Software Development

**Group 4**

Deliverable 4

John-Rey Juele (15167798)

Sophia Colgan (15159973)

Eoghan Casey (15160513)

Table of Contents

Introduction 3

Major Processes & Feature List 4

Registration 4

Login 4

Create Task 4

Task Stream 5

Claim Task 5

My Tasks 5

Flag Tasks 5

Moderator 6

Additional Features 6

Logout 6

Republish Task 6

Error Message 6

Potential Additions 7

Word Checker 7

Favourite Tags 7

Other Recourses Used 7

Template 7

Appendix 8

Website Files 8

Database E-R diagram 10

Database Create Schema 11

Introduction

This report presents the final state of the CS4014 Software Development project as completed by Group 4.

For the presentation, the group will be presenting on our own laptop, but the database was on the CSIS server provided. All files have since been copied to the group’s testweb folder, exactly as they were during the presentation.

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 section discussing features that could have potentially been added to the site if the time had been available, and a section of certain features that could have been handled differently if we were to do this project again.

This project can be cloned via GitHub: [https://github.com/Coding-Chicken-Lover/Web-Development-Project]

Additional resources used in the creation of the site are listed, where we have relied on free third-party code for the certain element 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 included a database E-R diagram and create statements for all database tables, to indicate foreign keys and data types.

Major Processes & Feature List

Relevant pages or scripts are listed in square brackets.

Registration

* We used a registration form in [index.php] requiring a student/staff number, first name, surname, email, password, a confirmation password and which subject are you doing . The formatting of the information provided in checked in java script and if valid sends the information to [insert\_user.php].
* In [insert\_user.php] it uses the code in [checkUserExistence.php] to see if the user has already been registered or banned. If it is available, the data provided will be logged into the databases using [insert\_user.php] and notified they registered successfully and redirected to [HomePage.php].

Login

* The login page is a form containing 2 data fields [login.php]. The user wil be asked to enter their user ID and password and then checked to see if they are of the correct format.
* If they it sends the information to [checkUserExistence.php] where it will check if the user is in the database already or banned. If the use user name or password is wrong, they will be notified and given a chance to login again. If the user is banned they will fail to login and need to contact an admin.
* If both the user name and password match the will user will be logged in and redirected to the homepage [HomePage.php].

Create Task

* For Task Creation the user must complete a form laid out with all the relevant information required for their task [task\_creation.php]. The information taken is the type of paper, description of the task, format of the document, tags associated with the document, date it needs to be claimed by and date it needs to be submitted by.
* If all the information is valid the information will be sent to [store\_task.php] and it will be assigned a Task number and stored into the database.

Task Stream

* The task stream is displayed on [HomePage.php] with the user being able to click into and view any of the tasks that are displayed.
* The task stream is personalised to each user by the which subject they associated with themselves when they registered. The tags applied to each task belong to a subject and only tags relevant to your subject will be displayed on the users task stream.

Claim Task

* From the Task stream created in [HomePage.php] the user is able to click into tasks and view them.
* Once a user clicks into a task it is redirected to [task\_details.php] which pulls up all the relevant information for that task included by the creator of the task in [task\_creation].
* If the user clicks on the claim task button it redirects to [claim\_task.php] and updates the database to have a claimed status and inserts the claimant’s id into the tasks information.
* It then lets you know you have claimed the task and gives you a link to redirect to the homepage.
* You also receive 10 reputation score for claiming a task.

My Tasks

* The user is able to click on my tasks on the navigation bar from [HomePage.php] and gets redirected to [my\_tasks.php].
* From here the user is able to view all tasks they have written and the current status of them.

Flag Tasks

* When the user views a task from the task stream[task\_details.php], if they find the content inappropriate they can flag the task by which category is most appropriate: spam, inappropriate, wrong\_subject and explicit.
* Once a task has been flagged it will be redirected to [flag\_task.php]. In [flag\_task.php] it will add the task to a database of flagged tasks.
* It will also increase the flaggers reputation score by 2 and inform of this and give them a link to redirect back to the home page.

