|  |
| --- |
| Hanoi university of science and technology  Document Managing System |
| E-Book Dungeon Report |
| Group:  *Trịnh Huy Hoàng ID: 20101586*  *Tăng Thế Cường ID: 20101226*  *Lê Hoàng Phương ID: 20101999* |
|  |
| **Instructor: Mrs. Nguyễn Thị Thu Trang and Mr. Trịnh Thành Trung** |
|  |
|  |
| **Hanoi, 05/2014** |

Table of Contents

[I. Work Assignment 2](#_Toc388485152)

[1. Trịnh Huy Hoàng: 2](#_Toc388485153)

[2. Tăng Thế Cường: 2](#_Toc388485154)

[3. Lê Hoàng Phương: 2](#_Toc388485155)

[II. Introduction 3](#_Toc388485156)

[1. Requirement 3](#_Toc388485157)

[2. Main functions of website 4](#_Toc388485158)

[2.1. Authentication 4](#_Toc388485159)

[2.2. Document Management 7](#_Toc388485160)

[2.3. Group Management 12](#_Toc388485161)

[2.4. User Management 14](#_Toc388485162)

[3. Difficulties: 15](#_Toc388485163)

[III. Technical solution 16](#_Toc388485164)

[1. Framework 16](#_Toc388485165)

[1.1. Phalcon PHP 16](#_Toc388485166)

[1.2. Application of Phalcon PHP 16](#_Toc388485167)

[2. Bootstrap 16](#_Toc388485168)

[3. Security 17](#_Toc388485169)

[4. Advanced: 17](#_Toc388485170)

[4.1. Ajax searching 17](#_Toc388485171)

[4.2. PHP Excel 18](#_Toc388485172)

[IV. Result 19](#_Toc388485173)

[1. Database 19](#_Toc388485174)

[2. Interfaces 20](#_Toc388485175)

[3. Evaluation 23](#_Toc388485176)

[3.1. Advantages 23](#_Toc388485177)

[3.2. Disadvantages 23](#_Toc388485178)

[V. Conclusion 24](#_Toc388485179)

[Reference 25](#_Toc388485180)

# Work Assignment

### Trịnh Huy Hoàng:

* Design database structure
* Implementing user server-side use-cases
* Making slide presentation

### Tăng Thế Cường:

* Implement admin server-side use-cases
* Testing
* Write report

### Lê Hoàng Phương:

* Implement client-side front-end (HTML, CSS, javascript,etc.)
* Resolve technological issues.

# Introduction

## Requirement

Build a website that allow user to:

* Upload document with format doc/docx, xls/xlsx, ppt/pptx, pdf, html, etc. to website.
* Create new document by HTML and be able to export to HTML or docx.
* Edit document attributes: name, description, author, year, etc.
* Edit document visibility: Only Me, Member Only, Public.
* Put document into one or many groups
* Search for document by attribute(s).
* View document directly on browser.
* Download document

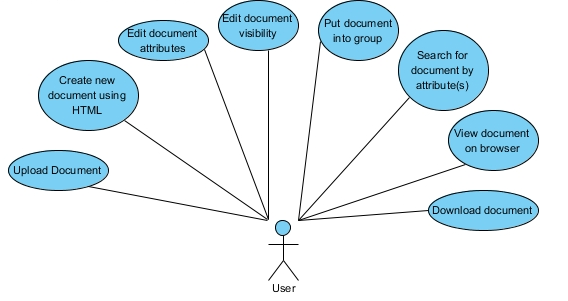


Figure 1: Use-Case Diagram of User according to requirement

Also, website allows administrator to:

* Manage user
* Manage document group. Be able to create child group.
* Download all documents belong to a group.

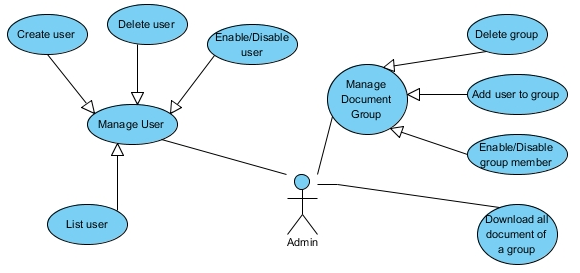


Figure 2: Use-Case Diagram of Administrator according to requirement

## Main functions of website

### Authentication

Authentication process allow user and admin to log in/out of website.

