Design Documentation Deliverable

# Date 3rd March 17

# Group Name/Number Group 16

|  |  |  |
| --- | --- | --- |
| **Student ID** | **Student Name** | **Signature** |
| **16118928** | **Matthew Riley Neil Timothy Paul McCarthy Richard O’Connell** |  |

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**

**Proofreading Website - PlayByPeer**

1. **Purpose of Website**

The purpose of PlayByPeer is to provide an interactive web platform to facilitate proofreading services for users. The service is available for all academic papers, such as theses, dissertations, assignments and research papers.

Currently if students wish to avail of a proofreading/review service, they may find themselves restricted in their ability to do so; either financially, or by subject specialisation. By creating a dedicated proofreading platform, such as PlayByPeer, users can experience the benefits of a collaborative effort from a community consisting of their peers.

In its simplest form, PlayByPeer will give registered users access to a secure platform on which they can upload, browse, or select academic documents which require proofreading or review.

* 1. **User Types**
     1. Students (basic user). Once registered, students will have the ability to upload a sample of their documentation which they require to be proofread/reviewed by the user community. Additionally, all students will be able to browse a catalogue of documents (tasks) which have not yet been ‘claimed’ by other users. If a user ‘claims’ a task, the site provides a platform to facilitates contact between the relevant users; i.e. the creator of the task and the claimant of the task.
     2. Moderators (enhanced user). In addition to the functions available to a student user, moderators have the additional functionality of being able to view a list of tasks which have been ‘flagged’ by other users as potentially being inappropriate; and remove such tasks if deemed necessary. Moderators will also have the ability to permanently ban other users from accessing the website.
     3. Administrator. Administrators will have the required permissions allowing them to alter/change any aspect of the site’s appearance, functionality or database records. Initially, the 4 group members listed above will be the only users with Administrator privileges.

**1.2 Justification of Functionality**

The 5 core functions of the system will be:

* Register as a User
* Browse the Catalogue of Available Tasks
* ‘Claim’ Tasks
* Create Tasks
* Leave Reviewer Ratings

1.2.1 Register as a User

On the landing page, visitors will only have 2 options: to Register, or to Log In. When registering, a user will be required to provide standard profile information, such as name, age, email address, etc. Additionally, users will be required to select their area of study/expertise/interest to be used by the system to display dynamic content that is user-targeted and relevant. Such registration processes are common and have been widely implemented by a large number of profile-based user applications, such as Facebook, LinkedIn, Amazon, etc.

Furthermore, In order to be eligible to register, a proposed user must hold a valid email account from a domain(s) specified by the site administrator; e.g. .ul.ie. By only allowing holders of email accounts from a specified domain(s) to register, some level of verification can be performed to check that users are indeed affiliated to the selected organisation or institution, e.g. Colleges, Universities, etc. This verification method is common in organisations where affiliation to a particular organisation is a prerequisite to obtaining a user account, where members are permitted to access particular information, such as library documents, employee information etc.

1.2.2 Browse the catalogue of available tasks

To have visibility of the task catalogue, all visitors to the website must first be registered (see 1.2.1) and be logged-in to the system. The justification of this is primarily to maintain security and confidentiality of users’ personal data; including uploaded academic data. This particular aspect of functionality is a common feature in sites such as dating websites, online (private) media catalogues, company-employee information, etc.

Once logged in, users can browse the entire catalogue of tasks, with the tasks most relevant to the users’ field of expertise displayed first. In addition to the areas of expertise selected in their profile, there will be the added functionality to modify the entries of the catalogue which are presented to the user based on their session/browsing habits of the site. This will be manifested by dynamically displaying tasks to the user on their homepage and/or when browsing the catalogue.

1.2.3 ‘Claim’ tasks

If a user would like to complete a task listed in the catalogue, they can do so by ‘claiming’ the particular task. Once claimed, a task is removed from the catalogue, and assigned to the user that claimed it. The website then facilitates contact (email) between the user who claimed the task and the and the creator (see 1.2.4).

1.2.4 Create Tasks

Once registered, users can create a proofreading/review task on the system. Details that users are required to provide in relation to the task include: Task title, description, deadlines, subject, etc. in addition to uploading a sample of the document that is to be proofread/reviewed.

1.2.5 Leave Reviewer Ratings

The ability for users to leave feedback or a rating system for other users is a widely-implemented methodology, examples of which include Amazon, boards.ie, etc. This process effectively allows users to ‘self-manage’ the online community of that site, highlighting to administrators and other users those who contribute to a high standard and, conversely, those who contribute to a low or negative standard. This functionally also has the potential to generate ‘prestige’ within the user community, thus an added incentive for users to make contributions to the service that are of a high standard.

**1.3 Potential Ramifications**

As with any online community collaboration platform, there will be potential benefits and ramifications to the users and, potentially, the wider community. It is therefore essential that benefits are maximised to their full potential where possible, whilst risks and adverse consequences are removed or mitigated to a level that is as low as reasonably practicable.

1.3.1 Benefits and advantages

The benefits of online community collaboration are evident with sites such as Wikipedia, boards.ie, etc. where vast amounts of information can be produced and maintained by an online community. The fact that people are volunteering to assist other users shows that, in the majority of cases, the intentions of the community are for the greater good and without ulterior motives.

Users that have their work proofread/reviewed will be the most obvious beneficiaries of the website, using the experience and abilities of other users to improve the quality of their work. Additionally, however, the users that perform the proofreading/reviewing of other users’ documents will also have the potential to benefit from participating in the community, as the site can help facilitate the development of their peer-review ability and critical thinking, whilst potentially building on their knowledge of new and existing fields of expertise.

1.3.2 Adverse and unintended consequences

By sharing academic documents with the site, or other users, there is the inherent risk to the confidentiality of the data. To address this issue, when registering, all users should have to read and agree to Terms and Conditions (T&Cs) of using the system. Within the T&Cs it should be stated explicitly that all media uploaded and/or shared on the site is done so at the user’s risk. Alternatively, it could be stated in the T&Cs that users are not to reproduce, use, publish etc any content obtained from the site or any other user. Whilst the latter suggestion may be the ideal approach, realistically the former would be the most suitable/feasible to implement.

