

Software Development Project

Deliverable 4

Group 20

Colm Le Gear – 15148823

David Kearney – 15169235

Liam Horgan – 14164736

Ryan Harris – 13150154

Contents

Introduction	4
Major Processes & Feature List.....	5
Login and Registration	5
Create Task.....	5
Task List.....	5
View Task Details	6
Reputation	6
Profile	6
Ban User	6
Flagged Tasks Page	6
Un-publish Task.....	7
Search	7
Tags	7
Flagging Tasks.....	7
Claimed Tasks	7
My Tasks.....	8
User Tasks.....	8
Potential Additions.....	9

Other Resources Used.....	9
Website Template.....	9
Bootstrap	9
Glyphicon Font	9
JQUERY	9
ER Diagram.....	10
Appendix	11
Site Files.....	11
Database Create Statements	13

Introduction

This report presents the final state of the CS4014 Software Development project as completed by Group 20.

This deliverable includes an outline of all major processes and features, identifying the files involved, and does the same for additional features present on the site. There is also a short section discussing worthwhile features that could potentially have been added if the time had been available.

The project can be cloned via github : <https://github.com/CS4084/CS4084>

Additional resources used in the creation of the site are listed, where we have relied on free third-party code for certain elements of the site.

Appendixes are included listing all main files used in the user and admin sites, with a brief description of each file. They also include a database Entity Relationship diagram and create statements for all database tables, to indicate foreign keys and data types.

Major Processes & Feature List

Login and Registration

- Login form [**index.php**] requiring email address and password.
 - If site visitor is not a member they can click a link that will bring them to the register page.
 - User will not be allowed to login if they have been banned.
- Site visitor can create an account with [**register.php**].
 - User can enter their details such as name, student ID, email, password to be stored in the database.
 - If user's data is entered correctly (e.g. matching passwords and valid UL email address) then they will be redirected back to the login page and shown a registered successfully message.

Create Task

- Registered users can create a task using [**newtask.php**].
 - This allows users to enter their task details and checks their input for errors. If their task is error free then the data are stored in the **task** table in the database and they are redirected to the task page for their new task.

Task List

- On the homepage [**dashboard.php**] displays a task list showing all available tasks to the user. A separate script [**displaytasks.php**] is used to create the HTML for displaying the tasks.
 - **All Tasks** – This will show details of all open tasks that are available.
 - **Subscribed Tags** – This will show tasks that include tags that the user is subscribed to.
 - **Most Viewed Tags** – This shows tasks that the user has viewed. The more they've viewed a task the higher up it will appear in this list.
 - **My Subject Stream** – This shows tasks that belong to the user's subject stream (faculty).

If the task is past its deadline to be claimed it will not be shown on the dashboard and instead will be added to the **unpublished_tasks** table.

View Task Details

- **[task.php]** is used when a user clicks on a task to view more task details.
 - This displays all task details.
 - Allows user to claim task.
 - Any user can download a sample of the document, but the user that has claimed the task will be able to download the full document.
 - Allows user to complete task using by submitting a review of the task which is handled by a separate script **[completetask.php]**.
 - Allows user to un-publish their own task and moderators to un-publish flagged tasks.
 - Users can cancel a task that they have claimed from this page. The task will be added back to the list of open tasks and their reputation score will be decreased by 15 points.
 - If a task owner views their completed task they will be able to see the review that has been left by the claimant and decide whether they are happy or not with the review. If they choose happy, the claimant will be awarded 5 reputation point, else they will be deducted 5 reputation points.

Reputation

- Reputation score is stored as **repScore** in the **users** table in the database. This score can be increased by claiming tasks, flagging tasks and receiving positive reviews on completed tasks. It can be decreased by failing to complete tasks, cancelling tasks and receiving a negative review for completing a task. If a user achieves a score of 40 or greater they are considered a moderator and have access to features such as viewing flagged tasks, un-publishing tasks and banning users.

Profile

- **[profile.php]** displays user's details
 - A moderator has the ability to ban user using **[ban.php]** from the profile page. Note: The ban user button is not visible to users who are not moderators.
 - Users can click a link on this page that redirects them to the user's available tasks. If the user is on their own profile page then they will be redirected to **Open Tasks** in the **My Tasks** section. This uses **[usertasks.php]**.