#### Login

Get POST request from view and then use sha1 to encrypt password. After that, check in database for user\_name, password\_hash and status = ‘active’. If true then register in session and user (or admin) can login.

public function startAction() {

**if** ($this->request->isPost()) {

$username = $this->request->getPost('login\_username', 'alphanum');

$password = $this->request->getPost('login\_password');

$password = sha1($password);

$user = **Users::**findFirst("user\_name='$username' AND password\_hash='$password'");

**if** ($user != false && $user->status == 'active') {

$group = **Groups::**findFirst("owner\_id='$user->user\_id'");

**if** ($group == false) {

$this->\_registerSession($user);

**return** $this->redirect("contents/index");

} **else** {

$this->\_registerSession($user, $group->group\_id);

**return** $this->redirect("lobby/index");

}

//**return** $this->forward('contents/index');

} **else** **if** ($user != false && $user->status != 'active')

$this->flash->error('Your account has not been activated!');

**else** **if** ($user == false)

$this->flash->error('Wrong username/password');

}

**return** $this->forward('session/index');

}

#### Logout

Clear the session and return to the index page.

#### Register

Receive information from register form. If all field validated, create new account.

public function registerAction() {

$validate = TRUE;

$request = $this->request;

**if** ($request->isPost()) {

**if** (isset($\_POST['register\_submit'])) {

//...//

$verification\_hash = md5(uniqid(rand(), **1**));

$user = **new** Users();

$user->user\_name = $username;

$user->password\_hash = sha1($password);

$user->verification\_hash = $verification\_hash;

$user->email = $email;

$user->first\_name = $firstname;

$user->last\_name = $lastname;

$user->phone = $mobilephone;

$user->expiration = **new** Phalcon\Db\RawValue('now()');

$user->extra\_time = **0**;

$user->status = 'active';

$user->maximum = **new** Phalcon\Db\RawValue('default');

$user->used = **new** Phalcon\Db\RawValue('default');

;

$user->available = **new** Phalcon\Db\RawValue('default');

//$user->created\_at = **new** Phalcon\Db\RawValue('now()');

//...//

}

}

**return** $this->forward('session/index');

}

### Document Management

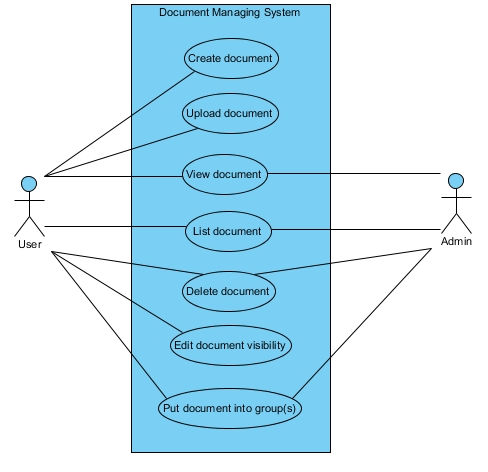


Figure 3: Use-Case Diagram of all document managing activities

#### Create document

Forward user to “create” page, which has a mini editor (enough for regular usage). After an user complete the document, click Save and new document will be saved to database and physical storage. This use TinyMCE library as editor and Htmltodocx library to create docx file

public function saveAction()

{

**if** (!$this->request->isPost()) {

**return** $this->dispatcher->forward(array(

"controller" => "contents",

"action" => "create"

));

}

$auth = $this->session->get('auth');

$userId = $auth['id'];

$userName = $auth['name'];

$dirPath = $this->config->upload\_dir . $userName . '/';

//...//

$html = $this->request->getPost("content");

$phpword\_object = **new** PHPWord();

$section = $phpword\_object->createSection();

$html\_dom = **new** simple\_html\_dom();

$html\_dom->load('<html><body>' . $html . '</body></html>');

$html\_dom\_array = $html\_dom->find('html', **0**)->children();

$paths = htmltodocx\_paths();

// ... //

if (!$content->save()) {

$messages = $content->getMessages();

foreach ($messages as $m) {

$errors[] = (string)$m;

$this->logger->error("Save file upload error: " . (string)$m);

}

}

rename($h2d\_file\_uri, ($dirPath . $content->information));

$this->flash->success("Your document is created successfully");

**return** $this->forward("contents/index");

}

#### Upload document

Allow users to select multiple documents from their computer and upload to system. The documents info will be saved to database and the physical files will be saved in /upload/username folder

public function uploadAction()

{

$auth = $this->session->get('auth');

$userId = $auth['id'];

$userName = $auth['name'];

$dirPath = $this->config->upload\_dir . $userName;

try {

**if** ($this->request->hasFiles() == true) {

$errors = array();

$manager = $this->transactions;

$transaction = $manager->get();

**//...//**

**if** (!$file->moveTo($dirPath . '/' . $file->getName())) {

$err = error\_get\_last();

$errors[] = $err['message'];

