

**REVIEW 1**

1. **Introduction**

**MasterCook** is an organization founded by YoungChefs, a group of young successful cooks in Vietnam. they always think of how to serve best meals for customers, as a result, they are insearch of new favourite recipes.

In early days, they started to found a club of cooking lover. As it turns out, cooks had a huge appetite for sharing recipes; the club was an instant success. Over a few years later, YoungChefs wanted to expand the club and social network allowing cooks of all types and interests to connect through recipes, cooking tips, personal collections, and more. What started with a single cookie recipe has grown organically to become the large food-focused social network with a community of over 50 home cooks consuming more than 1000 recipes.

“Everyone can cook”, that is the slogan of YoungChefs. They want to share all the recipes of meals for passtionate people and make an online magazine about food. This is the idea of **MasterCook.**

1. **Existing Scenario**

The club of cooking lovers founded by YoungChefs has been showed many disadvantages. The number of members in the group is one of them, YoungChefs can not handle and manage such a large quantity of cooks every offline meetings.

Besides, all the knowledge made by this club is not organized and shared with every members as well as all people in Vietnam. Moreover, as more than 1000 recipes are made, that is really difficult to store and find out one of them.

Finally, YoungChefs decides to make an online club, which is a website that always upload news and recipes for everyone to have access to newest cooking knowledge.

# Customer Requirement Specification

The website contains new recipes of these young cooks and collaborators, as well as some best suggested restaurants to customers, besides that, the website provides some up to date cooking news.

Guest who visit web-site firstly will see the home page. The home page contains a banner on header for our introduction. Guest will see the latest posts and pinned posts at first.

Guest can choose to see posts by choosing categories from a menu. Besides, they can search to see posts by name.

When guest see post in detail, they can see some related posts, related news. They are also able to leave their comment, which will see directly after they submit.

***If a user log in, there are 4 roles: collaborator, employee, manager and administrator. Notice that they can not register for new account, they must contact page if they want to be a collaborator.***

* **Collaborator**

Main task of collaborator is posting news to web-site, collaborator can only upload posts in the category they are assigned, which will be reviewed by employeed. Each post must have required following informations such as title, date, author, etc.

Collaborator can search their own posts which were accepted, and these post at first will show in order of latest to oldest. Collaborator can see which posts are still pending, which post were rejected. The rejected posts come with the reason why they were rejected.

Collaborator can edit the post if necessary. Besides, they can check and delete comments in the post.

Collaborator can see their own profile. They can see some information of their supervisor such as the contact, name, etc.

* **Employee**

Employee can do everything that collaborator do.

Employee can also review posts of collaborator. Employee have the right to review their collaborator’s posts, reject it or accept it. The posts which are accepted by employees will be passed to manager for final review before showing in web-site. If employee rejects task then they are required to add reason why the post is rejected.

* **Manager**

Manager will receive posts which are accepted from employees. These posts will be reviewed finally by manager.

Manager can reject post with some following reason why the post were rejected. Manager can accept post and then the post will be shown in web-site at the begin of the home page as the latest new.

Manager have the right to create new employee or collaborator. Besides, they are able to edit information of collaborators and employees.

Manager can see their own profile and some information about administrator, their supervisor, such as the contact, name, etc.

* **Administrator**

After logged in, administrator will see the main page for his tasks.

Administrator can manage categories of posts. He can remove existed one from web-site, update, or insert new category.

Administrator can manage all staff account such as create new account, delete account, update information of account, change role.

# Functional Requirement Specification

***4.1. Search posts for user***

**4.1.1.Description**

Guest type in a text box to search for the title of post they want to see. If search box empty, do nothing. Having no result then show an announcement, else display posts in order of date from latest to oldest.

**4.1.2.Functional Requirements**

REQ-01. A search text box and a submit button

REQ-02. All searched posts must be not in deleted status.

REQ-03. Category of the posts must be not in deleted status.

***4.2. GUI attractive***

**4.2.1.Description**

System should provide to Guest an attractive home page.

**4.2.2.Functional Requirements**

REQ-01. An attractive home page

***4.3. Select category of posts from menu***

**4.3.1.Description**

Guest chooses options in menu and subcategory in each option.

After choosing specific category, all post belong to this category will be displayed in order of date and the priority state.

**4.3.2.Functional Requirements**

REQ-01. A menu bar

REQ-02. All displayed categories must be not in deleted status.

REQ-03. All displayed menu options must be not in deleted status.

REQ-04. All displayed posts must be not in deleted status.

***4.4. Leave comment in a post***

**4.4.1.Description**

Guest will type his comment into a text area. After click submit, a form for guest to enter info of guest like name, email, phone, gender.

**4.4.2.Functional Requirements**

REQ-01. A text area for guest to type comments.