As online plagiarism checks are becoming commonplace in Colleges and Universities, there is a possibility that a user who has uploaded a sample of their document onto the website could find themselves accused by their institution of plagiarism. The most effective ways to mitigate this issue would be to prevent online plagiarism checkers from accessing sample documentation, and/or inform the institutions that the PlayByPeer website should be excluded from the plagiarism search.

Although the intentions of users may be well placed, there is always the potential that an issue will arise regarding the quality of the work undertaken, missed deadlines, disagreements, etc. Other than the functionality of Reviewer Ratings (see 1.2.5), users have little to no recourse to address their grievances with another user.

Although the concept of a ‘self-regulating’ community using moderators is an efficient model (see 1.1.2 and 2.0), there are also inherent risks involved with this practice. The main risk would be of ‘rogue’ moderators, who act in their own, possibly malicious, interest. This could include banning users unnecessarily, banning other moderators to maintain dominance, removing legitimate documentation or tasks from the website, etc. In addition to the discontent this would generate within the users, the actions of a rogue moderator could significantly undermine the credibility and reputation of the website. To address this, control measures should be implemented, such as limiting the number of actions a moderator can perform in a certain time, e.g. the ability to only ban a limited number of users within a 24hr period. Furthermore, moderators should not be able to individually ban other moderators without the approval of other moderators and/or a site administrator.

**2.0 Detailed Description**

2.0.1 Landing page

When PlayByPeer is accessed, visitors are presented with a landing page similar to that shown in Appendix 1 figure 1. As there are only 2 options available to all visitors on this page, i.e. sign up (register) or Sign-in (login), no processes are implemented at this point.

2.0.2 Registration

If a user has not yet registered, the ‘Sign Up’ button should be selected, where the user will then be directed to a page similar to that shown in Appendix 1 figure 4. By completing and submitting the required form, users initiate the process of creating a user account/profile (Appendix 3 - P1). Assuming all details entered by the user are valid, a user account is created and all information is stored in the ‘User’ Database table, as outlined in Appendix 2.

2.0.3 Main Features

Users that are already registered should select the ‘Sign in’ option presented on the landing page. This will activate the login process (Appendix 3 - P2), where the credentials supplied by the user, i.e. username and password, will be checked against the details stored in the ‘User’ database table. If the credentials are successfully validated, the user is directed to their personalised homepage (Appendix 1 figure 2) and will have access to a variety of site options. One of the options available to the user at this point will be the ability to create a task (Appendix 3 – P3) and publish it in the site catalogue of open tasks. In order to do this, various information is required to populate the fields of the Tasks database table (Appendix 2). Some attributes, such as TaskID and CreatorID, will be populated automatically, whilst the remaining fields, such as Task title, type, description etc. must be populated by the user, and a sample of the document also uploaded. Additionally, users must attach ‘Tags’ to the task which detail the subject area(s) that the task relates to, where tags will be ‘tied’ to tasks using the database table Task\_Tags as defined in Appendix 2. Once created, all tasks are added to the Task catalogue and will be visible to other users.

Another feature accessible from a user’s homepage is the ability to browse unclaimed tasks contained in the task catalogue. When a user selects this feature, the ‘Browse Tasks’ process (Appendix 3 - P4) is executed and the user is presented with a list displaying a summary of each unclaimed task. These tasks are filtered according to the user’s subject field(s), and sorted according to deadline; where tasks with closest deadline are displayed first. Should a user wish to view additional information of a particular task, they can do so by clicking/expanding the task, which then executes the Expand Task Details process (Appendix 3 – P5). This process will direct the user to a more detailed view of the selected task, and will be presented to the user in a format such as that shown in the web-page mock-up at Appendix 1, figure 3. If, after completing P5, a user would like to claim the task, they can do so by completing the necessary step(s), which requires the Claim Task process (Appendix 3 – P6). When a user claims a task, their userID is automatically entered into the ClaimID field of the Tasks database table shown in Appendix 2. The task then no longer visible to other users, and the task creator is notified that a user has claimed the task. Conversely, as well as claiming a task, users also have the option to ‘un-claim’ any task(s) assigned to them. This is also accomplished using process P6, which results in the task being returned (added) to the task catalogue and is once again visible to other users.

Once a task has been claimed, the creator of the task will be provided with the contact details (email address) of the user who claimed the task. It is then the responsibility of the task creator to supply the task claimant with the full document that is to be proofread/reviewed. This transfer of the document is a process that is external to the websites’ functionality or responsibility.

2.0.4 User feedback, moderators and other functions

As a collaborative-community environment, one of core the components of the system is the ability to leave user feedback (see section 1.2.5). This model users to self-manage the community, with users being rewarded for positive contributions by being promoted to moderator status. To have moderator rights, a user must first pass the required score threshold of 40 points which, once achieved, executes the Promote to Moderator process (Appendix 3 – P11). The user’s score is recorded in the UserRep attribute of the User database table, and can be increased or decreased for a number of reasons. The most effective method of increasing a user’s score is claiming tasks (+10 points) and completing the task to the creators’ satisfaction (+5 points). Another method of increasing a user’s score is ‘flagging’ inappropriate tasks, where users are awarder +2 points per task flagged. Also, users can have their score reduced for various infractions, such as missing a deadline (-30 points), cancelling a task (-15 points) and a task not being completed to the creator’s satisfaction (-5 points). Once a moderator drops below the 40-point threshold, they revert to becoming a standard and forfeit their moderator rights.

All users have the ability to ‘flag’ any task they deem to be inappropriate. When a task has been ‘flagged’ by a user, accomplished by executing P7 in Appendix 3, the task is added to a list which is only visible to moderators. If, after review, it is deemed that the content is inappropriate or breached the terms of use of the website, moderators have the authority to remove the inappropriate material and/or ban the user responsible.