$this->logger->error($err['message']);

} **else** {

**if** (!$content->save()) {

$messages = $content->getMessages();

$transaction->rollback((string)$messages[**0**]);

**foreach** ($messages as $m) {

$errors[] = (string)$m;

$this->logger->error("Save file upload error: " . (string)$m);

}

} **else** {

$user->used += $fileSize;

$user->available -= $fileSize;

$user->setTransaction($transaction);

**//...//**

$transaction->commit();

$this->flash->success($this->config->message->success->upload); }

}

} **else** { $errors[] = "Available space is not enough"; }

//...//

}

} catch (Exception $ex) {

$this->logger->error('Upload exception: ' . $ex->getMessage());

}

}

#### View document

At this moment, our system can only view docx, doc, xls, xlsx, pdf files. With xls and xlsx, we use PHPExcel library. With doc and docx, we use custom script. With pdf, it is “browser support”.

public function showAction($content\_id = **0**)

{

$auth = $this->session->get('auth');

$userId = $auth['id'];

$userName = $auth['name'];

try {

$content = **Contents::**findFirst($content\_id);

**if** ($content) {

**if** ($content->status == 'private' && $content->owner\_id != $userId) {

$this->flash->error("You don't have the right to view this document!");

**return** $this->forward("contents/index");

}

**if** ($content->content\_extension == 'docx') {

//...//

} elseif ($content->content\_extension == 'doc') {

//...//

} elseif ($content->content\_extension == 'xlsx' || $content->content\_extension == 'xls') {

//...//

} **else** {

$this->flash->error("Content was not found");

**return** $this->forward('contents/index');

}

} catch (Exception $ex) {

$this->logger->error('Show content exception: ' . $ex->getMessage());

**return** $this->forward('contents/index');

}

}

#### List document

In this function, we select all documents properties from database, and then print all in a table.

#### Delete document

In this function, we can select one or more document to delete. Documents info will be deleted from database and also on physical storage. Transaction must be used in this function to keep integrity

#### Edit document visibility

In this function, if a document is private (“Only Me”), it can be shared to other groups or make public. If a document is public, it can be made back to private.

#### Put document into group(s)

This function is a part of 2.2.6, which let user select groups to share documents.

public function updateAction()

{

**if** (!$this->request->isPost()) {

**return** $this->dispatcher->forward(array(

"controller" => "contents",

"action" => "index"

));

}

//...//

}

### Group Management

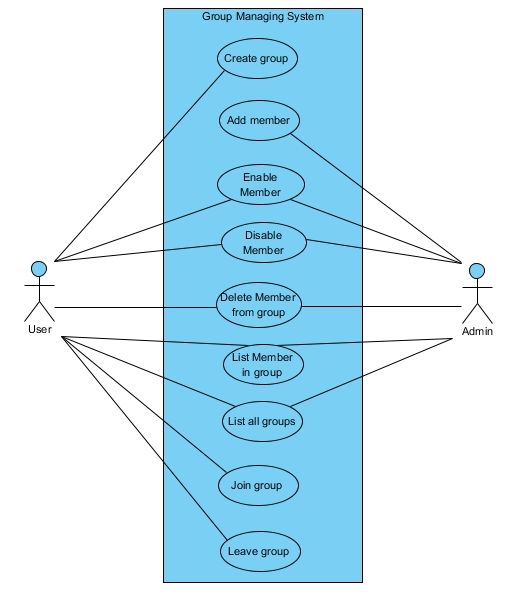


Figure 4: Use-Case Diagram of all group managing activities

#### Create group

When register, if an user select Group in account type, his group will be created at that moment. Otherwise, if user select Individual, then after login, he can create group in account tab.

// Create **new** group

$group = **new** Groups();

$group->owner\_id = $userId;

$group->group\_name = $groupName;

**if** ($group->save() == false) {

$messages = $group->getMessages();

$this->flash->error((string) $messages[**0**]);

#### Add member

Only admin can do this function. Admin can add members to an existing group.

#### Enable member

Group Admin can enable a member (make member status ‘active’) after an user want to join group or if member status is ‘rejected’.

#### Disable member

Group Admin can disable a member (make member status ‘rejected). That member is still in group but can not see any documents from group anymore. If user want to be enabled back, he must send join request again (and of course, Group Admin accept)

#### Delete member from group

Group Admin can delete a member from his group. If user want to be back in group, he must send join request again (and of course, Group Admin accept)

#### List member in group

Group Admin can list all members of his group to manage (enable, disable or delete).

#### List all group

List all of document group currently present in the system.