Moderator

* Each time a session starts it will check the users current reputation score. If the user has a reputation score of over 40 they will be given access to moderator status.
* On [HomePage.php] you will be able to see moderator functions that will allow you to view flagged tasks and ban users.

Additional Features:

Logout **-** On [HomePage.php] there is a logout button and when clicked it will redirect to [logout.php] which will end the session and redirects to [login.php].

Republish Task­ **-** If a task you created has timed out, you are given the option to republish the task [republish\_task.php] without having to create a new task.

Error Message **-** If an issue that we had not accounted for occurs, they will be redirected to [error.php].

Potential Additions

Word Checker **–** Currently, when the user is giving a description of the task it is up to them to spell everything correctly. If the time had been available perhaps we could have added this feature to our site.

Favourite Tags **–** Every user is going to have tags specific to them but an addition we could have had would have been a ‘favourite tags’ option. This option would be a separate page where the user can easily search through the tags they have made their favourite and find a task they want to claim. This would be a big help to the generated tags as we would have more information on the user to provide them tasks more suited to their taste.

Other Resources that were used

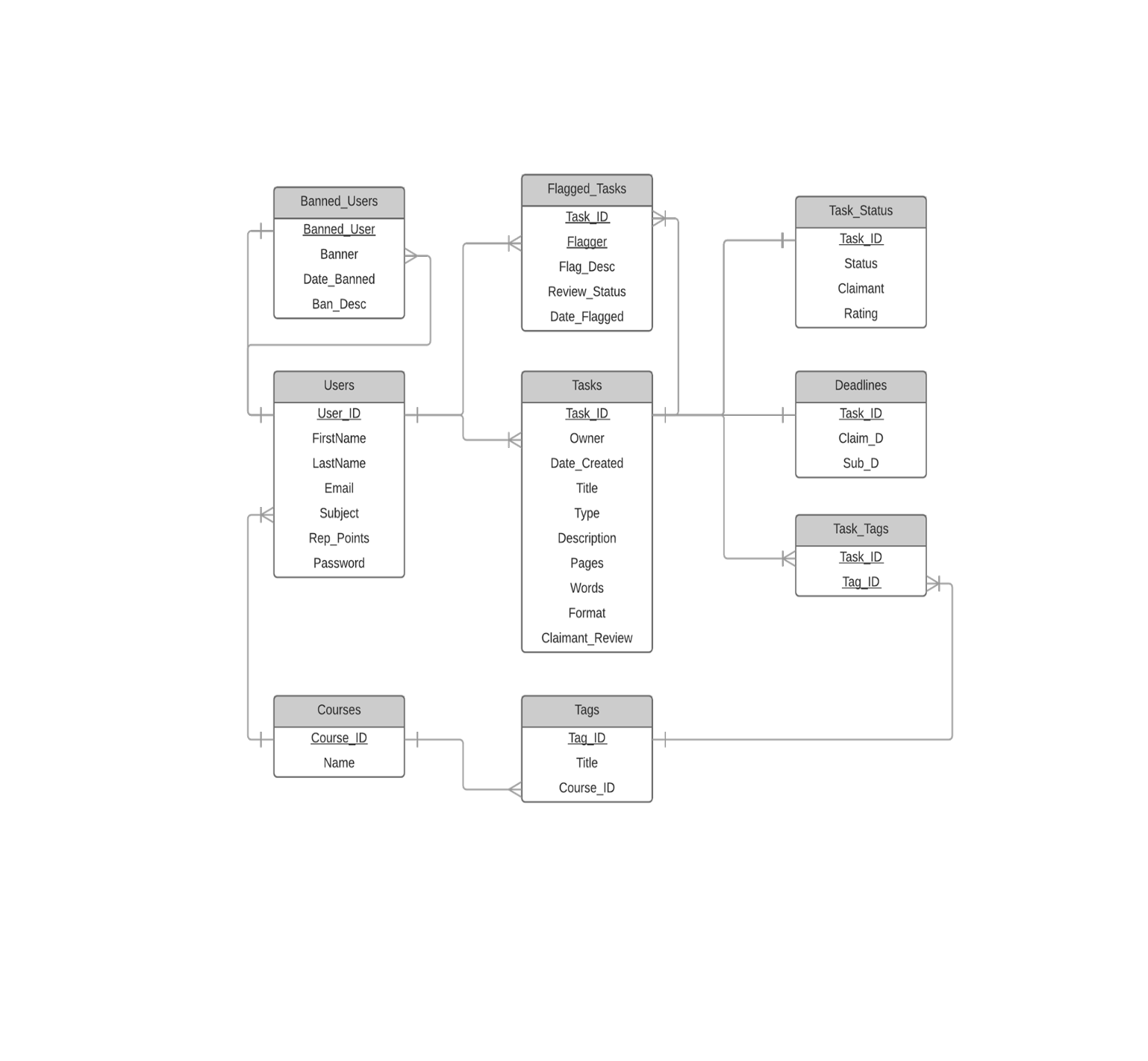
Template **-** We used a CSS/HTML template for the design of the website, it is called textured\_orbs and there is a copyright stamp on the footer of each page. The template was used as a based which we than edited to suit our style.

