2014

Rule Based Modeling Annotation

[Type the document subtitle]

Dennis A. Simpson

Rule Based Modeling Annotation Site

Table of Contents

[Introduction 3](#_Toc369431671)

[Structure 3](#_Toc369431672)

[Installation 3](#_Toc369431673)

[Prerequisites 3](#_Toc369431674)

[Setup 4](#_Toc369431675)

[Future Needs & Known Bugs 6](#_Toc369431676)

[Needed Features 6](#_Toc369431677)

[Bugs 7](#_Toc369431678)

[Using the Site 7](#_Toc369431679)

[Edit Users 8](#_Toc369431680)

[Login 8](#_Toc369431681)

[ECM Figures 9](#_Toc369431682)

[Diagrams 11](#_Toc369431683)

[Future ECM Figures 11](#_Toc369431684)

[Search Functions 11](#_Toc369431685)

[Future Search Additions 12](#_Toc369431686)

[Daily Builds 12](#_Toc369431687)

[List Daily Build Records 12](#_Toc369431688)

[Bugs and Future Considerations 12](#_Toc369431689)

[Operations 12](#_Toc369431690)

[Molecule 12](#_Toc369431691)

[Add External Links 13](#_Toc369431692)

[Add Components 13](#_Toc369431693)

[Future Molecule Considerations 14](#_Toc369431694)

[List and Enter Rules 14](#_Toc369431695)

[Link Rules to Molecules 14](#_Toc369431696)

[Future Rule Operations 14](#_Toc369431697)

[ECM Notes 14](#_Toc369431698)

[Link ECM Notes to Publications 15](#_Toc369431699)

[Future Contact Map Operations 15](#_Toc369431700)

[User Functions 15](#_Toc369431701)

[Edit User Profile 15](#_Toc369431702)

[List and Edit Projects 15](#_Toc369431703)

[User Function Issues 15](#_Toc369431704)

[System Administration 16](#_Toc369431705)

[List Users 16](#_Toc369431706)

[List User Groups 16](#_Toc369431707)

[List and Edit Projects 16](#_Toc369431708)

[PHP Schema 16](#_Toc369431709)

[PHP Schema Future Needs 18](#_Toc369431710)

[MySQL Schema 19](#_Toc369431711)

[Tables 19](#_Toc369431712)

[ci\_sessions: 19](#_Toc369431713)

[components: 19](#_Toc369431714)

[daily\_build: 20](#_Toc369431715)

[doi: 20](#_Toc369431716)

[ecmnote: 20](#_Toc369431717)

[group\_key: 20](#_Toc369431718)

[login\_attempts: 20](#_Toc369431719)

[molecule: 20](#_Toc369431720)

[projects: 21](#_Toc369431721)

[pubauth: 21](#_Toc369431722)

[pubmed: 21](#_Toc369431723)

[rulemol: 21](#_Toc369431724)

[rules: 21](#_Toc369431725)

[user\_autologin: 22](#_Toc369431726)

[user\_profiles: 22](#_Toc369431727)

[users: 22](#_Toc369431728)

# Introduction

Rule based modeling approaches greatly simplify the task of creating a mathematical model. This package was written with the intention to aid in annotating rule based mathematical models of the cell cycle, specifically models based on BioNetGen, NFsim, or KaSim. In practice there is no reason that this package could not be used to annotate any mathematical model of biology since rules and custom equations both should have certain logic as to their creation and supporting publications justifying that logic. Throughout this site rule based modeling nomenclature follows that of BioNetGen Language (BNGL) syntax, unless noted otherwise. This version of the site was designed and written by Dennis A. Simpson.

# Structure

This site is built on the Codeigniter v2.2 PHP framework using iScaffold v2.2 to generate the default controller, model, and view files. The framework files where then extensively altered to better suit the needs of the project. Site login is handled with Tank Auth and the user groups are handled with Tank Auth Groups. The image zoom script is from FWDesign. The style sheets are mostly Twitter Bootstrap and the java is your basic jquery 2.x. The site has been tested to run with Firefox ≥ v19 and Microsoft Explorer (32 and 64 bit versions) ≥ v9. It should work on Chrome and Safari however the layout might be off on those two.

The layout of the site is licensed under Creative Commons and the schema and code is licensed under Apache 2.0. The copyright is held by Dennis Simpson. By default the installed package will not allow users to register. This was chosen as a security measure. During the installation and setup you will need to decide how to handle this. Also, by default, CAPTCHA’s are turned off. The built-in CAPTCHA function leaves a lot to be desired so if you wish to use them the recommendation is to use re- CAPTCHA.

One very important note about site user passwords, they are server dependent. That means you cannot move the MySQL DB to another server and have the passwords remain valid. This security feature can be turned off but that is not recommended. The passwords are stored in the DB as salted, encrypted strings. It is not possible to recover lost passwords.

# Installation

Make sure you have the prerequisites listed below before attempting to install this site. This package is designed to run on an Apache server on a Linux box. There should be no issues with others as long as the web server can be tied into PHP and MySQL. During development this package was tested on a Windows7 box running Apache with no issues. All the files required to run the site are in the compressed archive.

## Prerequisites

* Server to host site. Can also be installed on a single machine and run as localhost. Installing Apache on Windows boxes can be a bit tricky. I used the Windows binary from Zend to install both Apache and PHP. Tested with Apache v2.2
* MySQL server installation. Need a version ≥5.5, community server edition works so I assume the enterprise edition would work as well.
* PHP ≥5.4 that has been configured to work with the server. Make sure the php.ini file codes the correct region for the time function.
* A good text editor. I like Notepad++ for Windows. The xml files used to display the images do not appear to load correctly if saved from PHP Designer, my favorite IDE.

\*\*The next packages are only required if you plan on editing any of the components of the site. The Tank\_Auth, phpass, and Tank\_Auth\_Groups have not been updated in quite some time.

* Codeigniter installation. I recommend using the latest version and fixing any syntax changes that may have occurred. Future support for CodeIgniter is currently in doubt.
* An IDE is very helpful but if you are hacking the code you should already know this. I used PHP Designer v8.0 when designing this site.
* The latest version of Tank\_Auth for Codeigniter. This provides the basic login functions.
* The latest version of phpass. Tank\_Auth comes with this but does not use the latest version. If you install a different version of this than the one included with Tank\_Auth then make sure to edit Tank\_auth.php line #3 to include the correct phpass.
* The latest version of Tank\_Auth\_Groups for Codeigniter. This is an extension of the Tank\_Auth package that allows for user groups in the login.