#### Join group

Any user can send join request in order to join a group. This request will be send to Group Admin of that group, and user need to wait for Group Admin to accept

#### Leave group

Member of a group can leave that group at any time.

### User Management

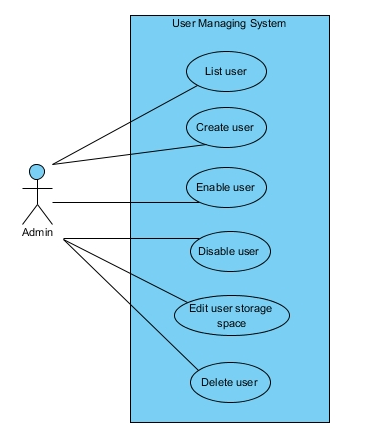


Figure 5: Use-Case Diagram of all user managing activities

#### List user

Admin can list all users in the system to manage (enable, disable, edit user storage and delete user).

#### Create user

Admin can create new user.

#### Enable user

Admin can enable user (make user status ‘active’).

#### Disable user

Admin can disable user (make user status ‘inactive’).

#### Edit user storage space

Admin can edit maximum user storage.

#### Delete user

Admin can delete users. This action can not be undone.

## Difficulties:

The use-case "Download all documents in a group" prove to be difficult, due to the way documents are stored on server.

For admin to be able to download multiple documents at a time, server must first compress those documents into archive. However each document stored in different directory (correspond to its owner), so compress them is difficult. We were unable to solve this problem in time, so this use-case is left unfinished.

Viewing document on browser is much more difficult than expected. Images, plaintext, pdf, etc. can be render by browser by itself, but special format such as doc, xls, ppt,, etc. require helps of additional libraries to read. We only able to find PHP Excel to help read xls/xlsx file, while other format is almost completely unsupported.

Render them is also difficult, because those format have different markup from HTML. Some special effect were hard to reproduce using HTML, so we render those document in plaintext form.

# Technical solution

## Framework

### Phalcon PHP

Phalcon is an open source, full stack framework for PHP 5 written as a C-extension, optimized for high performance. You don’t need to learn or use the C language, since the functionality is exposed as PHP classes ready for you to use. Phalcon also is loosely coupled, allowing you to use its objects as glue components based on the needs of your application.

Phalcon is not only about performance, our goal is to make it robust, rich in features and easy to use!

We choose because it’s fast, rich in features and easy to use.

### Application of Phalcon PHP

The most important function in our project is listing documents

In order to list all documents, we use Phalcon Controller to access database (through Model Contents), select all properly documents and pass to the View. In View, we list all documents in table

Another important function is create document.

In order to create document, we use TinyMCE as an editor. After user click on “Create new document”, createAction in ContentsController will be called and then redirect user to “create” View. In this View, we integrate TinyMCE library, so user can use this mini editor to create the document they want. When user click on Save button, saveAction in ContentsController will be called. In this action, htmltodocx library is used to create docx document from the “textarea” in “create” View. After that, we use Phalcon to create a new instance of Contents Model, apply all attributes and then save in database. In Phalcon, when we need to create/update record in database, we just need to use save() function.

## Bootstrap

Using bootstrap helps a lot when creating good-looking UI. It contain profession-looking CSS designed by expert.

It's also easy to use, esay to customize while greatly improve website's user experience.

Bootstrap also contain script to validate input at client-side, which is much better than validate it manually.

## Security

We try our best to avoid learned security vunerables. They are:

* Password encrypting: password store in database is hashed using SHA hash function. Even if attacker successfully obtain database, they still don't know the account password and can not damage the system right away.
* SQL injection: Phalcon PHP support prepared statement and thus prevent malicious SQL script being injected. Most input field are also validated before sending to the server to process.
* XSS: Content of document are all escaped. This preventing user to inject malicious script through their document.

## Advanced:

### Ajax searching

We apply ajax to searching function of document listing pages.

jQuery(document).ready(**function**() {

**var** is\_searching = **false**;

jQuery('#search\_documents').on('keyup', **function**() {

**var** search\_link = **this**.form.action;

**if** (is\_searching == **true**) {

**return**;

}

is\_searching = **true**;

jQuery.post(search\_link, jQuery(**this**.form).serialize(), **function**(data) {

jQuery('#documents\_table').replaceWith(jQuery(data).find('#documents\_table')) ;

is\_searching = **false**;

});

});

});

With this function, whenever user type any key into searching textbox, the textbox content will instantly be sent to the server. The table display the result also immediately updated.

### PHP Excel

PHP Excel is used to help reading xls/xlsx file. It comes with a calculation engine, but we weren't able to make use of it.

