2014

Software Engineering Project

Team F

**ONLINE RESOURCE PORTAL**

**TEAM MEMBERS**

Navjeet Singh Sandhu

)

Team Contact

(

Bashanta Dahal

Dirghayu Lakhey

Himmat Shrestha

Shankar Giri

**Lecturer:**

Dr. Torab

Torabi

**Supervisor:**

Suzan Majeed

System Maintenance Document submitted to La Trobe Universit

y,

Department of Computer Science and Computer Engineering

A

s a part of Software Engineering Project

**Copyright © May 2014**



# Copyright and Warranty Information

The information contained in this document is subject to change without notice.

The vendor assumes no accountability or liability for any errors that may appear in this document.

No warranty or representation, either expressed or implied, is made with regard to the quality, accuracy, or fitness of any part of this document. The manufacturer shall not be liable for any direct, indirect, special, incidental or consequential damages arising from any defect or error in this document or product. Product names appearing in this document are for identification purposes only.

Trademarks and product or brand names appearing in this document are the property of their respective owners.

Copyright on all material and all intellectual property residing in material in this document, including (but not limited to) graphics, book titles, text, layout, logos, trademarks and samples is owned by Quantum Victoria unless otherwise indicated. No reproduction, modification, translation, transmission, distribution, or use for commercial purposes of this document without written approval from Quantum

Victoria is permitted.

This document contains material protected under International Copyright Laws. All rights reserved.

These terms are subject to the conditions prescribed under the Australian Copyright Act 1968.

Online Resource Portal

2014 Industry Project

Department of Computer Science & Computer Engineering

La Trobe University

Bundoora, Victoria, Australia

# Table of Contents

Copyright and Warranty Information .................................................................................... 1 Table of Figures .................................................................................................................. 4

1.0 Introduction............................................................................................................... 6

1.1 System Overview ............................................................................................................ 6

1.2 Audience Description ..................................................................................................... 7

1.3 Applicability Statement .................................................................................................. 7

1.4 Purpose Statement ......................................................................................................... 7

1.5 Document Usage Description ....................................................................................... 8

1.6 Conventions..................................................................................................................... 8

1.7 Change Log ..................................................................................................................... 9

2.0 Software Design Scope ...........................................................................................10

2.1 Major Software Functions ........................................................................................... 10

2.2 Major Design Constraints and other Requirements ................................................ 10

3.0 Reference Documents .............................................................................................11

3.1 Existing Software Documentation .............................................................................. 11

3.2 System Documentation ............................................................................................... 11

3.3 Vendor Documentation ................................................................................................ 11

3.4 Other Documentation ................................................................................................... 13

4.0 User Stories .............................................................................................................14

4.1 User Story Dictionary ................................................................................................... 14

4.2 Iterative User Story Documents ................................................................................. 16

5.0 Object-Oriented Design ...........................................................................................48

5.1 High Level System Architecture ................................................................................. 48

5.2 High Level Package Diagram and Components ...................................................... 49

5.3 Use Case Analysis ....................................................................................................... 50

5.4 Domain Model and Class Diagram ............................................................................ 58

5.5 Establishment of the Database Objects and Data Access Strategy ..................... 61

5.6 Sequence diagram ....................................................................................................... 62

5.7 Object Dictionary .......................................................................................................... 72

6.0 Software release report ...........................................................................................74

6.1 Usability Test Report .................................................................................................... 74

6.2 System Test Report ..................................................................................................... 76

7.0 Additional User Interface Design .............................................................................78

GLOSSARY .......................................................................................................................80

INDEX ................................................................................................................................82

# Table of Figures

Figure 1: FIN Diagram - Login to the system ....................................................................................... 17

Figure 2: FIN Diagram - Add Resource ................................................................................................ 18

Figure 3: FIN Diagram - Add Post ........................................................................................................ 19

Figure 4: FIN Diagram - Add Tag .......................................................................................................... 20

Figure 5: FIN Diagram - Edit Resource................................................................................................. 21

Figure 6: FIN Diagram - Edit Post ........................................................................................................ 22

Figure 7: FIN Diagram - Edit Tag .......................................................................................................... 23

Figure 8: FIN Diagram - Delete Resource ............................................................................................ 24

Figure 9: FIN Diagram - Delete Post .................................................................................................... 25

Figure 10: FIN Diagram - Delete Tag ................................................................................................... 26

Figure 11: FIN Diagram - Like/Share Resource .................................................................................... 27

Figure 12: FIN Diagram - Submit Enquiry ............................................................................................ 28

Figure 13: Wireframe - Login Screen ................................................................................................... 43

Figure 14: Wireframe - Contact Admin ............................................................................................... 44

Figure 15: Wireframe - Admin Approval ............................................................................................. 44

Figure 16: Wireframe - Admin Resource View ................................................................................... 45

Figure 17: Wireframe - User Resource View ....................................................................................... 45

Figure 18: Wireframe - Drag and Drop ............................................................................................... 46

Figure 19: Wireframe - Add New Tags ................................................................................................ 46

Figure 20: Wireframe - Tagged Images ............................................................................................... 47

Figure 21: System Architecture Diagram ............................................................................................ 48

Figure 22: Use Case Model .................................................................................................................. 50

Figure 23: Class Diagram ..................................................................................................................... 58

Figure 24: Entity Relationship Diagram ............................................................................................... 61

Figure 25: Sequence Diagram – Login ................................................................................................. 62

Figure 26: Sequence Diagram – View Resources ................................................................................ 63

Figure 27: Sequence Diagram – Like Resources .................................................................................. 64

Figure 28: Sequence Diagram – Share Resources ............................................................................... 65 Figure 29: Sequence Diagram – Create Resource ............................................................................... 66

Figure 30: Sequence Diagram – Edit/Delete Resource ....................................................................... 67

Figure 31: Sequence Diagram – Add Photos to resource ................................................................... 68

Figure 32: Sequence Diagram – View Photos in Resource .................................................................. 69

Figure 33: Sequence Diagram – Submit Enquiry ................................................................................. 70

Figure 34: Sequence Diagram – Logout ............................................................................................ 705

Figure 35: Confirm Delete Popup Window ......................................................................................... 78

Figure 36: Error message while adding new resource with blank name, description and url field .... 78

Figure 37: Error message while adding new tag with blank title and description .............................. 79 Figure 38: Error message while creating new post with blank title and body field ............................ 79

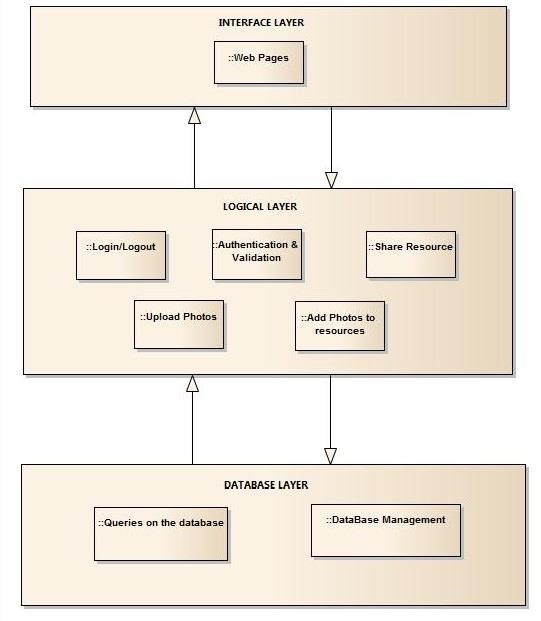
## 1.0 Introduction

### 1.1 System Overview

The Online Resource Portal will be a one stop portal to view, access and use the online resources. The portal can be accessed by the users through internet on their web-enabled devices like Laptop/Desktop, Phone and Tablet. Once logged in, the users will be taken to the homepage containing the various other functions of the system like viewing resources, uploading resources, uploading photos, completing assignments etc.

The system is divided into three layers:

1. **Database Layer**, which handles all the queries made on the database and manage the data within the database
2. **Logical Layer**, which comprises of all the logical tasks that a user can carry out on the system
3. **Interface Layer,** which features the interface which is being used for interaction between user and the system



### 1.2 Audience Description

The various users of our systems will be:

* **Admins** refer to the Quantum Victoria staffs whose major role will be updating and maintaining the system by adding various resources and approving them. They will need to know the working structure of the system and its functions. In our system, we have used a Boolean value as ‘is\_admin’ in our database to determine whether a user is admin or not. If ‘is\_admin’ = true then the user is an admin otherwise the user is a general user.

* **Students** who will be viewing the various resources like wiki-pages, external links, videos and photos. The will also have the ability to upload photos and complete various tasks and assignment. These users should have the basic knowledge of navigating through web pages.

* **Teachers** with the abilities to view resources and also create assignments and assign them to their students. They also need to have the basic knowledge of navigating through web pages

**1.3 Applicability Statement**

The system is designed to run on any system that has a browser.

### 1.4 Purpose Statement

The major purpose of this project is to provide a one stop portal to view, access and use the online resources created by the staff at Quantum Victoria. Quantum Victoria has various digital online resources scattered across various websites. This portal will centralize all these resources.

**The major applications of the portal are**:

1. To assist the learning process of the students by making various online resources available on a single portal in an interactive way.
2. Enhancing the knowledge of students by completing tasks and assignments assigned to them and hence broaden their horizon
3. Social interaction with other users by uploading photos to various resources and sharing those resources across various social websites such as Facebook and Twitter.
4. Help teachers can create various assignments and assign them to their students and keep a track of it.
5. Assist parents in viewing the progress of their children.

### 1.5 Document Usage Description

The Document contains the following section

1. **Introduction**

It gives a brief introduction to the project and its various purposes and its intended users.

1. **Software Design Scope**

It covers the major software functions, its benefits and the various design constraints.

1. **Reference Documents**

This section consists of the various supporting documents like the project description document and various vendor documents.

1. **User Stories**

This section describes all the completed user stories related to the project and also the user stories in progress. In addition it also provides unit testing and the wireframes.

1. **Object Oriented Design**

It covers the blueprint for the implementation of the various concepts developed. It contains the Architectural Diagram along with the Class Diagram and The ER diagram and the Sequence Diagram.

1. **Software Release Report**

It outlines the procedure for usability testing and all system testing reports.

1. **Additional User Interface Design**

It contains any interfaces discovered during the design stage like warning screen, error message and help window.

### 1.6 Conventions

In order to prepare System Maintenance Document, we have used some of the design conventions. We have used blue color and different font size for headings to uniquely identify them. We also used the table of content provided by Microsoft Word to easily find and navigate to the required heading.

Each figure was captioned properly with figure number and description of the figure and list of figures was added right after the table of content to find out the figure easily.

The major part of this document is written with Arial font. While caption of figure is in Calibri font and project title in the cover page is written in Times New Roman font.

### 1.7 Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Changes | Made by |

First Release

1.0 19 May 2014  Cover Page and Document Layout Himmat Shrestha

 Table of Contents and Table of Figures

Added System Overview and Audience

1.1 22 May 2014 Dirghayu Lakhey

Description

1.2 24 May 2014 Added Software Design Scope Shankar Giri

1.3 26 May 2014 Added Use Case Model and Use Cases Navjeet Sandhu

1.4 27 May 2014 Added Integration Testing Shankar Giri

1.5 28 May 2014 Added Flow of Interaction Diagram Himmat Shrestha

Added Class Diagram and Sequence

1.6 29 May 2014 Navjeet Sandhu

Diagram

1.7 29 May 2014 Added ER Diagram Navjeet Sandhu

Added Unit Testing and Usability Test

1.8 29 May 2014 Bashanta Dahal

Report

1.9 30 May 2014 Added Additional User Interface Design Himmat Shrestha

2.0 30 May 2014 Refined Class Diagram Navjeet Sandhu

2.1 30 May 2014 Added High Level Package Diagram Navjeet Sandhu

2.2 30 May 2014 Added INDEX Dirghayu Lakhey

2.3 30 May 2014 Changed System Overview Content Navjeet Sandhu

## 2.0 Software Design Scope

### 2.1 Major Software Functions

#### Admin

* Approve photos uploaded by general users
* Can edit tags, resources and galleries

#### General Users

* Upload photos and tag them
* Search resources and photos
* Provides the facility of data entry for resources, galleries and tags in hand-held devices

#### System

* Captcha implementation for Human Error checking
* Stores created resources, tags in the database
* Stores added photos in the database
* Resources are sharable in social medias
* Resources can be saved in PDF format
* Resources can be printed

### 2.2 Major Design Constraints and other Requirements

* Users should be logged in to view system contents **Constraints:**
* WordPress API to authenticate users
* Unix hosted LAMP server

## 3.0 Reference Documents

### 3.1 Existing Software Documentation

The existing documentation consists of the documentation provided by the client Quantum Victoria as a description of the project.

### 3.2 System Documentation

There are no system documents applicable as website is not being embedded in any existing system.

### 3.3 Vendor Documentation

The various vendor documents are listed below.

#### PHP

PHP is a server side scripting language for web development. The PHP code can be mixed with HTML code or can be used in combination with various web frameworks.

The Manual for PHP can be found here:

<http://www.php.net/manual/en/>

##### Laravel

Laravel is an open source PHP web application framework to assist the development of MVC web applications. Right now, it is the most popular PHP framework.

The API for Laravel can be found here:

<http://laravel.com/api/4.1/>

##### MySQL

MySQL is the second most used open source relational database management system (RDBMS). It is used to create and manage databases, build database structures, backup data, inspect status and work with data records.

The Manual can be found here:

<http://dev.mysql.com/doc/>

##### Word Press

Word Press is a free and open source blogging tool and a content management tool based on PHP and MySQL which runs on web hosting services.

The manual for Word Press can be found here:

<http://codex.wordpress.org/Main_Page>

The manual for WordPress authentication can be found here: <https://bitbucket.org/hampel/wordpress-auth-driver-for-laravel>

##### Facebook – Share and like plug-in

Facebook is a social networking site where the users can interact with their friends and family by updating status and sharing photos. For our portal, we have used the plug-in for sharing and liking contents on Facebook which is provided by Facebook.

The plugin can be found here:

<https://developers.facebook.com/docs/plugins/share-button/><https://developers.facebook.com/docs/plugins/like-button>

##### BlueImp jQuery-File-upload

It is a jQuery plug-in for uploading single or multiple photos/files on a website.

The manual can be found here: <https://github.com/blueimp/jQuery-File-Upload/wiki>

##### Twitter Bootstrap

It is a free collection of tools for creating websites and contains HTML and CSS based design templates for typography, forms, buttons and other interfaces.

<http://getbootstrap.com/css/>

##### Font Awesome

It is a website which provides a lot of user-friendly icons that can be used for various functions.

<http://fortawesome.github.io/Font-Awesome/get-started/>

### 3.4 Other Documentation

This is the first documentation of this system. There is no other documentation yet made for this system.

## 4.0 User Stories

In software development and product management, a user story is one or more sentences in the everyday or business language of the end user or user of a system that captures what a user does or needs to do as part of his or her job function.[[1]](#footnote-13) As a part of doing project in agile, it is important to write User Stories (US) very effectively. User Stories are generally written in the form of:

“As <role/who><when><where>, I want <what/goal> because <why/benefit>.”

JIRA provided the agile environment where User Stories were created and assigned to each team members. Each member would focus on at least one User Story at a time and start working on it. Agile focuses more on coding aspects such as code clarity while writing it by the project developers.

