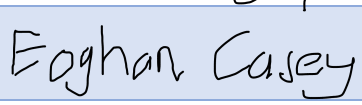
Design Documentation Deliverable: Academic Paper Proofreading Website

Date 03/03/2017

Group Name/Number Enter Name Here xD/Group 4

Student ID	Student Name	Signature
15167798	John Juele	
15159973	Sophia Colgan	
15160513	Eoghan Casey	
15162788	Donagh Kelleher	

Please use the checklist below to ensure that your report contains all the items required.

- ✓ **Purpose of the website (a brief introduction)**
- ✓ **User Types**
- ✓ **Justification of Functionality**
- ✓ **Detailed Description**
- ✓ **Appendices:**
  - ✓ **Web Pages (mock ups of main pages)**
  - ✓ **Process Descriptions**
  - ✓ **Table Design**
- ✓ **Technologies**

1.0 Website Purpose

The purpose of this website is to provide students with a means to have their academic documents (theses, dissertations, research papers etc.) proofread by fellow students and university staff. The user will have the ability to post a document onto the website including a small description of the task and other users can claim the task for proofreading.

This website aims to assist students in their progress and development throughout their 3rd level education. Due to the stressful and demanding nature of university life, students need as much assistance that they can possibly attain. The website helps alleviate the tension that a typical student encounters by providing them with a medium to ask for help from their peers.

One of the biggest benefit that this website can provide is that it encourages its users to cultivate confidence in their work. Students can have confidence in the fact that the other user, who may most likely be proficient in their field of work, has proofread the student's document. This in turn can increase one's assurance in the quality of the document that the student will eventually submit.

1.1 User Types

Users of this website can be of three different types:

1. Task Owner
2. Task Claimant
3. Moderator

Task Owner

A task owner is a user that uploads a document onto the site for proofreading. The website would provide these users with the opportunity to create a task, fill in the necessary details and the website will upload the task to the task stream so that others can claim it.

Task Claimant

A task claimant is a user who has taken the responsibility to proofread a document belonging from the task owner. A user's main page will contain a task stream which would display all the available tasks to claim. Upon claiming a task, the claimant can request the full document from the document/task owner and the website will automatically send an email to the task owner.

Moderator

A moderator is basically a task claimer and/or task claimant with additional privileges. A user becomes a moderator whenever they obtain 40 reputation points. As a moderator, the user has the responsibility to maintain order and control in the website. The site can provide the mod a list of flagged tasks (Task that have been flagged by other site users for being inappropriate) and can choose to unpublish a task and/or ban the user owning that task.

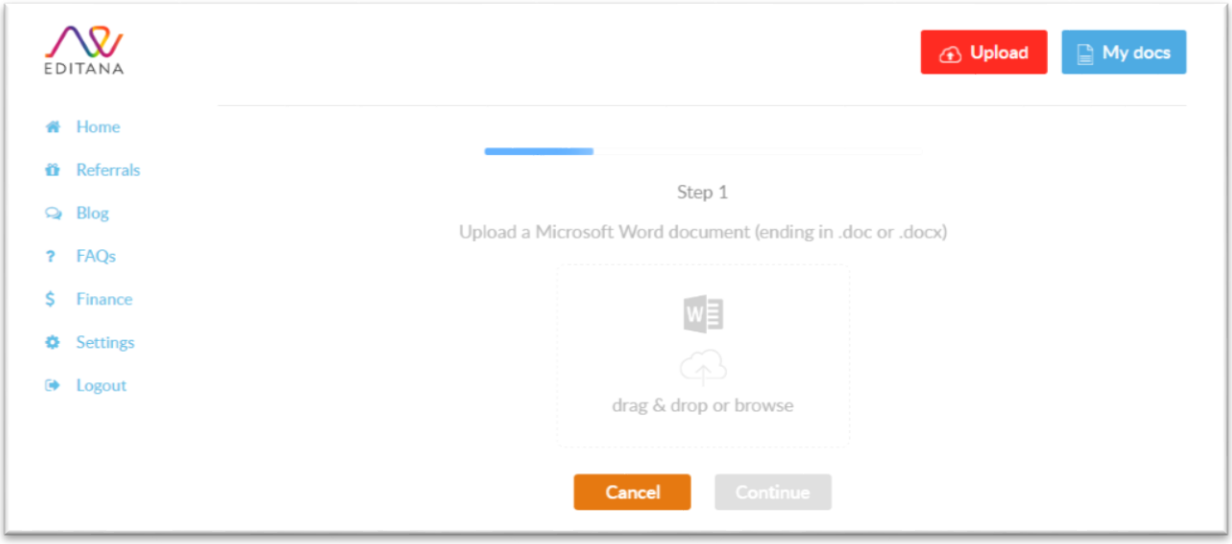
1.2 Justification of Functionality

The functionality provided by the website for the user is quite simple as most of the processes can be done with just a couple of link presses and form filling. Upon entering the website, the user can either register or sign into the website by filling in the displayed form. After submitting the form, the user will earn the access to their main page.

