

ASSESSMENT AND INTERNAL VERIFICATION FRONT SHEET (Individual Criteria)

Course Title	MCAST BSc. Software Year 2		Lecturer Name & Surname	Thomas Gatt	
Unit Number & Title	IFSFT-506-1612: Server Side Scripting				
Assignment Number, Title / Type	2 – Develop a web application for a given scenario - Home				
Date Set	18/12/2020	Deadline Date	22/01/2021		
Student Name		ID Number		Class / Group	

<input type="checkbox"/>	Student's declaration prior to handing-in of assignment: ❖ I certify that the work submitted for this assignment is my own and that I have read and understood the respective Plagiarism Policy		
<input type="checkbox"/>	Student's declaration on assessment special arrangements (Tick only if applicable) ❖ I certify that adequate support was given to me during the assignment through the Institute and/or the Inclusive Education Unit.		
<input type="checkbox"/>	❖ I declare that I refused the special support offered by the Institute.		
Student Signature:		Date :	

Assessment Criteria	Maximum Mark	Mark Achieved
KU6 Show understanding of the validation techniques available in a framework	5	
KU7 Show understanding of model commands which can be used to interact with data in a database	5	
AA2 Examine various methods of debugging	7	
AA3 Develop a web application using the features of a server side framework	7	
AA5 Create table associations to be used in an application using the features of a server side framework	7	
SE1 Develop and design an application programming interface using the features of server side framework.	10	
SE2 Design URLs in a SEO friendly way	10	
SE3 Modify a web application in order to make it safer using the framework's security features	10	
Total Marks	61	

Assessor's feedback to student
<i>(If necessary, use reverse side of page for IV feedback on assignment brief / sample of assessment decisions)</i>

	Name & Surname	Signature	Date
Internal Verifier : Approval of <u>assignment brief</u>	Gerard Said	G. Said (ELEC SIG)	
Lecturer / Assessor : Issue of results and feedback to student	Thomas Gatt	T. Gatt (ELEC SIG)	
Internal Verifier : Approval of <u>assessment decisions</u> (Sample)			
Learner's signature upon collection of corrected assignment.			

Develop a web application for a given scenario

Assignment 2

Assignment Guidelines

Read the following instructions carefully before you start the assignment.

If you do not understand any of them, ask your lecturer.

- The assignment coversheet should be the first sheet in your assignment. Moreover, the coversheet should be fully completed with all the necessary details.
- You are required to use the **CakePHP v4** framework for this assignment. **'Bake' or similar CLI commands are not allowed.**
- All text/code *must be properly referenced*. In the absence of proper referencing, the assignment will be regarded as plagiarised.
- **Copying is strictly prohibited and will be penalized** in line with the College's disciplinary procedures.
- The deadline should be specified by your lecturer. When the deadline is due, you shall hand in 2 deliverables:
 - A zip file containing the application, all source code and a database SQL file.
 - You are also required to create a word document and **include all the code** from your controllers, models and views in **plain text**. You are to submit this document on the specified platform on the same deadline. Your lecturer will forward you details in order to submit your assignment before the deadline.
- **Marks assigned to each criteria will be determined by the interview performance. i.e. You must be able to answer any questions related to each criterion. You might not be awarded marks associated to a criterion if you fail to answer any questions correctly.**

Contents

Assignment Guidelines.....	3
Assignment Criteria	5
Introduction	6
Tasks.....	7
Assignment Rubric	11

Assignment Criteria

Criteria	Description	Marks
AA2	Examine various methods of debugging	7
AA3	Develop a web application using the features of a server side framework	7
SE1	Develop and design an application programming interface using the features of server side framework.	10
KU6	Show understanding of the validation techniques available in a framework	5
SE2	Design URLs in a SEO friendly way	10
KU7	Show understanding of model commands which can be used to interact with data in a database	5
AA5	Create table associations to be used in an application using the features of a server side framework	7
SE3	Modify a web application in order to make it safer using the framework's security features	10

Introduction

File Solutions LTD just hired you to create a platform that should allow users to share files on the web. They named this concept 'ShareIT'. The idea is very similar to how Wetransfer works: to allow users to upload and share files with each other. Users should be able to upload their files to this platform in order to make them visible to their friends or to the public.

File Solutions LTD, just like any company, needs this to be developed as fast as possible yet they want it to be secure and scalable. Therefore, you are being requested to make use of CakePHP version 4 as a PHP framework. This not only makes development faster but also safer and easier to maintain.