## Setup

Once you have MySQL set up on a server that can recognize PHP, you are ready to build the site. To set up the site first unpack the archive to the root of your server as a user with root permissions. This should create a directory called model with several subdirectories under it. Make sure the /model/application/compiled, /model/uploads, and the /model/application/logs folders are writable. The MySQL dump file is found in the /model directory and is called model.sql. Use phpMyadmin or MySQL command line to load this file into your MySQL installation. This file will create a DB called “model” and all the tables required for the Site. It is highly recommended that you log into your MySQL installation as root and set a user ID and password for the model DB that does not have root privileges. Using the root ID and password for DB access from the site is a tremendous security risk. All the package files should now be installed but you are not quite ready to start using it. Check and edit the following:

/model/application/config/config.php

* Line 17 Set the base URL for your site.
* Line 227 Enter a new session encryption key.
* Line 231 Confirm that the Session Variables are all set as you want.

/model/application/config/database.php

* Edit this file for the host name as well as correct user name, and password for MySQL

/model/application/config/tank\_auth.php

* Lines 10 & 11 Edit this with your information.
* Lines 22 & 23. Highly recommend that line 22 remain “FALSE”. Some blogs discussing phpass recommend line 23 being set to >50.
* Lines 42 – 52. This is the registration section. By default the captcha is off. I recommend that the “allow registration” be set to off after the initial user is set up unless there is someone to monitor all new users. By default the site adds new users with Guest privileges only. New users should only be added with the “allow registration” set to true.

/model/application/view/login\_html

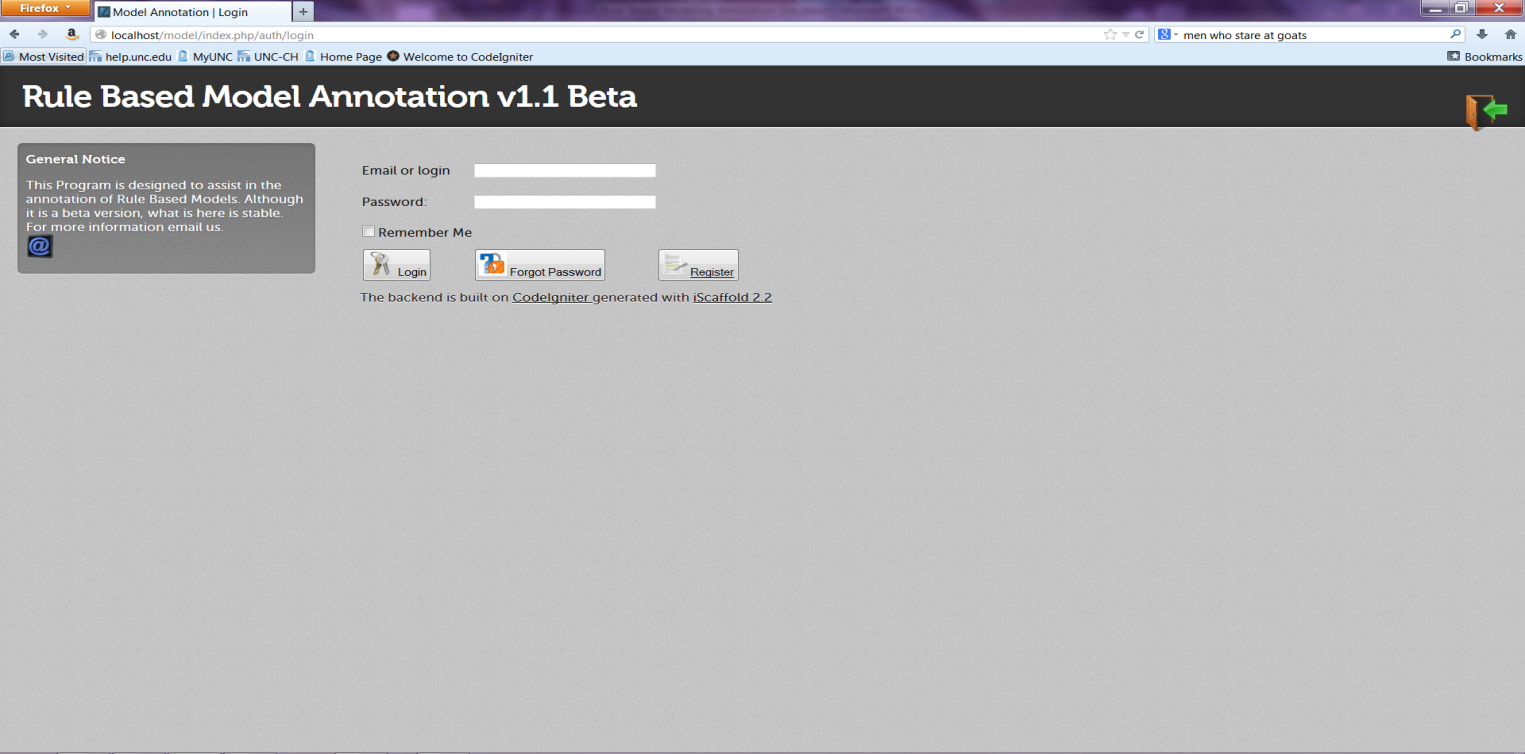
* Since Tank\_auth does not really follow the framework MVC paradigm and doesn’t use smarty templates, I have created a completely separate login page.
* Line 208 The contents of the “General Notice” box is here.
* Line 213 Enter your contact email.

/model/application/view/frame\_admin.tpl

* This file is the main display. All pages are displayed within this page. This file contains all links to the css files and everything to make the page look like what you see.

Now pointing your browser to <http://yourhost/model> should load the login page which should look something like figure 1.

Figure



If the DB connection is not set up correctly you will get an error message stating that a DB error occured. Confirm that the DB user ID and password is correct in the database.php file. Test the DB connection by logging in with the MySQL command line. Depending on the edits made to /model/application/config/tank\_auth.php lines 42 – 52 the “Register” button may or may not be visible and a CAPTCHA section may be seen. For the first run you need the “Register” button to be visible.