### 4.1 User Story Dictionary

#### Role: Student

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPTION** | **ACCEPTANCE CRITERIA** | **STORY POINTS** | **CONTRIBUTER** |
| As a student, I want to login to the portal to use the various features of the website | Wordpress authentication should be fulfilled. | 5 | Navjeet Sandhu |
| As a student, I want to view resources because I want to learn about various subject in an interactive manner | The resources should be presented in an interactive design and be kids friendly. | 3 | Bashanta Dahal |
| As a student, I want to  Like/Share resources because I want to interact with my friends by sharing resources on social networking websites | The resources should be shareable on Facebook. | 3 | Dirghayu Lakhey |
| As a student, I want to print resources because I want to keep that for record | The resources should be printed in PDF format. | 3 | Himmat Shrestha |
| As a student, I want to upload photos because I want to sharing it with everyone | The user should be able to drag and drop multiple photos into the browsers and upload them. | 5 | Bashanta Dahal |
| As a student, I want to tag photos so that one can find the photo easily using tag name | The users should be able to use only the available tags. | 3 | Shankar Giri |

#### Table 1: User Story Dictionary for Students

##### Role: Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPTION** | **ACCEPTANCE CRITERIA** | **STORY POINTS** | **CONTRIBUTER** |
| As an admin, I want to create resources so that the resources can be viewed by other users for their educational purposes | The resources should have title, description and links associated with it | 3 | Bashanta Dahal |
| As an admin, I want to create tags so that the users can tag photos | Tags should be related to resources not random | 3 | Shankar Giri |
| As an admin, I want to receive email notifications because I want to know when a user uploads a photo | Notifications should be received through email | 3 | Navjeet Sandhu |
| As an admin, I want to approve photos in order to prevent users from uploading unrelated and offensive photos | The photos should be appropriate | 3 | Himmat Shrestha |
| As an admin, I want to share photos so that photos can be shared on social networking sites | The photos should be approved by the admin and available to share. | 3 | Dirghayu Lakhey |

**Table 2: User Story Dictionary for Admins**

### 4.2 Iterative User Story Documents

***4.2.1 User Story Definition***

####  Login to the website using Wordpress authentication

The first screen every user will see while visiting the website will be the login screen. The login interface uses wordpress authentication for successful login to the system. After the successful login, a user can do the desired tasks that are available to them.

####  View different resources

After successful login to the system, users can view different resources available in the system.

####  Like/Share Resources

A user can like and/or share resources on social networking sites like Facebook, Twitter, LinkedIn and Google Plus.

####  Upload Photos

Users can upload photos that are related to the resources. After successful upload and approval by admin, they can share photos to social networking sites.

* **Tag Photos**

After uploading photos, users can tag the photos to any of the available tags.

* **Create resources – admins only**

Only admins of Quantum Victoria can create the resources.

* **Create tags – admins only**

Only admins of Quantum Victoria can create the tags.

* **Print resources**

The resources can be exported to PDF format and can be viewed offline.

####  Drag and Drop

Users can drag and drop one or more than one photos at a time using drag and drop feature. The simple upload feature by clicking browse button is still there.

####  Email notification

If a user uploads new photo, it will be first in the list of pending photos. The admins need to approve if the photo uploaded is related or not to any of the resources. An email notification will be sent to admins of Quantum Victoria when a user uploads a photo and the admins can approve or reject them.

##### *4.2.2 Flow of Interaction Diagram*

User Type

Admin

User Type

Student

User Type

Teacher

Login to website

Display Home Page

Enter

Username and Password

Verify

Details

Display suitable

message

Wrong

Credentials

Correct

Credentials

**Figure 1: FIN Diagram - Login to the system**

Add Resource

Return to List of Resources

Enter

Title

,

Description

and Link of Resource

Verify

Details

Display suitable

message according to

the error

Incorrect

Input

(

s

)

Correct

Inputs

Add Resource to

database

Clicks

Submit

Resets all the field

Clicks

Cancel

Clicks

Reset

**Figure 2: FIN Diagram - Add Resource**

Add Post

Return to List of Posts

Enter

Title and Body of Post

Verify

Details

Display suitable

message according to

the error

Incorrect

Input

(

s

)

Correct

Inputs

Add Post to database

Clicks

Submit

Resets all the field

Clicks

Cancel

Clicks

Reset

**Figure 3: FIN Diagram - Add Post**

Add Tag

Return to List of Tags

Enter

Name and Description of Tag

Verify

Details

Display suitable

message according to

the error

Incorrect

Input

(

s

)

Correct

Inputs

Add Tag to database

Clicks

Submit

Resets all the field

Clicks

Cancel

Clicks

Reset

**Figure 4: FIN Diagram - Add Tag**

Edit

Resource

Return to List of

Resources

E

dit

Title

,

Description

and

/

or

URL

of

Resource

Verify

Details

Display suitable

message according to

the error

Incorrect

Input

(

s

)

Correct

Inputs

Edit the contents of

the Resource

In

database

Clicks

“Submit”

Resets all the field

Clicks “Cancel”

Clicks “Reset”

**Figure 5: FIN Diagram - Edit Resource**

Edit

Post

Return to List of Posts

E

dit

Title and

/

or

Body of Post

Verify

Details

Display suitable

message according to

the error

Incorrect

Input

(

s

)

Correct

Inputs

Edit the contents of

the

Post

In

database

Clicks

“Submit”

Resets all the field

Clicks “Cancel”

Clicks “Reset”

**Figure 6: FIN Diagram - Edit Post**

Edit

Tag

Return to List of

Tags

E

dit

Name and Description

of

Tag

Verify

Details

Display suitable

message according to

the error

Incorrect

Input

(

s

)

Correct

Inputs

Edit the contents of

the Tag

In

database

Clicks

“Submit”

Resets all the field

Clicks “Cancel”

Clicks “Reset”

**Figure 7: FIN Diagram - Edit Tag**

Confirm Delete

?

Delete Resource

from database

Yes

No

List of

Resources

Return to

List of Resources

Clicks

Delete

**Figure 8: FIN Diagram - Delete Resource**

Confirm Delete

?

Delete Post

from database

Yes

No

List of Posts

Return to

List of Posts

Clicks

Delete

**Figure 9: FIN Diagram - Delete Post**

Confirm Delete

?

Delete Tag

from database

Yes

No

List of Tags

Return to

List of Tags

Clicks

Delete

**Figure 10: FIN Diagram - Delete Tag**

List of

Resources

Return to

Resource Details Page

Clicks on a

resource

Opens Resource

details page

Clicks on like

/

share

on Facebook

Already Logged In

to Facebook

Like

/

Share Resource on

Facebook Newsfeed

Enter Username and

Password of Facebook

account

Yes

No

Clicks

Login

**Figure 11: FIN Diagram - Like/Share Resource**

Login Page

Return to

Contact Us page

Clicks on

Contact Us

tab

Open Contact Page

Send message to

admins

Yes

No

Enter Name

,

Email

,

Message and Captcha

Input Correct

Display suitable

message

**Figure 12: FIN Diagram - Submit Enquiry**

##### *4.2.3 Unit Testing*

As we are using PHP laravel framework, we are planning to use PHPUnit for unit testing. However, as we are not following Test Driven Development (TDD) approach, we didn’t write any test cases this semester. Instead, we will write the entire unit tests needed for the major functionalities in the following semester. In MVC[[2]](#footnote-14)-design pattern, following testing are significantly important to make the system robust.

1. Model Testing
2. Route Testing
3. Controller Testing

Some of the methods we would be using from PHPUnit framework are as follows

1. assertTrue()
2. assertEquals()
3. assertFalse()

On top of PHPUnit, there are some test methods built in laravel as follows

1. assertResponseOk()
2. assertResponseStatus()
3. assertRedirectedTo()
4. assertRedirectedToRoute() etc.