The users should be able to:

- Register/Login with Facebook and/or Google
- Upload an file (accessible to Everyone (public), all your friends or specific friends)
- View files available (homepage)
- Add a comment to a file
- Delete own comments
- Delete own file entries
- Edit own file entries
- Mark files as available or unavailable
- Add/Remove Friends

The web app should also cater for two different roles within the system:

- User
 - All the above
 - Admin
 - All the above
 - View all files available
 - View all files added by a user
 - Delete comments on a file entry
 - Delete any file entry
-

Tasks

File Solutions LTD have written this user requirement specification document which should be very easy to follow. The web application is divided into various pages, each page responsible for its tasks.

1. Homepage (Add new File AND List files)

- a. Ability for any user (even non authenticated users) to add file entries
 - i. File title (required) *for example: Jake's Photos*
 - ii. File description (required, minimum of 20 characters)
 - iii. One file attachment (required, **allowed file types**: .zip or .rar only)
 - iv. To: *Everyone (Public)* OR *All Friends* OR *Specific Friends*
 - The *Everyone* option is assumed if the user is not authenticated. This option makes the file visible to all users (including non-authenticated ones)
 - If the *Friends* option is selected, the file will be available to the user's friends only. Authentication is required for this feature.
 - If the *Specific Friends* option is selected, the user needs to select from a list the friend(s) that he/she wants to provide access to (*You might want to use the chosen.js plugin for this*). Authentication is required for this feature.
- b. In this page you should also display the files (not marked as unavailable) that the user has access to:
 - i. Files shared to *Everyone* (these should be displayed irrespective of the login status).
 - ii. Files shared to *Friends* that the authenticated user has access to.
 - iii. You are expected to show **one whole list**, not two separate ones.
- c. Each file entry should show the following details:
 - i. File title
 - ii. Uploader name and surname (if it was added by an authenticated user, otherwise show 'Anonymous')
 - *Admin only*: The Uploader needs to be linked to a page that shows all file entries added by the user (see point 2c)
 - iii. File download link button (only if the file is marked as 'available' – by default it should be available unless otherwise marked as unavailable by the uploader – see point 3e)
 - iv. Count of comments on the file entry (if any otherwise show 0)
 - v. Date added shown in this format: 8th December 2020 15:20
- d. The file name should link to `/file/[random-alphanumeric-string]/[file-name]` (see point number 3 for more details)

2. List of Users + Friend/Unfriend

- a. Authenticated users can view list of users (first name and last name) registered with the system
- b. The menu item 'User List' should be visible in the menu if the user is authenticated.
- c. *Admin only*: Clicking on the user should allow admins to view list of file entries added by the user.
- d. Friend/Unfriend button next to each user
 - i. There is no need for the user to accept the friend request.
 - ii. The friendship should be **two-way**. This means that when user A adds user B as a friend, both users become friends with each other and they can share file entries between them.
 - iii. The Friend button **should only be shown** if the authenticated user is not a friend with the user
 - iv. The Unfriend button **should only be shown** if the authenticated user is a friend with the user

3. File page & Comments

- a. This page should show the details as defined in point 1c as well as the file description
- b. It can only be accessed by users who access to it. Possible cases include:
 - i. The role of the authenticated user is 'Admin' so he/she has access to all file entries.
 - ii. The user accessing the file entry is the owner of it.
 - iii. The file was shared with the authenticated user because he/she is a friend of the uploader.
 - iv. The privacy setting for the file was set to 'Everyone'
- c. Comments
 - i. Only authenticated users can add comments to the file entry
 - ii. The following should be stored:
 - Comment text
 - Comment owner (user_id)
 - Date and Time
 - iii. List of comments should be shown after the file details. Each comment should consist of:
 - Comment owner (name and surname). This should be linked as defined in point 1c ii
 - Date added shown in this format: 8th December 2020 15:20
 - Comment text
 - Delete button (see next point)
 - iv. Only the admin or comment owner can delete the comment. The delete button should be visible to admins or comment owners only.

- d. Link to this page should be in the following format: */file/[random-alphanumeric-string]/[file-name]*
 - i. *Example: /file/uwiehuw882/jakes-photos*
 - ii. See the term 'slug' on the cookbook to format 'Jake's photos' to 'jakes-photos'
- e. 'Mark as unavailable' or 'Mark as available' button should be visible if the authenticated user is the uploader. The status (Available/Unavailable) should be shown in this page too.
- f. 'Edit' button should be visible if the authenticated user is the uploader.
- g. 'Delete' button should be visible if the authenticated user is the uploader or an admin.