Other functions available to users include viewing the list of tasks that the user has claimed (Appendix 3 – P8), and viewing a list of tasks that the user has created; including both those currently listed in the catalogue and those that have been claimed by a user (Appendix 3 – P10). The lists presented to the users in both of these instances (P8 and P10), will be sorted according to date, so that the task with the closest deadline is situated at the top of the list.

**3.0 Technologies**

**TO BE COMPLETED**

Operating System

PHP

HTML & CSS

JavaScript

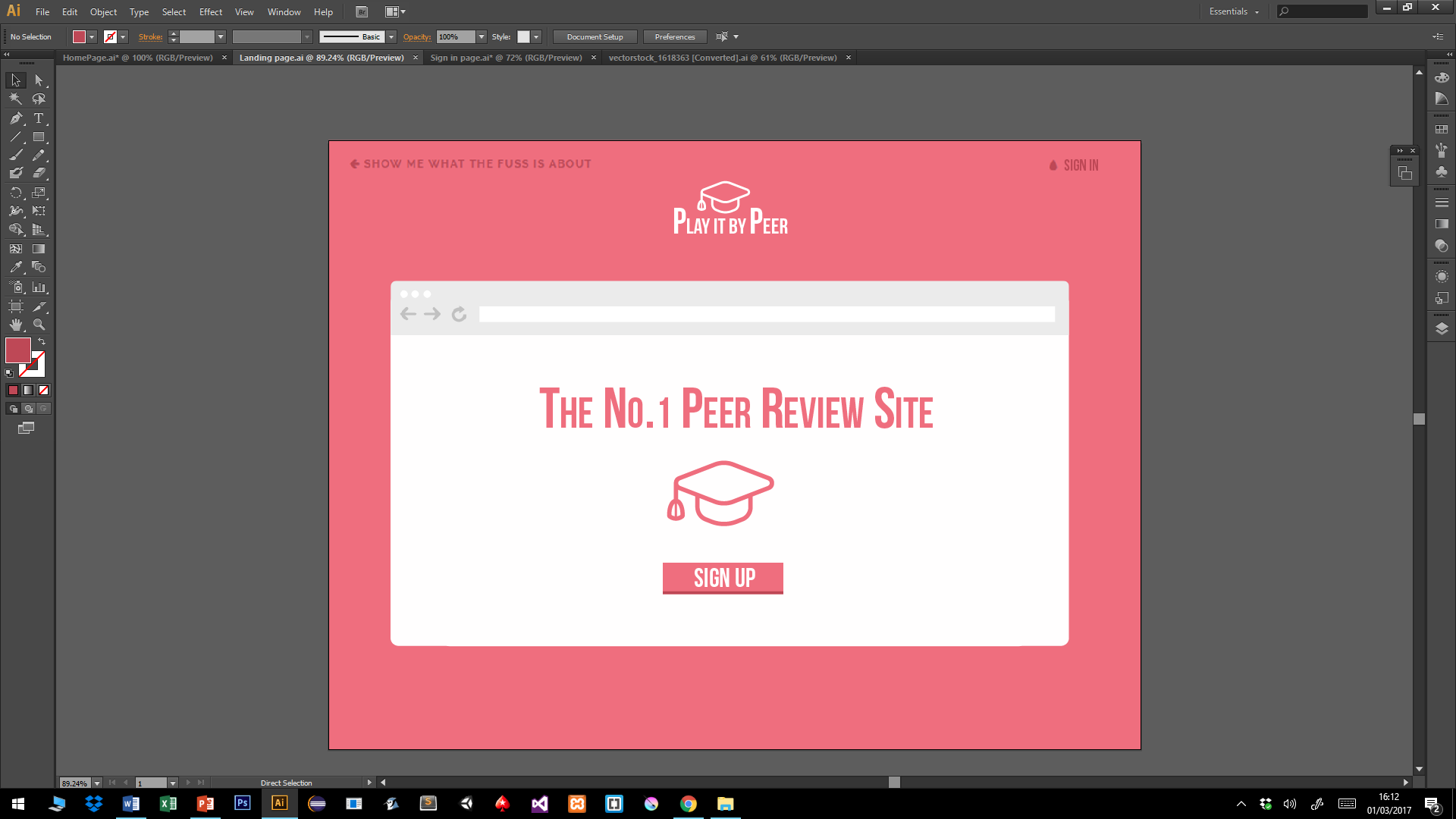
JQuery

MySQL

Github. Several repositories will be used for this project. The main repository is at https://github.com/CS4065Group16/Main.git

