**Introduction to Databases**

**Programming Project**

# 

# **Project Name: Post ITT**

# **Group**

# MM

# **Contributors**

Helen O’Brien

Katie Griffiths

Anthony Bruhl

Aoife Sayers

# **Date of Submission**

Phase 1 Analysis: Thursday 13th October 20%

Phase 2: Thursday 10th November 30%

Phase 3: Thursday December 8 20%

# **Introduction**

PostITT will be a ‘Community of Practice Group’\*, where users can share knowledge and information. A user can post information, responses, questions and answers on various categories, tags can also be applied to assist searching. Posts can be filtered and displayed by category, tag or user. Posts can be rated and from this users will receive a ranking, a leader board can be displayed for users . The posts will be displayed in order af their ranking. Media such as documents and images can be uploaded, displayed or links shared within posts. The posts will be overseen by a modulator, the modulator will receive alerts when content contains flagged words. Image content must be approved by Modulator.

\* A **community of practice** (**CoP**) is a group of people who share a craft and/or a profession. It is through the process of sharing information and experiences with the group that members learn from each other, and have an opportunity to develop personally and professionally ([Lave & Wenger 1991](https://en.wikipedia.org/wiki/Community_of_practice#CITEREFLaveWenger1991)). (Wiki )

# **Approach**

As a team we have chosen a Top down, Iterative approach. Firstly, we started off with a few high-level entities and relationships. Gradually, we then made top-down changes to identify lower-level entities, relationships and associated attributes. We also kept older versions of each part in the document so we could rollback to an older version if needed.

In top down approach, when the higher level parts are completed, we have a deeper, more mature idea of the overall system by working from general to specific. Whereas, if we were working with a bottom up approach we would have to analyse the more complex lower parts and develop upwards, while only gaining a limited knowledge of the overall system little by little.

In real life scenarios, top down is a more cost effective approach if the different components of the system have been evaluated well.