## Future Needs & Known Bugs

While there are some bugs in the site (listed below and in their respective sections), none cause the site to be unusable. More work is needed on the workflow within the site. A potential modification to the user activation that I have seen would be to require the System Administrator to approve sending the activation email. The user privileges could be set at the same time. This has the potential to simplify the registration process. Another change that would be useful would be the addition of user groups such as labs or teams. This would require editing all the controller files to check for group permissions. The user registration page should be altered to force the registrant to add a phone number, affiliation, and location. Consideration should be given to storing the reaction rate parameter (not the number itself) and definition in the DB. This would be most useful if the site was linked to a program the writes and executes the model such as SIMple\_Cell.

### Needed Features

#### Documentation

* Page specific help files need to be written and associated with respective pages. These should be downloadable or viewable in new tabs. In lieu of this type of help file some type of bubble or other popup specific to each section of a page needs to be written.
* PDF version of this file should be available for download by users and System Administrators. A stripped down version without setup, code blocks, and schemas should be available for Guests. My thinking is if you don’t qualify as a user you should not be able to gain knowledge of how the site is constructed.
* A document should be written describing my view on how to construct these models. This would not be so much about how the programs work but the thought process and methods to successfully construct these.

#### File Uploads, Directory Structure

* Need a method to upload images or new versions of existing images. This should include a method to create a new subdirectory, if needed, for new image. Images should be in directories unique to project. Creation should be automatic and based on Project ID. Alternatively all the images could stay in the image directory and be renamed upon upload to have the Project ID added as a prefix.
* Upload and download method for Java Script file that defines the display properties for the image. This will need to be integrated closely with image upload because the image name and path is required inside of this file. This will also require modification of frame\_admin.tpl so that the dropdown menus can be easily changed. Possibly have the diagram image group be a separate file that can be uploaded/downloaded.
* These types of file changes should be logged. This will require adding a file\_log table to the DB. This table is envisioned to be accessible to System Administrators only. Record entry would be fully automated.

#### General Work-Flow

* Still need some more thought into making working with rules simpler.
* As of May 29, 2014 CodeIgniter is no longer actively maintained by Ellis Labs, the owners. It has been reported that they are attempting to sell it. This will eventually become an issue as PHP is changed. Serious consideration needs to be given to refactoring the site for another framework such as Zend or Laravel.
* **IMPORTANT**. CodeIgniter Upload library file. Public function: validate\_upload\_path. A change is required in this line of code; if (function\_exists('realpath') AND @realpath($this->upload\_path) !== FALSE) needed to be changed to if (method\_exists('realpath',$this->upload\_path) AND @realpath($this->upload\_path) !== FALSE) due to changes in PHP. Change made by Dennis Simpson June 8, 2014.
* The image pages need sprucing up. The ability to start at some initial zoom level other than full zoomed out will help. Tested Corel Draw and found it will make drawing these images simpler. Start thinking redraw. As of Oct. 10, 2013 the new FWD zoomer plugin has been obtained. This version has changed enough that it will require a bit of time to make it work.
* **CRITICAL.** Need the ability to define the parameters and functions. I am thinking a separate table for each.
* Would like to have some code that will extract all the required components needed to recreate a BNGL file. This would require having the parameter values stored somewhere in the system. This is not strictly necessary since the user should be entering daily builds of the rule files.

### Bugs

* Deleting a project will result in orphan records in the DB. The intent when deleting records is to have the deletion cascade through the DB to prevent orphan records.
* Rule deletion has a foreign key failure only on server. May need terminal access to fix this. Have not been able to trace this on localhost.
* Display Bug when retrieving a list of publications associated with molecules. Only the first author is retrieved. This has something to do with the need to query the DB with an array selecting only unique records. If the requirement for unique records is removed it will return all authors. This causes no performance issues. To fix this thought should be given to changing the DB schema here. Reading out the authors from the XML file into a string that is then inserted into a column in the pubmed table might be a better way to go.
* **MAJOR.** Selecting “Show Molecules Associated with a Rule” does not gracefully handle the display when there is nothing to return. Should send user to the associate molecule with rule page with the rule preselected and a message saying there are no molecules currently associated.
* When adding external links to a molecule, the molecule should automatically select like the “add components” work flow.

# Using the Site

To begin using the site you must create a System Administrator as described below. To enter, update, or delete data, login into the site with as user who is a member of the Users or System Administrators group. Guest access allows viewing, searching, and external links only. Checking the “remember me” box will create an entry in the user\_autologin table specific for the IP address of the computer you are using, logging in from a different computer or IP address will require your ID and password. By default a user will be logged out after 2 hours of inactivity. The icons used in the site are shown and defined in table 1. If a page does not show a particular icon or menu option then that function is not available. Care should always be exercised when deleting records in this type of DB. A conformation box will pop-up whenever the delete icon is clicked. Selecting “Ok” or “Yes” will send the delete query to the DB. Deletes occur immediately and can NEVER be undone.

Table

| **Icon** | **Function** |
| --- | --- |
|  | Logout. Visible in upper right corner of every page. |
|  | Delete Record. Not available for Guests. |
|  | Edit Record. Not available for Guests. |
|  | Publications. Generally returns a list of PubMed references. |
|  | External link to PubMed record. Opens in new browser tab. |
|  | Show Rules. Found on ECM marker pins and List ECM Notes. |
|  | Components. Clicking will bring up a list of components. |
|  | Molecule. Clicking will bring up molecule. Gold star to those who recognize what molecule this is. Hint: the author of this site and document could not have designed and built it without this. |
|  | External links that are not publications. Opens in new browser tab. |
|  | Email webmaster/site designer. Found in footer of every page. |
|  | Download file icon. Clicking it will get you a prize. |
|  | Open file icon. Clicking on this will display the file in a new browser tab. Be aware that text files do not contain much formatting information and therefore are not so pretty to read on a browser. |