##### *4.2.4 Integration Test*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Test Purpose** | **Input**  **Parameters** | **Actual Data Input** | **Expected Output** | **Test Status** |
| **1.0** | To test that login screen appears as the app loads up. | N/A | N/A | The login screen should appear when the application loads up. | Successful |
| **2.0** | To test the functionality of LOGIN button with correct username and password | Username | “root” | The system will allow the access to the specific logged in user and grant access to specific contents | Successful |
| Password | “ubuntu” |
| **2.1** | To test the functionality of LOGIN button with correct username and incorrect password | Username | “root” | The system should not allow access further | Successful |
| Password | “abcd123” |
| **2.2** | To test the functionality of LOGIN button with correct incorrect username and password | Username | “quantum” | The system should not allow access further | Successful |
| Password | “ubuntu” |
| **2.3** | To test the functionality of viewing resources with a click on Resources icon on dashboard | N/A | N/A | The system should display all the available resources | Successful |
| **2.4** | To test the functionality of “view all resource” from resource  dropdown in navbar | N/A | N/A | The system should display all the available resources | Successful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **2.5** | To test the functionality of creating new resources from resource dropdown in navbar | N/A | N/A | The system should display a form to create a new  resource | Successful |
| **2.6** | To test the functionality of creating new  resources | Name | “Sports” | Admin should be able to create a new resource | Successful |
| Description | “Summer  Games” |
| URL | “http://spor ts.com.au” |
| **2.7** | To test the functionality of creating a new resources with empty inputs | Name | N/A | Admin should be prompted with inputs required. | Successful |
| Description | N/A |
| URL | N/A |
| **2.8** | To test the functionality of creating a new resources with two  fields empty | Name | “Education” | Admin should be prompted with required fields missing for description and url. | Successful. |
| Description | N/A |
| URL | N/A |
| **2.9** | To test the functionality of creating a new resources with one  fields empty | Name | N/A | Admin should be prompted with required fields missing for name and url. | Successful |
| Description | “Early education for kids” |
| URL | N/A |
| **2.10** | To test the functionality of CANCEL button that opens when creating a new resource | N/A | N/A | The system should re-direct to resources page | Successful |
| **2.11** | To test the functionality of adding tag to a new resource | Name | “kids” | Admin should be able to create a new tag with a tag | Successful |
| Description | “This is  about kids” |
| URL | http:/quant um.com.au |
| Tagname | “kids” |
| **3.2** | To test the functionality of Add Button new  resource | N/A | N/A | Admin should be displayed with a form to create new resource | Successful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **3.3** | To test the functionality of EDIT BUTTON | N/A | N/A | Admin should be displayed with a form to edit particular resource with  values in input fields | Successful |
| **3.4** | To test the functionality of UPDATE button | N/A | N/A | Existing input values will remain as it is | Successful |
| **3.5** | To test the functionality of UPDATE with name change | Name | “Quantum” | Name of the resource should changed and remaining input values should remain as it was  before | Successful |
| Description | N/A |
| URL | N/A |
| Tagname | N/A |
| **3.5** | To test the functionality of UPDATE with  description change | Name | N/A | Description of that particular resource should change besides other input fields. | Successful |
| Description | “about quantum” |
| URL | N/A |
| Tagname | N/A |
| **3.6** | To test the functionally of deleting tagname with UPDATE button | Name | N/A | Quantum tag name should be deleted after updating the resource | Successful |
| Description | N/A |
| URL | N/A |
| Tagname | “quantum” |
| **3.7** | To test the functionally of adding tagname with  UPDATE button | Name | N/A | Resource should be updated with newly added tagname | Successful |
| Description | N/A |
| URL | N/A |
| Tagname | “games” |
| **3.8** | To test the functionality of PRINT button | N/A | N/A | A new print view window will be opened | Successful |
| **3.9** | To test the print functionality of browser print function | N/A | N/A | A new window with printing facility opens | Successful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **3.10** | To test the functionality of CANCEL button | N/A | N/A | Print window will be closed and print preview page will be displayed | Successful |
| **3.11** | To test the functionality of EXPORT TO PDF | N/A | N/A | A new window for saving the resource in PDF will be opened | Successful |
| **3.12** | To test the functionality of COPY button | N/A | N/A | An alert will display to inform the user that data has been copied to  clipboard | Successful |
| **3.13** | To test the functionality of resource on click | N/A | N/A | Resource details with an option to share to various social media should be  displayed | Successful |
| **3.14** | To test the functionality of facebook “LIKE”  button of resource | N/A | N/A | If an user is not logged in facebook, a facebook login windo should appear asking user to login to facebook to like the resource | Successful |
| **3.15** | To test the functionality of CANCEL button of facebook window | N/A | N/A | Facebook login window  should close | Successful |
| **3.16** | To test the functionality of logging in facebook in local server | Email | Rayx.\*\*v@g mail.com | After an user is loged in with valid facebook credentials, facebook window closes and resource should be liked | Failure |
| Password | \*\*\*\*\*\*\*\*\*\* |
| **3.17** | To test the functionality of facebook “LIKE” button when the user is logged in facebook in local server | N/A | N/A | The resource must be displayed in user’s facebook timeline | Failure |
| **3.18** | To test the functionality of facebook “LIKE” on  live server | N/A | N/A | The resource must be displayed in user’s facebook timeline | Successful |
| **3.19** | To test the functionality of facebook “LIKE” button when user is  logged in on live server | N/A | N/A | The resources is liked and displayed on user’s timeline | Successful |
| **3.19** | To test the functionality of facebook “SHARE”  button on local server | N./A | N/A | A new browser should be opened prompting for user to enter Email and Password if the user is not logged in | Successful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **3.20** | To test the functionality of facebook “SHARE” on local server when the user is logged in | N/A | N/A | Resource should be shared on user’s timeline | Failure |
| **3.21** | To test the functionality of facebook “SHARE” button on live server when the user is not logged in yet | N/A | N/A | A new browser should be opened prompting for user to enter Email and Password if the user is not logged in | Successful |
| **3.22** | To test the functionality of loggine in to facebook to share a resource | Email | Rayx.\*\*v@g mail.com | User is provided a link to share on user’s timeline when facebook’s valid login credentials are provided | Successful |
| Password | \*\*\*\*\*\* |
| **3.23** | To test the functionality of loggine in to facebook to share a resource after login  verification | N/A | N/A | User is prompted with an input box to say something about the link that is going to be shared | Successful |
| **3.24** | To test the functionality of sharing the resource link writing something about it on timeline | Say something about resource | “This is a resource from  Quantum  Victoria” | Resource should be published along with the description provided | Successful |
| **3.25** | To test the functionality of CANCEL button provided with facebook sharing browser windoe | N/A | N/A | Facebook sharing window should be closed | Successful |
| **3.26** | To test the functionality of twitter sharing button when the user is not already logged in | N/A | N/A | A new browser window will appear prompting user to login to twitter account | Successful |
| **3.27** | To test the functionality of twitter sharing after user promoted with  login window | Username | Ray.\*\*v@g mail.com | Resource should be shared in twitter | Succesful |
| Password | \*\*\*\*\*\* |
| **3.28** | To test the functionality of twitter sharing with | Username | Ray.\*\*v@g mail.com | User shouldnot be able to share the resource in the | Successful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | fake twitter account | Password | “abc124” | twitter |  |
| **3.30** | To test the functionality of google plus | N/A | N/A | User should be prompted with asking for email and password if not already  logged in | Successful |
| **3.31** | To test the functionality of google plus when the user is prompted to log in before sharing | Email | Ray.\*\*v@g mail.com | User should be able to share the resource in his/her  google plus profile | Successful |
| password | \*\*\*\*\*\* |
| **3.32** | To test the functionality of google plus with fake google account | Email | “randome @gmail.co m” | User should not be able to share the resource | Successful |
| Password | “randompw  ” |
| **3.33** | To test the functionality of Linkedin sharing button when the user is not already logged in | N/A | N/A | A new browser should open prompting user to log in | Successful |
| **3.34** | To test the functionality of sharing resource with LinkedIn after the usrs provides login  credentials | Email | Rayx.\*\*v@g mail.com | A new browser should open to let the user share the resource with different  audience | Successful |
| Password | \*\*\*\*\*\* |
| **3.34** | To test the functionality of sharing resource with LinkedIn with share  button on local server | N/A | N/A | Resource should be shared in the LinkedIn | Successful |
| **3.35** | To test the functionality of sharing resource with LinkedIn with share  button on live server | N/A | N/A | Resource should be shared in the LinkedIn | Successful |
| **3.36** | To test the functionality of LinkedIn CLOSE button after a resource s successfully shared | N/A | N/A | LinkedIn browser should closed | Sucessful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **4.0** | To test the functionality of viewing tags fromtags dropdown in navbar | N/A | N/A | The system will display all the available tags | Successful |
| **4.1** | To test the functionality of creating tag from Tags dropdown in  navbar | N/A | N/A | A form with input fields for  Name and Description  should load up | Successful |
| **4.2** | To test the functionality of CANCEL button | N/A | N/A | The system will re-direct to tags page | Successful |
| **4.3** | To test the functionality of AddNew Tag button | N/A | N/A | A form with input fields for  Name and Description  should load up | Successful |
| **4.4** | To test the functionality of Edit Tag | N/A | N/A | Form with editable input fields should load up | Successful |
| **4.5** | To test the functionality of CANCEL button | N/A | N/A | The system will re-direct to tags page | Successful |
| **4.6** | To test the functionality of UPDATE with name change | Name | “Quantum2  ” | Tag name will be changed  leaving description as it is | Successful |
| Description | N/A |
| **4.7** | To test the functionality of UPDATE with  description change | Name | N/A | Description of the tag will be changed | Successful |
| Description | “Descriptio  n for  Quantum2” |
| **4.8** | To test the functionality of Delete button on tags pags | N/A | N/A | A pop up will be alerted for confirming deletion | Successful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **4.9** | To test the functionality Delete confirmation  Button | N/A | N/A | Tag will be deleted | Successful |
| **4.10** | To test the functionality of CANCEL button | N/A | N/A | Tag will not be deleted and page will be re-directed to all tags | Successful |
| **5.0** | To test the functionality of Gallery on navbar | N/A | NA | It should be directed to a page where user can  add/upload/delete photos | Successful |
| **5.1** | To test the functionality of Add Files Button | N/A | N/A | A new window will pop up and user can select the images to be added | Successful |
| **5.2** | To test the functionality of Start Upload | N/A | N/A | Images added with Add Files can be now uploaded to system | Successful |
| **5.3** | To test the functionality of Cancel Update | N/A | N/A | Before starting to upload, added files should be able to be canceled uploading | Successful |
| **5.4** | To test the functionality of Add Photos to  Resources | N/A | N/A | A new window appears and photos from right panel can be dragged and dropped to specific resources to the left | Failed |
| **5.5** | To test the functionality of Delete All button | N/A | N/A | After selecting a check box right to Delete All button, all photos should be deleted with a click on Delete All button | Successful |
| **5.6** | To test the functionality of Delete | N/A | N/A | Resource should be deleted | Successful |
| **5.7** | To test the functionality of image in full screen | N/A | N/A | Image should be displayed  in full screen | Successful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **5.8** | To test the functionality of canceling full image and back to thumbnail view | N/A | N/A | User should be directed to thumbnail views of images | Successful |
| **5.9** | To test the functionality of carousal on full screen image view | N/A | N/A | Images should be displayed in slide show view with click on arrow | Successful |
| **6.0** | To test the functionality of Logout under  Account in dropdown in navbar | N/A | N/A | Pages redirects to login screen | Successful |
| **7.0** | To test the functionality of Contact Us nav menu | N/A | N/A | User should be prompted with a new UI and form to fill up with email, full name and message | Successful |
| **7.1** | To test the functionality of Send Message with  fields empty | Email | N/A | User should be displayed with errors and should be prompted saying these  fields are required | Failure |
| Fullname | N/A |
| Message | N/A |
| Captcha | N/A |
| **7.2** | To test the functionality of Send Message with  email field filled | Email | “aba@gmail  .com” | User should be prompted with errors to fill the remaining required fields | Failure |
| Fullname | N/A |
| Message | N/A |
| Captcha | N/A |
| **7.3** | To test the functionality of Send Message with email and Fullname  field filled | Email | “aba@gmail  .com” | User should be displayed with error message and prompted to fill the  remaining required fields | Failure |
| Fullname | “Sarun G” |
| Message | N/A |
| Captcha | N/A |
| **7.4** | To test the functionality of Send Message button with email, fullname and message filed  besides captcha | Email | “aba@gmail  .com” | User should be displayed with error message and prompted saying the  captach is incorrect | Successful |
| Fullname | “Sarun G” |
| Message | “Test message” |
| Captcha | N/A |
| Fullname | “Sarun G” |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **7.5** | To test the functionality of re-directing user after sending message | N/A | N/A | User should be directed to home page | Failure |
| **7.6** | To test the functionality of Home button on nav menu | N/A | N/A | User should be directed to Home page | Failure |
| **8.0** | To test the functionality of browser Back Button after a user is logged in | N/A | N/A | User should be displayed with login and should not be allowed to access dashboard unless logged in | Failure |
| **8.1** | To test if the user can access the dashboard contents access via browser back button  after logged out | N/A | N/A | User should not be able to access contents from the dashboard | Successful |
| **9.0** | To test the functionality of number of views on resource | N/A | N/A | Each time a resource is clicked, number of hits should be increased by one | Successful |
| **9.1** | To test the functionality of number of hits  increase on resource | N/A | N/A | Number of hits on the resource should keep on  increasing on each click | Successful |
| **10.0** | To test the functionality of pagination | Show | 5 | User should be able to see five resources and remaining resources should be paginated to next page and give user an option to paginate to NEXT page | Successful |
| **10.1** | To test the functionality of pagination | Show | 10 | User should be able to see ten resources and remaining resources should be paginated to next page and give user an option to paginate to NEXT page | Successful |
| **10.2** | To test the functionality of pagination | Show | 20 | User should be able to see 20 resources and remaining resources should be paginated to next page and give user an option to paginate to NEXT page | Successful |
| **10.3** | To test the functionality of pagination | Show | 50 | User should be able to see 50 resources and remaining resources should be  paginated to next page | Successful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **10.4** | To test the functionality of pagination | Show | 100 | User should be able to see  100 resources and remaining resources should be paginated to next page and give user an option to paginate to NEXT page | Successful |
| **10.5** | To test the functionality of ordering the tile of resource | N/A | N/A | Resources title’s should be arranged in alphabetical order | Successful |
| **10.6** | To test the functionality of ordering the  description of resource | N/A | N/A | Resources title’s should be arranged | Successful |
| **11.0** | To test the functionality of search input box in resources | Resourcena me | “test5” | It should display test5 resource | Successful |
| **11.1** | To test the functionality of search input box in tag | Tagname | “quantum” | It should display quantum tag | Successful |
| **12.0** | To test the functionality of pagination in tags | Show | 5 | User should be able to see five tags and remaining tags should be paginated to next page and give user an option to paginate to NEXT page | Successful |
| **12.1** | To test the functionality of pagination | Show | 10 | User should be able to see ten tags and remaining tags should be paginated to next page and give user an option to paginate to NEXT page | Successful |
| **12.2** | To test the functionality of pagination | Show | 20 | User should be able to see 20 tags and remaining tags should be paginated to next page and give user an option to paginate to NEXT page | Successful |
| **12.3** | To test the functionality of pagination | Show | 50 | User should be able to see 50 tags and remaining tags should be paginated to next page and give user an option to paginate to NEXT page | Successful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **12.4** | To test the functionality of pagination | Show | 100 | User should be able to see 100 tags and remaining tags should be paginated to next page and give user an option to paginate to NEXT page | Successful |
| **12.5** | To test the functionality of ordering the tile of tags | N/A | N/A | Tags title’s should be arranged in alphabetical order | Successful |
| **12.6** | To test the functionality of ordering the  description of tags | N/A | N/A | Tags title’s should be arranged | Successful |
| **13.0** | To test the functionality of accessing the contents manually typing URL without  being logged in | URL | http://local host:8000/t ags | User should be directed to login page | Successful |
| **13.1** | To test the functionality of Brand in navbar | N/A | N/A | User should be directed to home page | Successful |
| **13.2** | To test the functionality of Brand in navbar when a user is in resources page and is  logged in | N/A | N/A | User should be directed to home page | Successful |
| **13.3** | To test the functionality of Brand in navbar when a user is in tags page and is logged in | N/A | N/A | User should be directed to home page | Successful |
| **13.4** | To test the functionality of Brand in navbar when a user is in gallary page and is logged in | N/A | N/A | User should be directed to home page | Successful |
| **13.4** | To test the functionality of Brand in navbar when a user is in Edit Resrouce page and is logged in | N/A | N/A | User should be directed to home page | Successful |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **13.5** | To test the functionality of navigating to view all tags when a user in resource editing page | N/A | N/A | User should be directed to viewing all tags page | Failure |
| **13.6** | To test the functionality of navigating to view all resources when a user in resource editing page | N/A | N/A | User should be directed to viewing all resources page | Failure |
| **13.7** | To test the functionality of navigating to gallery when a user in resource editing page | N/A | N/A | User should be directed to gallery page | Successful |
| **13.8** | To test the functionality of navigating to view all tags when a user in tag editing page | N/A | N/A | User should be directed to viewing all tags page | Successful |

##### *4.2.5 WireFrames*

Wireframes are images or set of images which displays the functional elements of a website or page, typically used for planning a site's structure and functionality. For our project, we designed some wireframes for planning purpose which are shown below.