REQ-02. A form to fill informations of guest. Form have name field, email field, phone field, gender button.

REQ-03. After send comments, the current page will be updated.

REQ-04. Comment will be checked before being shown in web.

***4.5. See detail of post***

**4.5.1.Description**

When guest click a post, the post’s detail will be shown with the following structure:

* header
  + title
  + author
  + date modified
* body
  + text
  + images
* footer
  + related post
  + comments

**4.5.2.Functional Requirements**

REQ-01. increase view of post implicitly.

REQ-02. The post must be not in deleted status.

REQ-03. The related post is same sub-category with the current post.

REQ-04. The comments must be not in deleted status.

## ***4.6. User can log in the system***

**4.6.1. Description**

The system provide a page to allow user login. The system must distinguish Collaborator, Employee, Manager and Administrator accounts.

**4.6.2. Function Requirements**

REQ-01. User must provide Username and Password to log in into the website.

REQ-02. System must distinguish User role to display pages which users can view.

REQ-03. User can only access to the pages they have permission.

## ***4.7. Management of User***

### **4.7.1.Description**

System must provide for User (Collaborator, Employee, Manager and Administrator) a management page

### **4.7.2.Functional Requirements**

1. User can modify their profiles

## ***4.8. Collaborator can post articles to system for pending***

### **4.8.1.Description**

System must provide for Collaborator an upload page. In this page, Collaborator can write their articles and submit. Submission should provide these information:

1. Title
2. Content (text/picture)
3. Subcategory

### **4.8.2.Functional Requirements**

1. Collaborator must provide Title, Content (text and picture), Subcategory to submit article.
2. Collaborator can only have new posts for subcategory he/she manages.
3. Title is unique
4. Submission must be sent to Employee

## ***4.9. Collaborator can view their own articles***

### **4.9.1.Description**

System must provide for Collaborator an article management page. Collaborator can view their article list in order latest to oldest by default and can sort by status: accepted, pending, rejected. The rejected articles come with the reason why they were rejected. Collaborator can choose to view an article

### **4.9.2.Functional Requirements**

1. System must provide Article Management page for Collaborator, they can view their article list in order latest to oldest
2. Collaborator can sort their articles by status
3. Collaborator can view the reason of rejected articles
4. Collaborator can choose an article in the list to view

## ***4.10. Collaborator can edit their articles***

### **4.10.1.Description**

In Article Management page, Collaborator can choose to edit their articles. Modified articles are sent to Employee.

### **4.10.2.Functional Requirements**

1. Collaborator can edit their articles
2. Modified articles must be sent to Employee

## ***4.11. Collaborator can delete comments in their articles***

### **4.11.1.Description**

When viewing their own article, Collaborator can delete comments in their articles

### **4.11.2.Functional Requirements**

1. Collaborator can delete comments in their articles

***4.12. Employee can review posts of collaborator.***

**4.12.1.Description.**

Employee can view pending posts of collaborator in the specific subcategories he/she manages. Employee has to read and decides to approve the post or reject it. After approving the post, it will be passed to manager.

**4.12.2.Functional Requirements.**

REQ-01. Employee can view list of new pending posts.

REQ-02. Employee can approve or reject the pending posts.

REQ-03. Employee can edit post immediately if necessary.

REQ-04. Employee must have a reason why the post is rejected.

***4.13. Manager can review posts after they are passed by employees.***

**4.13.1.Description.**

Manager can view pending posts of collaborator in the specific category he/she manages. Manager has to read and decides to approve the post or reject it. After approving the post, it will be on the website as lastest news.

**4.13.2.Functional Requirements.**

REQ-01. Manager can view list of new pending posts.

REQ-02. Manager can approve or reject the pending posts.

REQ-03. Manager can edit post immediately if necessary.

REQ-04. Manager must have a reason why the post is rejected.

***4.14. Manage accounts.***

**4.14.1.Description.**

Manage and administrator can manage account of employees and collaborators, include:

* View how many accepted posts, pending posts and rejected posts of collaborator.
* View how frequent that collaborator have a new post.
* View the average views and comments each post of collaborator.
* View how many posts accepted and rejected by employee.
* View how frequent that employee check a new post.
* View the average views and comments each post approved by that employee.
* Create new accounts for employees, collaborators and managers, include: First name, last name, username, password, phone number, email, date of birth, address, gender, working categories, role. (Only administrator can create account for managers)
* Edit information of employees, collaborators and managers, include: First name, last name, password, phone number, email, date of birth, address, gender, working categories, role. (Only administrator can edit information of managers)

**4.14.2.Functional Requirements.**

REQ.01 Must have an interface for managing accounts.