# Appendix

## Website Files

|  |  |
| --- | --- |
| **File** | **Description** |
| cancel\_task.php | This script is run when the task claimant decides to cancel his/her claimed task with the help of the User.class.php file. It also deducts 15 rep points of the claimant. |
| checkUserExistence.php | This file contains functions that are used in other scripts, mainly for the registration and login files. The “checkUserIdExistence” function checks if a user with the entered id exists. The “checkUserExistence” function checks If a user with their entered id and password exists. The “checkIfBanned” function checks if a user is banned. |
| claim\_task.php | This script executes when the user decides to claim a task on the task stream. He/she gains 10 reputation points. It updates the info in the database. |
| complete\_task.php | This script runs when a user decides that he is finished with his claimed task. It updates the info in the database |
| connect.php | This script helps the server connect to the database. It is used in many of the php scripts that utilizes the database. |
| download\_file.php | The script is used whenever a file preview belonging to a certain task is downloaded. |
| email\_request.php | This script is used to display the email template that will be used when the task claimant wishes to email the task owner for the full document. The user can also click a link to automatically input the template to the email client installed in their computer. |
| error.php | This script displays error messages from the PDO library and to notify the user that he is banned. |
| flag\_task.php | This script is run whenever a user wants to flag a task. |
| flagged\_tasks.php | PHP scripts are embedded in this file. It helps display tasks that has been flagged (not including tasks that you have flagged or flagged tasks that you own) and its information. It also gives you the option to unpublish the task/Mark as checked/ban the owner of the task. |
| HomePage.php | PHP scripts are embedded in this file. It is the user’s home page when they first register/login. PHP scripts help the display the user’s data along with the task stream and the tasks that the user claimed. PHP scripts also helps in ordering the list closest to the current date and checks if a task is expired. |
| index.html | This the first page the user enters in the website. It is the registration page of the website. |
| insert\_user.php | Enters the details of the registering user into the database. |
| login.html | This is the login form of the website. |
| login\_user.php | Scripts in this file checks if the entered id and password of user is in the database. If so, it allows user to enter the website. |
| logout.php | Logs the user out of the website. |
| mod\_features.php | This file contains PHP functions that fulfil a particular mod task such as unpublishing a task, marking a task as seen, banning a user, unbanning a user. All of these connect to the database and executes SQL queries. |
| my\_tasks.php | This file displays all the tasks owned by the user and all its data. |
| republish\_task.php | This file allows the user to republish a task that was unclaimed during its time in the task stream. |
| review\_rating.php | This scripts runs when a user submits the task owner’s rating of the claimant’s task. It grants 5 points to the claimant if the owner of the task is happy. Otherwise 5 points is removed from the claimant. Updates the details in the database. |
| store\_task.php | PHP code in this file adds the created task (from task\_creation.php) into the database and also uploads the files. |
| task\_creation.php | Contains a form for creating a task. |
| task\_details.php | This is run to display the details of a certain task to the potential claimant. It also provides an option to flag the task or claim the task. Upon claiming, the user is given an option to ask the owner for the full file, complete the task, cancel the task and download the task preview |
| upload\_preview.php | This file is executed whenever the owner of a task wants to upload the task preview from the My Tasks page. |
| user.class.php | This file contains a class that represents to user of the website and helps obtain details of the current user easier that having the write individual queries. |
| view\_banned\_users.php | This page displays all the users of the website that has been banned by other mods and gives you the option to unban a user. |
|  |  |