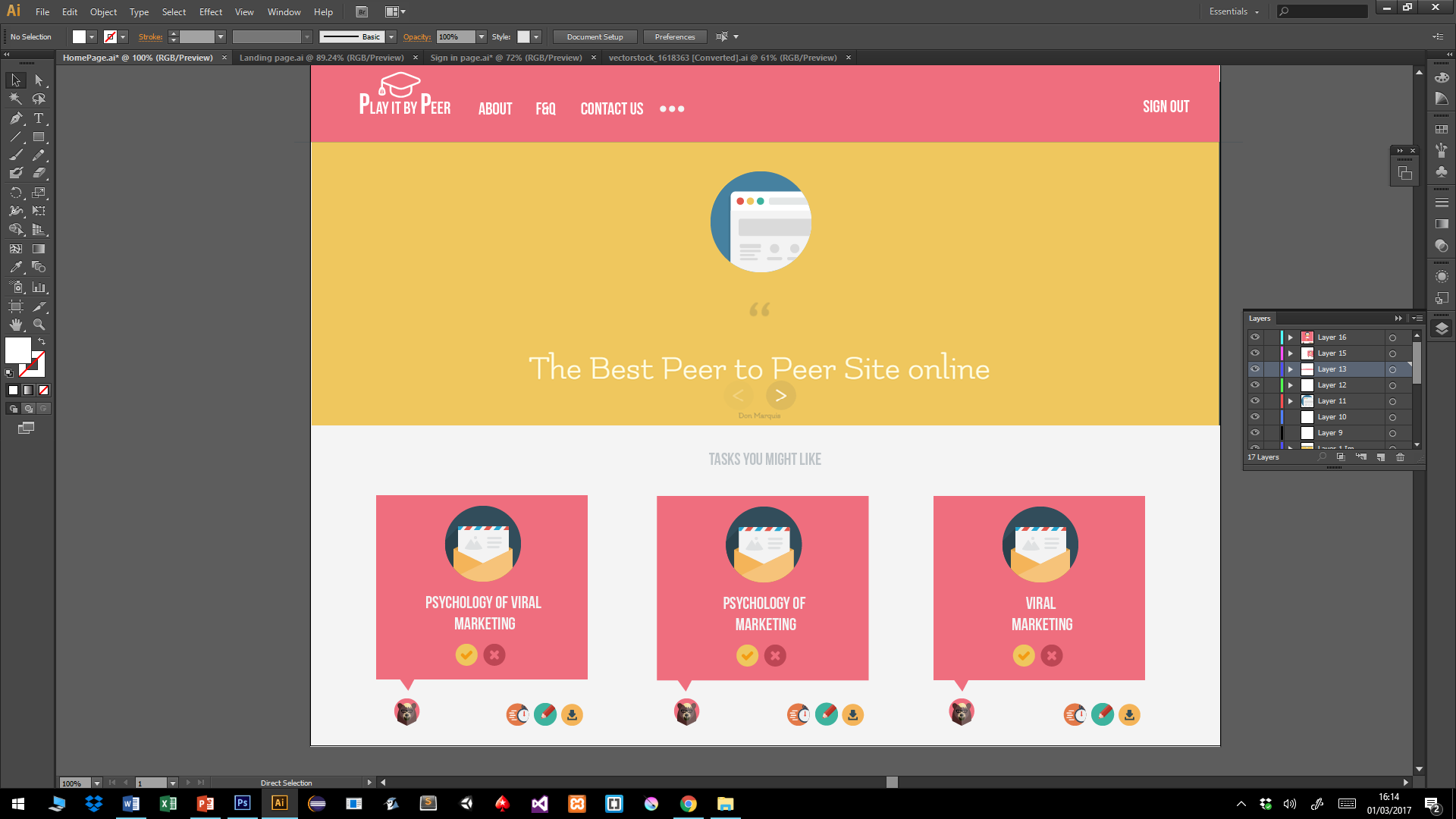
**Appendix 1: Web Pages**

**Landing Page**



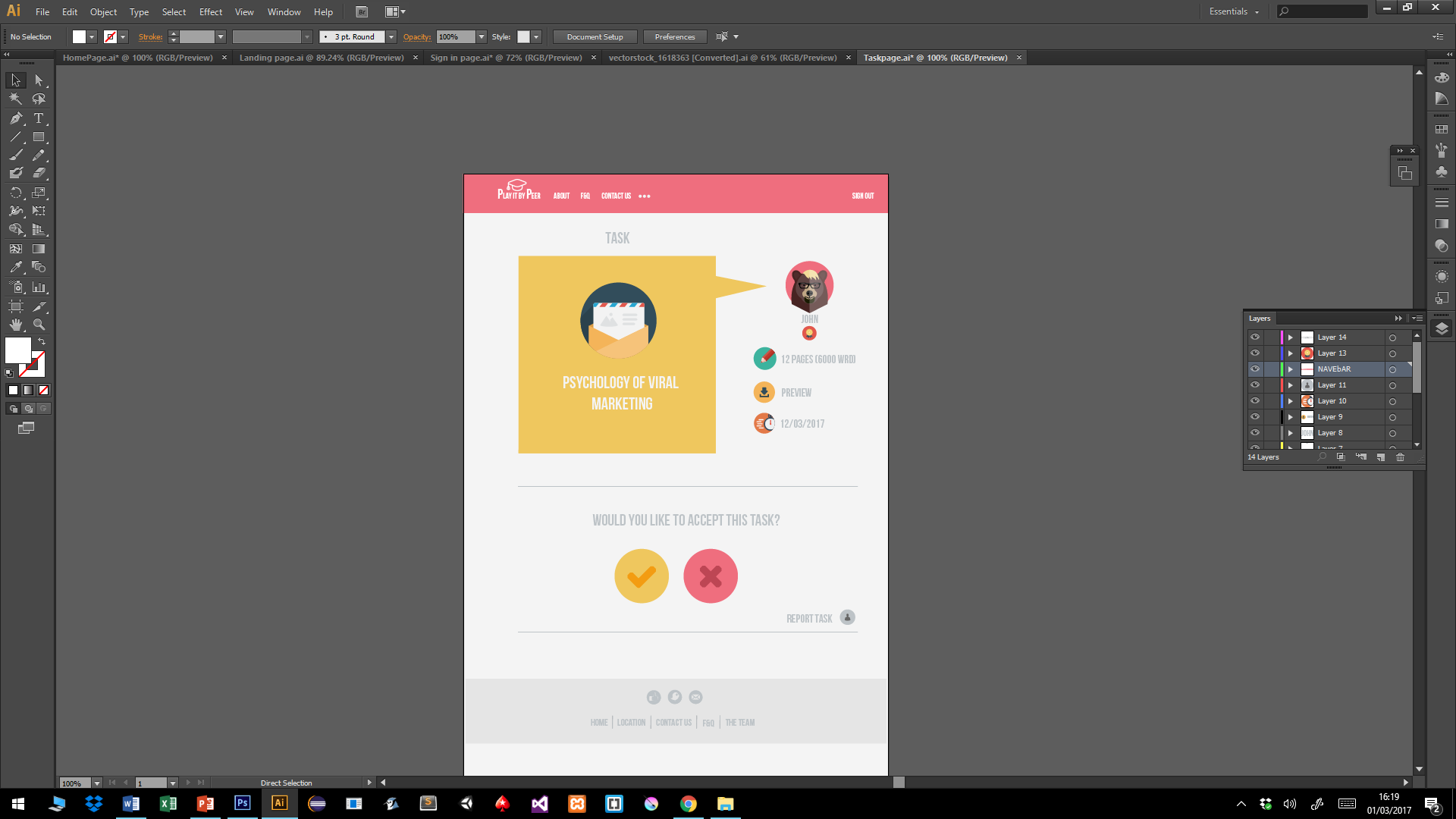
*Figure 1 – Proposed Landing Page*

**Home page**



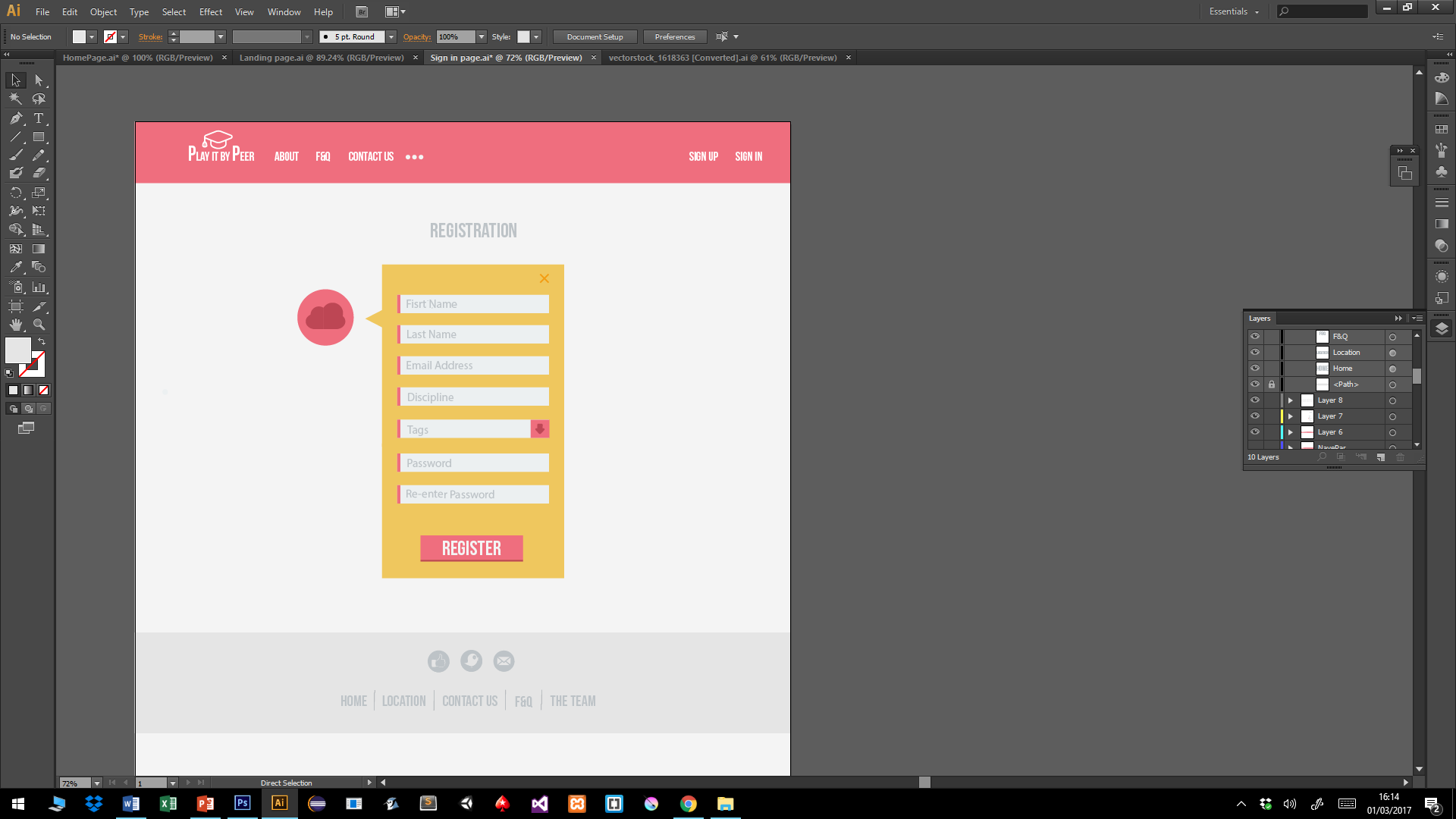
*Figure 2 – Proposed Home Page*

**Task Page**



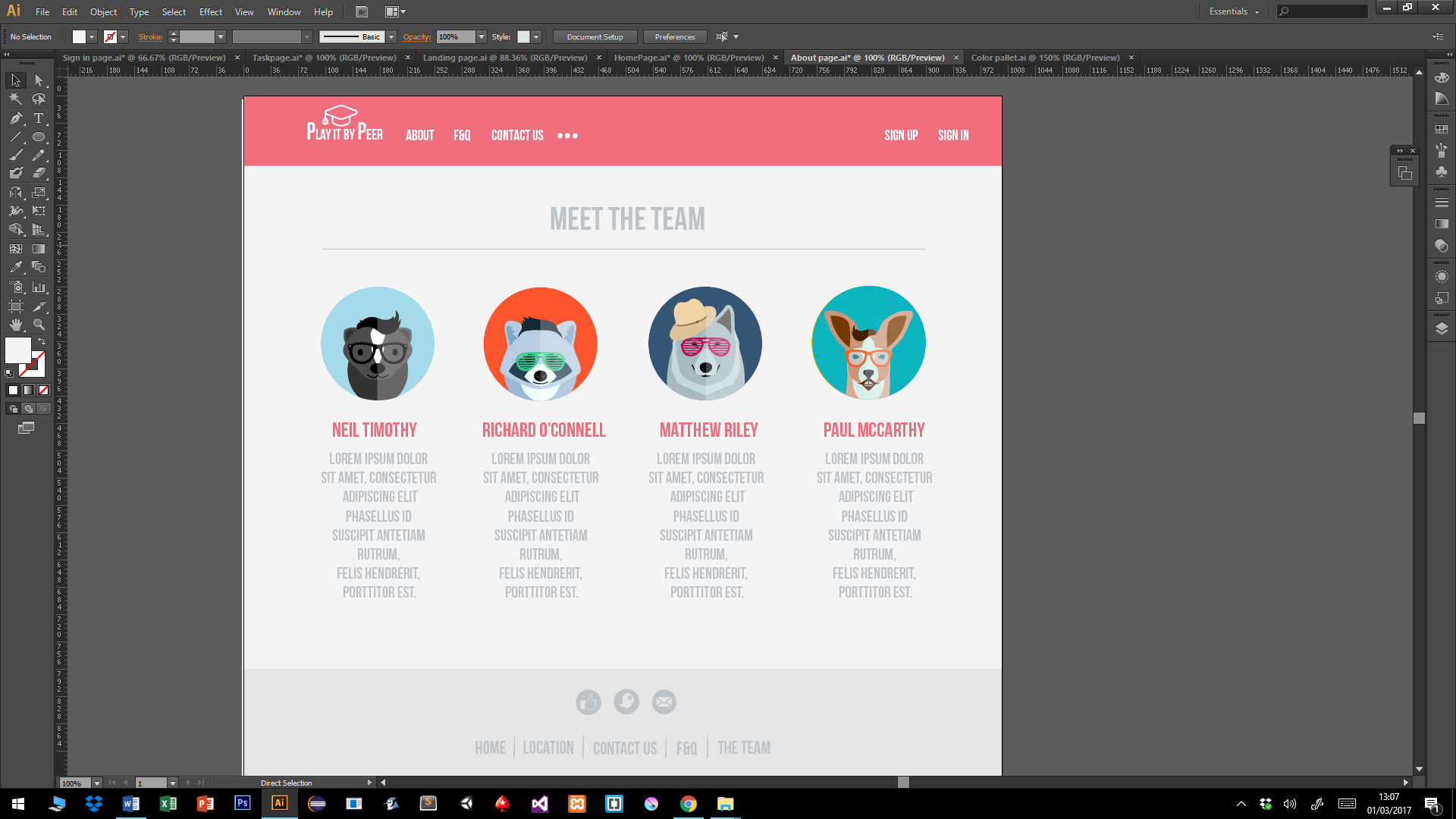
*Figure 3 – Proposed Task Details page*

**Registration Page**



*Figure 4 – Proposed Registration Page*

**Team /About Page**



*Figure 5 – Proposed ‘About’ page*

**Appendix 2: Tables**

**Database Tables**

= Primary Key

*Please note tables may be modified and amended during the course of the project depending on functionality and user feedback.*

**Table Name: User**

*Primary Key : UserID*

Description: The User table stores general information about the each user such as their name, email and subject area.

|  |  |  |
| --- | --- | --- |
| Field Name | Data Types | Example |
| UserID | INT(10) | 9832224 |
| FirstName | VARCHAR(32) | RICHARD |
| LastName | VARCHAR(32) | JONES |
| UserEmail | VARCHAR(32) | [RichardJo@yahoo.ie](mailto:RichardJo@yahoo.ie) |
| UserSubject | VARCHAR(32) | Math |
| UserRep | INT | 20 |
| PromoteToMod | BOOLEAN | False |
| UserTags | TEXT | html |
| Password | VARCHAR(32) (ENCRYPTED) | 3@#$%100%2 |

* **UserID** is the primary key identifier for every user. Every user has one unique ID(e.g. 9832224).