# 

# **Proposed Names for Project**

CollegePostIt

PostHelp

CollegeChitChat

UseMe

CollegeTalk

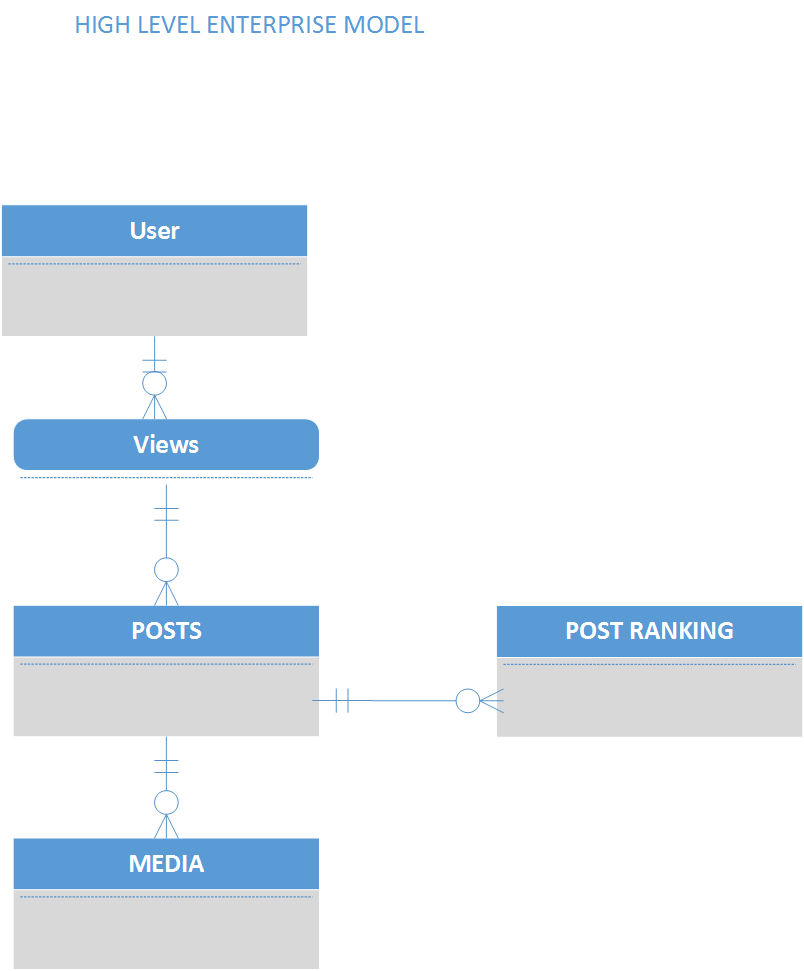
PostMe

Post ITT

**Unanimous Choice: PostITT**

# **Enterprise Model**

# **Version 1**



# **Enterprise Model**

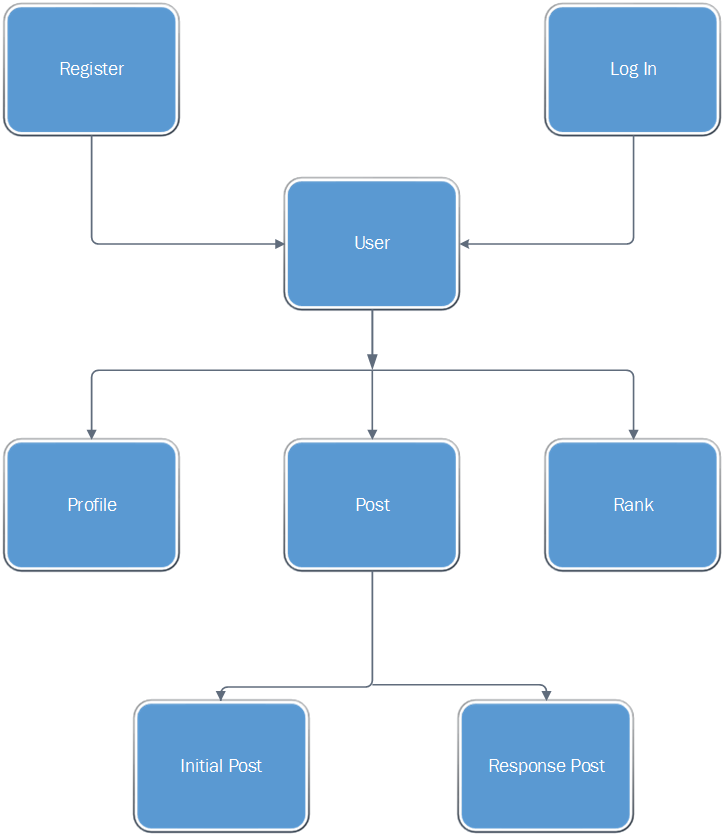
# **Version 2**

# 

# 

# 

# **Functional Decomposition Diagram**



# **User Requirements Version 1**

**User:** user is a person, who uses the website. He can be different type: standard user or moderator user.

* Standard user can be a student, a lecturer or society member... in ITT
* Moderator is a user who has the role to authorize a post

A person who want to use the website can register himself and become a user of the website

User can log in/log out

User can see the post page

User can send a post on post page about a category

User can respond to a post

User can give a rank to a post

User have a score, which is defined by the number of his posts time the sum of all his post ranks

User can visit his profile page

**Post:** post is a text send by a user, he can have several medias (pictures, or videos).

A post can have different type: Informative, reply, question, answer

* Informative post is a post which give a information about college life
* Reply post is a answer or a comment for a informative post
* Question post is a post to request something about courses
* Answer post is the solution to answer a question post

A post have rank, which can increase or decrease. The rank can be negative

**Profile:** profile is a summary of user information. He can see his personal information, his score, his post number

Some information on the profile page can be see only by his proprietary such as personal information.

**User registration:** To register person must fill out a form. Information asked are : first name, surname, date of birth, email.