**Figure**

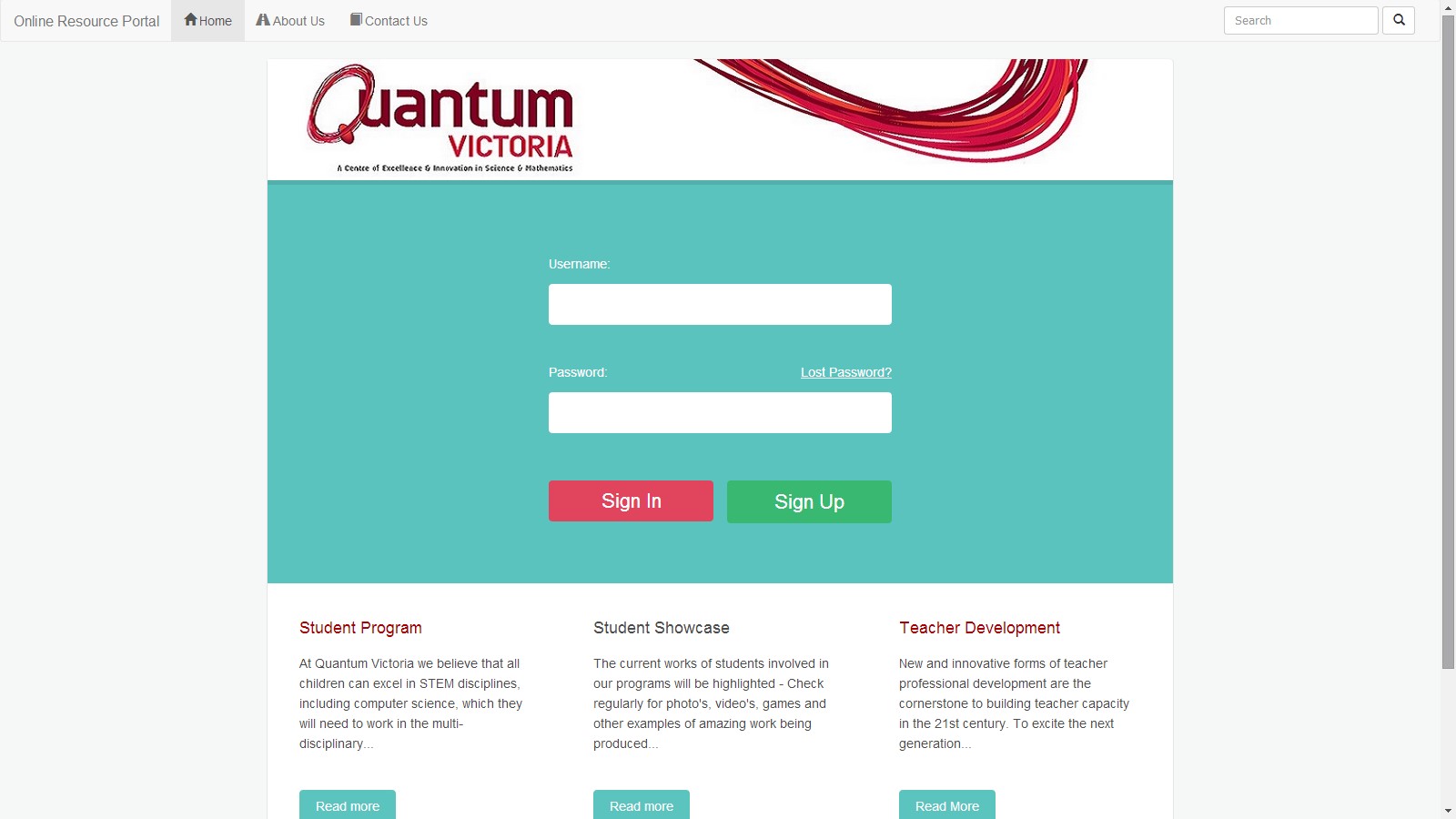
**13**

**:**

**Wireframe**

**-**

**Login Screen**



**Figure**

**14**

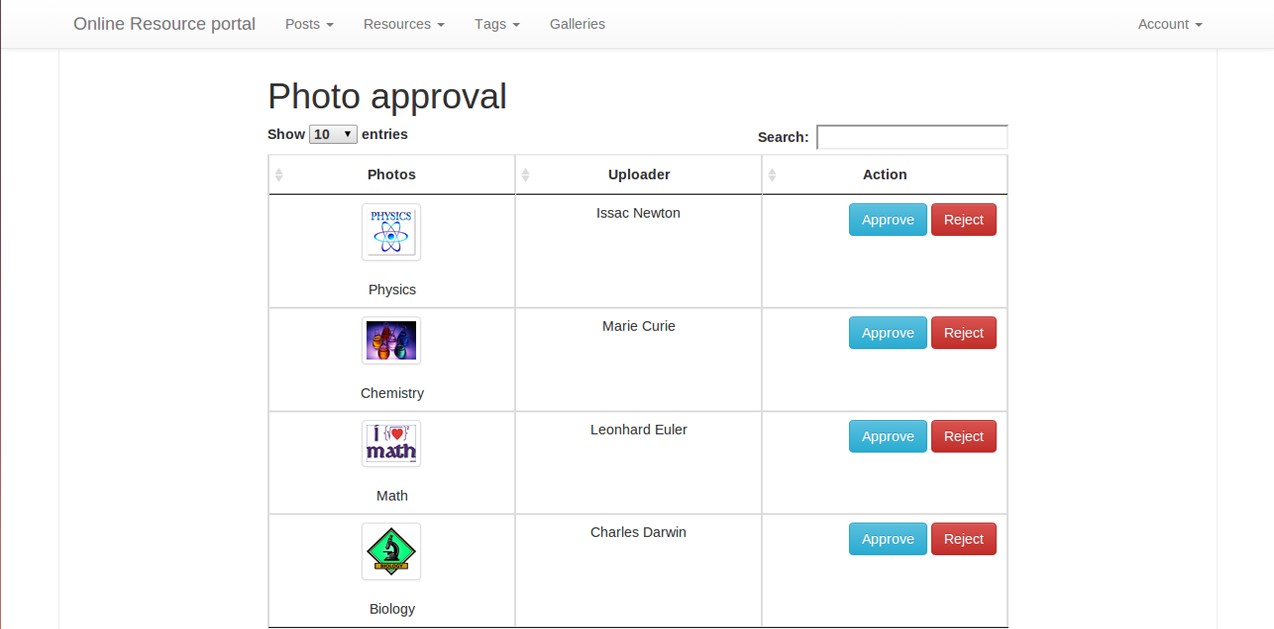
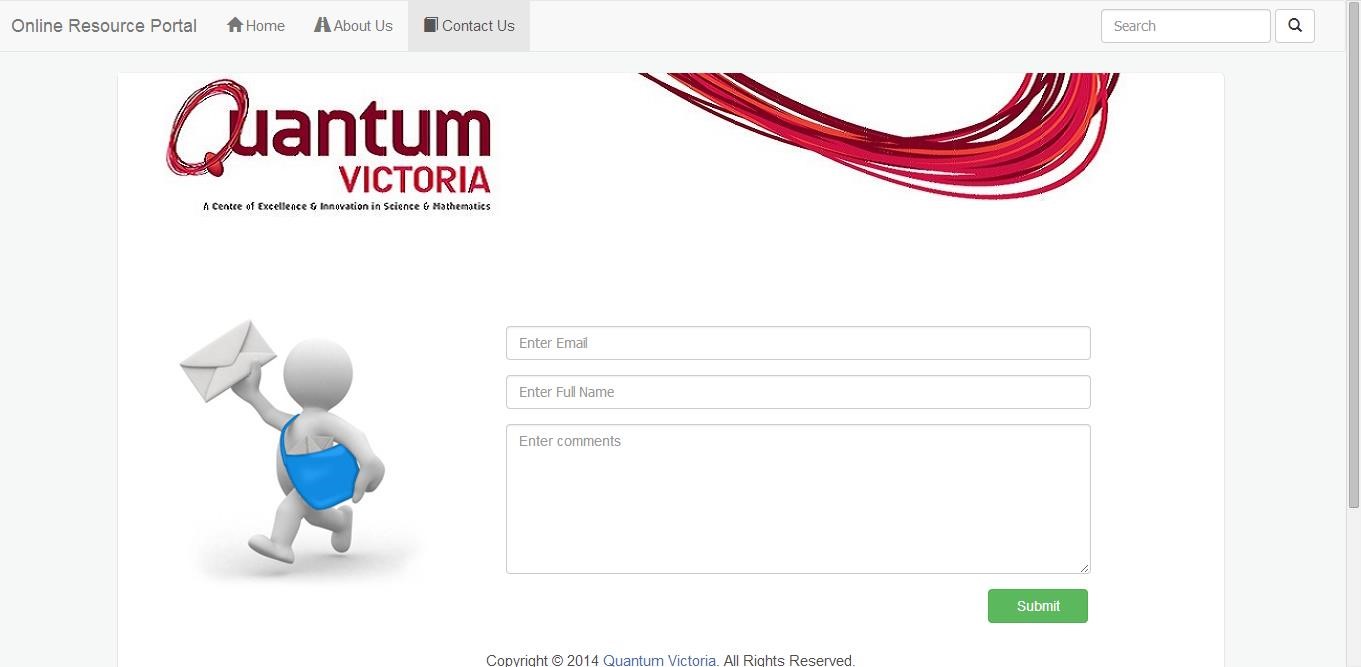
**:**

**Wireframe**

**-**

**Contact**

**Admin**



**Figure 15: Wireframe - Admin Approval**

**Figure**

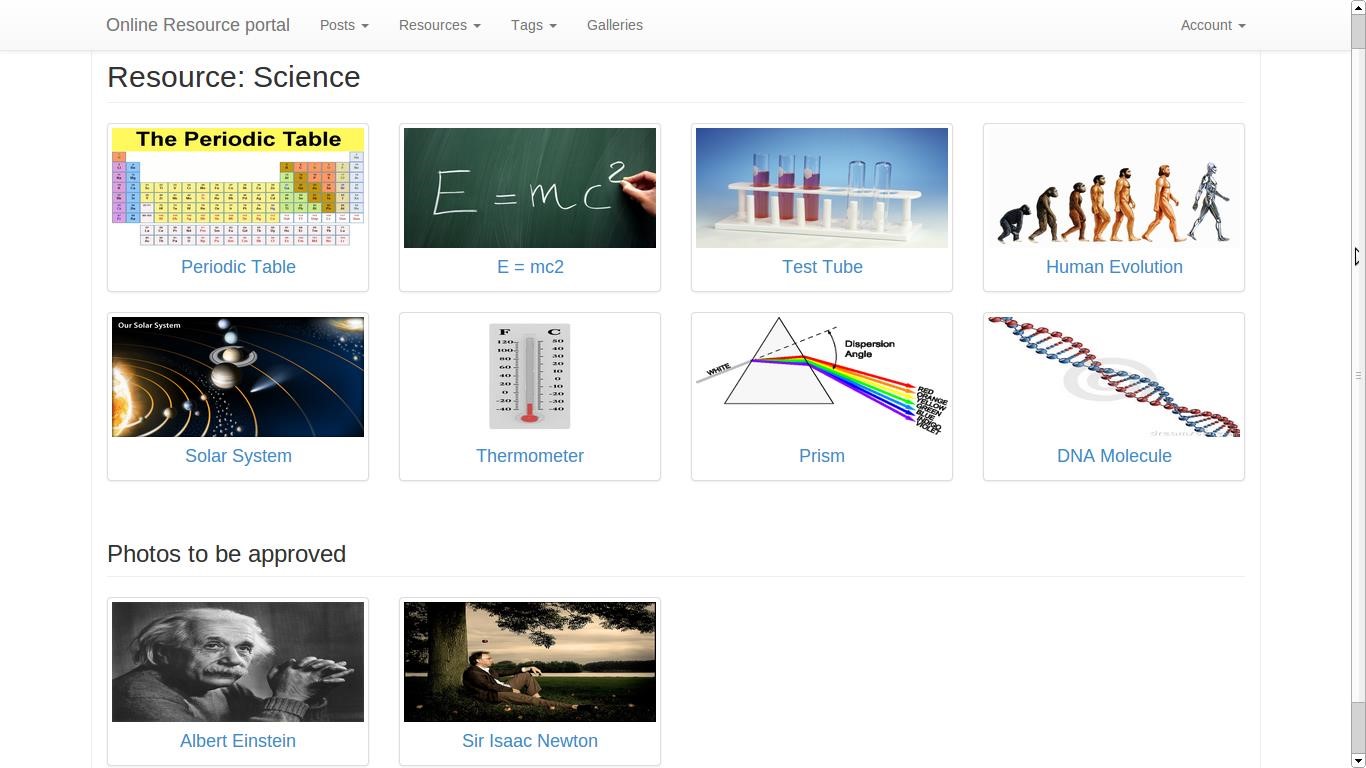
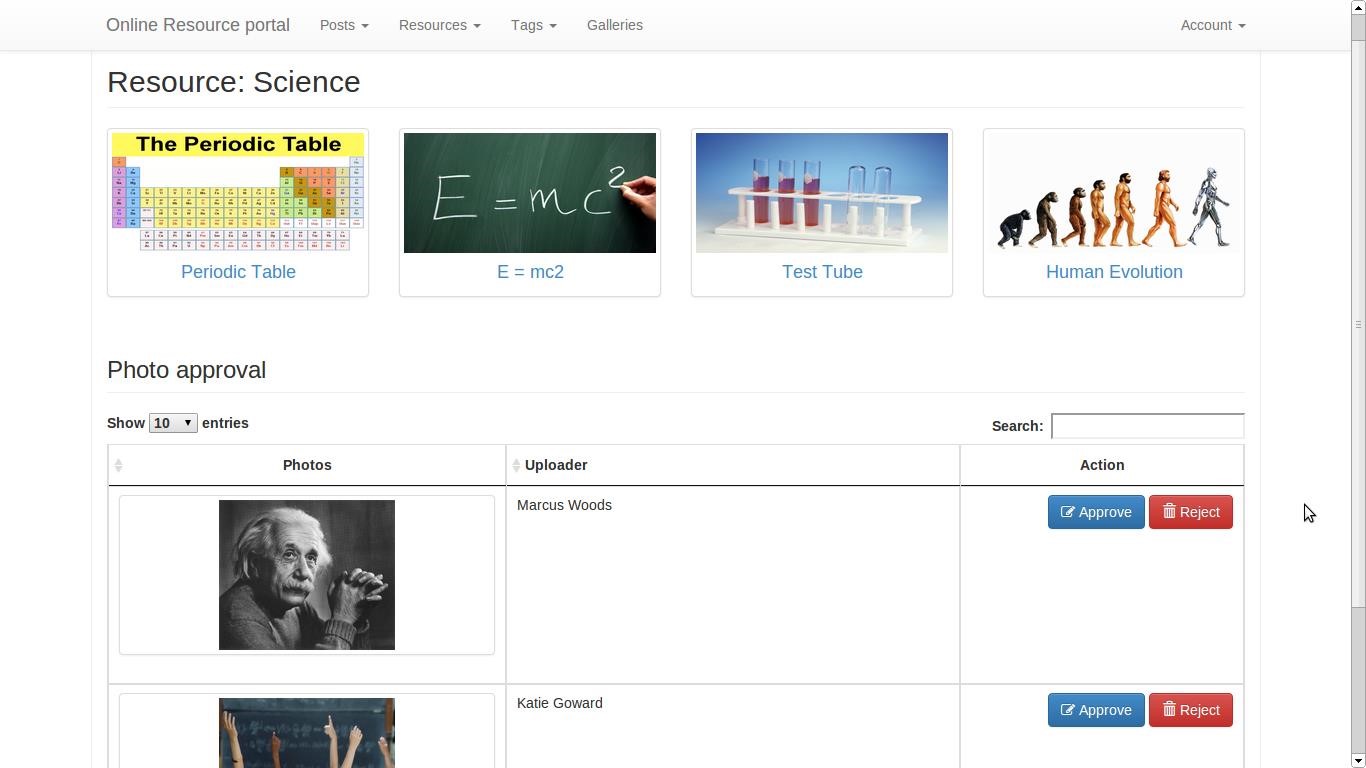
**16**