On the main page, the user will have the opportunity to utilize the main features/functionality of the website. For claiming a website there will be a task stream/list of all available tasks. The display of this stream is similar to how social media websites like Facebook.com and Twitter.com displays the “News Feed” to its users (Twitter snippet on left).

A claimant can click on a task in the task stream and view the details of the task. If they are willing to proofread their chosen document, they can request the full document from the owner and the website will automatically send an email to the owner.

Publishing a task is simple process. The user fills in a form to fill the details about their task. The owner would be required to upload a small preview of their paper to present to potential claimants. We have taken this feature as inspiration from the website www.editana.com where their users can upload files to present to other users of their site.



1.3 Potential Ramifications

As we have outlined in Section 1.0, a consequence that we greatly appreciate from this website is how it may ease the life of a typical student. As students, we can easily see the potential practicality of this website. It can drastically cut down on the workload of the students especially on the area of analysis and refinement. A student can make very effective use of his available time in increasing the quality and excellence of one's work. "Quality is greater than quantity" is a very popular saying and is clearly appropriate to this website's theme.

This website provides a great service to students and we greatly believe that it is a step forward in the improvement of relations between students and education. Students can be more confident in their work having obtained a second opinion of a person with greater proficiency in that certain field. The site can provide students with a way to mitigate the pressure of producing academic documents and can assist students in their success in university.

There are a couple of issues, however, that can potentially occur within the website. A very serious issue that we foresee is the abuse of the moderator system. It is inevitable that there may be users of the site with bad intentions that have the objective to become moderator and obtain power over other users. Currently, it is too easy to become moderator and upon earning this authority, a moderator can easily ban a user by just flagging a task themselves and removing the user their right to use the website. This is certainly a serious matter and we plan on increasing the difficult of achieving moderator status and revisiting and improving upon the punishment system.

2.0 Detailed Description

When the user first enters the website they will arrive to the landing page which may appear similar to the designed mock up pages (see Appendix 1: Web Page Mock-ups/Ideas). This page may display a brief overview of the website and provides the user with the opportunity to either register themselves or log-in as an existing user.

P1 (Login) occurs on the landing page of the website. If the user is already an existing user (data on user is stored in the database), they may enter their email address/Student ID and password to gain entrance to their home page. Their input is the validated. If their input is equal to their credentials stored in the database, the user is directed to their home page. Otherwise they are denied access to their profile but will be given a chance to re-enter their credentials.

P2 (Registration) may also take place in the landing page (see Appendix 1: Web Page Mock-ups/Ideas). Here the user may fill in the displayed input boxes to store their details into the database. Their input is also checked according to the appropriate input boxes. If the input is incorrect, the user is given the chance to amend their mistakes. Their ID may be checked to figure out if the user already exists. If this is the case, the site redirects the user back in order to log in as an existing user. Otherwise the user is redirected into their new home page.

P3 (Create a task) is executed after the user clicks on the link that sends them into the Task Creation Page. Here the user is presented with a form in which the user must fill in (see Appendix 3: Processes for form details). The user input is validated and makes sure the data is correct. The user is then allowed to upload 3 pages of their document as a preview to the claimants. If everything is complete, the data is inserted into the database and the task is displayed on the task stream on the home pages of all users.

P4 (Claim a task) takes place whenever a user clicks on a task located in the task stream of their main page. They are then presented with all the details the task owner has inputted for that task, including the three-page preview of the paper. The user will then have a choice whether they would like to accept to claim the task or decline it. If they choose to claim the task, the user will be presented with a text area to input a message for the task owner, mainly to request the entire document for proofreading. When they press on the send button, the claimant's message will be sent to the owner via e-mail.