However, PHP Excel doesn't actually help rendering xls/xlsx file. It read data, but rendering is totally up to us, which is quite difficult to simulate Excel using only HTML and javascript.

# Result

## Database

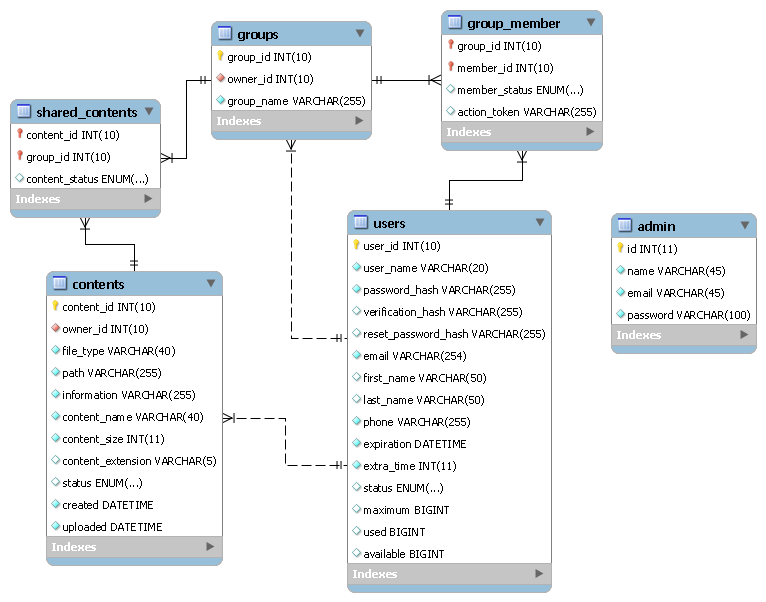
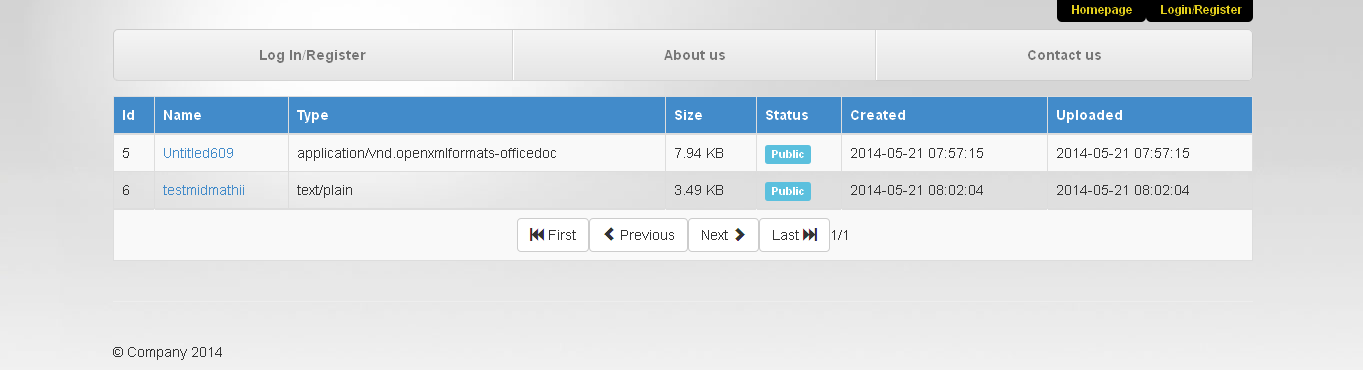