* **FirstName** is the actual first name of the user stored in the table.
* **LastName** is the actual surname of the user stored in the table.
* **UserEmail** is the user’s own email address.
* **UserSubject** is the user’s major discipline
* **UserRep** is the user’s reputation on a scale of 1-100 stored in the table.
* **PromoteToMod** is accessible if the user’s reputation is greater than 40. It will allow them to gain moderator privileges.
* **UserTags** is a list of the user’s favourite and subscribed tags.
* **Password** is the user’s encrypted password.

**Table Name: Tasks**

*Primary Key: TaskID*

Description: A table containing a list of tasks uploaded. The user from here can add a query to create a task, browse for a task, claim a task and flag a task. Users can also filter the database here and view only claimed or unclaimed tasks here. They can also filter the database and view flagged tasks.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Types** | **Example** |
| TaskID | INT(10) | 3562705 |
| CreatorID | INT(10) | 00022224 |
| ClaimID | INT(10) | 45453334 |
| TaskTitle | VARCHAR(32) | English Spellcheck |
| TaskType | VARCHAR(32) | MSc |
| TaskDesc | VARCHAR(32) | Hamlet report |
| PageCount | INT | 5 |
| WordCount | INT | 5000 |
| FileFormat | VARCHAR(10) | .docx |
| ClaimDeadline | DATE | 15/05/2017 |
| SubmissionDeadine | DATE | 10/10/2017 |
| ClaimedStatus | BOOLEAN | True |
| FlaggedStatus | BOOLEAN | False |
| TaskTags | TEXT | html |

* **TaskID** is the primary key identifier unique to every task. Every task has one unique ID. (e.g. 2705550).
* **CreatorID** is the ID of the user that posted the task.
* **ClaimID** is the ID of the user that claims the task. When it is null or blank the task is still listed but once a user has claimed it, it is no longer listed in the catalogue.
* **TaskTitle** is the name of the task as specified by the creator.
* **TaskType** is the type of task as specified by the creator.
* **TaskDesc** is a description of the task as specified by the creator.
* **PageCount** is a counter referring to the amount of pages in the task as specified by the creator.
* **WordCount** is a counter referring to the amount of words in the task as specified by the creator.
* **FileFormat** is the file type of the task as specified by the creator.
* **ClaimDeadline** is the final date this task can be claimed as specified by the creator.
* **SubmissionDeadline** is the final date a task can be submitted as specified by the creator.
* **ClaimedStatus** is a current status of the task. It refers to whether or not the current task has been claimed by a user.
* **FlaggedStaus** is a current status of the task. It refers to whether or not the current task has been flagged by a user.
* **TaskTags** is the tags for that particular task.