**:**

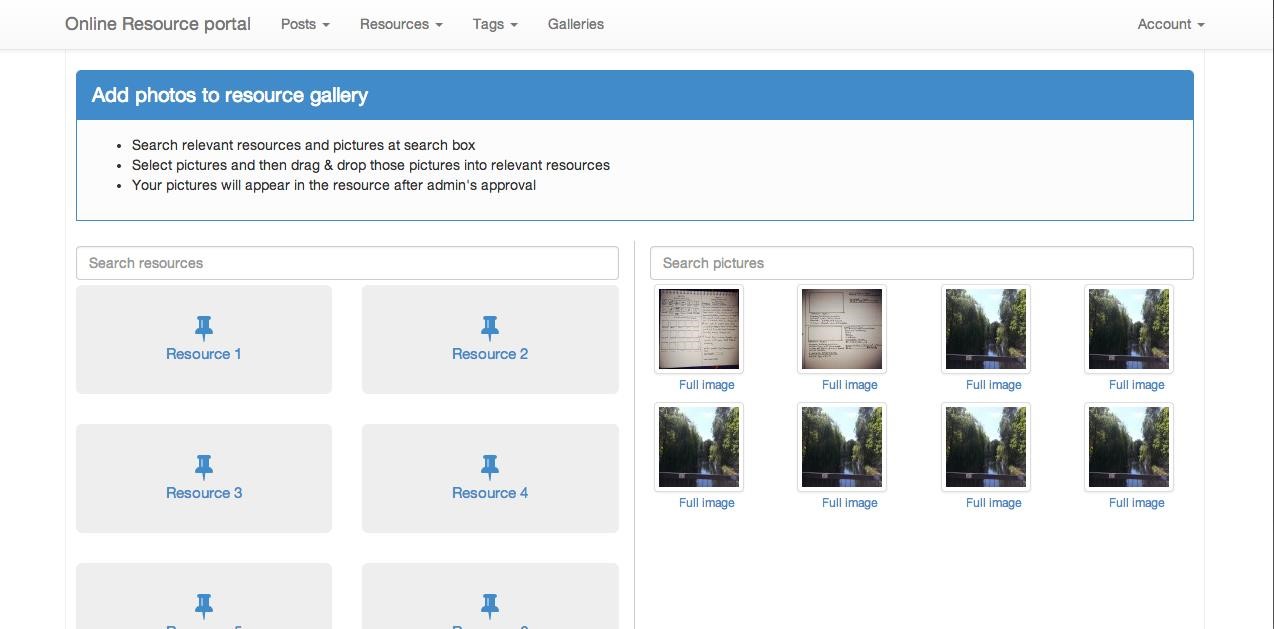
**Wireframe**

**-**

**Admin Resource View**



**Figure 17: Wireframe - User Resource View**



**Figure 18: Wireframe - Drag and Drop**

**Figure**

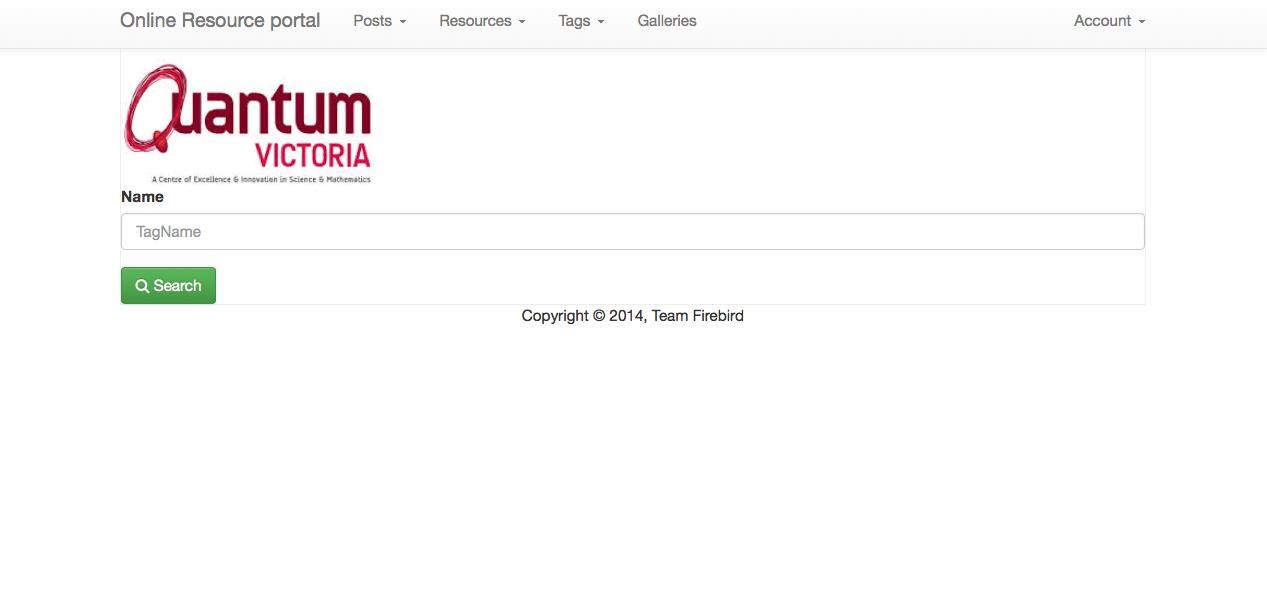
**19**

**:**

**Wireframe**

**-**

**Add New Tags**



**Figure**

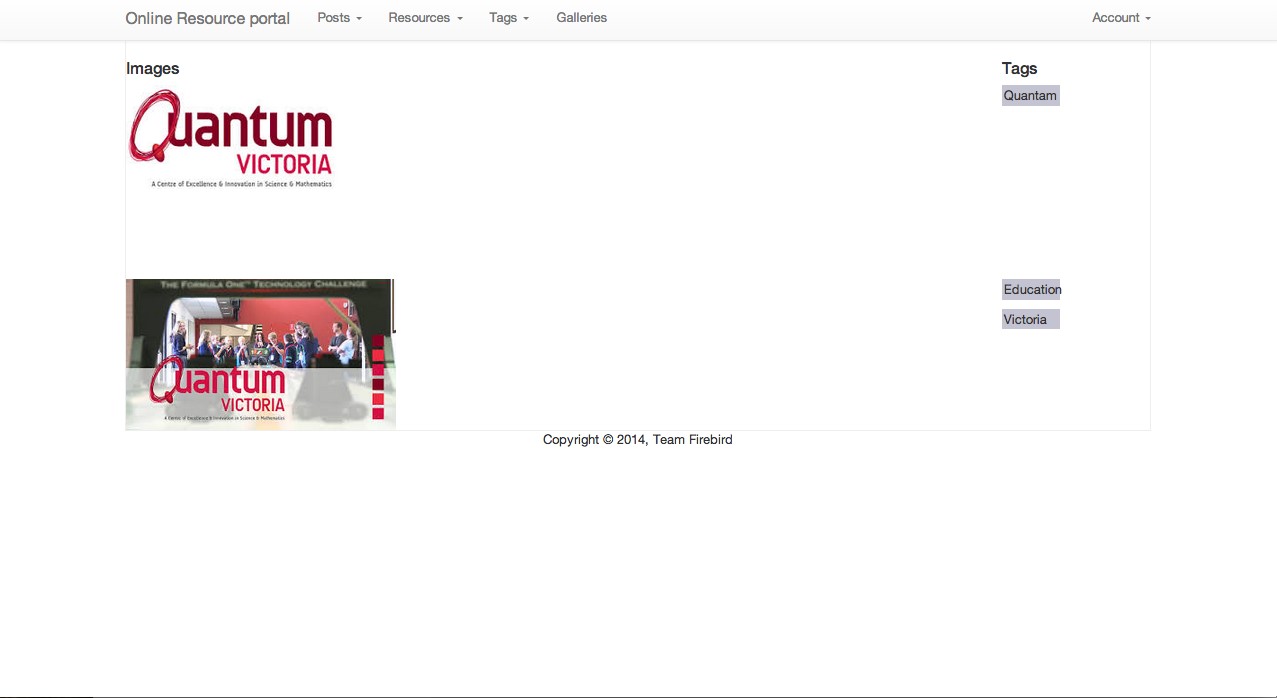
**20**

**:**

**Wireframe**

**-**

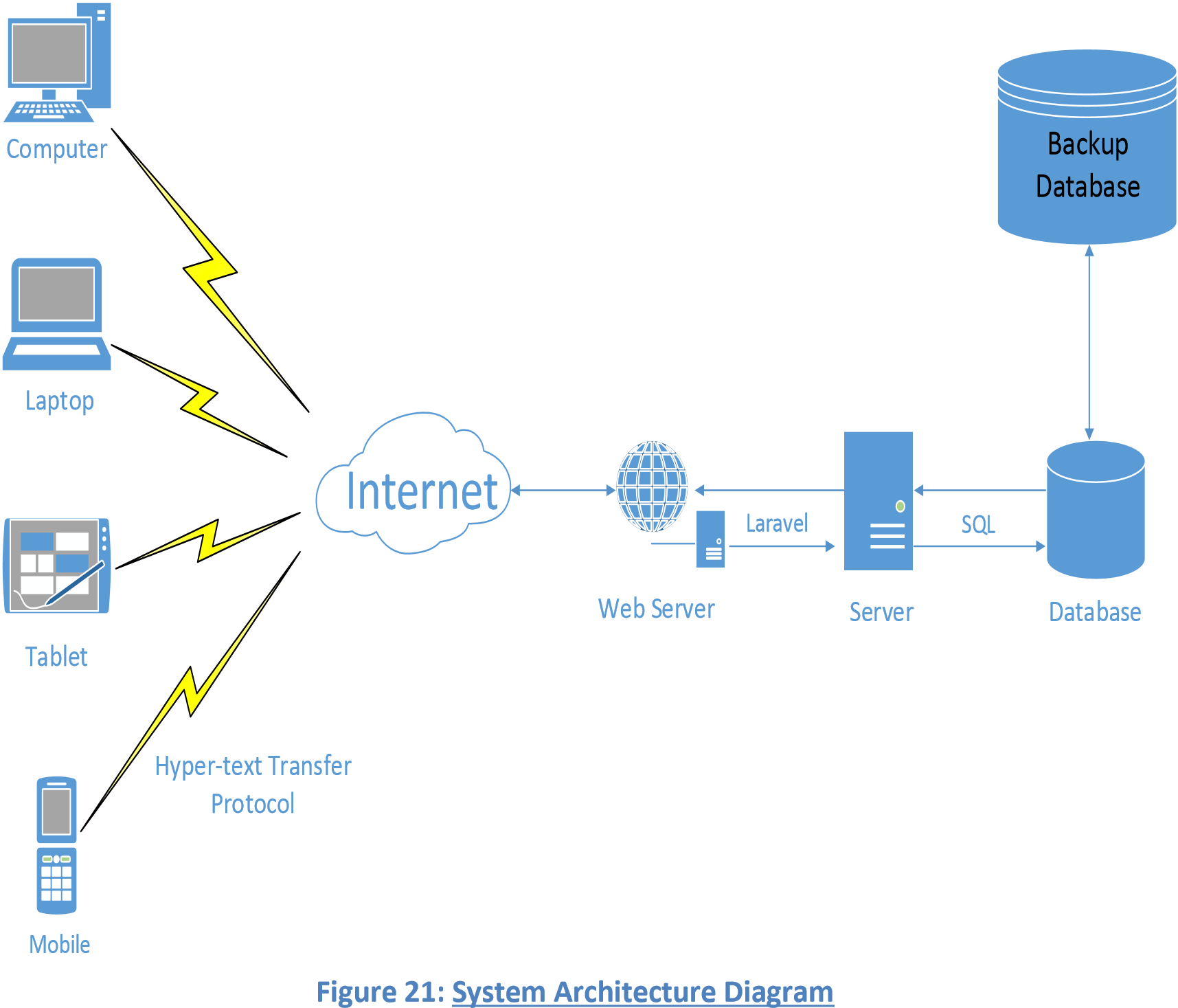
**Tagged Images**



## 5.0 Object-Oriented Design

### 5.1 High Level System Architecture

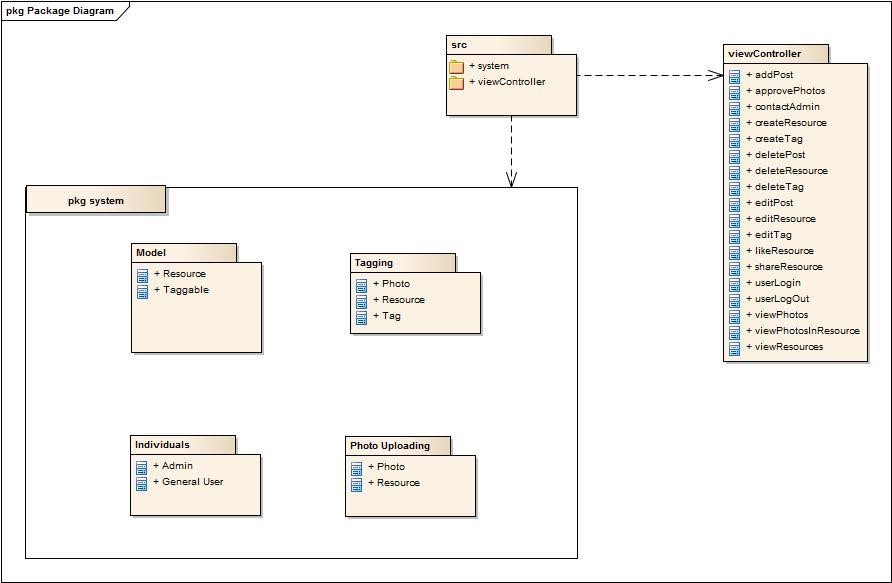
6.0 The system **architecture diagram** enables us to graphically model the applications of a system, and the externals that they interface with and data stores that they use or provide information to. It comprises the software elements, the relations between them, and the properties of both elements and relations.



### 6.1 High Level Package Diagram and Components

A **package diagram** in the [Unified Modeling Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language) depicts the [dependencies](http://en.wikipedia.org/wiki/Dependency_(UML)) between the [packages](http://en.wikipedia.org/wiki/Package_(UML)) that make up a model. It comprises of

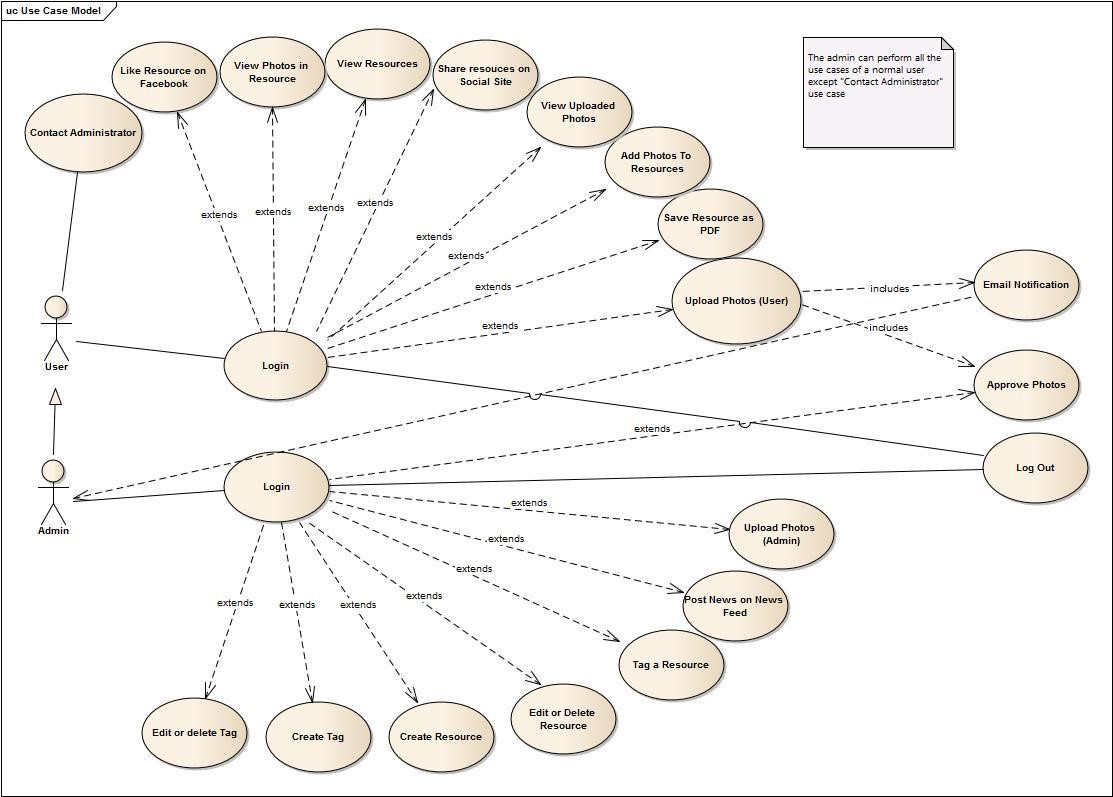
1. Packages
2. Classes
3. Interface
4. Objects



### 6.2 Use Case Analysis

#### USE CASE DIAGRAM INTRO

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a [use case.](http://en.wikipedia.org/wiki/Use_Case) A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual [use case.](http://en.wikipedia.org/wiki/Use_Case)



**Figure 22: Use Case Model**

#### USES CASE

In [software](http://en.wikipedia.org/wiki/Software_engineering) and [systems engineering,](http://en.wikipedia.org/wiki/Systems_engineering) a **use case** is a list of steps, typically defining interactions between an actor (user) and a system, to achieve a goal. The actor can be a human or an external system.

**USE CASE 1**: User Login

**TRIGGER/GOAL**: A user has entered the website and wants to log in

**ACTOR**: Admin/User

##### MAIN FLOW

1. Actor enters username into the username field and password into the password field 2. User is redirected to the users home page

##### EXTENSIONS

2a - User clicks on the “Back” button on the browser instead 1. Login Screen terminates and User returns to home page.

**USE CASE 2**: View Resources

**TRIGGER/GOAL**: The user/admin wants to see the available resources

**ACTOR**: Admin/User

##### MAIN FLOW

1. The actor selects the resource
2. The actor is redirected to the resource description page with all the contents of the resource.

##### EXTENSIONS

2a - User clicks on the “Back Button” on the browser instead 1. Resource Page terminates.