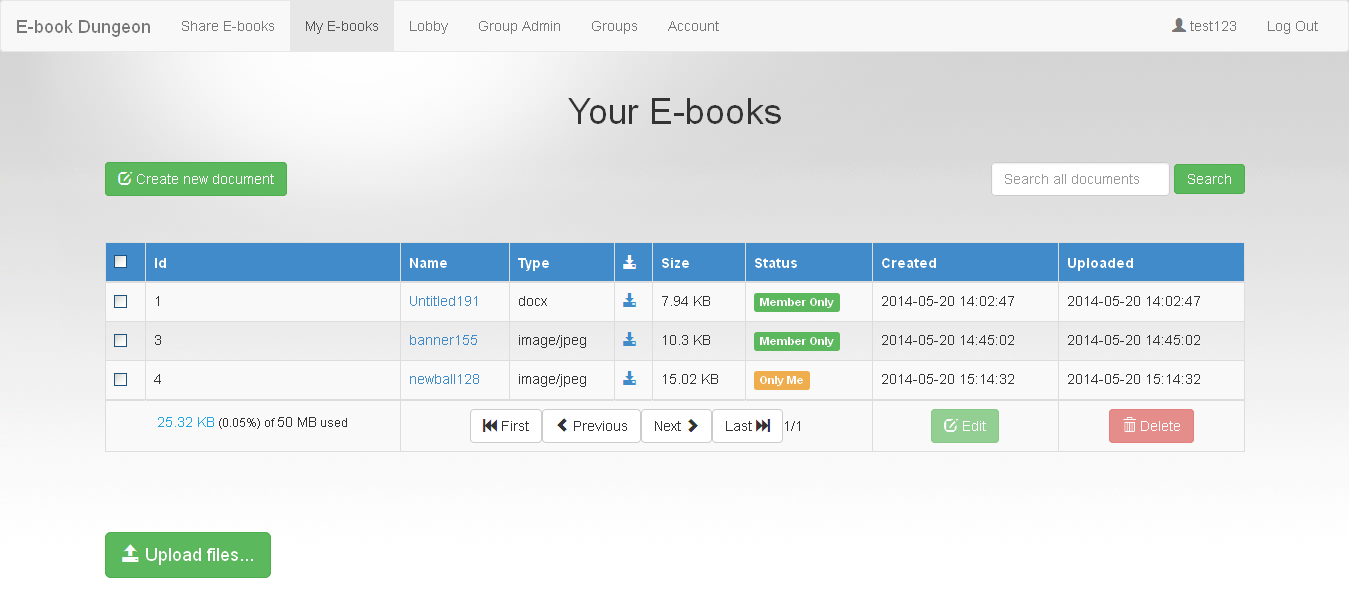
Figure 6 : Database structure

## Interfaces

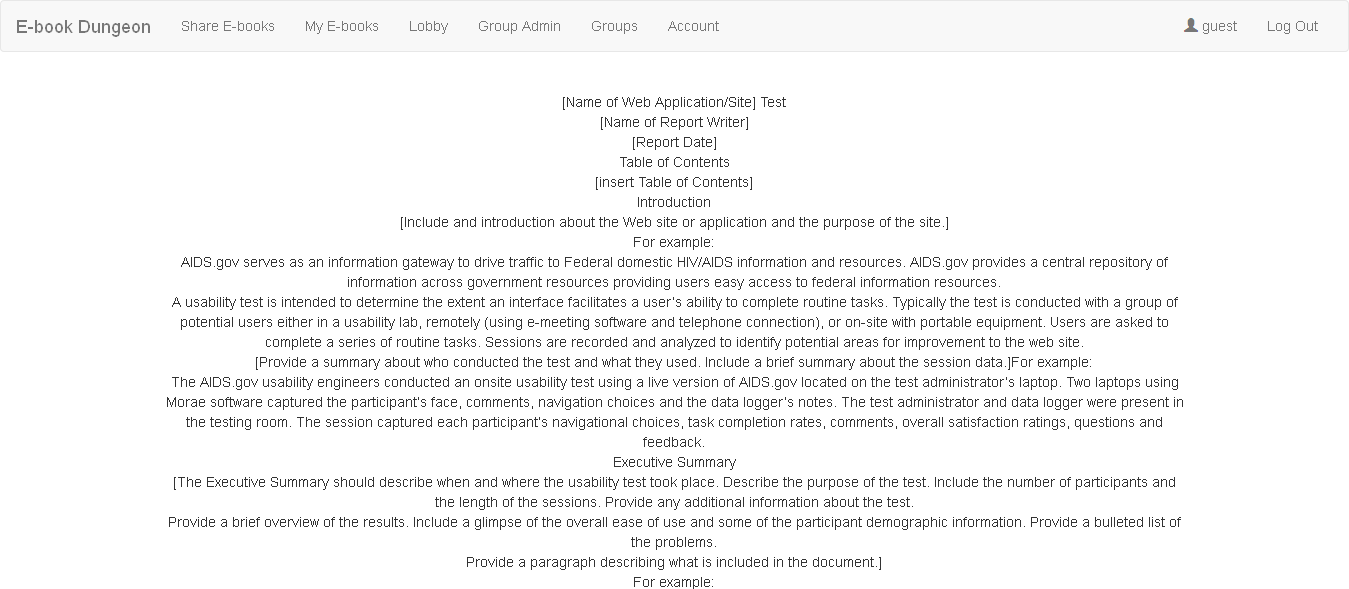
Using bootstrap, we tried to create a simple, friendly-looking interface.