**Table Name: Task\_Tags**

Description: Task\_Tags creates the relational link tying tasks and tags together (junction table).

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Types** | **Example** |
| TaskID | INT (10) | 2705033 |
| TagID | INT (10) | 2701111 |

**Table Name: User\_Tags**

Description: Task\_Tags creates the relational link tying user and tags together (junction table).

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Types** | **Example** |
| UserID | INT (10) | 2705033 |
| TagID | INT (10) | 2701111 |

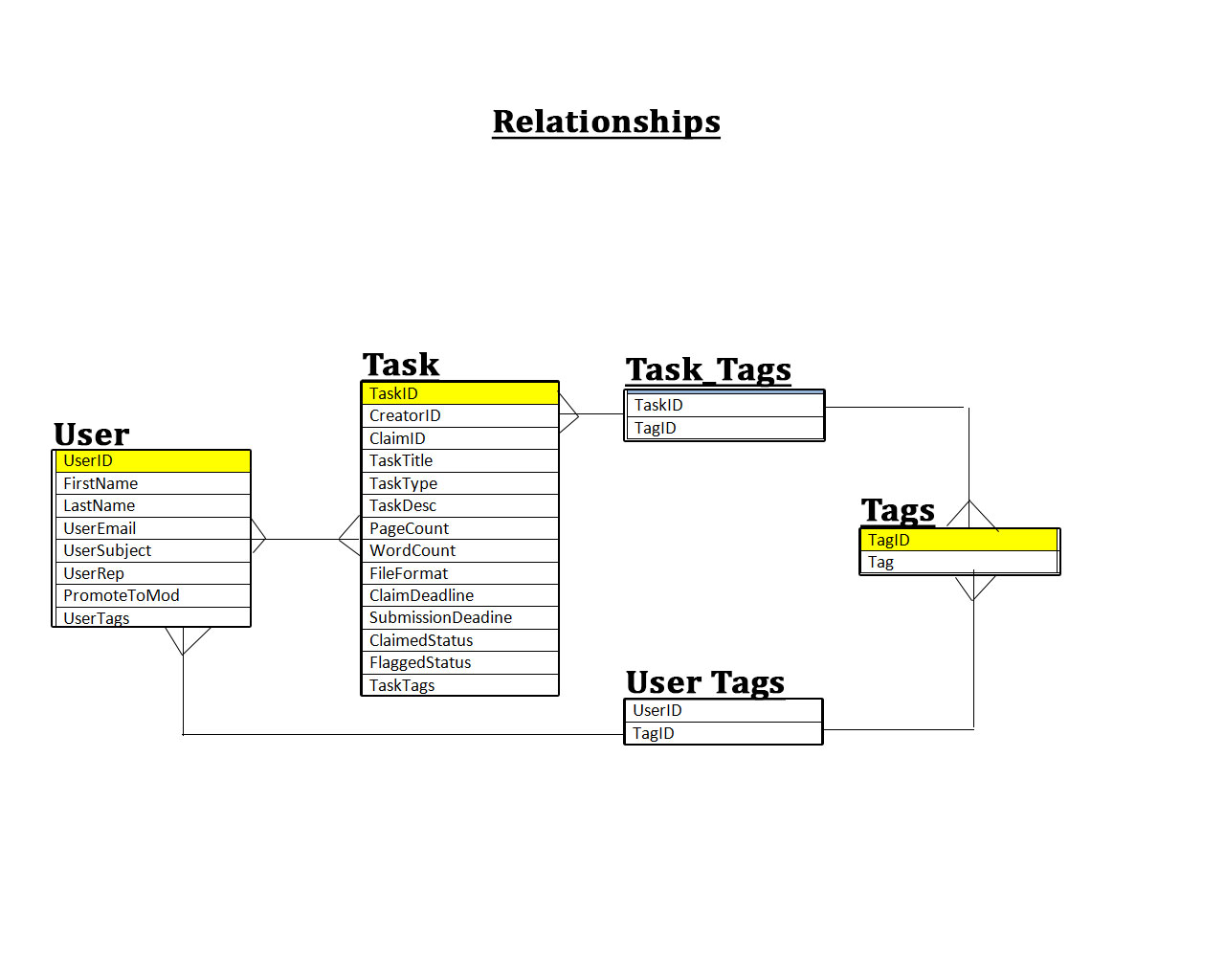
**Table Name: Tags**

Description: The tags table stores all tags listed with a TagID.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Types** | **Example** |
| TagID | INT (10) | 67449 |
| Tag | VARCHAR (255) | html |
|  |  |  |

**Relationships within the database**

* Many users can create/view many tasks. Many tasks can have many users.
* Many tasks can have many tags. Many tags can have many tasks.