REQ.02 All fields in creating and editing accounts must be validated:

* First name and last name must not be empty.
* Username is unique.
* Password must have 6-32 characters.
* Phone number must have 10-11 numbers.
* Email must have right format.
* Gender must be checked.
* Can have more than 1 subcategories or empty.
* Role must be checked as collaborator, employee or manager (manager can only be checked by administrators).

***4.15. Administrator, manager manage posts***

**4.15.1.Description**

Administrator and manager after logging in can modify all posts.

**4.15.2.Functional Requirements**

REQ-01. System provide a page for modifying the post.

REQ-02. System provide a form with all attribute which are allowed to be modified.

REQ-03. Some post’s attributes can be modified is:

* the title
* Post’ state:
  + accepted
  + pending
  + rejected
  + deleted
* The modified date is automatically updated to current date after they click submit changes.
* The priority.

REQ-04. If some attributes that manager not allow to modified it will be disable.

***4.16. Administrator manage subcategories***

**4.16.1.Description**

Administrator after logging in can modify all subcategories, include add new, edit and remove.

**4.16.2.Functional Requirements**

REQ-01. System provide a page for modifying the post.

REQ-02. System provide a form with all attribute which are allowed to be modified.

REQ-03. Some subcategory’s attributes can be modified is:

* the name
* state:
  + active
  + deleted
* The category in which this belongs to.

***4.17. Making statistics for Administrator***

**4.17.1 Description**

System provides many options for administrator to make statistics based on all web-site’s activites.

**4.17.2 Functional Requirements**

REQ-01. System provides a menu for administrator to choose options for what statistics he want.

REQ-02. There’s a option for total posts of web-site in a specific time such as:

* total accepted post of web site from date X to date Y.
* total rejected post of web site from date X to date Y.
* total pending post of web site from date X to date Y.

REQ-03. There’s a option for finding out list of posts based on number of views in descending or ascending order in a specific time.

# System Requirements

## ***5.1. Hardware***

### **5.1.1.Web Server**

|  |  |
| --- | --- |
| Processor | Pentium IV or higher. |
| Memory | 512MB RAM or greater. |
| Modem/ADSL | Connecting to internet 24/24 |

### **5.1.2.Client**

|  |  |
| --- | --- |
| Processor | Pentium III or higher. |
| Memory | 64MB RAM or greater. |
| Monitor | Super VGA (800x600) or higher resolution with 256 colors. |
| Pointing Device | Microsoft Mouse, Microsoft IntelliMouse®, or compatible pointing device. |
| Modem/ADSL | Connecting to internet |

## ***5.2. Software***

### **5.2.1.Web Server**

|  |  |
| --- | --- |
| Operation System | Windows 2000/XP/2003 |
| Browser | Internet Explorer 6.0 or higher. |
| Database | Microsoft SQL Server 2014 |
| Software | Java SDK |
|  | Apache Tomcat 8.0.27 |

### **5.2.2.Client**

|  |  |
| --- | --- |
| Operation System | Windows XP or later. |
| Browser | Internet Explorer 6.0 or higher. |

# Development Software

* Microsoft Windows 10
* Netbeans 8.2
* Java SDK
* Apache Tomcat 8.0.27
* Microsoft SQL Server 2014
* Google Chrome

# Techonology

* J2EE
* Java Web Service
* JSP, Servlet
* HTML

1. **Use Case and Sequence Diagram**
2. ***. Collaborator***

**Collaborator**

1. **Post article**

|  |  |  |
| --- | --- | --- |
| **Author** | Vo Gia Vu | |
| **Use Case Name** | Post articles | |
| **Actors** | Collaborator | |
| **Description** | Collaborator wants to post articles to system for pending | |
| **Requirements** | Collaborator provide articles information | |
| **Pre-conditions** | Actor has logged in as Collaborator | |
| **Post-conditions** | Success: Collaborator creates an article and sends to system in pending status  Fail: Refill information | |
| **Basic flow** | Actor Actions:  1. Actor click ‘Upload’ button on Home page menu.  3. Actor input article’s information, select ‘Subcategory’ combo box and click ‘Submit’ button. [Alternative 1] | System Responses:  2. System redirects to ‘Upload’ page and display Article form with following control:  -‘Title’ text field  -‘Content’ text area  -‘Subcategory’ combo box  -‘Submit’ button  4. System checks the information [Exception 1] |
| **Alternative flow** | Actor Actions:  [Alternative 1]  1. Actor click ‘Home’ button | System Responses:  2. System returns to Home page. |
| **Exception** | Actor Actions:  [Exception 1]  1. Actor left ‘Title’ text field blank.  2. Title has already exists  3. Actor left ‘Content’ text area blank. | System Responses:  System display message to Actor:  - ‘Title cannot be blank.’  - ‘This Title has already exist.’  - ‘Content cannot be blank.’ |