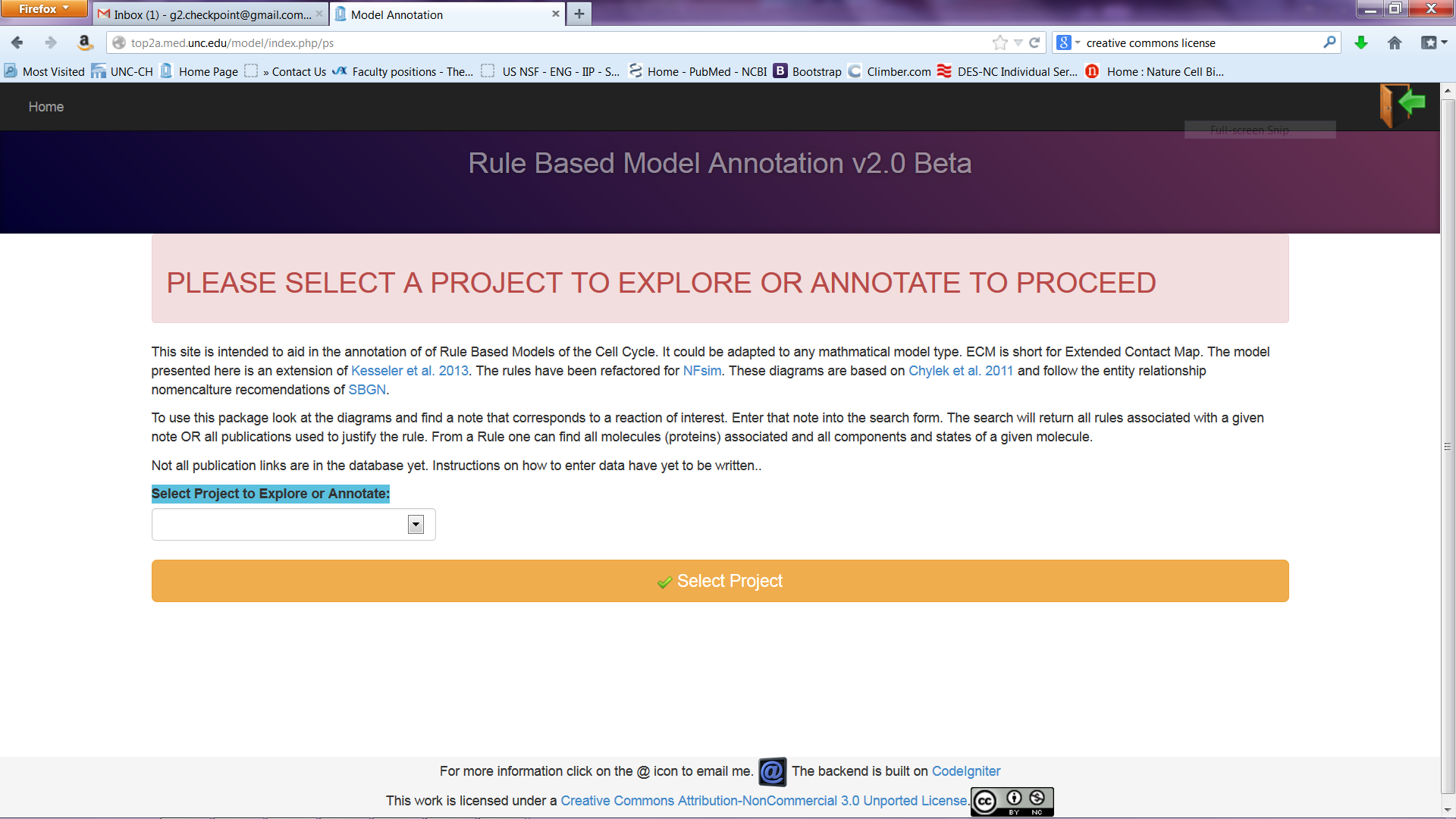
## Edit Users

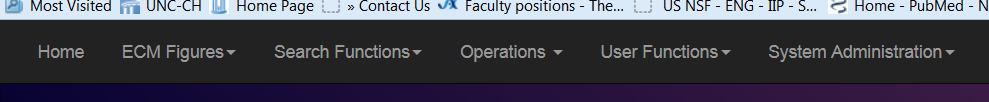
NEVER insert users from within the site or from MySQL. Doing so may result in new user not being registered with the site properly. Users should only be added through the “Registration” button. To create a new user click on the “Register” button on the login screen and fill in the fields listed on the new screen. This button will only be visible if “allow registration” is set to true. The Tank\_Auth validation requires unique user-id’s, and emails for registration. By default all new users are part of the “Guest” group with a group\_id of 300. By default an activation email will be sent to the registrant if the server allows smtp. The email will contain a link that will activate the registrant and log them in. Localhost implementations do not allow this. For localhost installations the only mechanism to activate a registrant is to edit the “users” table in MySQL changing the “activated” field from 0 to 1. This can be done from a MySQL command line or from phpMyAdmin. Never activate a non-localhost account directly through MySQL.

# Login

To begin to use the site set up a System Administrator by clicking on the “Register” button seen on the login page (Figure 1), and entering an ID, email and password in the spaces provided on the registration page. If it is a production server follow the instructions in the email that will be sent to initially activate the account. If it is a localhost installation, activate the account as described above. Using phpMyAdmin or use the MySQL command line to change the users.group\_id field from 300 to 100. Anyone with System Administration access can now do this from within the site. Now logging into the site with the System Administrators ID and password should bring up the page shown in figure 2. If this is a clean installation there will be no project to select. System Administrators can jump to the “System Administration” menu group without selecting a project. This allows a rapid method to create a new project or edit users. The site currently requires all users to select a project prior to entering. Once a project is selected the page will change and a dropdown menu bar will appear as seen in figure 3.

Figure





Figure

# ECM Figures

Clicking on this will open a drop-down menu that has links to the diagrams in the site. Hovering over any of the markers will open a fly-out that allows the user to click on either of the icons or on the “Note” hyperlink for more information. Clicking directly on a marker will magnify the image to 20%. The image can be repositioned by dragging.

There can be any number of figures in the site. All of these are accessible by clicking on “ECM Figures” and then clicking on the submenu that contains the figure of interest. These figures can be high resolution JPEG or SVG type images stored in the /model/ecm folder. Tiff files will also work but tend to be too large to be useful. To change the title of an image or add additional figures or new figures to the site first copy the image into the /model/ecm folder. This folder should require root level access. The code for the dropdown menu is found at or near line 97 in model/application/views/form\_admin.tpl containing the lines in Code Block 1.