4. Register/Login & Logout

- a. Implement Firebase's Facebook and/or Google Login for Web
 - i. Save the user's credentials in your table if they registered for the first time
 - ii. Otherwise, retrieve data from the table and authenticate the user
- b. Logout (should be visible if the user is logged in only)
- c. You still need to implement the manual Register/Login to test your system with different users unless you want to create multiple Facebook accounts. This will be covered during the sessions.
- d. You should at least store the following data:
 - i. First Name
 - ii. Last Name
 - iii. Email

5. My List / Files by User

- a. 'My List' menu item should be visible if the user is logged in
- b. It should contain a list of file entries (file name only) added to the website by the uploader. This list should include all file entries (available and unavailable). Each file entry should be linked as described in point 3d.
- c. Users not authorised to view the file entry should not be able to access them (**even if the direct link is used**) – see point 3b.
- d. *Admin only:* View list of file entries added by a user - see point 2c.

6. API

File Solutions LTD would also like you to provide easy access to the logic you've created in your application so that others can make use of this data. You should make use of CakePHP's REST function to expose your API via REST.

- a) Get list of file entries and their comments added by a **specific user** (GET)
- b) Delete **specific** file entry (DELETE)
- c) **Add** a new file entry including attachment (POST)

7. Debugging

While CakePHP's core Configure Class settings can really help you see what's happening under the hood, there are certain times that you'll need to log data to the disk in order to find out what's going on.

Make use of CakePHP's Log class to log specific events related to files and its comments. The logs should include both successful and failed actions. By creating **specific log scopes**, make use of the correct logging levels and log the following events:

1. New file
2. Download a file (when the button to download the file is clicked)
3. New Comment
4. Logins
5. Logout

Details to log

- Date & Time
- User (if logged in)
- IP Address
- Message
- User/File ID if applicable
- Any other information you believe is important to log

Design

You are expected to present a simple yet neat design. Feel free to use Bootstrap. Please **refrain** from using the `<table>` tag to display lists. Use cards or lists instead.

--End of Assignment--

Assignment Rubric

Criteria	Description	Marks	Awarded
AA2	Examine various methods of debugging		
	Log Scope for the FilesController set up	2	
	Correct use of logging levels used in Log Scope	2	
	Correct use of general debug logging throughout application	2	
	'Mark file as unavailable' implemented correctly and as described	1	
AA3	Develop a web application using the features of a server side framework		
	'New File' implemented correctly and as described	3	
	Ability to share a file with Everyone as described	2	
	'Delete Comment' action implemented correctly and as described	2	
SE1	Develop and design an application programming interface using the features of server side framework.		
	Implement a REST API method to get list of files and their comments for a specific user	3	
	Implement a REST API method to delete a specific file entry	3	
	Implement a REST API method to add a new file entry (including attachment)	4	

Criteria	Description	Marks	Awarded
KU6	Show understanding of the validation techniques available in a framework		
	Correct use of <i>Validator</i> Object as described	1	
	Ability to Delete a file as described	1	
	Ability to Edit a file as described	1	
	File comments can be added and viewed	2	
SE2	Design URLs in a SEO friendly way		
	Correct use of the <i>Router</i> & <i>Inflector</i> class for accessing a specific file	2	
	'My List' implemented as described	2	
	Admin actions implemented as described	2	
	File entry shared correctly with Friends as described	4	
KU7	Show understanding of model commands which can be used to interact with data in a database		
	'New file' – file uploaded, checked for extension and stored correctly	2	
	Homepage implemented as described	3	
AA5	Create table associations to be used in an application using the features of a server side framework		
	Correct association types used in Users and Files controllers (1 mark each)	2	
	Correct use of <i>contain()</i> ORM query function	1	
	File entries in 'Home' and 'File' pages shown as described and according to access levels. Users are not allowed to access a specific file entry if they have no access to it.	4	
SE3	Modify a web application in order to make it safer using the framework's security features		
	Firestore Google/Facebook Login implemented correctly	5	
	Correct use of the <i>Authenticator</i> component	1	
	Ability to Friend/Unfriend users. Implemented as described.	4	

Total Mark (out of 61)	
Comments	