Ban User

- A moderator can ban a member by viewing their profile and clicking the "Ban User" button which uses **[ban.php]**. Their email will be blacklisted from the site so that they won't be able to login or register using that email again and the bannedUser ID and moderator ID will be stored in the **user_banned** table.

Flagged Tasks Page

- Only moderators can access this section. They will be able to see a list of all flagged tasks.
- Moderators can un-flag a task from this page by clicking the link.
- If a moderator visits the task page of a flagged task they will be able to unpublish it.
- Moderators are able to ban users who have submitted flagged tasks by visiting their profile page.

Un-Publish Task

- Moderators can un-publish flagged tasks using [**task.php**]. The un-published task is removed from open tasks and added to the **unpublished_tasks** table in the database.
- Users can un-publish their own tasks from the [**task.php**] page.

Search

- Searches are performed using [**search.php**]
 - This php file runs an SQL query that returns available tasks (not claimed, not un-published, not completed) that match the search string.
 - [**searchpredict.php**] is used as the source for the JQuery autocomplete function in the search box. This queries the database to find the matching results and display them underneath the search text box.

Tags

- Subscribing and unsubscribing to tags is handled by [**tag.php**]
- Users can subscribe to a tag by clicking the tag on the task page or the task list.
- Users can also subscribe to tags by searching for an existing tag on the dashboard.
- [**tagsearch.php**] is used as the source for the JQuery autocomplete function on the tag search box. This queries the database for an existing tag matching the users query and displays them under the text box.
- If the user tries to subscribe to a tag that doesn't exist an appropriate error message will be displayed on the dashboard.
- Users can unsubscribe from a tag by clicking the tag name in the 'subscribed tags' section on the dashboard.

Flagging Tasks

- Users can flag a task by clicking the "Flag as inappropriate" link in the task list. Users are awarded 2 reputation points for flagging a task.
- This is handled by the [**flag.php**] script.
- When a task is flagged it will be added to the **task_flagged** table, and will appear in the "Flagged Tasks" list for moderators to review.
- Users are not able to flag their own tasks.

Claimed Tasks

- Users can view a list of all tasks that have been claimed by them in the [**claimedtasks.php**] page.
- If the task is past its completion deadline it will not be displayed and instead will be added to the **unpublished_tasks** table. The claimant's reputation will also be reduced by 30 points.

My Tasks

- Users can view a list of all the tasks they have created by clicking the “My Tasks” link. This is handled by `[usertasks.php]`.
- Users can view their tasks in four separate categories:
 - **Open Tasks** – All currently open tasks submitted by the user.
 - **Claimed Tasks** – Tasks the user has submitted that have been claimed by other users.
 - **Completed Tasks** - Tasks the user has submitted that have been completed by other users.
 - **Unpublished Tasks** – Tasks that have been unpublished by either the user or a moderator.

User Tasks

- Users can view a list of all the open tasks submitted by a user by clicking the link on their profile page. This is handled by `[usertasks.php]`.

Potential Additions

[Editing and Deleting Profiles](#) – Our group had hoped to add functionality that would allow users to edit (change email, password, etc) and delete their profiles. However it wasn't on the top of our agenda and it ended up being a feature that we decided not to include in the interest of time.

[Recaptcha](#) – We had hoped to feature a Recaptcha to prevent spammers from abusing forms but it ended up being a feature we didn't devote enough time to and ultimately had to be left out.