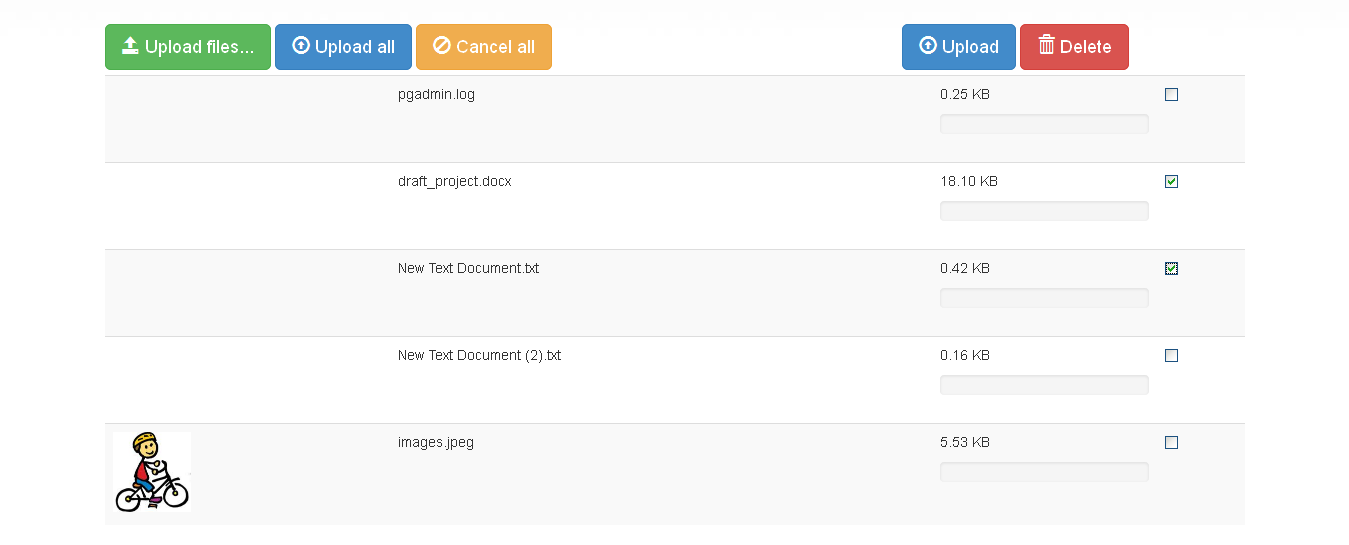
Picture 1: Index page



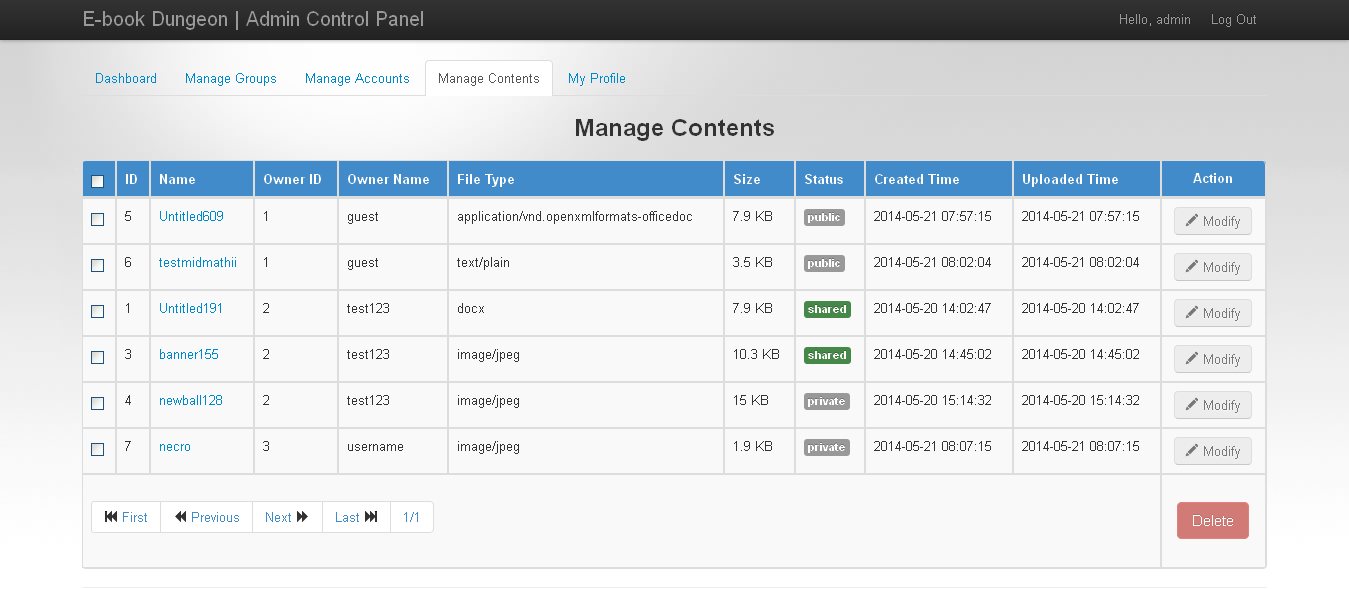
Picture 2: Personal Ebook page



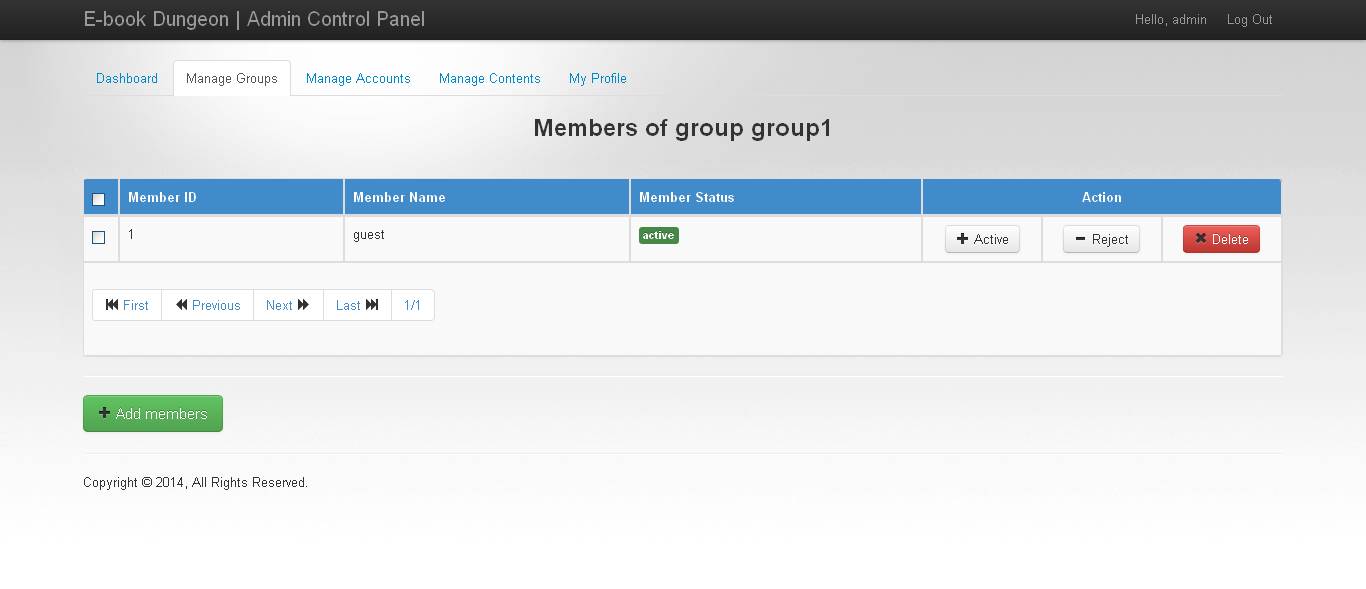
Picture 3: Viewing document by browser



Picture 4: Uploading



Picture 5: Admin's manage contents page



Picture 6: Admin's manage group member page

## Evaluation

### Advantages

Thanks to Phalcon PHP, server processing is very fast.

The website also has no security issue detected (at the moment), as SHA hashed password, anti SQL injection, anti XSS were implemented.

The website UI is clean-looking and easy to use.

Successfully using some advanced component or technique such as MVC model, ajax, Bootstrap, PHP Excel, etc.

### Disadvantages

Some of the intended function were not completed.

Viewing document can only display plaintext, due to custom docx rendering script. The same to xls/xlsx document can only display as table.

# Conclusion

The website has most of its essential functions working. Some functions of initial requirement were not completed, for example, "Edit document atrributes", or "Download all document from a group", etc. due to either time constraint or technical difficulties. However, the develop team has successfully applied web programming techniques learned through the course into this website.

In the future, this website can be further developed to be actually used as a commercial product. After finishing above uncompleted functions, some new features may be added, such as editing document online, online payment, document renting, etc. Also, security and interface still have lot of room to improve.

# Reference

[1] <http://docs.phalconphp.com/en/latest/>

Official Phalcon PHP documentation.

[2] <https://github.com/PHPOffice/PHPExcel/wiki/User%20Documentation>

Official Phalcon PHP Excel.

[3] <http://stackoverflow.com/>

Various helpful Q&A.

[4] <http://www.tutorialspoint.com/bootstrap/bootstrap_quick_guide.htm>

Quick tutorial to Bootstrap.