The person must be register in ITT

**User login:** To log in website, user must give email and password

# **User Requirements Version 2**

User can register to use the application

User can deregister

User can log in

User can log out

User can view his profile

User can edit profile page

User can add a profile picture

User can view leaderboard

User can view the post page

User can put up a post on post page

User can respond to a post

User can give a post a category; Informative, reply, question, answer

User can give a post one or multiple tags

User can add media to a post

User can edit a post

User can delete a post

User can rank a post

User has a score, the number of his posts multiplied by the all his post ranking

# 

# 

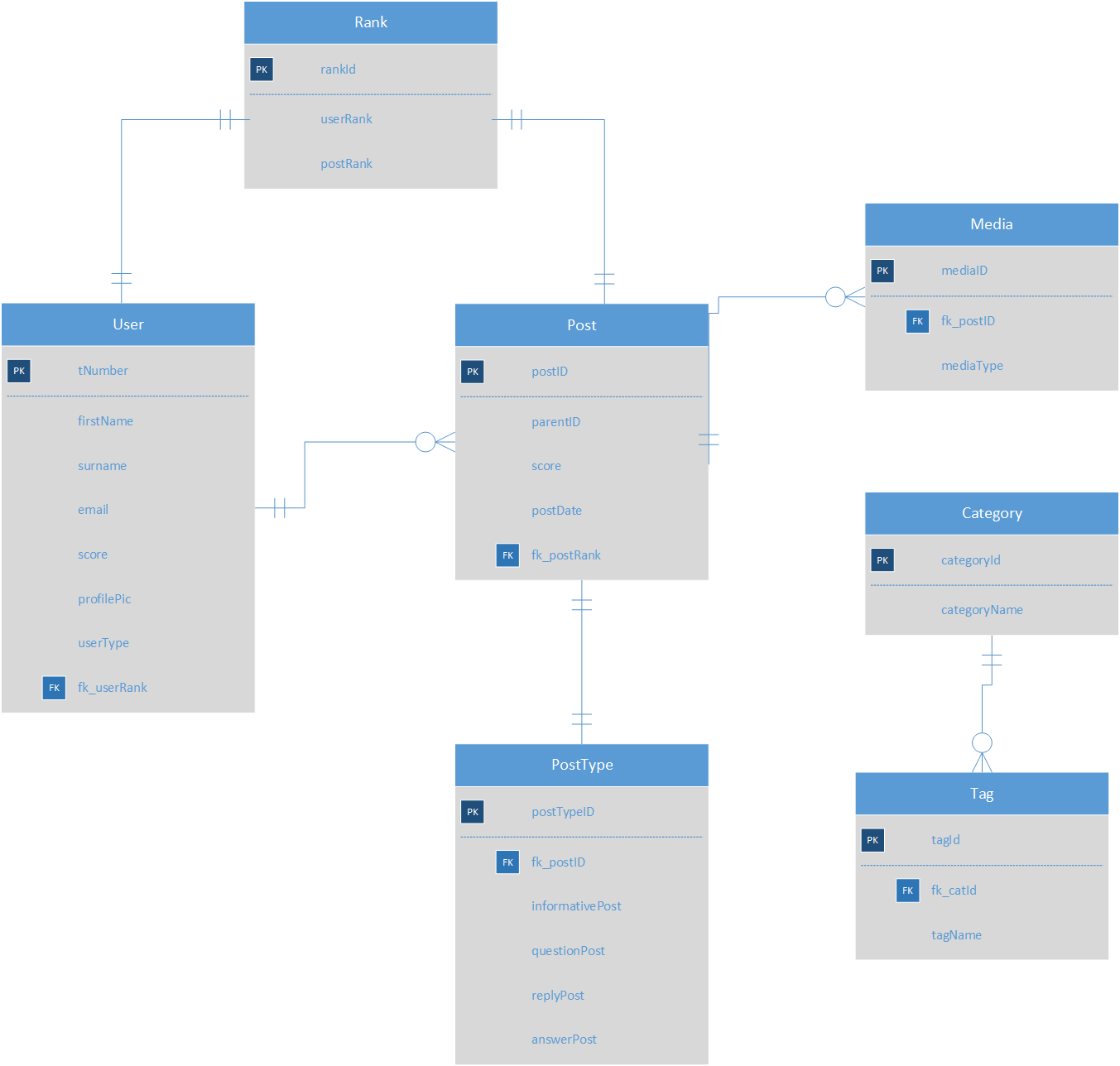
# 

# 

# 

# **Entity Relationship Diagram v1**

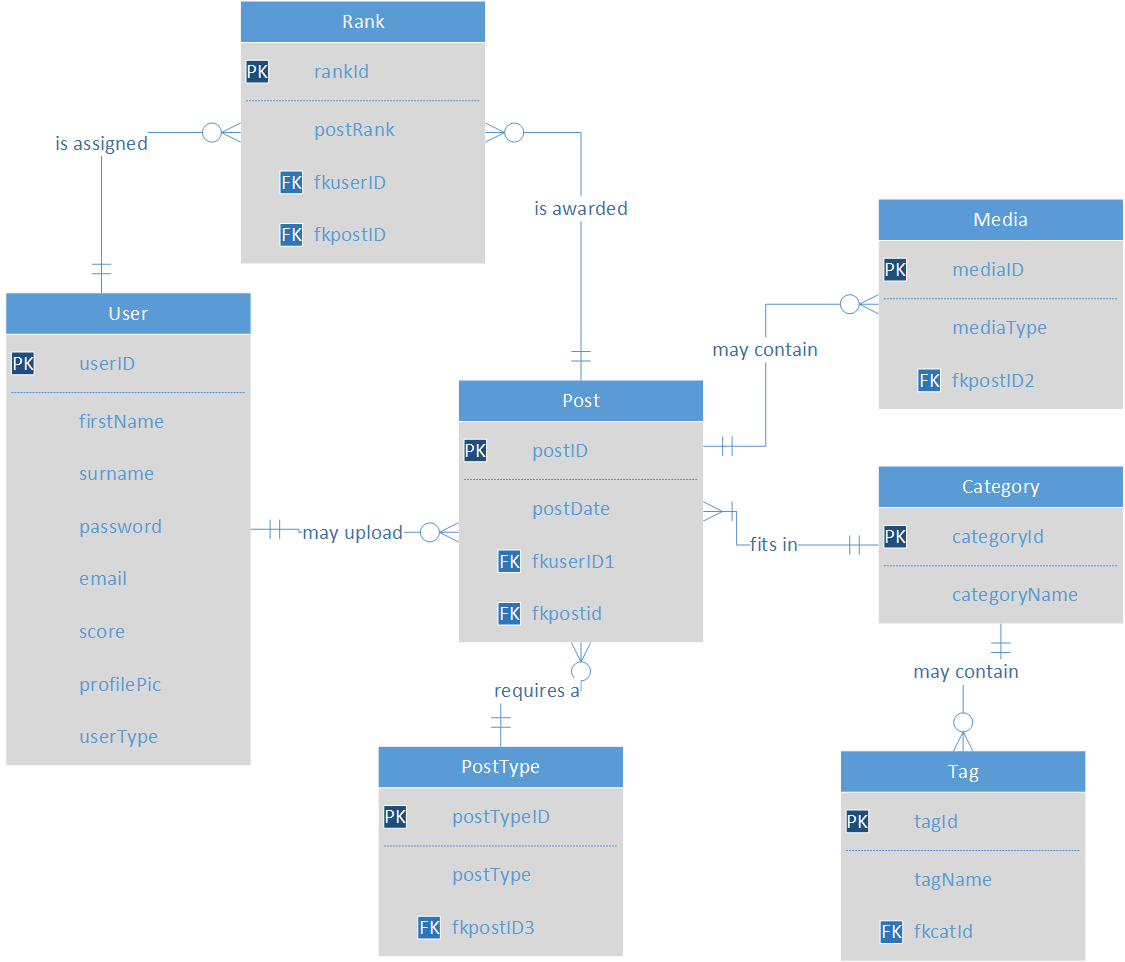
# 



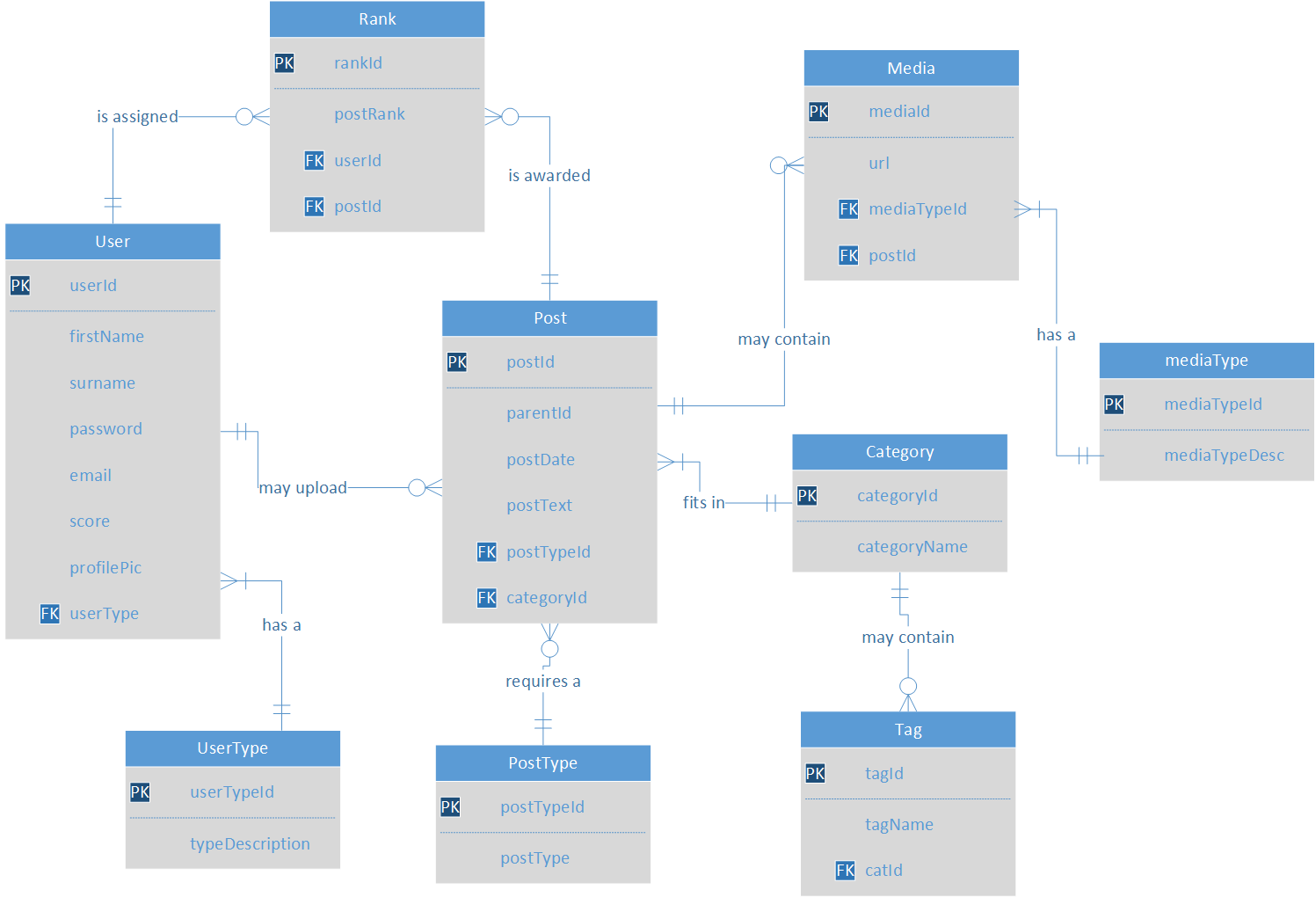
# 

# 

# **Entity Relationship Diagram v2**



# **Entity Relationship Diagram v3**



# 

# 

**\*parentId** = foreign key + a relationship from post to post table

Unique identifier for parent post - signifies an addition to a main post

# **Data Entity Matrix**

# Data Entity Matrix.PNG

# 

# **Business Rules**

**Structural**

Only current students can register for the website

User must be registered to use application

User’s email can be used for one account and one account only

The Modulator cannot participate in posting

**Constraint on a field**

# Registration details cannot be null

Login username must be valid

Login password must be valid

# A post cannot be null

# A post must have a category

# A post can only have one category

# A post can have no more than 10 tags

# 

# Users can only vote once per post

# No more than 2 pieces of media per post

# Media file can be no larger than 5 Mb

**Trigger**

Users ranking is calculated from number of posts multiplied by post ranking

User has only 3 attempts to login

Moderator will be alerted, triggered by keywords - inappropriate words – ducking

Breaches in protocol causing intervention by the Modulator will result in loss of ranking points

Password must achieve a certain strength - diversity of characters

Posts will be displayed in order af their ranking

Searches are performed by category, tag or user

User must give the video/image/document a title /description

Media must be validated and approved by the moderator

**Action**

Three failed login attempts results in being locked out for one hour

Users will be penalised for inappropriate language by losing points

A post that has been reported will be temporarily removed and must be approved by Moderator