**Code Block 1**  
<li class="dropdown">  
 <a href="#" class="dropdown-toggle" data-toggle="dropdown">ECM Figures<b class="caret"></b></a>  
 <ul class="dropdown-menu">  
 <li><a href='index.php/dashboard/ecm\_1'>**Main G2 Figure**</a></li>  
 <li><a href='index.php/dashboard/ecm\_2'>**Generic Transport**</a></li>  
 <li><a href='index.php/dashboard/ecm\_3'>**Glyph Definition**</a></li>  
 </ul>  
</li>

The bold text is what is seen in the dropdown menu. Edit this to suit your needs. The highlighted text calls the PHP function to load everything. Read the Zoomer documentation found at model/zoom/documentation/index.html to learn about how the zoomer functions and the available options. By default the site is set to display the image in a lightBox. The view file to set this up is ecm.tpl. When using a lightBox viewer this file has no real function but is required to be present. To load more than three images requires changes to four files. First, open model/application/view/image.tpl and duplicate the code block shown in Code Block 2.

**Code Block 2**   
{if $image == **"ecm\_1"**}  
 <script type="text/javascript">  
 $(document).ready(function() {  
 var zoomer = new FWDZoomer("lightBox", "zoom/load/**config1.xml**");  
 });  
 </script>  
 {/if}

The “ecm\_1” on the first line is the unique identifier for each image. The other file that is unique for each image is the config1.xml file. There will be more about the XML file later. The second file that needs editing is model/application/controller/dashboard.php. Open the file and find the section that contains the functions for the FWZoom maps as seen in Code Block 3. The function label, in this case “ecm\_1” and the ‘ecm\_1’ next to ‘image’ must correspond to the highlighted text shown in the form\_admin.tpl and in the image.tpl code above. Each image must have its own config.xml file that loads it.

**Code Block 3**function ecm\_1()  
 {  
 $this->template->assign( 'template', 'ecm' );  
 $this->template->assign( 'image', 'ecm\_1' );  
 $this->template->display( 'frame\_admin.tpl' );  
 }

The zoomer works by calling the config.xml file shown in image.tpl and parsing it with Java. The XML file has the name and path to the images (a low and high resolution image is required for each), and instructions on displaying the image and markers.

## Diagrams

There is no automated or even simple way to draw the entity diagrams needed for Rule Based Model annotations. Process diagrams are not sufficient for RB models. Diagrams should follow as closely as possible the guidelines of the [SBGN](http://www.sbgn.org/Main_Page). If you have access to a vector based graphics program such as Adobe Illustrator or Corel Draw you will have the best bet at drawing the diagrams. A reasonable open source alternative to these is [InkScape](http://inkscape.org/). The printed size of the image used on the site can easily exceed five or six pages or greater than 6,000 x 6,000 pixels. When drawing it do not try and shrink everything so tiny that you need to magnify the image by 3 or 4 times just to see things while drawing. As discussed above, images on the site can be TIF, JPG or SVG format. SVG has some advantages over JPG; just make sure your Apache mime types are configured to handle these.

Currently the coordinates for the markers must be determined manually and then typed into the config.xml file for the image. There is a trick to doing this. FWZoom uses the window size set in the config.xml file to determine the location of the markers when fully zoomed out. To find the coordinates for the markers load the image that is going to be used on the site into InkScape and resize the image and page to the size of the display window. It takes some tweaking to get the image set in InkScape in a way that the coordinates line up with FWZoom. I have not been able to make this work with Adobe Photoshop or ImageJ. Both should work if fiddled with a bit. At this point simply move the pointer to where you want a marker and note the coordinates then type them into the config.xml file. I recommend that every three or four markers you load the image to make sure the markers are where you want them.

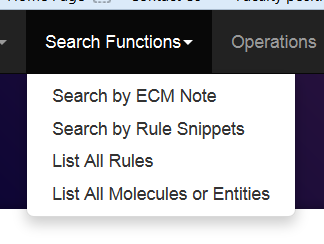
## Future ECM Figures

* A graphics program more dedicated to drawing entity diagrams would be helpful. At the time of writing this there was none available that were any good.
* A new Zoomer plugin is supposed to be available in October 2013. This will allow the initial zoom level to be set and will allow the markers to better link to sub maps.
* The figures are not explicitly associated with a given project. This should be changed.
* A figure upload feature should be built into the system. Each project will require its own directory under /uploads.

# Search Functions

All items in this list are available to all users. Clicking on the search function menu will open a drop down with the options seen in figure 4. Selecting “Search by ECM Note” will open a window that allows the user to select an ECM not from a drop down list and then show all associated publications or all associated Rules.

**Figure 4**



Selecting the “Search by Rule Snippets” option will open a window that allows a user to enter any part or the entire rule into the search box. Clicking the search button will return the rule. Very useful to find all the rules associated with one or more molecules.

The final two options are self-explanatory, listing all Rules or Molecules in the DB respectively.

# Daily Builds

This section is designed to allow the upload of rule files and to write notes about what was changed and why on that day. This menu group is not available to guests.

## List Daily Build Records

This is currently the only item available in this group. Clicking on it will open a page that displays a list of all builds for the working file. If there are no builds available you will get a notice to that. If there are builds available they can be edited or deleted by clicking on the appropriate icon on the right. If there is a rule file uploaded with the build notes icons that allow the files to be viewed in a new tab or downloaded will also be available. From this page the user can click on the “New Record” tab to open a page that allows notes to be entered and a file uploaded. The notes are required. There is no real limit to the amount of information that can be entered into the note field. The recommended way to use this is to have this open while you are working on the model, entering the notes as you go. If you need to leave the page just save the record. To come back to the page, find it in the list and click edit. When you are done for the day select and upload your rule file. The file must be text. The name does not matter but needs a .bngl suffix. When the file is uploaded it will be renamed using your user ID and a UNIX timestamp. If a file has already been uploaded it must be deleted prior to uploading a new one. This can be done from the edit page. The site only allows one file per record.

### Bugs and Future Considerations

* There are no bugs in this section. Everything works as intended.
* Files must end in .bngl, this is restrictive and needs to be altered. For security should not be wide open but maybe a list that will cover NFsim and KaSIM models.
* It is reasonable to assume that more than one upload file will be required. Modestly complex coding change.
* I think ideally a normal user should only be able to see their build files and notes in order to keep the display clean and reduce accidental deletes and edits. System Administrators should be able to see all by project. This would be a fairly simple code change.