P5 (Flag Task) happens when a user discovers a task in their task stream that they suspect to be inappropriate. That particular task will be flagged in the database. After flagging the task, the flagger will earn 2 reputation points.That task can then be reviewed by the moderator. If the moderator decides that the task should exist, he has the power to unpublish it.

Appendix 1: Web Page Mock-ups/Ideas

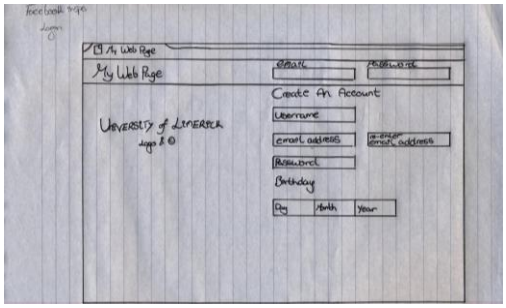


Figure 1: Landing Page

- Facebook inspiration

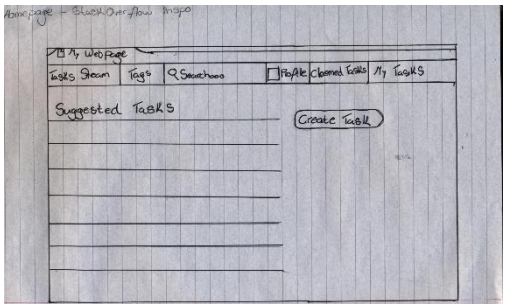


Figure 2: User's Home Page (Idea 1)

- StackOverflow inspiration

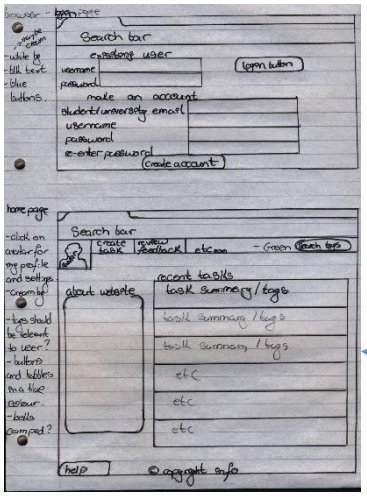


Figure 3:

Landing Page (Idea 2)

User's Home Page (Idea 2)

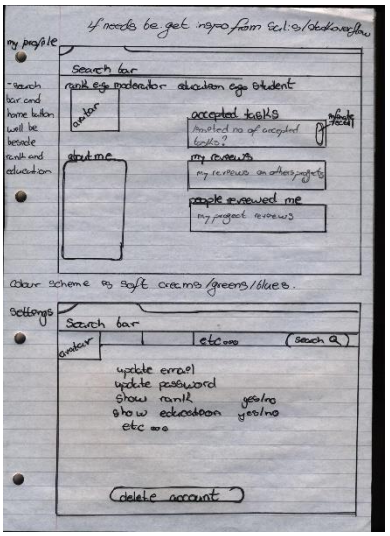


Figure 3:

User Profile

Profile update

Appendix 2: Database Tables

 = Primary Key

Table name: Users

Primary Key: User_ID

Description: The Users table stores the information about each user of the site such as their name, email, course of study and their hashed password.

Users

Table Attributes	**Data Type**	**Example**
User_ID	Integer	15167798
FirstName	Text	John
LastName	Text	Juele
Email	Text	15167798@studentmail.ul.ie
Subject	Text	Computer Systems
Pass_word	Text	adkja8774hac

- **User_ID** is the attribute to identify each website user. The user's UL student ID is the data stored in this column.
- **FirstName** is the forename of the user.
- **LastName** is the surname of the user.
- **Email** is the UL e-mail address of the user.
- **Subject** is the course of study the user is currently pursuing.
- **Pass_Word** is the password that will be used to login to their homepage.

Table name: Tasks

Primary Key: Task_ID

Description: The Tasks table is used to store information on each created task such as the **Type**, **Description**, **Page** and **Word Counts** etc.….