2. User returns to the session home Page

**USE CASE 3**: View photos in a resource

**TRIGGER/GOAL**: User wants to view a photo in the resource

**ACTOR**: Admin/User

##### MAIN FLOW

1. The user clicks on the photo thumbnail
2. The user is redirected to the page that contains the full size image

##### EXTENSIONS

2a - The image is not displayed due to some error

1. The user is shown an alternate text to the image

**USE CASE 4**: Share resources on social website

**TRIGGER/GOAL**: The user wants to share the resource on their social networking sites

**ACTOR**: Admin/User

##### MAIN FLOW

1. The user selects the social site from the options
2. The user is prompted to login to the selected social site in a different page
3. The user is authenticated using the social site API
4. The link to the resource is shared on the user’s home screen
5. The user is redirected to the resource description page

##### EXTENSIONS

3a - The user clicks on the cancel button instead

1. The login screen terminates

**USE CASE 5**: Like Resource on Facebook

**TRIGGER/GOAL**: The user/admin wants to like the resource

**ACTOR**: Admin/User

##### MAIN FLOW

1. The user clicks on the like button on the resource description page 2.

1. If the user is not logged in

i. The user is prompted to log in to the Facebook account ii. The user is authenticated by Facebook API

iii. The resource is liked on their Facebook page and resource description page

1. If the user is already logged in the Facebook

i. The resource is liked on their Facebook page and the resource description page

##### EXTENSIONS

2a - The user clicks on cancel button during log in prompt

1. The user login page terminates

**USE CASE 6**: Save the resource page as PDF

**TRIGGER/GOAL**: The user wants to save the resource page as PDF on their system

**ACTOR**: Admin/User

##### MAIN FLOW

1. The user clicks on the “Import as PDF” link on the resource description page 2. The user is directed to a new screen

3. An alert box appears that display the information to print the page

**USE CASE 7**: Create Resource

**TRIGGER/GOAL**: The admin wants to create a resource

**ACTOR**: Admin

##### MAIN FLOW

1. The admin fills up the required information
2. Admin clicks on create button
3. The resource is created and admin is redirected to view all resources page

##### EXTENSIONS

2a - The admin clicks on cancel rather than clicking on create button

1. The create resource screen terminates and the user is redirected to view all resources page

2b - The admin fills in the wrong or invalid information

1. An error is shown in a flash message and admin stays on the create resource page

**USE CASE 8**: Edit or delete a resource

**TRIGGER/GOAL**: The admin wants to edit or delete the resource

**ACTOR**: Admin

##### MAIN FLOW

1. Admin clicks on the resources link from his session home page
2. Admin is directed to the resources page
3. a) The admin clicks on edit button next to the desired resource

i. The admin is directed to the edit resource page ii. The admin updates the information and click on save

iii. The information is updated in the database and the admin is redirected to the resource page

b) The admin clicks on delete button next to the desired resource

i. An alert box pops up for confirmation of the deletion ii. The admin clicks on the delete button

iii. The alert box terminate, resource is deleted and the resource page is refreshes

##### EXTENSIONS

3a ii. The admin clicks on cancel button

1. The edit screen terminates, the information is discarded and the user is redirected to the resources screen

3b. ii. The admin clicks on cancel button

1. The alert box terminates and user returns back to the resource screen without any changes

**USE CASE 9**: Tag a resource

**TRIGGER/GOAL**: The admin wants to add a tag to the resource

**ACTOR**: Admin

##### MAIN FLOW

1. Add fills in the tag information
2. The autofill option helps the admin choose the tag from existing resources
3. Admin clicks on the save button
4. The tag is saved with the resource and the admin is redirected to the resource page

##### EXTENSIONS

3a - The admin clicks on cancel rather than clicking on save button

1. The create resource screen terminates and the user is redirected to view all resources page

**USE CASE 10**: Contact Administrator

**TRIGGER/GOAL**: The user wants to contact the website administrators

**ACTOR**: User

##### MAIN FLOW

1. The user fills up the required information and enter his/her query
2. The user clicks on the “Send Message” button
3. The information is saved and the user is redirected to the home screen

##### EXTENSIONS

2a - The captcha entered by user does not match the displayed captcha

1. The user is redirected to the contact page with an appropriate error message

**USE CASE 11**: Create a Tag

**TRIGGER/GOAL**: The admin wants to create a new tag

**ACTOR**: Admin

##### MAIN FLOW

1. The admin enters the required information and clicks on “Create Tag” button
2. The information is saved and the admin is redirected to the tags page

##### EXTENSIONS

1a - The admin clicks on “Cancel” button rather than “Create Tag” button

1. The information is discarded and the admin is redirected to the tags page

**USE CASE 12**: Upload Photos

**TRIGGER/GOAL**: The user wants to upload a photo

**ACTOR**: User / Admin

##### MAIN FLOW

1. i. The actor clicks on add files and browse the photo from his device to add to the galleryii. The actor use the drag and drop feature of the website to upload photos
2. The photos appear on the website and actor click on “Start upload” button to upload the photos
3. The photos are uploaded on the website and the gallery page refreshes

##### EXTENSIONS

2a - The actor clicks on “Cancel Upload” or “Delete Button” rather than “Start Upload” button

1. The photos are removed from the gallery page

**USE CASE 13**: Add Photos to Resources (User)

**TRIGGER/GOAL**: The user wants to add uploaded photos to resources

**ACTOR**: User

##### MAIN FLOW

1. The user clicks on the “Add photos to resources” link on the gallery page
2. The user is redirected to the “Add Photos” page
3. The user drags and drops the photo in a resource
4. The photo appears on the admin approval screen and needs approval before adding to the resource
5. The photo is added in that resource and an alert box pops up confirming the operation

**USE CASE 14**: Add Photos to Resources (Admin)

**TRIGGER/GOAL**: The admin wants to add uploaded photos to resources

**ACTOR**: Admin

##### MAIN FLOW

1. The admin clicks on the “Add photos to resources” link on the gallery page
2. The admin is redirected to the “Add Photos” page
3. The admin drags and drops the photo in a resource
4. The photo is added in that resource and an alert box pops up confirming the operation

**USE CASE 15**: View Uploaded Photos

**TRIGGER/GOAL**: User wants to see the photos uploaded by him

**ACTOR**: User

##### MAIN FLOW

1. The user clicks on the “Gallery” link from the navigation bar
2. The user is directed to the gallery page where he can see photos uploaded by that user

**USE CASE 16**: Email Notification of the photo upload

**TRIGGER/GOAL**: The user wants to add a photo to a resource

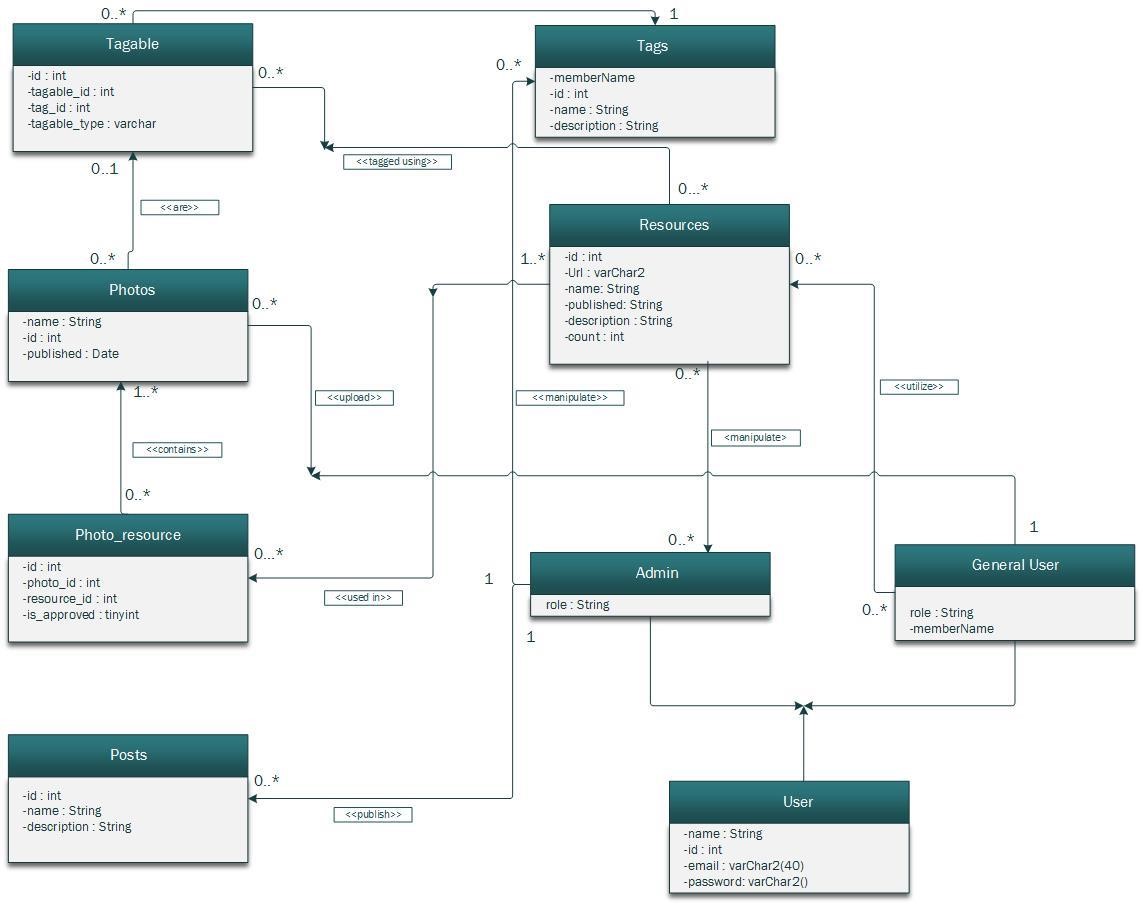
**ACTOR**: Admin/User

##### MAIN FLOW

1. The user drags and drops the photo in a resource
2. The photo appears on the admin approval screen and needs approval before adding to the resource
3. The admin is notified about the upload request through an email

### 6.3 Domain Model and Class Diagram

The domain model is created in order to represent the vocabulary and key concepts of the problem domain. The domain model also identifies the relationships among all the entities within the scope of the problem domain, and commonly identifies their attributes.



#### Figure 23: Class Diagram

***FINAL RELATIONAL TABLES:***

**PHOTO\_RESOURCES** (id, photo\_id, resource\_id, is\_approved, created\_at, update\_a)

**PHOTOS** (id, name, size, type, url, title, description, created\_at, updated\_at, user\_id)

**POSTS** (id, title, body, created\_at, updated\_at)

**TAGGABLE** (id, taggable\_id, tag\_id, taggable\_type, created\_at, updated\_at)

**TAGS** (id, name, description, created\_at, updated\_at)

**USERS** (ID, user\_login, user\_pass, user\_email, user\_registered, display\_name, updated\_at) **RESOURCES** (id, name, description, created\_at, updated\_at)

|  |  |  |  |
| --- | --- | --- | --- |
| ENTITY | NAME | NULL | OTHER CONSTRAINTS |
| PHOTO\_RESOURCES | Id photo\_id,    resource\_id    approved created\_at updated\_at | NOT NULL  NOT NULL  NOT NULL  NOT NULL | Primary Key  Foreign Key, References  PHOTOS (id)  Foreign Key, References  Resources (id) |

|  |  |  |  |
| --- | --- | --- | --- |
| ENTITY | NAME | NULL | OTHER CONSTRAINTS |
| PHOTOS | Id  name  description created\_at updated\_at | NOT NULL  NOT NULL | Primary Key |

|  |  |  |  |
| --- | --- | --- | --- |
| ENTITY | NAME | NULL | OTHER CONSTRAINTS |
| TAGGABLE | id  taggable\_id    tag\_id    taggable\_type created\_at updated\_at | NOT NULL  NOT NULL  NOT NULL  NOT NULL | Primary Key  Foreign Key, References  RESOURCES (id)  Foreign Key, References  TAGS (id) |

|  |  |  |  |
| --- | --- | --- | --- |
| ENTITY | NAME | NULL | OTHER  CONSTRAINTS |
| TAGS | id  name  description created\_at updated\_at | NOT NULL  NOT NULL | Primary Key |

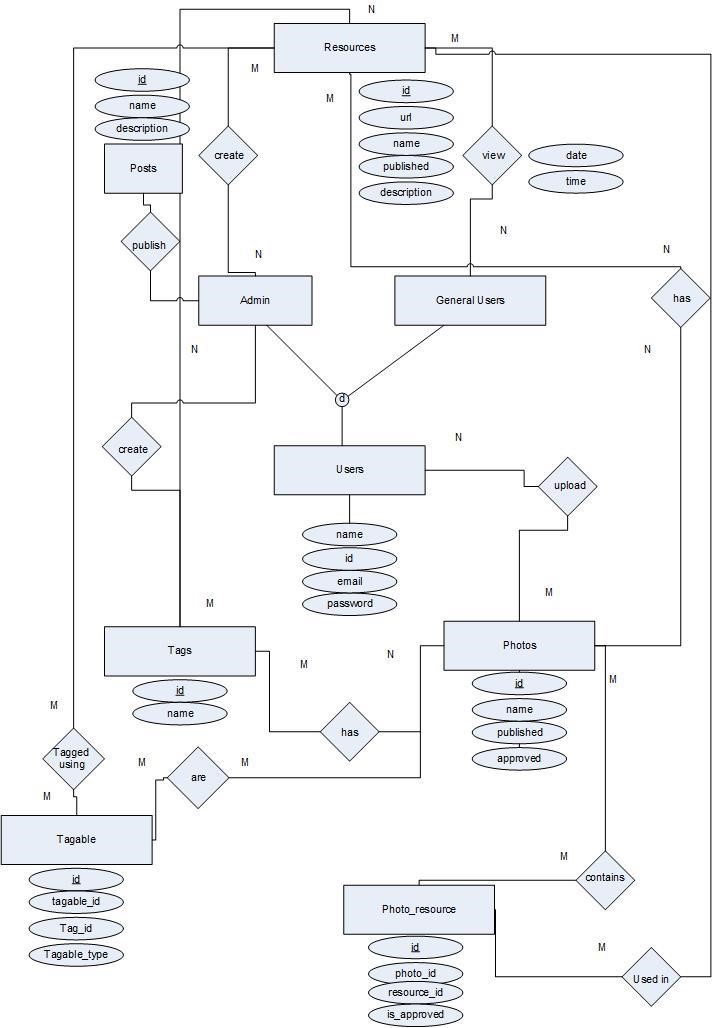
|  |  |  |  |
| --- | --- | --- | --- |
| ENTITY | NAME | NULL | OTHER  CONSTRAINTS |
| RESOURCES | id  name  description created\_at updated\_at | NOT NULL  NOT NULL | Primary Key |

|  |  |  |  |
| --- | --- | --- | --- |
| ENTITY | NAME | NULL | OTHER  CONSTRAINTS |
| USERS | ID user\_login user\_pass user\_email registered display\_name  updated\_at | NOT NULL  NOT NULL  NOT NULL | Primary Key      type email |

|  |  |  |  |
| --- | --- | --- | --- |
| ENTITY | NAME | NULL | OTHER  CONSTRAINTS |
| POSTS | id  name  description created\_at updated\_at | NOT NULL  NOT NULL | Primary Key |

### 6.4 Establishment of the Database Objects and Data Access Strategy

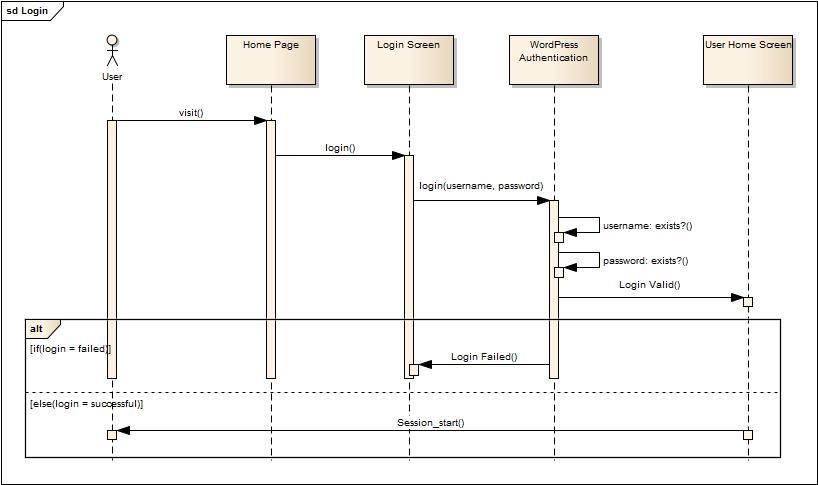
**Entity–relationship model** (**ER model**) is a [data model](http://en.wikipedia.org/wiki/Data_modeling) for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a [database](http://en.wikipedia.org/wiki/Database) such as a [relational database.](http://en.wikipedia.org/wiki/Relational_database) The main components of ER models are [entities](http://en.wikipedia.org/wiki/Entities) and the relationships that can exist among them.