# Operations

This section is where users can view, edit, or enter information about the model in the DB. The menu is divided into three sections; Molecule Operations, Rule Operations, Contact Map Operations. Guests have access to the data lists but not to the edit or delete functions. In the drop-down menus these are grouped and divided by main function.

## Molecule

This brings up a list of all molecules in the DB for the working project selected. From this page the user can use the icons (table 1) found on the right to show all the components or external links associated with the molecule. Non-Guest users can also edit, delete, or by clicking on the “New Record” tab, create a molecule. Keep in mind that deletion of a molecule will also delete the associated components.

Clicking on the New Record tab will take you to a data entry page shown in Figure 5. Molecules can be proteins, protein complexes, or something else. The majority of molecules found in rule based models are proteins and protein complexes. The recommended syntax for a molecule name is the HUGO nomenclature (http://www.genenames.org/). For example, cyclin B1 should be listed as CCNB1. This form is intended for entering one molecule at a time. Type the molecule name into the molecule entry box without any components or state variables (BNGL syntax or Kappa syntax) exactly as written in the rule based model file. Any comments about the molecule can be typed into the Molecule Comment box. It is recommended that some comment or note about the molecule be entered. The “Molecule” and “Associated Project” fields are required. As seen in Figure 5, the “Associated Project” field should be preselected. There is no real limit to the size of the molecules or molecule comment entries in the DB because the Molecule is stored as tinytext and the Molecule Comment is stored as text.

When editing a molecule only the “Molecule” and “Molecule Comment” fields are available. To edit a molecule, find the molecule you wish to edit in the list and click on the edit icon. A molecule can be deleted by clicking on the delete icon.

Figure 5



## Add External Links

Clicking on this menu item opens a window that allows the user to add external links to a molecule. There can be multiple external links for each molecule; because of this the link must be the full URL. There is a reminder for this in the entry field. Overall this page should be fairly self-explanatory.

# Add Components

Clicking on this menu item opens a page that allows components, component states, and component definitions to be entered for each molecule. The component and states must be entered exactly as found in the model. All these fields are required. Like the molecule names, component names and states have no practical length limitation in the DB. They will have practical limits in the model.

Components can only be entered one at a time. This means that a molecule that has, for example, 4 components will need to be selected four times, entering a new component and component states each time.

### Future Molecule Considerations

* When entering components there should be hints at least in the states box suggesting the format for the state.

# List and Enter Rules

Selecting this menu item will open a page that lists all rules. Each row shows a rule, its associated ECM Note, and any comments about the rule. All users can select the molecule icon on the right and see the molecules associated with the rule. Non-Guests can also edit or delete a rule. Clicking on the “New Record” tab (non-Guests only) will open a window that allows users to enter a new rule and link it to an ECM Note. Rules, including rate variables, are entered according to the syntax of the system used (BNGL, Kappa, etc.). In order for the rules to display better, it is a good idea to add a space after the “->”. Rules that are very long may also require a space elsewhere to display in a more appealing manner. This is not technically required. As currently envisioned each rule can only be associated with one ECM Note.

## Link Rules to Molecules

Selecting this menu option will open a window that contains a dropdown select for molecules and a dropdown select for rules. Use this to select the rule and molecule you want to associate with it. This step needs to be repeated for each molecule found in a rule.

### Future Rule Operations

* There should be a tag with the rules to say which language the rule was written for. This may not be strictly necessary since that information could be written into the project description.

# ECM Notes

These notes are intended as a way to group all the rules associated with a single function. These should also be entered BEFORE the rules are entered. Selecting this option from the dropdown menu will open a page that contains a list of all the ECM Notes for the working project as well as the comments associated with each note. On the right of the table all users of the site can click on either the publication icon or the rule icon to get a list of publications or rules associated with the note. Non-Guest users also have the option to delete or edit the note as well as clicking on the “New Record” tab to enter a new ECM Note. While the “Comment” section is not strictly required at this time, it is advisable to enter some text here to describe what the note is about.

## Link ECM Notes to Publications

This is intended as a way to add any publications that justify the both the reactions associated with the note and to provide justification for the underlying rules associated with a given note. This menu item is not available to Guests. Selecting this item from the dropdown menu opens a page that allows the user to select the PubMed document and the ECM Note. First you will need a file to import. At present only PubMed XML files can be imported into the DB. These XML documents can only contain a single PubMed record. To save a XML file from PubMed first do a search. When you have a single record displayed on your screen look to the upper right hand region of the screen. There should be a “send to” option there. Click on that and select file and XML. Note the location that the file saved on your computer. Once the file is on your computer, click on the “Browse” button and select the file. There will be a slight pause while the file is processed. After the file name appears on the screen, select the ECM Note from the drop list to associate the PubMed record with. Both a valid XML file and an ECM Note are required. The XML file that is uploaded is not saved on the server however the local copy is still on your computer.

### Future Contact Map Operations

* Thought should be given as to how to pass the XML file from PubMed directly into the DB.
* Other file types could be allowed with changes to the code.
* Explore the possibility of having the uploaded file deleted from localhost as well. I think this might be explicitly forbidden in PHP due to the potential security risk.

# User Functions

No “User Functions” are available to guest users.

## Edit User Profile

Selecting this option will bring up a page showing the profile of the user that is logged in. The display on this page is different depending on the privilege of the user. Regular users can only view and change their user name, First and Last names, Affiliation, and email. System Administrators can list all users, select, and edit any user. The System Administrators can ban a user, give the reason for banning them as well as activate or inactivate any user. The other ability a System Administrator has is to delete a user. Users that are banned should not be deleted since deleting them will allow them to create a new account.

## List and Edit Projects

Selecting this option from the menu will bring open a page showing all projects in the DB. Both users and System Administrators can create a new project by clicking on the “Enter New Project” tab, edit existing projects by clicking on the edit icon, and delete projects by clicking on the delete icon. Remember that deletions are permanent. Deletion of projects is intended to wipe out all rules, molecules, ECM Notes, everything associated with the project.

### User Function Issues

* Deleting projects results in orphaned fields in the DB.
* More thought needs to be given to the security of this section. Consider requiring System Administrators to be the only ones to delete projects.
* Possibly restrict different projects to different users. This will require implementation of another level of permissions for the Site as a whole.

# System Administration

This menu group is only available/visible to the System Administrators.

## List Users

Selection of this item will open a page that lists all users and information about the users such as last login, last IP, etc. The users listed can be edited by selecting the edit icon or deleted by selecting the delete icon.