Tasks

Table Attributes	Data Type	Examples
Task_ID	Text	bAh1yd98
Type	Text	Dissertation
Description	Text	Lorem ipsum dolor sit amet….
Tags	Text	computer
Pages	Integer	5
Words	Integer	2000
Claimed	Boolean/TinyInt	FALSE
Flagged	Boolean/TinyInt	FALSE

- **Task_ID** is used to identify each task. It is the primary key of the table and therefore must be unique.
- **Type** displays what type of document is being published
- **Description** stores the description that the owner has inserted so that claimants can get a glimpse of what the task is about.
- **Tags** helps categorize each task
- **Pages** stores the number of pages the task has.
- **Words** stores the number of words the task has.
- **Claimed** refers to whether or not the task has been claimed. Data for this field is not inserted on creation of the task and is automatically defaulted as FALSE.
- **Flagged** refers to whether or not the task has been discovered as inappropriate. Data for this field is not inserted on creation of the task and is automatically defaulted as FALSE.

Table name: Claims

Primary Key: Task_ID

Description: The Claims table is used to store information on what task has been claimed, who owns the task and who claimed the task.

Claims

Table Attributes	Data Type	Examples
Task_ID	Text	bAh1yd98
Owner	Integer	15167798
Claimant	Integer	11111111

- **Task_ID** is used to identify the claimed task. It references to the Task_ID from the Tasks table.
- **Owner** refers to the User_ID of the owner of the task.
- **Claimant** refers to the User_ID of the user that claimed the task.

Table name: File_Paths

Primary Key: Task_ID

Description: The File_Paths table is used to store information on the file type and the location of where the file has been stored on the server.

Claims

Table Attributes	Data Type	Examples
Task_ID	Text	bAh1yd98
File_Type	Text	PDF
Path	Text	./folder/file.php

- **Task_ID** is used to identify the claimed task. It references to the Task_ID from the Tasks table.
- **File_Type** refers to type of the preview file.
- **Path** refers to the file directory path to the location of the file on the server.

NOTE: THIS TABLE IS SUBJECT TO CHANGE OR MAY BE REMOVED ENTIRELY. FILE UPLOADING IS CURRENTLY BEING RESEARCHED

Table name: Reputation

Primary Key: User_ID

Description: The Reputations table is used to keep account on the reputation points each user has and whether or not a user is a moderator.

Reputations

Table Attributes	Data Type	Examples
User_ID	Integer	15167798
Rep	Integer	40
Mod	Boolean/TinyInt	TRUE

- **User_ID** is used to identify the user. It references to the User_ID from the Users table.
- **Rep** refers to the accumulated reputation points the user has gathered.
- **Mod** determines whether or not the user is a moderator.

Table name: Deadlines

Primary Key: Task_ID

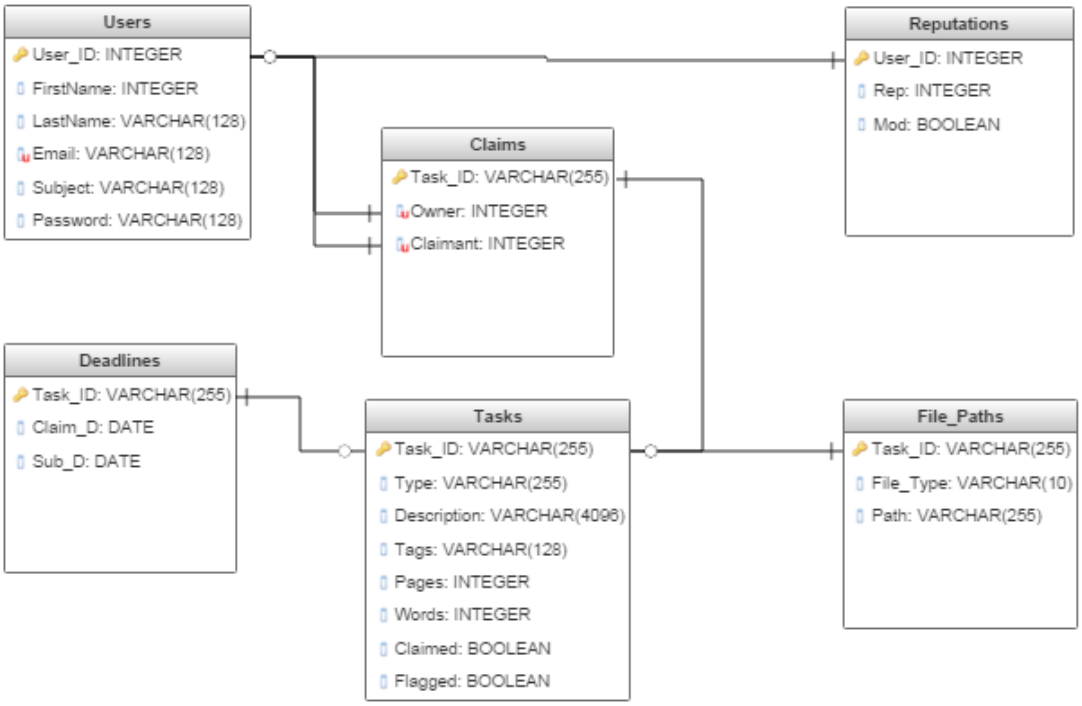
Description: The Deadlines table is used to store information on the deadlines of the task, deadlines such as the date at which the task expires from the task stream and the date at which the task is due to be resubmitted back to the owner.