Other Resources Used

The following are various resources used in the creation of our site. Files for these resources

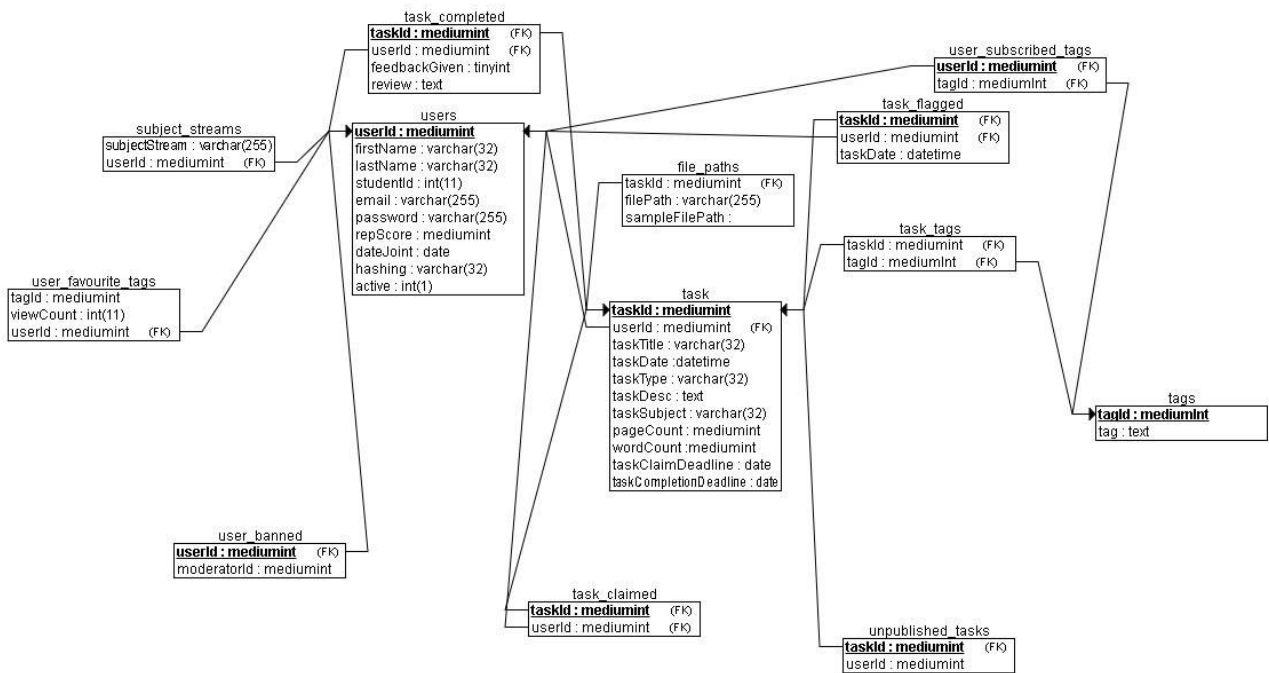
[Website Template](#) – The original website template was taken from (www.quackit.com/html/templates/). Our website is based on this template; however we have modified it to suit our needs. Custom CSS for the site is contained in the custom.css file.

[Bootstrap](#) – The front end of the website was designed using Bootstrap, which is a CSS and JavaScript framework for creating responsive websites.

[Glyphicon Font](#) – Is used to display icons throughout the site. This is included in Bootstrap.

[jQuery](#) – Is used for certain interactive elements of the site, such as the drop down menu for navigation when viewing the site on mobile and for the search and tag auto completion. A JQuery CSS file is also used for the auto completion styling. The code for the JQuery autocomplete and the basic structure for the [tagsearch.php] and [searchpredict.php] files was taken from (www.codexworld.com/autocomplete-textbox-using-jquery-php-mysql/)

Entity Relationship Diagram



Appendix

Site Files

The following are all display pages and scripts used in the user section of the site. CSS and JS file are not listed in detail, though relevant CSS files are mentioned in the display page listing. No external JavaScript files were created by the group for this project, so all JavaScript files are part of 3rd-party components, such as Bootstrap and JQuery.