**Data Dictionary**

# **Entity - USER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Size** | **Data Type** | **Description** | **Example** |
| userId | 8 | NUMBER(8) auto\_increment | Unique identifier for user | 1 |
| firstName | 20 | VARCHAR(20) | Users First Name | Foxy |
| surname | 20 | VARCHAR(20) | Users Surname/Last name | Lady |
| password | >=5 & <= 30 | VARCHAR(20) | Used for logging in & registering for an account  Mixed Case -Contains numbers & characters | Password123@ |
| email | 30 | VARCHAR(20) | Users email address, used to log into the account | t00170881@gmail.com |
| score | 5 | NUMBER(5) | Users score calculated by  userRanking X COUNT(postid/No. Posts) | 123 |
| profilePic | 50 | VARCHAR(50) | Users profile picture for display | ../picture/mypicture.jpeg |
| userType | 1 | VARCHAR(1) | User or Moderator  DOMAIN - ‘U’ OR ‘M’  Foreign key referencing UserType | ‘U’ OR ‘M’ |

**Entity - USERTYPE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Size** | **Data Type** | **Description** | **Example** |
| userTypeId | 5 | NUMBER(5) | Unique identifier for a user type  CONSTRAINT | 2 |
| typeDescription | 2 | VARCHAR(2) | Describes the user type | User, Modulator |

**Entity - POST**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Field Size | Data Type | Description | Example |
| postID | 5 | NUMBER(5) | Unique identifier for post  Generated using auto-increment - CONSTRAINT | 055 |
| parentId | 5 | NUMBER(5) | Unique identifier for parent post - signifies a response or answer to a post | 34 |
| postDate | 9 | DATE | Date of post in format dd-mmm-yy | 05-OCT-16 |
| postText | 100 | VARCHAR | Text of a post | Hello World! It’s me Joe Soap |
| postTypeId | 5 | NUMBER(5) | Foreign key referencing postType | 343 |
| categoryId | 5 | NUMBER(5) | Foreign key referencing category | 321 |

**Entity - POSTTYPE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Size** | **Data Type** | **Description** | **Example** |
| postTypeId | 5 | NUMBER(5) | Unique identifier for post type  Generated using auto-increment - CONSTRAINT | 034 |
| postType | 20 | VARCHAR(20) | Determines whether the post is Informative, a question, an answer, or a response.  DOMAINS - Defined as domains in database | Informative |

# **Entity - CATEGORY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Size** | **Data Type** | **Description** | **Example** |
| categoryId | 5 | NUMBER(5) | Unique identifier for category | 321 |
| categoryName | 20 | VARCHAR(20) | Name of Category  Category should be the general area name, ie. Computing, Business, Nursing | Computing |

# **Entity - TAG**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Size** | **Data Type** | **Description** | **Example** |
| tagId | 5 | NUMBER(5) | Unique identifier for tag | 050 |
| tagName | 20 | VARCHAR(20) | Name of Tag  Tag names are sub categories entered by the user ie. Web Development, Java etc.  Entering a tag for a category is not mandatory. | Java |
| catId | 5 | VARCHAR(5) | Foreign key reference for categoryId in the Category Entity | 321 |

**Entity - MEDIA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Size** | **Data Type** | **Description** | **Example** |
| mediaId | 5 | NUMBER(5) | Unique identifier for media | 055 |
| url | 30 | varchar(30) | Url address for accessing media | \images\mypic.jpg |
| mediaTypeId | 5 | NUMBER(5) | Foreign key reference for mediaTypeId in the Mediatype Entity | 055 |
| postId | 5 | NUMBER(5) | Foreign key reference for postId in the Post Entity | 350 |

**Entity - MEDIATYPE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Size** | **Data Type** | **Description** | **Example** |
| mediaTypeId | 5 | NUMBER(5) | Unique identifier for media type | 055 |
| mediaTypeDesc | 20 | VARCHAR(20) | Types of Media allowed on posts  Images - jpg, jpeg, png, gif  Documents - .docx, .doc, pdf, .java, .html, .txt, .htm | myFile.txt  ../documents/myFile.txt |

**Entity - RANK**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Size** | **Data Type** | **Description** | **Example** |
| rankId | 5 | NUMBER(5) | Unique identifier for rank | 343 |
| postRank | 5 | NUMBER(5) | Score calculated by user voting system | 9 |
| userId | 5 | NUMBER(5) | Foreign key reference for userId from Users table | 43 |
| postId | 5 | NUMBER(5) | Foreign key reference for postId from Posts table | 23 |