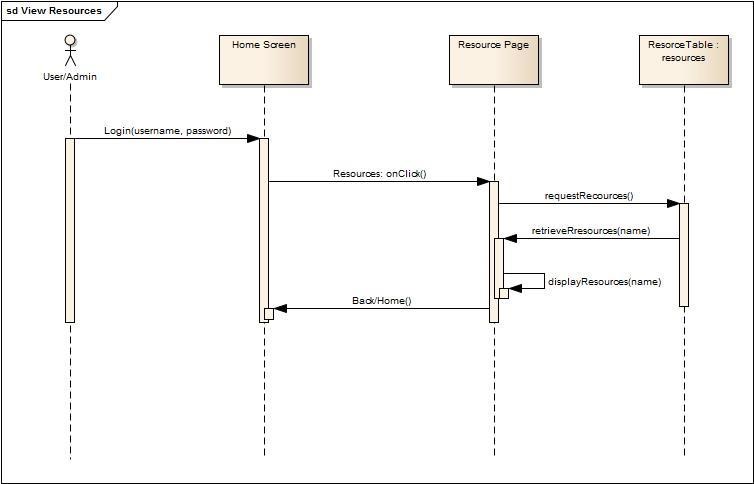
**Figure 24: Entity Relationship Diagram**

### 6.5 Sequence diagram

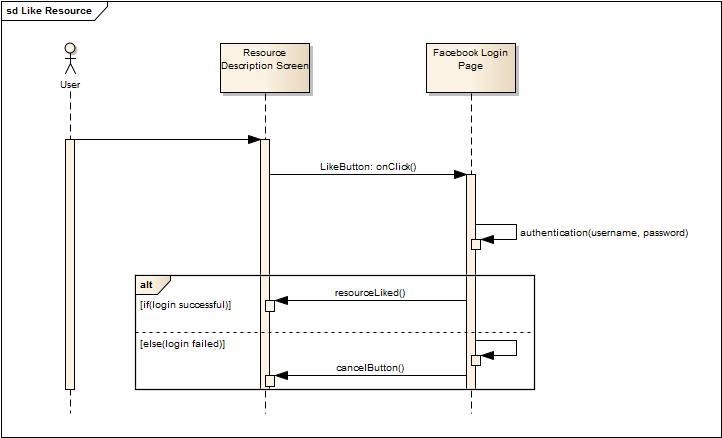
A **sequence diagram** is an [interaction diagram](http://en.wikipedia.org/wiki/Interaction_diagram) that shows how processes operate with one another and in what order. It is a construct of a [Message Sequence Chart.](http://en.wikipedia.org/wiki/Message_Sequence_Chart) A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.



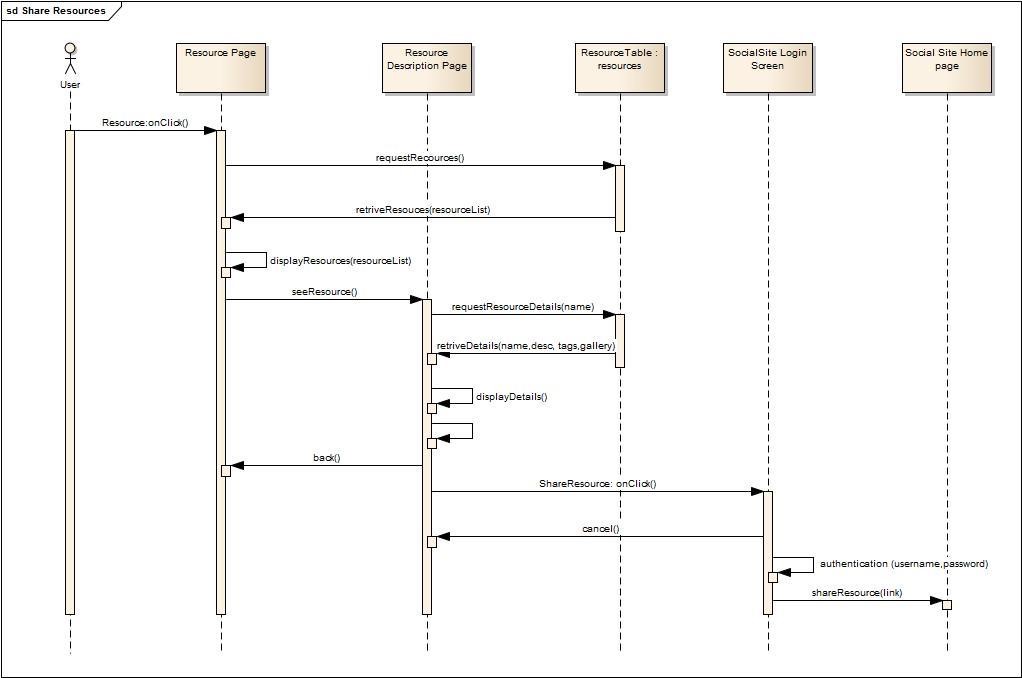
#### Figure 25: Sequence Diagram – Login



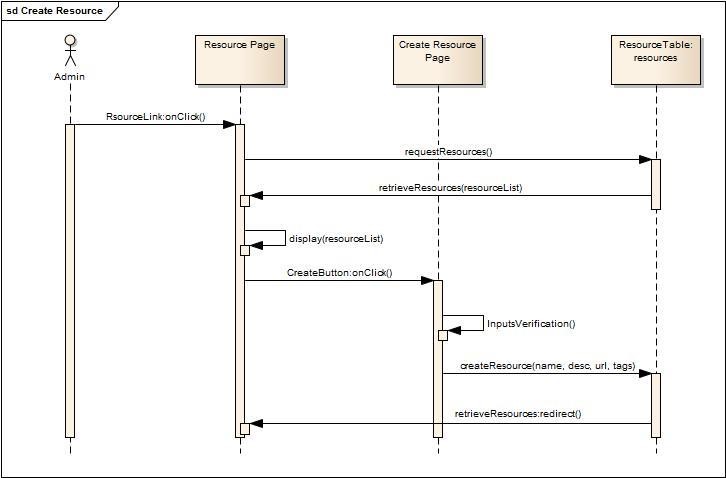
**Figure 26: Sequence Diagram – View Resources**



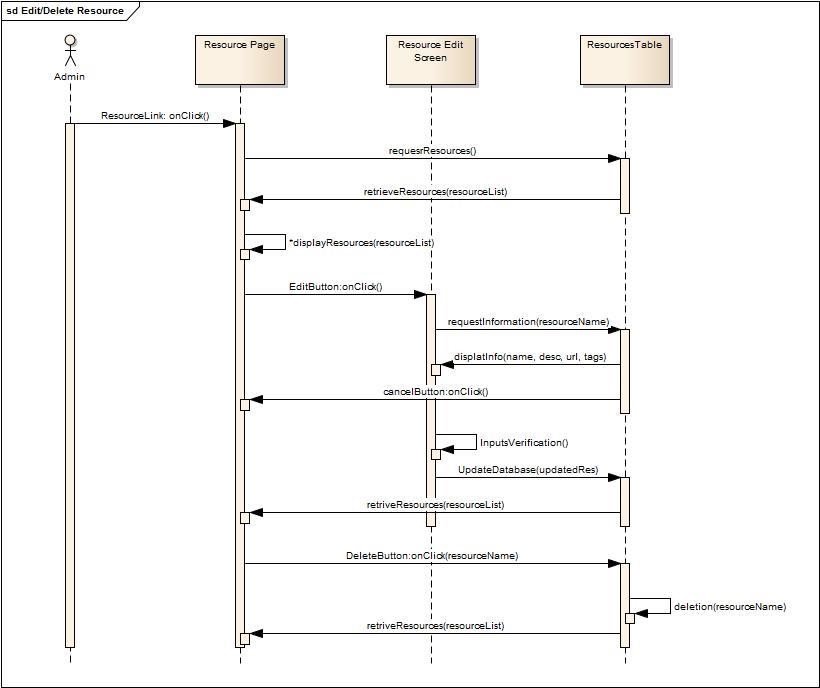
**Figure 27: Sequence Diagram – Like Resources**



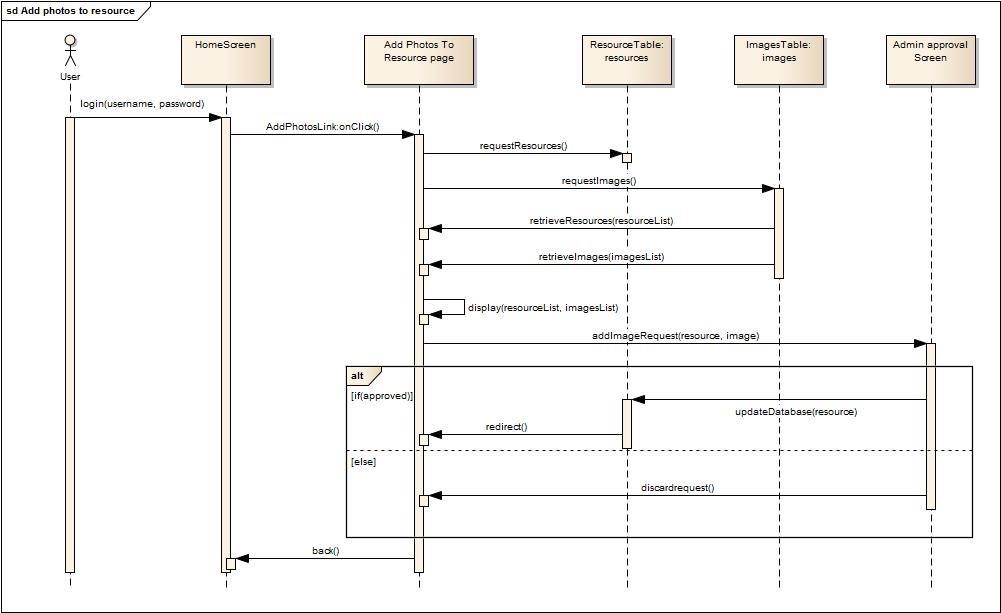
**Figure 28: Sequence Diagram – Share Resources**



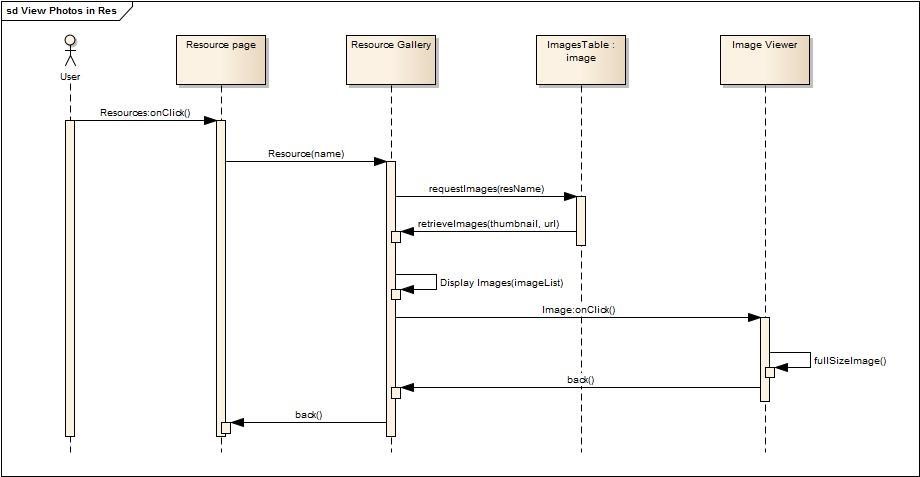
**Figure 29: Sequence Diagram – Create Resource**



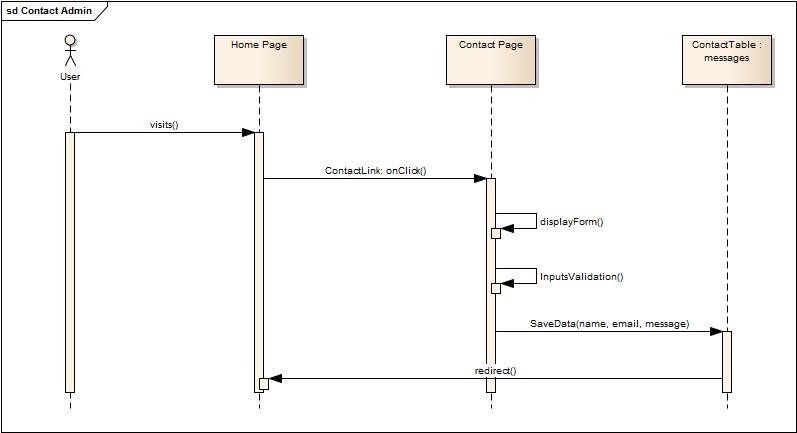
**Figure 30: Sequence Diagram – Edit/Delete Resource**



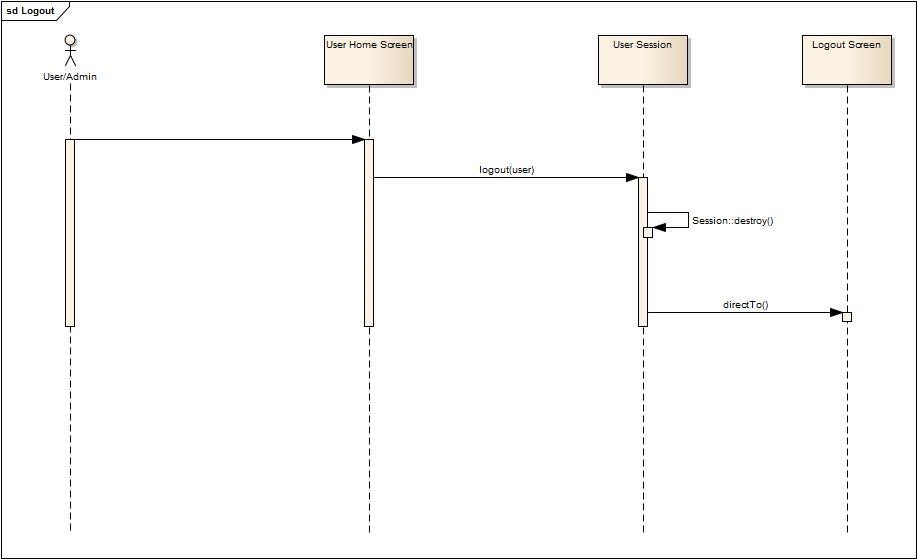
**Figure 31: Sequence Diagram – Add Photos to resource**



**Figure 32: Sequence Diagram – View Photos in Resource**



**Figure 33: Sequence Diagram – Submit Enquiry**



**Figure 34: Sequence Diagram – Logout**

### 6.6 Object Dictionary

Below listed are all the Objects and their methods developed for our Online Portal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Object | Package | Type | Description | Signature |
| create() | Resources | Resource  Management | Method | Show the form for creating a new resource | create() |
| index() | Resources | Resource  Management | Method | Display a listing of the resource | index() |
| store() | Resources | Resource  Management | Method | Store a newly created resource | store() |
| show() | Resources | Resource  Management | Method | Display the  specified resource. | show() |
| edit() | Resources | Resource  Management | Method | Show the form for editing the  specified resource. | edit() |
| update() | Resources | Resource  Management | Method | Update the  specified  resource in storage. | update() |
| destroy() | Resources | Resource  Management | Method | Remove the  specified resource from storage. | destroy() |
| galleries() | Resources | Resource  Management | Method | Show the  gallery related to the resource | galleries() |
| addPhotos() | Resources | Resource  Management | Method | Add photos to the  resource | addPhotos() |
| removePhotos() | Resources | Resource  Management | Method | Remove photos from the resource | removePhotos() |
| index() | Users | User  Management | Method | Show all the users | index() |
| create() | Users | User  Management | Method | Create new  users | create() |
| show() | Users | User  Management | Method | Show the  user details | show($username) |
| store() | Users | User | Method | Store the | store() |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Management |  | details of a user |  |
| index() | Photos | Photo management | Method | Show all the photos | index() |
| store() | Photos |  |  | Store the  uploaded  photos in  storage | store() |
| show() | Photos | Photo management | Method | Show the  photo | show() |
| destroy() | Photos | Photo management | Method | Remove the photo from  storage | destroy() |
| create() | Session | Session  Management | Method | Show the  login page | create() |
| store() | Session | Session  Management | Method | Input and  Authenticate  the login  details | store() |
| destroy() | Session | Session  Management | Method | Logout from  the portal | destroy() |
| create() | Post | Post management | Method | Show the form for creating a new post | create() |
| index() | Post | Post management | Method | Display a listing of the post | index() |
| store() | Post | Post management | Method | Store a newly created post | store() |
| show() | Post | Post management | Method | Display the  specified post. | show() |
| edit() | Post | Post management | Method | Show the form for editing the specified post. | edit() |
| update() | Post | Post management | Method | Update the specified post in storage. | update() |
| destroy() | Post | Post management | Method | Remove the specified post from storage. | destroy() |

## 7.0 Software release report

### 7.1 Usability Test Report

For usability, we will test our application with different people of different age group. We will ask them to navigate through our site and give their review and rating for the major functionalities of the system.