FILE	DESCRIPTION
ban.php	Script for banning users via their profile page [profile.php].
claimedtasks.php	Script for user to view their claimed tasks.
completetask.php	Script for completion of a task. Inserts task completion data such as the review, into task_completed table in the database.
config.php	Script to connect to database and sets it to variable \$db .
dashboard.php	Contains SQL statements to display all available tasks, subscribed tags tasks, most viewed tasks and faculty/subject tasks.
displaytasks.php	Creates HTML elements for displaying tasks and optionally removes tasks if they are past the claim deadline.
flag.php	Handles flagging tasks as inappropriate. Does not allow a user to flag their own task.
flaggedtasks.php	Lists all tasks that have been flagged by users for moderators to review.
index.php	Handles the user login. Doesn't allow banned users to log in.
logout.php	Script that logs the user out and destroys their session.
newtask.php	Page to create a new task. Users input the task information and a new task is added to the task table if the information is correct.

profile.php	Displays a user's profile information. Moderators can ban user's that have had their tasks flagged from this page.
register.php	Page to handle registration of a user. This page also sends a verification email to the user's email.
search.php	Page that displays the search results for a user's query.
searchpredict.php	Script that handles the auto completion for the search box.
tag.php	Handles subscribing and unsubscribing to tags.
tagsearch.php	Script that handles the auto completion for the tag search box.
task.php	Page Where the user can view details about a task. Moderators and task owners can un-publish tasks. Users can claim and complete tasks from this page.
unflag.php	Handles unflagging a task for moderators.
usertasks.php	Displays a list of open tasks for a user, or a list of all tasks sorted into categories for the logged in user.
verify.php	Script for verifying the user's email address via the link sent to their email.

Database Create Statements

```
CREATE TABLE `file_paths` (  
  `taskId` mediumint(9) NOT NULL,  
  `filePath` varchar(255) NOT NULL,  
  `sampleFilePath` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `subject_streams` (  
  `userId` mediumint(9) NOT NULL,  
  `subjectStream` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=COMPACT;
```

```
CREATE TABLE `tags` (  
  `tag` text NOT NULL,  
  `tagId` mediumint(9) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `task` (  
  `taskId` mediumint(9) NOT NULL,  
  `userId` mediumint(9) NOT NULL,  
  `taskTitle` varchar(32) NOT NULL,  
  `taskDate` datetime NOT NULL,  
  `taskType` varchar(32) NOT NULL,  
  `taskDesc` text NOT NULL,  
  `taskSubject` varchar(32) NOT NULL,
```

```
`pageCount` mediumint(9) NOT NULL,  
`wordCount` mediumint(9) NOT NULL,  
`taskClaimDeadline` date NOT NULL,  
`taskCompletionDeadline` date NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `task_claimed` (  
  `taskId` mediumint(9) NOT NULL,  
  `userId` mediumint(9) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `task_completed` (  
  `taskId` int(11) NOT NULL,  
  `userId` int(11) NOT NULL,  
  `feedbackGiven` tinyint(1) NOT NULL DEFAULT '0',  
  `review` text NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `task_flagged` (  
  `taskId` mediumint(9) NOT NULL,  
  `userId` mediumint(9) NOT NULL,  
  `taskDate` datetime NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `task_tags` (  
  `taskId` mediumint(9) NOT NULL,  
  `tag` varchar(255) NOT NULL
```

```
`taskId` mediumint(9) NOT NULL,  
`tagId` mediumint(9) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `unpublished_tasks` (  
`taskId` mediumint(9) NOT NULL,  
`userId` mediumint(9) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `users` (  
`userId` mediumint(9) NOT NULL,  
`firstName` varchar(32) NOT NULL,  
`lastName` varchar(32) NOT NULL,  
`studentId` int(11) NOT NULL,  
`email` varchar(255) NOT NULL,  
`password` varchar(255) NOT NULL,  
`repScore` mediumint(9) NOT NULL DEFAULT '0',  
`dateJoined` date NOT NULL,  
`hashing` varchar(32) NOT NULL,  
`active` int(1) NOT NULL DEFAULT '0'  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=COMPACT;
```

```
CREATE TABLE `user_banned` (  
`moderatorId` mediumint(9) NOT NULL,  
`bannedUserId` mediumint(9) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `user_favourite_tags` (
```

```
  `userId` mediumint(9) NOT NULL,
```

```
  `tagId` mediumint(9) NOT NULL,
```

```
  `viewCount` int(11) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `user_subscribed_tags` (
```

```
  `userId` mediumint(9) NOT NULL,
```

```
  `tagId` mediumint(9) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```