# 

# 

# 

# 

# 

# 

# **Relational Schema**

**User** (userId, firstName, surname, password, email, score, profilePic, userType)

**UserType**(userTypeId,typeDescription)

**Post** (postId, parentId, postdate, postText, postTypeId, categoryId)

**PostType** (postTypeId, postType)

**Category** (categoryId, categoryName)

**Tag** (tagId, tagName, categoryId)

**Rank** (rankId, postRank, userId, postId)

**Media** (mediaId, url, mediaTypeId, postId)

**MediaType**(mediaTypeId, mediaTypeDesc)

# **NORMALISATION**

|  |  |  |  |
| --- | --- | --- | --- |
| **UNF** | **1NF** | **2NF** | **3NF** |
| userId  firstName  surname  password  email  score  profilePic  userType  postdate  postText  postType  categoryName  tagName  postRank  mediaType  Url | userId  firstName  surname  password  email  score  profilePic  userTypeId  userTypeDesc  postId  postdate  postText  postTypeId  postTypeDesc  categoryId  categoryName  tagId  tagName  postRankId  postRank  mediaId  url  mediaTypeId  mediaTypeDesc | userId  firstName  surname  password  email  score  profilePic  userTypeId  userTypeDesc  postId  postdate  postText  postTypeId  postTypeDesc  categoryId  categoryName  tagId  tagName  postRankId  postRank  mediaId  url  mediaTypeId  mediaTypeDesc | userId  firstName  surname  password  email  Score  profilePic  userType fk  userTypeId  userTypeDesc  postId  parentId fk  postdate  postText  postTypeId fk  categoryId fk  tagId fk  postTypeId  postType  tagId  tagName  categoryName fk  categoryId  categoryName  rankId  postRank  userId fk  postId fk  mediaTypeId  mediaTypeDesc  mediaId  mediaType  url  postId fk |

# 

# 

# **Database Schema**

CREATE TABLE User (

userId Integer(8) autoIncrement not null,

firstName Varchar2(20) not null,

surname Varchar2(20) not null,

password Varchar2(30) not null,

email Varchar2(30) not null unique,

score Integer(5) default 0,

profilePic Varchar2(50),

userType(1) Integer not null,

Constraint pk\_first\_name PRIMARY KEY (userId),

Constraint fk\_user\_type FOREIGN KEY (userType) REFERENCES UserType (UserTypeID)

)

## CREATE TABLE UserType(

userTypeId Integer(5) autoIncrement not null,

typeDescription Varchar2(1) not null unique,

Constraint pk\_first\_name PRIMARY KEY (userTypeId),

)

CREATE TABLE Post(

postId Integer(5) autoIncrement not null,

parentId Integer(5),

postdate date not null,

postText Varchar2(100) not null,

postTypeId Integer(5) not null,

categoryId Integer(5) not null,

Constraint pk\_post\_id PRIMARY KEY (postId),

Constraint fk\_parent\_id FOREIGN KEY (parentId ) REFERENCES Post(postId),

Constraint fk\_post\_type\_id FOREIGN KEY (postTypeId) REFERENCES PostType(postTypeId),

Constraint fk\_category\_id FOREIGN KEY (categoryId) REFERENCES Category(categoryId)

)

CREATE TABLE PostType (

postTypeId Integer(5) autoIncrement not null,

postType Varchar2(20) not null unique,

Constraint pk\_post\_type\_ id PRIMARY KEY (postTypeId )

)

CREATE TABLE Category(

categoryId Integer(5) autoIncrement not null,

categoryName Varchar2(20) not null unique,

Constraint pk\_category\_ id PRIMARY KEY (categoryId)

)

CREATE TABLE Tag(

tagId Integer(5) autoIncrement not null,

tagName Varchar2(20) not null unique,

catId Integer(5) not null,

Constraint pk\_tag\_ id PRIMARY KEY (tagId),

Constraint fk\_cat\_id FOREIGN KEY (catId) REFERENCES Category(categoryId)

)

CREATE TABLE Rank(

rankId Integer(5) autoIncrement not null,

postRank Integer(5) not null,

userId Integer(5) not null,

postId Integer(5) not null,

Constraint pk\_rank\_ id PRIMARY KEY (rankId),

Constraint fk\_user\_id FOREIGN KEY (userId) REFERENCES User(userId),

Constraint fk\_post\_id FOREIGN KEY (postId) REFERENCES Post(postId )

)

CREATE TABLE Media(

mediaId Integer(5) autoIncrement not null,

mediaTypeId(5) Integer not null,

url Varchar2(30) not null,

postId integer(5) not null,

Constraint pk\_media\_ id PRIMARY KEY (mediaId),

Constraint fk\_media\_type\_id FOREIGN KEY (mediaTypeId ) REFERENCES MediaType(mediaTypeId),

Constraint fk\_post\_id FOREIGN KEY (postId) REFERENCES Post(postId )

)

CREATE TABLE MediaType(

mediaTypeId Integer(5) autoIncrement not null,

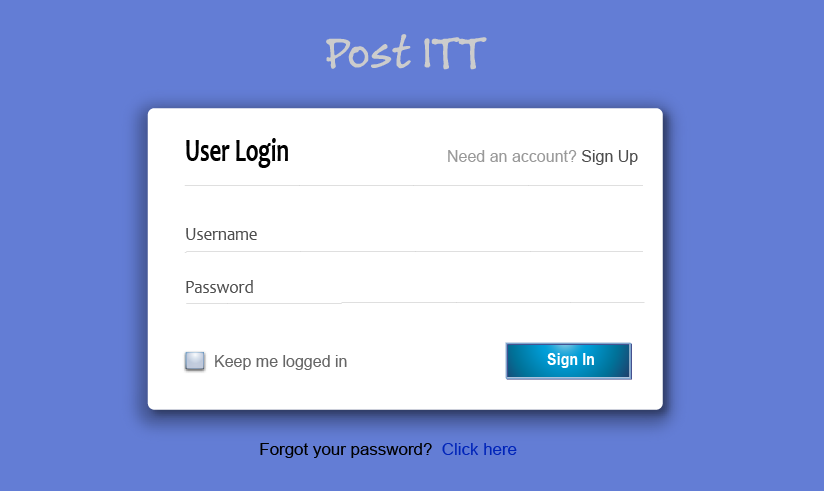
MediaTypeDesc Varchar2(20) not null unique,

Constraint pk\_media\_type\_ id PRIMARY KEY (mediaTypeId)

)

**UI Samples**

**Log In**



The Post ITT login page will allow the user to successfully log in to the site or it will allow them to sign up/register to gain access to the site. The following is needed for a successful log in/registration:

Log in:

A current user will input their username and password that was validated when first registered.