## List User Groups

This feature is not yet implemented. The idea is to have the users grouped into labs, projects, or some other system.

## List and Edit Projects

This is currently redundant with the same function in the User Functions menu. Eventually there will need to be project functions that are only available to the System Administrators and those will be here.

# PHP Schema

The site uses a CodeIgniter Framework for the PHP. Frameworks all attempt to follow a “Model”, “Views”, “Controller” (MVC) layout. Some of the site specific config files that need to bed edited are discussed above. While I did attempt to name the files in a way that made some sense as to the function performed, it was not always possible. For example, files with the “molecule” in the name are all involved in displaying, or editing, etc. molecules. However the molecule controller uses the components model to find associated components. Short of writing a long essay on what each file does there is no way to fully describe them. All view files in the site use the Smarty Templating Engine to render. The best way to understand them is to read this and read the code in the files. The file names are case sensitive so do not change it.

| File | MVC Component | Function |
| --- | --- | --- |
| components.php | Controller | All components. |
| dashboard.php | Controller | Page seen upon login success. |
| daily\_build | Controller | Coordinated Daily Build record functions. |
| doi.php | Controller | Coordinates External Links. Name an unfortunate historical hold-over. |
| ecm.php | Controller | Setup display of images and directs link inputs from image markers. |
| ecmnote.php | Controller | Coordinates the ECM Notes. |
| molecule.php | Controller | Coordinates Molecule functions. |
| projects.php | Controller | Coordinates Projects functions. |
| ps.php | Controller | Coordinates initial Login functions. |
| pubmed.php | Controller | Coordinates getting PubMed XML into DB. |
| rulemol.php | Controller | Coordinates linking Rules to Molecules. |
| rules.php | Controller | Coordinates Rules |
| sysadmin.php | Controller | Coordinates unique System Admin Functions. |
| model\_components.php | Model | Functions primarily associated with components. |
| model\_daily\_build | Model | Functions associated with Daily Build Records |
| model\_doi.php | Model | Functions primarily associated with External Links. |
| model\_ecm.php | Model | Only ECM table metadata which is not used. |
| model\_ecmnote.php | Model | Functions primarily associated with the ECM Notes. |
| model\_molecule.php | Model | Functions primarily associated with the Molecules. |
| model\_projects.php | Model | Functions primarily associated with Projects. |
| model\_pubmed.php | Model | Functions primarily associated with manipulating PubMed XML files. |
| model\_rulemol.php | Model | Functions primarily associated with linking Rules to Molecules. |
| model\_rules.php | Model | Functions primarily associated with Rules. |
| model\_sysadmin.php | Model | Functions primarily associated with unique System Administration. |
| model\_utilities.php | Model | Contains all the common DB utilities such as delete function. |
| dashboard.tpl | Views | Generates the page seen after a successful login, the home page. |
| ecm.tpl | Views | Required to view images. Doesn’t actually do much. |
| ecm\_search\_results.tpl | Views | Generates page showing all rules associated with an ECM Note. |
| form\_components.tpl | Views | Generates a page to enter or edit Components associated with a Molecule |
| from\_daily\_build | Views | Shows the Create and Edit pages for Daily Builds |
| form\_doi.tpl | Views | Generates page to enter or edit External Links to a Molecule. |
| form\_ecmnote.tpl | Views | Generates page to enter or edit ECM Notes. |
| form\_molecule.tpl | Views | Generates page to enter or edit Molecules. |
| form\_projects.tpl | Views | Generates page to enter or edit Projects. |
| form\_rules.tpl | Views | Generates page to enter or edit Rules. |
| form\_users.tpl | Views | Generates a page to allow User Profiles to be edited or deleted. |
| frame\_admin.tpl | Views | This is the code that wraps around all the other pages. Generates header, footer, menus, etc. |
| image.tpl | Views | This is a helper file for displaying the images. Since it is all java scripts should be loaded differently. |
| list\_components.tpl | Views | Generates the pages that have lists of Components associated with a Molecule. |
| list\_daily\_builds | Views | Displays a list of Daily Build Records |
| list\_ecmnote.tpl | Views | Generates the page that shows all ECM Notes associated with the working Project. |
| list\_molecule.tpl | Views | Generates page that shows all Moleucles associated with the working Project. |
| list\_projects.tpl | Views | Generates a page showing all Projects in DB. |
| list\_components.tpl | Views | Shows a list of Components associated with a Molecule. |
| list\_rules.tpl | Views | Generates page that lists all Rules associated with working Project. |
| list\_sysadmin.tpl | Views | Generates page that lists all users of the site. |
| login.html | Views | This is our login page. Does not use the frame\_admin.tpl page. |
| rule\_find\_input.tpl | Views | Generates the page to search for Rules based on Rule snippets. |
| rules.tpl | Views | Generates page that displays a single rule for editing. |
| searchinput.tpl | Views | Generates page that allows the search for Rules or Publications associated with an ECM Note via manual selection of the ECM Note. |
| searchresults.tpl | Views | Generates a page to display all Molecules associated with a Rule. |
| select\_file.tpl | Views | Generates the page to select the PubMed XML file to upload and ECM Note to link it to. |
| show\_components.tpl | Views | Shows a single Component associated with a Molecule. Used to show successful insertion of Component in DB and after an edit of a Component. |
| show\_doi.tpl | Views | Generates a page that either displays all External Links associated with a Molecule or the newly inserted External Link. |
| show\_ecmnote.tpl | Views | Generates a page showing a single ECM Note. |
| show\_molecule.tpl | Views | Generates a page showing a single Molecule. |
| test.tpl | Views | Generates the page that displays PubMed records. Name should be changed. |

## PHP Schema Future Needs

* The Views file test.tpl needs a name change. This will require edits in some of the controller files as well.
* An additional effort should be made to streamline this code. Some of the files could be combined and it is possible that some functions are duplicates and could be moved into a single separate file as was done with model\_utilities.php.