Claims

Table Attributes	Data Type	Examples
Task_ID	Text	bAh1yd98
Claim_D	Date	2017-03-02
Sub_D	Date	2017-03-30

- **Task_ID** is used to identify the claimed task. It references to the Task_ID from the Tasks table.
- **Claim_D** refers to the date at which the task is due to be removed from the task stream if the task was not claimed.
- **Sub_D** refers to the date at which the task is due to be submitted back to the task owner.

Relationships within the database



- Each user can only have one User_ID. Each user can only have 1 record of their reputation and the task that they claim/own.
- Each task can only have one Task_ID. Each Task_ID has only 1 record of deadlines, its file path and claimants/owner.

Appendix 3: Processes

Process Number	P1
Process Title	Login
Brief Description	Validates user details and grants or denies access to their profile.
Inputs	User email/User ID and password.
Detailed Description	User will enter either their student email (UL) or student id into the first field. The user will then enter their own password into the second field. The user will then press enter or login and it will validate the two inputs. If they are valid it will bring them to the home screen. If they are, not it will inform them the login information is invalid and deny them access. If the user does not already have an account there will be an option to register and direct them to that page.
Output	Directs the user to the home page or to Registration

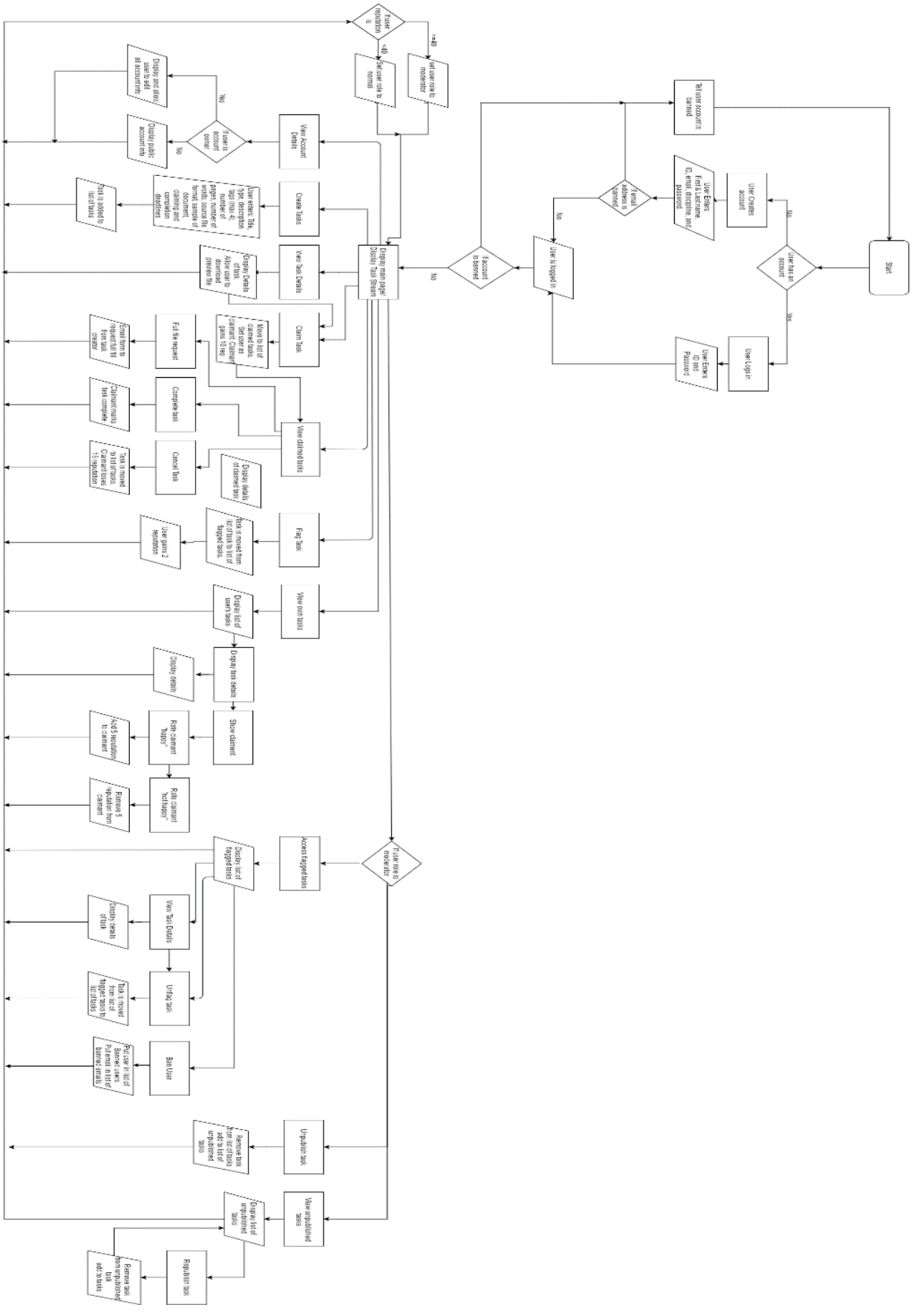
Process Number	P2
Process Title	Registration
Brief Description	Accepts registration requests from users. Directs user to their new profile if supplied information is correct or rejects request if information supplied is incorrect.
Inputs	User information (First name, Last name, student or staff ID, email, discipline of study, password and re-enter password)
Detailed Description	User will enter their information into all the fields above. If all the information is valid for each field the user will have registered successfully. If any of the information is invalid it will tell the user that their registration was unsuccessful and which fields did not provide the correct information.
Output	Registers the user and directs the user to the home page