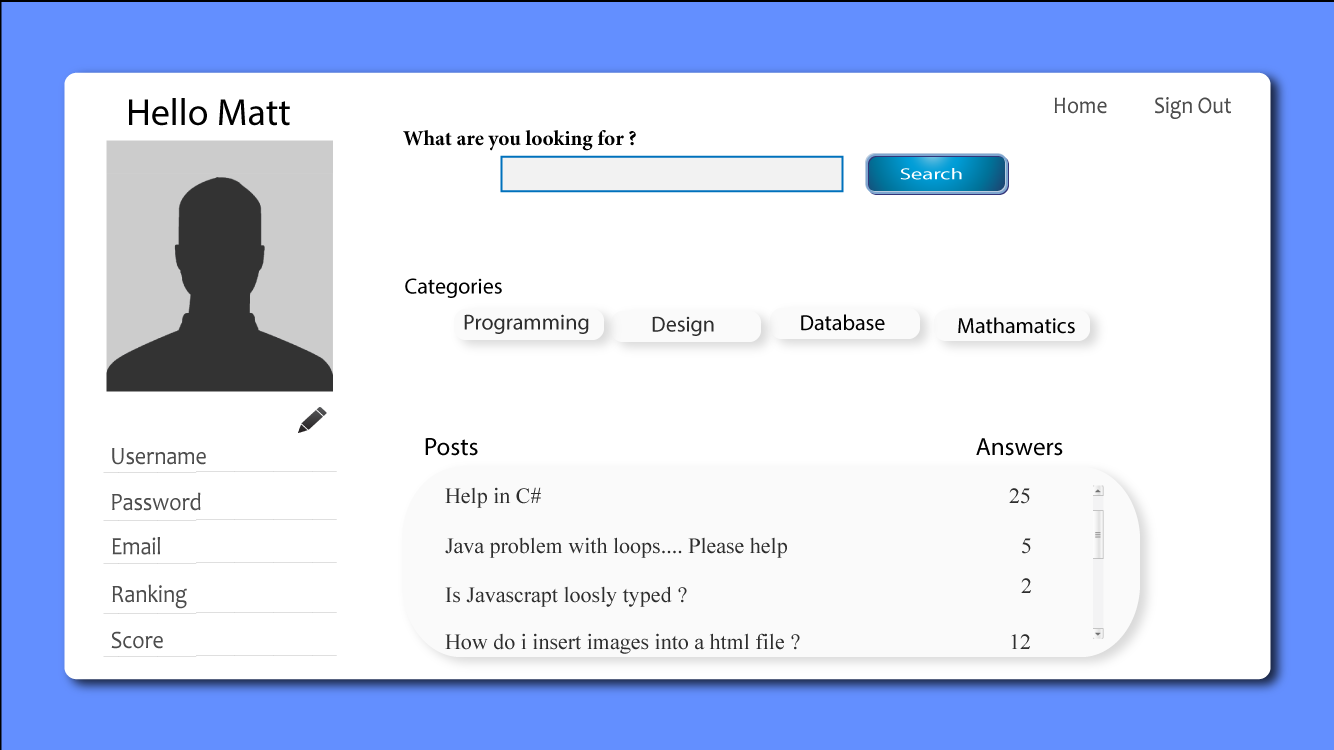
Register:

A user wishing to join the site will need to fill in their first name and surname, a valid email address and a valid password. Once completed they will then have access to the site and there profile.

**Business Rules for log in/register screen**

1. The user will only have 3 attempts to gain access to the site, if it fails they are locked out for one hour till access can be gained again.
2. The email address must be a valid email and not used for any other account on the site.
3. When registering all fields must have a value.
4. First name and last name must only contain letters
5. First and last name must be less than 20 characters long.
6. Password must be minimum 5 and less than 20 characters long.

**Profile Page**



The PostITT Profile page will allow the user to view their profile they created when the registered to be a member. On this page the user will be allowed to edit and view current status on their profile in the following ways:

1. Update and edit personal details, Username,Password,Email etc.
2. View categories that they are signed up for.
3. View posts that they have wrote and have been ranked.
4. Search across the platform of the site.
5. Navigate back to the home page.
6. Change the profile picture.

**Business Rules for profile page**

1.The user's password must be between 5 and 20 characters long..

2. User's email must not be longer than 20 characters long and contain the @ symbol.

3. User's profile picture must not be larger than 2mb in size.

4. A user can have no more than 5 categories selected.

**Post Page/Main Page/Home Page UI**



The Main Page/Home Page will allow the user to navigate through the site, they will be able to see the posts of different users also the rankings that have been given to each post. They will be able to access:

1. The posts that have been added to the site.
2. Answer any post that the user has interest in.
3. Write a post in question to other user’s on the site.
4. Search for a particular post, category or tag.
5. View their profile.
6. Search for a particular person.
7. View other people's rankings on posts.
8. Vote on a particular post.