**Appendix 3: Processes**

|  |  |
| --- | --- |
| **Process Number** | P1 |
| **Process Title** | Registration |
| **Brief Description** | Directs the user to a form. A new account/profile is created new if supplied with correct information. Form/details are rejected if information submitted is not in a valid format. |
| **Inputs** | User’s information for creation of new profile inputted via registration form. |
| **Detailed Description** | In order for new profile to be created, a registration form must be submitted. Users must complete the form with information in the required format. If correct input is supplied to all fields, form can be successfully submitted. If incorrect information is provided, users will be asked to resubmit their details. The information requested by the form is First Name, Second Name, Student or Staff Id, Email, Major Subject, Password and Password re-entry. If these details are valid, a user table is then uploaded. |
| **Output** | Creates a new profile. Directs users to their new profile (homepage). |

|  |  |
| --- | --- |
| **Process Number** | P2 |
| **Process Title** | Login |
| **Brief Description** | Validates username and password provided from the user, either restricts or provides access to user’s account/system. |
| **Inputs** | Username & Password. |
| **Detailed Description** | User will enter their unique username along with associated password in the relevant field. If the details are incorrect, the user does not progress any further and the login process is restarted. The process validates the input from two fields and only if they are correct will the user be able to progress further. |
| **Output** | Directs users to their profile. |

|  |  |
| --- | --- |
| **Process Number** | P3 |
| **Process Title** | Create task |
| **Brief Description** | Registered students create tasks to be proof read by users. |
| **Inputs** | Task details and documentation information submitted by user in task creation form in order to successfully publish new task. |
| **Detailed Description** | User enters details responding to task creation form, must correctly provide all task details – Task title, task type, task description, tags, number of paragraphs, word count, file format, file sample, deadline for claiming task/deadline for task completion. Process validates input and if correct task is created and published to reviewed. If input is not correct the user will be asked to resubmit the task details. If the input is correct the task database is updated. |
| **Output** | Creates a task, publishes task details in task catalogue. |

|  |  |
| --- | --- |
| **Process Number** | P4 |
| **Process Title** | Browse tasks |
| **Brief Description** | Retrieve and generate a list of tasks that have not yet been claimed. |
| **Inputs** | User’s discipline taken from profile, browsing habits of website. |
| **Detailed Description** | Query tasks database. Retrieve and generate a list of tasks published by other students that are available/unclaimed. Tasks appear which are relevant to student’s discipline and in order of nearest deadline first. |
| **Output** | A list of available and relevant tasks for the user, and displayed in order of nearest deadline first. |

|  |  |
| --- | --- |
| **Process Number** | P5 |
| **Process Title** | Expand Task Details |
| **Brief Description** | Select task in order to view more details |
| **Inputs** | Click/selection by user |
| **Detailed Description** | The registered user clicks onto a task, the task database is queried more details relating to the task ie number of pages, word count, associated tags ect. should appear. The user is then able to download a preview of documentation associated to that task. |
| **Output** | All information and data relating to that task, preview documentation of task. |

|  |  |
| --- | --- |
| **Process Number** | P6 |
| **Process Title** | Claim task |
| **Brief Description** | Registered users can claim an unclaimed/available task. |
| **Inputs** | Click/selection by user |
| **Detailed Description** | The registered user clicks onto a task and is then able to claim task if required. Once a user claims task, the task database is updated and the task is removed from the list to prevent another user claiming the same task. Once a task has been claimed, the user (claimant) is rewarded 10 marks to his/her reputation score. |
| **Output** | Task removed from task list, user rewarded 10 marks to reputation score. Task claimed message appears on screen to user. |

|  |  |
| --- | --- |
| **Process Number** | P7 |
| **Process Title** | Flag/report task |
| **Brief Description** | If a task judged as being inappropriate by a user, a task can be flagged/reported. |
| **Inputs** | Flagging (icon click) by user |
| **Detailed Description** | If a user is of an opinion that a task may be inappropriate, they can flag the task and will then be highlighted to the moderators. Users are awarded + 2 marks to their reputation score. Task and user database is updated. |
| **Output** | Marks awarded to user. Message confirming action displayed. Flagged items list updated and able to be viewed by moderators. |

|  |  |
| --- | --- |
| **Process Number** | P8 |
| **Process Title** | View all claimed tasks |
| **Brief Description** | Registered users can view a list of their claimed tasks |
| **Inputs** | Click/selection by user |
| **Detailed Description** | Task database queried. If registered user is logged on, the user can view a list of claimed tasks, sorted in order of nearest deadline first. |
| **Output** | A list of available tasks claimed by the user in order of nearest to deadline first. |

|  |  |
| --- | --- |
| **Process Number** | P9 |
| **Process Title** | Select an option to action with claimed task |
| **Brief Description** | Registered users can click into a task from claimed task list. Then user is provided with options to do with that task. |
| **Inputs** | The user must select option from dropdown list. |
| **Detailed Description** | When registered user is logged on and clicks into view a task, If the task deadline has not been reached the user can pick one of the options 1) request full file from task owner 2) Mark task as completed 3) Cancel task claim. If the task deadline is gone the status of task is altered to cancelled and can not pick any of these options, the claimant is then penalized 30 marks from their reputation score. Update task and user database. |
| **Output** | If task within deadline, A personal email is sent to task owner requesting full documentation of task or task marked as completed or task marked as cancelled and claimant deducted 15 marks. If deadline for task has gone status of task changed to cancelled and task claimant deducted 30 reputation points. |

|  |  |
| --- | --- |
| **Process Number** | P10 |
| **Process Title** | My tasks |
| **Brief Description** | Registered users can view a list of their created tasks. Contact and rate claimant. |
| **Inputs** | No user input |
| **Detailed Description** | If registered user is logged in and has created tasked the user can view a list of their tasks. The retrieved list will show current status of either pending, unclaimed, claimed, cancelled or completed. If task marked as complete, the user cant provide feedback (Happy or Unhappy). Task and user database queried and updated. |
| **Output** | Retrieve a list of users published tasks, if claimed, show claimants email address, first and last name. Reward/Penalise claimant with points based on users feedback rating. |

|  |  |
| --- | --- |
| **Process Number** | P11 |
| **Process Title** | Promote user to moderator |
| **Brief Description** | User gets automatically promoted to moderator based on reputation score. |
| **Inputs** | No user input |
| **Detailed Description** | If registered user gets a reputation score of 40 or more, they automatically get promoted to moderator. User database updated. |
| **Output** | Change status to users profile to moderator. |