Database E-R Diagram



# Database Create Statements

/\*

Courses table

\*/

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `Courses` **(**

`Course\_ID` **int** unsigned **NOT** **NULL,**

`Name` **varchar(**128**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`Course\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 **;**

/\*

Users table

\*/

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `Users` **(**

`User\_ID` **int** unsigned **NOT** **NULL,**

`FirstName` **varchar(**128**)** **NOT** **NULL,**

`LastName` **varchar(**128**)** **NOT** **NULL,**

`Email` **varchar(**128**)** **NOT** **NULL,**

`Subject` **int** unsigned **NOT** **NULL,**

`Rep\_Points` **int** **DEFAULT** '0'**,**

`Password` **varchar(**255**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`User\_ID`**),**

**UNIQUE** **KEY** **(**`Email`**),**

**FOREIGN** **KEY** **(**`Subject`**)** **REFERENCES** `Courses`**(**`Course\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 **;**

/\*

Tasks table

\*/

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `Tasks` **(**

`Task\_ID` **int** unsigned **NOT** **NULL** AUTO\_INCREMENT**,** /\* Auto generates task id \*/

`Owner` **int** unsigned **NOT** **NULL,** /\* References to the user's id \*/

`Date\_Created` **datetime** **NOT** **NULL,**

`Title` **varchar(**255**)** **DEFAULT** **NULL,**

`**Type**` **varchar(**20**)** **DEFAULT** **NULL,**

`Description` text **DEFAULT** **NULL,**

`Pages` **int** unsigned **DEFAULT** 0**,**

`Words` **int** unsigned **DEFAULT** 0**,**

`**Format**` **varchar(**10**)** **DEFAULT** **NULL,**

`Claimant\_Review` text **DEFAULT** **NULL,**

**PRIMARY** **KEY** **(**`Task\_ID`**,** `Owner`**),**

/\* A user's tasks should be deleted when a user is removed

so that potential claimants cannot ask a non-existent user for their task

\*/

**FOREIGN** **KEY** **(**`Owner`**)** **REFERENCES** `Users`**(**`User\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 **;**

/\*

Tags table

\*/

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `Tags` **(**

`Tag\_ID` **int** unsigned **NOT** **NULL** AUTO\_INCREMENT**,**

`Title` **varchar(**20**)** **NOT** **NULL,**

`Course\_ID` **int** unsigned **NOT** **NULL,**

**PRIMARY** **KEY** **(**`Tag\_ID`**),**

**FOREIGN** **KEY** **(**`Course\_ID`**)** **REFERENCES** `Courses`**(**`Course\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 **;**

/\*

Task\_Tags table

\*/

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `Task\_Tags` **(**

`Task\_ID` **int** unsigned **NOT** **NULL,**

`Tag\_ID` **int** unsigned **NOT** **NULL,**

**PRIMARY** **KEY** **(**`Task\_ID`**,** `Tag\_ID`**),**

**FOREIGN** **KEY** **(**`Task\_ID`**)** **REFERENCES** `Tasks`**(**`Task\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE,**

**FOREIGN** **KEY** **(**`Tag\_ID`**)** **REFERENCES** `Tags`**(**`Tag\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 **;**

/\*

Deadlines table

\*/

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `Deadlines` **(**

`Task\_ID` **int** unsigned **NOT** **NULL,**

`Claim\_D` **datetime** **NOT** **NULL,**

`Sub\_D` **datetime** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`Task\_ID`**),**

**FOREIGN** **KEY** **(**`Task\_ID`**)** **REFERENCES** `Tasks`**(**`Task\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 **;**