# 

# 

# **CASE Tools Evaluation**

**Case Tool Evaluation (Visio)**

The tool for creating the Enterprise Data Model Diagram (ERD), the Functional Decomposition Diagram and the Entity Relationship Diagram for this project is Microsoft Visio. With having experience in previous projects working with this software was reasonably intuitive. The software works well in creating diagrams of choice and has a familiar setting to other Microsoft office programs.

In selecting the software you are presented with a choice of templates to choose from, each ranging from Engineering, Database, Software, Flowcharts and Business, with there own set of shapes assigned to help in creating your diagram/chart. When the desired choice is selected you are presented with a blank canvas and a preset layout to the left in which you can view the shapes that are available.

We created the ERD Diagram using Crow’s Foot Database Notation template. We then had to simply select and drag onto the canvas the various types of stencils ranging from Entity, Primary Key Attribute, Primary Key Separator, Attribute and Relationship. If selecting the Entity you simply click and drag across onto the blank canvas from there you can resize by dragging the corners.

Once the Entity is in place you can double click on the Entity box it allows you to edit the Entity Name and the attribute names. With the first attribute name being marked as the primary key by default, once it has been assigned a value the attribute will be displayed as underlined in the diagram. Each Entity can be assigned a theme through the design tab in the menu, this gives a chance to make your diagram pop and be more visual.

When adding the relationship/connectors this can be a tricky task if you have little patience, once selected and brought over onto the canvas the aligning with each entity takes a bit of practice. Once in place you are then able to format the connectors, giving each start and end the desired crow’s feet symbol amongst other formatting to the shapes. A couple of options were not available but with add on’s available from online they would be easy to download and instal into visio.

With various options to include images and bring in files to add they were all optional which gives this software a one up on designing diagrams and charts based on the standard Microsoft software that are available.

In conclusion with this report in my eyes i would highly recommend using the visio software as it is an easy tool when developing a diagram/chart for any database/business design model, with a huge selection of templates it gives the user plenty of choices.

**Case Tool Evaluation (Google Docs)**

As a group it was agreed to use Google Docs to assist us in developing the project. It gave us the chance to add our own details and comments to put this project together. Each of us in the group had access to give us real time collaboration and enable us to provide comments in different sections as well as the chance to leave questions and to answer them when needed.

Google Docs is a free service available through Google using the url link: <http://docs.google.com/>

Google Docs allows the access and storage of files up to 15GB from any device enabling the user to edit or change files from any location.

In turn we as a group found Google Docs a very useful source in detailing the project and being able to insert diagrams produced in other software. The real time collaboration each day gave us the chance to understand and complete the project outlined above. If anything was added or changed in the document, comments were left to assist in the other parties of the team as to which information had been altered. While the project is still ongoing we still resort to group meetings during college time and out of college work is produced through this service. We also commented on areas that needed clarification. Other users who saw the comment could then reply.

# **Case Tool Evaluation (Adobe Illustrator CS6)**

As a group we decided to include illustrated UI designs for what the screen layouts would look like. For this a piece of software was used called Adobe Illustrator CS6. As multimedia students we are all familiar as to how this software is used.

Illustrator is a vector graphics editor and is a lot like photoshop but has an easier layout to enable you to make any shape/drawing or design contained in layers. This version was the first version to include 3 dimensional capabilities allowing users to extrude or resolve shapes in making 3D objects.

Each screen was designed using all the groups ideas as to what each screen would comprise of. Colours were kept as normal as possible and a few shadows were added to make the screens have a pop out effect. When each UI was being designed we wanted to keep the colour scheme that was chosen the same for each UI. Overall each person in the group was happy with the final designs.

# **Inspiration for PostITT**

* StackOverflow
  + Question & answer site for programmers.Users are working together to build a library of detailed answers to every question about programming or other topics - Graphic design etc.
* Edmodo
  + Online classroom in form of social media. With Edmodo, students and teachers can reach out to one another and connect by sharing ideas, problems, and helpful tips.
* Facebook
  + World wide known social media platform.
  + Ability to upload photos and video, send messages, ask questions and keep in touch with friends, family and colleagues.
* Blackboard
  + A tool that allows staff to add resources for students to access online.

**Additional Tasks included:**

* User Requirements
* Data Entity Matrix
* Data Dictionary
* Normalisation Table
* Database Schema
* UI Samples

**Conclusion**

In conclusion, the first phase of this project was successful. Everyone on the team managed to compromise on ideas for the project, work as a team on this document. Each team member contributed their input and queried any issues we weren’t sure of. Any issues had were resolved and this detailed document was compiled as a result.

# **References**

"Java Tutorial 20 : Connect To Mysql Database". *YouTube*. N.p., 2016. Web. 21 Sept. 2016.

"CASE Tools". *Umsl.edu*. N.p., 2016. Web. 4 Oct. 2016.

Reddit DB Design

Srinivas, Dasari. "Simple Reddit Database Design". *sodhanalibrary*. N.p., 2015. Web. 6 Oct. 2016.

Top Down vs Bottom Up

"Advantages And Disadvantages Of The Top-Down Andbottom-Up Implementation Approaches".*Publib.boulder.ibm.com*. N.p., 2016. Web. 6 Oct. 2016.