Process Number	P3
Process Title	Create Task
Brief Description	Allows user to create a proofreading task. The user will fill out all the fields provided and creates a task if the information is all valid.
Inputs	Information on the task being created (Title, Task type, brief description, Description Tags, number of pages, number of words, source file format, 3 pages for sample, Deadline for claiming task, Deadline for completing task)
Detailed Description	The user must fill in all of the information needed in the fields provided. The process will then check and validate all the information provided when completed. If any of the information is invalid the creation of the task will fail and the user will be told which fields they provided invalid information in. If all information is valid the task will be created.
Output	The user will be informed that their task has been created and will be given the option to return to the home page.
Process Number	P4
Process Title	Claim task.
Brief Description	The user will be able to select and claim tasks.
Inputs	User input on viewing and claiming tasks
Detailed Description	From the homepage the user will see a stream of available tasks, filtered to their field of study. The user will be able to click on these tasks and see more details about the task such as description, number of pages, associated tags, etc. The user will also be able to download the preview of the document which would be attached to the task. If the user wishes to they are then able to claim the task and that task will be taken off the stream.
Output	The user will be taken to a page will all their claimed tasks.

Process Number	P5
Process Title	Flag tasks

Brief Description	User are able to flag any tasks that they deem inappropriate.
Inputs	Tasks already submitted by users for review.
Detailed Description	Users are also given the option to flag tasks if they deem the content inappropriate. When they flag a task they will be rewarded 2 marks towards their reputation score. If the user is also a moderator, they will have access to all tasks that have been flagged and can un-publish the task.
Output	Reputation score for flagging a task.

Process Flowchart



3.0 Technologies

- **Windows 10 Operating System**
  - The main operating used for website development.
- **Atom/Notepad++**
  - The text editors used to write code for the creation of the website.
- **HTML5 & CSS**
  - The languages used to create the look and feel of the website. Html5 is used to generate the structure/skeleton of the website, while CSS helps create the appearance of the website.
- **PHP**
  - A server-side scripting language. This language will help create dynamic webpages. Is is the key language that will provide a connection to our databases using the PDO library.
- **Javascript**
  - Javascript will be used to add extra effects and functionality to the website. We will be using this language to validate our inputs and add interactivity throughout the website.
- **MySQL & PHPMyAdmin**
  - A relational database management system. By mean of PHPMyAdmin, we would use these to create our database.
- **Git & GitHub**
  - https://github.com/Coding-Chicken-Lover/Web-Development-Project
  - Our main version control system.