/\*

Task\_Status table

\*/

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `Task\_Status` **(**

`Task\_ID` **int** unsigned **NOT** **NULL,**

`Status` **varchar(**20**)** **NOT** **NULL** **DEFAULT** 'PENDING\_CLAIM'**,**

`Claimant` **int** unsigned **DEFAULT** **NULL,** /\* what happens when the user deletes their profile? \*/

`Rating` **varchar(**10**)** **DEFAULT** **NULL,**

**PRIMARY** **KEY** **(**`Task\_ID`**),**

**FOREIGN** **KEY** **(**`Task\_ID`**)** **REFERENCES** `Tasks`**(**`Task\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE,**

**FOREIGN** **KEY** **(**`Claimant`**)** **REFERENCES** `Users`**(**`User\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 **;**

/\*

Banned\_Users

\*/

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `Banned\_Users` **(**

`Banned\_User` **int** unsigned **NOT** **NULL,**

`Banner` **int** unsigned **NOT** **NULL,**

`Date\_Banned` **datetime** **NOT** **NULL,**

`Ban\_Desc` text **DEFAULT** **NULL,**

**PRIMARY** **KEY(**`Banned\_User`**),**

**FOREIGN** **KEY** **(**`Banned\_User`**)** **REFERENCES** `Users`**(**`User\_ID`**),**

**FOREIGN** **KEY** **(**`Banner`**)** **REFERENCES** `Users`**(**`User\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 **;**

/\*

Flagged\_Tasks

\*/

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `Flagged\_Tasks` **(**

`Task\_ID` **int** unsigned **NOT** **NULL,**

`Flagger` **int** unsigned **NOT** **NULL,**

`Flag\_Desc` **varchar(**20**)** **NOT** **NULL,**

`Review\_Status` **varchar(**10**)** **DEFAULT** 'UNCHECKED'**,**

`Date\_Flagged` **datetime** **NOT** **NULL,**

**PRIMARY** **KEY(**`Task\_ID`**,** `Flagger`**),**

**FOREIGN** **KEY(**`Task\_ID`**)** **REFERENCES** `Tasks`**(**`Task\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE,**

**FOREIGN** **KEY(**`Flagger`**)** **REFERENCES** `Users`**(**`User\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 **;**

/\*

These values has to be inserted otherwise the database will not function.

\*/

**INSERT** **INTO** Courses **(**Course\_ID**,** Name**)** **VALUES**

**(**111**,** 'Computer Science'**),**

**(**222**,** 'Engineering'**),**

**(**333**,** "Mathematics"**),**

**(**444**,** "Science"**),**

**(**555**,** "Law"**),**

**(**666**,** "Business"**);**

**INSERT** **INTO** Tags **(**Title**,** Course\_ID**)** **VALUES**

**(**'programming'**,** 111**),**

**(**'web design'**,** 111**),**

**(**'operating systems'**,** 111**),**

**(**'software testing'**,** 111**),**

**(**'graphic design'**,** 111**),**

**(**'games development'**,** 111**),**

**(**'electronics'**,** 222**),**

**(**'mechanics'**,** 222**),**

**(**'aeronautics'**,** 222**),**

**(**'civil'**,** 222**),**

**(**'circuits'**,** 222**),**

**(**'engineering maths'**,** 222**),**

**(**'algebra'**,** 333**),**

**(**'statistics'**,** 333**),**

**(**'R'**,** 333**),**

**(**'trigonometry'**,** 333**),**

**(**'calculus'**,** 333**),**

**(**'probability'**,** 333**),**

**(**'graph theory'**,** 333**),**

**(**'physics'**,** 444**),**

**(**'chemistry'**,** 444**),**

**(**'biology'**,** 444**),**

**(**'contract law'**,** 555**),**

**(**'civil rights law'**,** 555**),**

**(**'maritime law'**,** 555**),**

**(**'criminal law'**,** 555**),**

**(**'family law'**,** 555**),**

**(**'environmental law'**,** 555**),**

**(**'money'**,** 666**),**

**(**'accountancy'**,** 666**);**