Rough Concept Publisher

Todo

* Improve imbedded content.
* Create a link to quickly post a news story when new content is posted.
* RSS feed for news stories.
* Add an admin option to make a specific feature or a seed based on the day.
* Access to change admin options.

Database Specifications

# Database Information

# Naming

A sort of Hungarian notation is used, items are prefixed according to what they are. The name itself should describe what the object is, names should be as short as possible but long enough to uniquely describe the object.

## Notation

* tbl: Table, eg tblContent.
* txt: Text including blobs of text and character arrays, eg txtTitle.
* n: Integer, eg nNewItem.
* f: Float, size doesn’t matter, eg fPrice.
* b: Bool value.
* dt: Date, eg dtCreated.
* qry: Query (usually in php, not in the database itself).
* id: When refering to an id in another table, it should be follow by the name of the table without the leading tbl, eg idContent would refer to the id of the item in tblContent. Also a table id field may be “id” without a name so in the table Content the id is “id” instead of “idContent”, “idContent” is acceptable but not prefered. When there is more than one id linked from a table and underscore should separate the text that distingueshes what the two ids are, eg idUser\_From, idUser\_To, both of these ids link to the table tblUser so a dinstinction must be made as to what the two references are used for.

# Tables

## tblContent

This is is the main table, it holds the contents of all upload content.

tblContent(id, *idUser*, txtSeries, fVolume, txtTitle, txtAuthorLast, txtAuthorFirst, txtDesc, dtProduced, txtFile, dtPublished, dtUpdated, txtSort)

## tbl ContentGenres

This is the link between content items, and genres, for each genre that a content is part of there should be an entry in this table.

tblContentGenres(*idContent*, *idGenre*)

## tblGenre

A list of all genre’s available.

tblGenre(id, txtDesc)

## tblUser

All the users and their access levels.

tblUsers(id, txtUserName, txtPassword, txtAlias, nAccessLevel)

## tblComment

Comments posted, linked to content.

tblComments(id, *idContent*, *idUser*, txtName, txtEmail, txtComment, dtPosted, bApproved)

## tblNews

News items.

tblNews(id, *idUser*, dtPosted, txtTitle, txtBody)

CREATE TABLE `RCPublisher`.`tblNews` (

`id` INT NOT NULL ,  
`idUser` INT NOT NULL ,  
`dtPosted` DATETIME NOT NULL ,  
`txtTitle` CHAR( 20 ) NOT NULL ,  
`txtBody` TEXT NOT NULL ,  
PRIMARY KEY ( `id` )

) ENGINE = MYISAM

## tblMessage

Messages sent.

tblMessages(id, idUser\_To, idUser\_From, txtEmail, txtSubject, txtMessage, bRead, dtSent)

CREATE TABLE `RCPublisher`.`tblMessage` (

`id` INT NOT NULL ,  
`idUser\_To` INT NOT NULL ,  
`idUser\_From` INT NULL ,  
`txtEmail` CHAR( 20 ) NOT NULL ,  
`txtMessage` TEXT NOT NULL ,  
`bRead` BOOL NOT NULL ,  
`dtSent` DATETIME NOT NULL ,  
PRIMARY KEY ( `id` ) ,  
INDEX ( `idUser\_To` , `dtSent` )

) ENGINE = MYISAM

PHP Specifications

RC Publisher uses object oriented php. Each page has it’s own class which inherits from a base class. The base class is responsible for the website layout. The class for each page only manages what is displayed in the content section of each page.

# Naming

Variable names should be very descriptive of the variable, a hungarian type notation is used.

## Notation

* n: Integer types.
* b: Bool types.
* f: Floating point types.
* str: String types.
* qry: Strings that are specifically an sql query.
* m\_: Class member attributes.

Function and method names should be have each word capitalized, eg TheFunction($nValue).

Class names start with C and then have a name using the same format as functions. They should be in a file named after the class but in all lowercase letters and words separated by underscores. Only one class should be used per file. Class files themselves should not be webpages, and should be in a private folder, another php script should load the class and display the contents of it.

$\_GET and $\_POST variable names should generally be one word, and all lower case, if more than one word is needed to describe it then it should be separated by underscores. If a name might need two words, but it would make the name really long an abbreivation should be used for the name (eg u\_file instead of upload\_file).

Pages

# Main Page (index.php, index.php?content=home)

The main page displays only the newest content that has been uploaded. New comments are shown, as well as new content from the blog site. A detailed design spec has not been documented yet, but that will come shortly.

The basic idea is that the newest content, and links to all other content is presented to the viewer.

# Table of Contents Page (index.php?content=toc[&page=\*][&sort=\*])

The list page presents a list of all content within the database. The list page is controlled by the get parameters on the URL. It can be sorted by all the content displayed in the list (except for description). When sorting by genre the list appearance is different than all other list types. The list only presents 20 items per page, and multiple pages are available to view. The 20 items limit is performed differently when sorting by Genre.

# Upload Page (index.php?content=upload)

The upload page allows new content to be uploaded to the site. Only users with the appropriate access can upload content. The upload page performs various checks to make sure that the uploaded content is formatted correctly. The upload page will upload anything, but it shows a warning if the content is not in PDF format.

# Login Page (index.php?content=login[&logout])

The login page allows an administrator to log in. The page performs authentication. It also allows logged in users to logout with the appropriate parameters.

# Content Page (index.php?content=content&id=(?))

The content page displays the content of a particular piece of content. It also allows a logged in administrator to go to the edit content page for that particular item. Non-users may also post comments about the particular piece. Administrators may also delete comments.

# Edit Content Page index.php?content=editc&id=(?))

Allows an administrator to enter various content items, including series, volume, title, author, description, date written, and allows the user to replace the existing content with a new upload.

# Edit Genres Page (index.php?content=editg)

Allows an administrator to add or delete genres to the available genres. Note that if a genre is deleted, any titles that had that genre will no longer be assigned to it as it removes not only the entry from the Genres table, but also any genres with matching id from the PieceGenre table.

# Inbox (index.php?content=email[&message=(?)])

Allows a logged in user to see the messages that they have been sent, delete them, and read them.

# Contact Page (index.php?content=contact).

Allows someone to email a registered user.

User Access

User accounts have an access level. Each page is assigned an access level, with 1 being the lowest level, and all public pages have an access level of 1. If a browser tries to navigate to a restricted page, the page will be not be processed at all (POST or GET messages will not be processed). Only a message stating that the page is restricted will be displayed. User access is handled by PHP sessions with variables stored in $\_SESSION.

# Passwords

Passwords are stored in the database by their MD5 hash. Because the web server does not have an encryption certificate, the login is managed by javascript hasing the password into md5. Then a new string is created with a unique key appended to the hashed password, and that string is then hashed using sha1. In order to authenticate the user, the password is retrieved from the database, and then the same alogrithm for the sha1 hashing is applied to the retrieved password, and the two are compared. That way the actual password, and it’s md5 hash are not transmitted over the internet, it all the actual password exists only on the client, and only a hashed version of the md5 is transmitted. Not as secure as SSL but much cheaper than a certificate.