Since we have not implemented the system fully yet so we have not performed many tests until now.

The one test was performed to give an idea how the tests will look like.

**Name:** Gurinder Singh

#### Age: 16 Gender: Male

**Technical Knowledge:** (Internet Savvy) **Overall Rating:**

|  |  |  |
| --- | --- | --- |
| **FUNCTIONALITY** | **REVIEW** | **RATING (OUT OF 6)** |
| **Login System** | The login system has a simple yet eloquent look. Easy to navigate from login to home screen | 5 |
| **Homepage** | The home screen looks very attractive with lots of animation in it | 6 |
| **Contact Us page** | Contact us page is nice looking. The big icon on the left looks very ice | 5 |
| **Dashboard for admin, users** | It is quite easy to navigate through it. May be add some style to it | 4 |
| **Managing resources** | N/A |  |
| **Managing posts** | N/A |  |

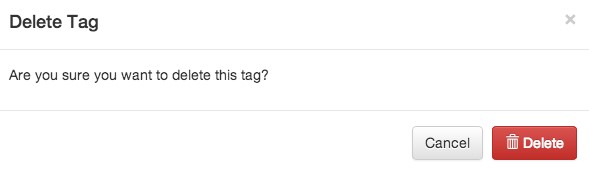
|  |  |  |
| --- | --- | --- |
| **Creating tags for admin** | N/A |  |
| **Photo Upload** | It is very easy to use once you understand how it works. The drag and drop feature is big plus | 6 |
| **Menu of the system** | Needs more style | 3 |
| **Share/Like Resources** | It has the standard procedure as the other sites. | 4 |
| **Tagging photos and resources** | It is quite difficult and keeps on give errors sometimes | 2 |
| **Print and export of data** | Works like piece of cake | 6 |
| **Drag and drop to use multiple resources** | The best feature of the website. Gets rid of the annoying traditional upload file interface of other sites | 6 |
| **Kid friendly design in non-admin section** | Quite attractive | 4 |
| **Ease in navigation** | Very easy | 5 |
| **Overall design** | Needs a bit more designing. Just too simple right now | 4 |
| **Nice to have features** | If I could play some games on the website too and make some friends |  |

### 7.2 System Test Report

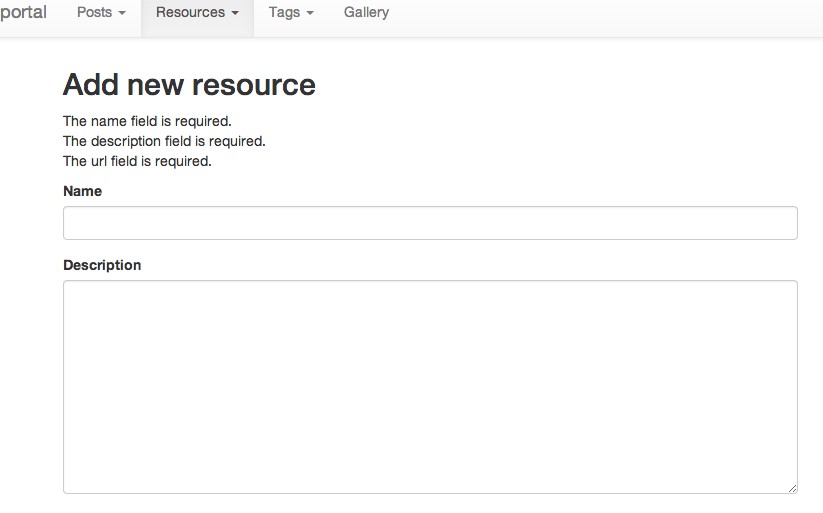
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tes**  **t Id** | **Test Purpose** | **Steps** | **Input Parameters** | **Actual**  **Data**  **Input** | **Expected Output** | **Test Status** |
| **1.0** | User Login | 1.Enter the username and password  2.Press the  Login button | “root” | “ubuntu” | The user should be logged into the website | Successful |
| **2.0** | To test if the users can view the resources | Click the resource icon on the dashboard | N/A | N/A | The user should be able to view all the available resources | Successful |
| **3.0** | To test if the user can view the photos in a resource | 1.Go to the resource page 2.Click on the galleries | N/A | N/A | The user can view the photos | Successful |
| **4.0** | To test if the user can share the resource on social websites | 1.Go to the resource page 2.Click on the share on Facebook button | N/A | N/A | The user can share the resource on Facebook | Successful |
| **5.0** | To test if the user can like the resource on social websites | 1.Go to the resource page 2.Click on the  like on  Facebook button | N/A | N/A | The user can like the resource on Facebook | Successful |
| **6.0** | To test if the user can save the resource as pdf | 1.Go to the resource page 2.Click on the export to pdf button | N/A | N/A | The user can save the resource as a pdf | Successful |
| **7.0** | To test if the admin can create a resource | 1.Go to the resource page 2.Click on the create resource button | Title    Description      Url | Physics    Physics    physics.c om | The admin can create a resource | Successful |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| **8.0** | To test if the user can upload a photo | 1.Go to resource page 2.Select the photo  3.Click on the upload button | N/A | N/A | The photo gets uploaded | Successful |
| **9.0** | To test if the user can upload the photo to a resource | 1. The user clicks on the  “Add photos to resources” link on the gallery page 2. The user drags and drops the photo in a resource | N/A | N/A | The photo gets uploaded | Successful |
| **10.** | The user wants to  View Uploaded  Photos by them | 1. The user clicks on the “Gallery” link from the navigation bar | N/A | N/A | The user can view the uploaded photos by them | Successful |
| **11.** | To test if the admin gets the Email Notification of the photo upload | 1. The user drags and drops the photo in a resource | N/A | N/A | The admin is notified via email about the photo upload | Successful |
| **12.** | The user wants to  Contact  Administrator | The user fills up the required information and enter his/her query 2. The user clicks on the  “Send  Message”  button | N/A | N/A | The information is saved and sent to the admin | Successful |

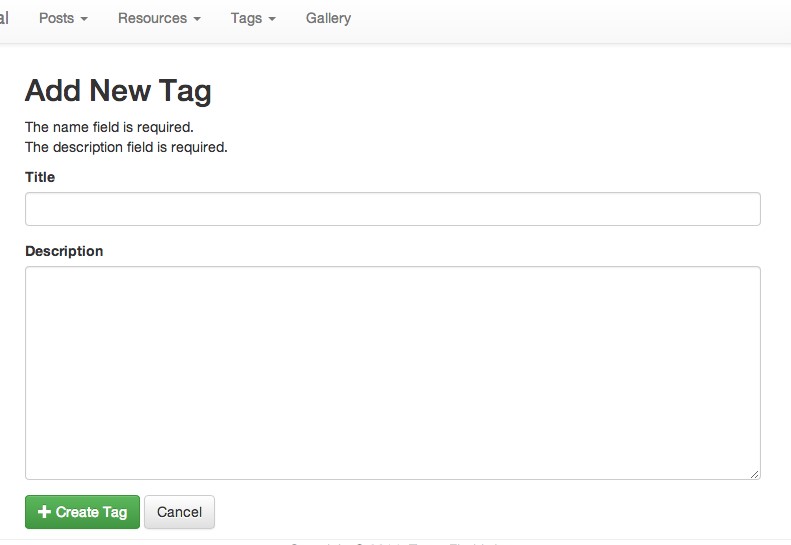
## 8.0 Additional User Interface Design



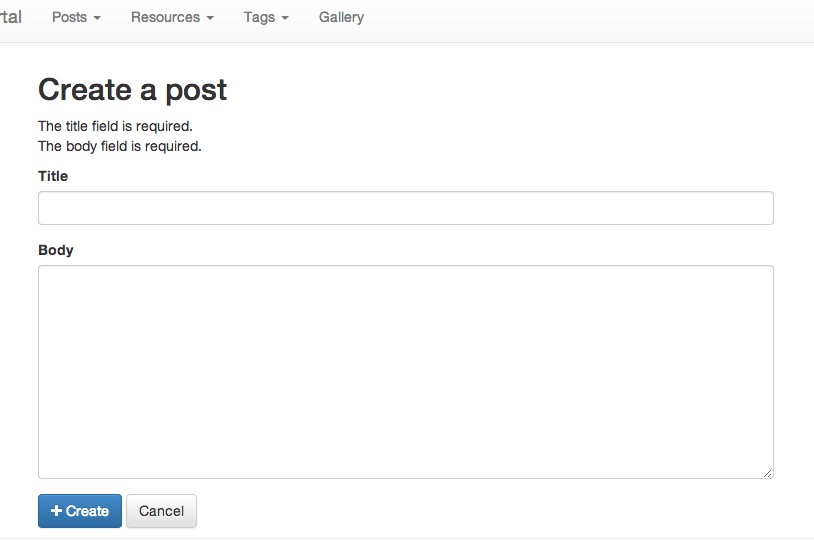
### Figure 35: Confirm Delete Popup Window



**Figure 36: Error message while adding new resource with blank name, description and url field**



**Figure37: Error message while adding new tag with blank title and description**



**Figure 38: Error message while creating new post with blank title and body field**

## GLOSSARY

### Agile

It is a software development method based on iterative and incremental development, in which requirements and solutions evolve through collaboration between selforganizing, cross-functional teams.

### API – Application Programming Interface

In computer programming, an application programming interface (API) specifies how some software components should interact with each other

### Architectural Diagram

A graphical representation of all the components of the system, the connections and interfaces between them.

### Bootstrap

Bootstrap is a free collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

### Class Diagram

It is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

### CRUD

A computing term meaning Create, Read, Update and Delete operations.

**CSS-** **Cascading Style Sheets**

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language

#### ER – Entity Relationship

A graphical representation of entities and their relationship to each other, typically used in computing in regard to data within a database.

#### HTML- HyperText Markup Language

HTML or HyperText Markup Language is the standard markup language used to create web pages.

### HTTP

The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.

### LAMP

The acronym LAMP refers to first letters of the four components of a solution stack, composed entirely of free and open-source software, suitable for building high-availability heavy-duty dynamic web sites, and capable of serving tens of thousands of requests simultaneously.

#### Laravel

Laravel is a free, open source PHP web application framework, designed for the development of MVC web applications. Laravel is released under the MIT license, with its source code hosted on GitHub

#### MySQL

It is the world's second most widely used open-source relational database management system (RDBMS).

**Navbar**

It refers to the navigation bar of the webpage.

#### PDF - Portable Document Format

Portable Document Format (PDF) is a file format used to present documents in a manner independent of application software, hardware, and operating systems.

#### PHP - Hypertext PreProcessor

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language.

**jQuery** jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML.

### UNIX

Unix is a multitasking, multiuser computer operating system that exists in many variants. The original Unix was developed at [AT&T's](http://en.wikipedia.org/wiki/American_Telephone_%26_Telegraph) [Bell Labs](http://en.wikipedia.org/wiki/Bell_Labs) research center by Ken Thompson, Dennis Ritchie, and others

**WordPress**

WordPress is web software you can use to create a beautiful website or blog.

#### WireFrame

A website wireframe, also known as a page schematic or screen blueprint, is a visual guide that represents the skeletal framework of a website.

## INDEX

**A**

|  |  |  |  |
| --- | --- | --- | --- |
| Admins | |  | 7 |
| Agile | |  | 15 |
| API – Application Programming Interface | |  | 11 |
| Architectural Diagram  **B** |  |  | 8 |
| Bootstrap  **C** |  |  | 13 |
| Class Diagram |  |  | 51 |
| Create Resource |  |  | 17 |
| Create Tags |  |  | 17 |
| CSS- Cascading Style Sheets  **D** |  |  | 13 |
| Drag And Drop |  |  | 17 |
| Domain Model  **E** |  |  | 51 |
| Email Notification |  |  | 17 |
| ER – Entity Relationship  **F** |  |  | 52 |
| FontAwesome  **H** |  |  | 13 |
| HTML- HyperText Markup Language | |  | 13 |
| HTTP  **I** | |  | 11 |
| Integration Testing  **J** | |  | 31 |
| jQuery | |  | 13 |

**L**

|  |  |  |  |
| --- | --- | --- | --- |
| Laravel | | | 12 |
| Like Resources  **M** | | | 16 |
| MySQL  **N** | | | 12 |
| Navbar | | | 31 |
| Object Dictionary  **P** | | | 58 |
| PDF - Portable Document Format | | | 11 |
| PHP - Hypertext PreProcessor | | | 12 |
| PHPUnit | |  | 30 |
| Print Resources  R | |  | 17 |
| Relational Database Table | |  | 52 |
| **S** |  |
| Sequence diagram |  |  | 53 |
| Share Resources |  |  | 13 |
| Students  **T** |  |  | 7 |
| Tag Photos |  |  | 17 |
| Teachers  **U** |  |  | 7 |
| Usability Test |  |  | 60 |
| Unit Testing |  |  | 30 |
| Upload Photos |  |  | 16 |
| Use Cases |  |  | 44 |
| User Stories |  |  | 15 |
| UNIX  **W** |  |  | 11 |
| WordPress |  |  | 12 |

WireFrame 38

1. http://en.wikipedia.org/wiki/User\_story [↑](#footnote-ref-13)
2. MVC - Model View Controller [↑](#footnote-ref-14)