# MySQL Schema

## Tables

This section contains information that should help in understanding the MySQL database schema employed. The Model database currently contains the 16 tables shown in table 1.

| Tables in DB Model | |
| --- | --- |
| Table Name | **Description** |
| ci\_sessions | Stores session information from CodeIgniter. Required for login status and function of some of the pages. |
| components | The Components (BNGL syntax) and the possible States of the Components that make up a Molecule are stored here. |
| daily\_builds | Notes about daily builds and links to uploaded rule file. |
| doi | Stores External Links for Molecules. Can store any external link. |
| ecmnote | Stores the Notes and comments from the ECM figures. |
| group\_key | Defines Guest, User, and System Administrator ID’s |
| login\_attempts | Part of authentication, keeps track of login attempts by user and IP. |
| molecule | Molecule ID’s and definitions are stored here. |
| projects | Project ID’s and definitions. |
| pubauth | Stores the authors for the PubMed documents. |
| pubmed | Stores the PubMed documents minus the authors. |
| rule\_mol | Allows Many-to-Many type link of Rules and Molecules. |
| rules | Stores the Rules. |
| user\_autologin | If a user chooses auto login option those variables to accomplish that are stored here. |
| user\_profiles | User Profile such as affiliation. |
| users | User ID and passwords are stored here as well as the activation status. Passwords are encrypted and salted. |

Let’s discuss these tables in a little more detail.

### ci\_sessions:

session\_id Encrypted ID that is also stored in a cookie defining current session.

ip\_address IP of computer you are logged in from.

user\_agent Information about what is browsing your site. Browser, if browser which one, robot, ect.

last\_activity Date and timestamp that is used in conjunction with the cookie to determine whether or not to keep the connection alive.

user\_data Stores information about project and login status that the site references.

### components:

idcomponents Primary key for the table. Auto incremented, unique integer.

component Actual component of a molecule. Written exactly as defined in rule file. Column is defined as varchar with a length of 45. Not allowed to be null.

definition What is the component? For example component B is defined as a generic binding site. Stored as text.

states Defines the possible states a component can exist in. For example component B can be Bound or Unbound. Stored as tinytext because all possible states are stored and defined in a single column.

idmolecule This is the foreign key storing the idmolecules primary key linking the components to the molecules. Stored as an integer with a length of 11.

projects\_id Stores the foreign key linking the projects to the components. Stored as an integer with a length of 11.

### daily\_build:

id Primary key for table. Auto incremented, unique integer.

notes Notes about build. Text type.

user\_id Foreign key linking to users.

project\_id Stores the foreign key linking the projects to the components. Stored as an integer with a length of 11.

created Date and time record was entered.

updated Date and time record was last edited.

file\_link Name of uploaded file.

### doi:

iddoi Primary key for table. Auto incremented, unique integer.

doi Contains the full URL to the external link about a molecule. Stored as a varchar with a length of 100.

doi\_idecm This is the foreign key linking to the ecmnote table. This does not make any sense to have. I need to check this.

doi\_idmolecule

Foreign key linking to molecules. Stored as an integer.

projects\_id This appears to be a hold-over from earlier work.

### ecmnote:

idecm Primary key for the table. Auto incremented, unique integer.

ecmnote Stores the note as found in the ECM figure. Stored as a varchar(45).

comment Stores information user wishes to write about a given note. Stored as text.

project\_id Foreign key linking to project. Stored at an int(11).

### group\_key:

group\_key Primary key for the table. Auto incremented, unique integer.

group\_type Guest, User, System Administrator. Stored at a varchar(30).

### login\_attempts:

id Primary key for the table. Auto incremented, unique integer.

ip\_address IP address user is attempting to login from. Stored as a varchar(40).

login Stores login count. Stored as a varchar(50).

time Time login was attempted. Used if user is locked out.

### molecule:

idmolecule Primary key for the table. Auto incremented, unique integer.

molecule Stores the molecule exactly as written in the rule. Stored as tinytext.

comment Allows user to write a description about the molecule. Stored as text.

projects\_id Stores project link. Not defined as foreign key. Stored as integer.

### projects:

id Primary key for the table. Auto incremented, unique integer.

name Project name. Stored as varchar(100).

description Think of this as an abstract describing the project. Stored as text.

created Automatic field shows when project was created. Timestamp.

updated Automatic field showing when project record was modified. Timestamp.

### pubauth:

id Primary key for the table. Auto incremented, unique integer.

LastName Author last name. Stored as varchar(45).

ForeName Author first name. Stored as varchar(45).

Initials Author initials. Stored as varchar(5).

pubmed\_id Foreign key linking this table to pubmed table.

### pubmed:

id Primary key for the table. Auto incremented, unique integer.

Title Title of journal. Stored as varchar(50).

PubDate Publication date. Stored as year.

Volume Publication volume. Stored as an INT(6).

Issue Publication issue. Stored as an INT(6).

idecm Foreign key linking to the ECM note. Stored as INT(11).

PMID PubMed ID. Stored as INT(11).

ELocationID This is the Digital Identifier that PubMed usually includes. Stored as varchar(45).

AbstractText PubMed abstract. Stored as text.

ArticleTitle Title of article. Stored as varchar(175).

MedlinePgn Page numbers. Stored as varchar(20).

### rulemol:

idrulemol Primary key for the table. Auto incremented, unique integer.

rulemol\_idrules

Foreign key linking to rule. Stored as INT(11).

rulemol\_idmolecule

Foreign key linking to molecule. Stored as INT(11).

### rules:

idrules Primary key for the table. Auto incremented, unique integer.

rule This is our rule. Stored as text(100). Might need to consider removing the size limit.

rulenote Allows users to write a nice story about the rule. Stored as text.

rules\_idecm Foreign key linking rule to ecm note. Stored as INT(11).

projects\_id Foreign key linking rule to projects. Stored as INT(11).

updated Automatic field that shows when the record was last updated. Timestamp.

### user\_autologin:

key\_id Primary key for the table. From the ci\_sessions table. Stored as Char(32).

user\_id Link to user table. Stored as INT(11).

user\_agent From the ci\_sessions table. Stored as varchar(150).

last\_ip Last IP address user logged in from. Stored as varchar(40).

last\_login Stores the date and time of last login. Stored as timestamp.

### user\_profiles:

id Primary key for the table. Auto incremented, unique integer.

user\_id Foreign key linking to users table. Stored as INT(11).

country Not currently used.

website URL associated with user. Stored as varchar(225).

updated Timestamp of when record was changed.

surname Users first name. Stored as varchar(25).

givenname Users last name. Stored as varchar(20).

organization Affiliation of user.

### users:

id Primary key for the table. Auto incremented, unique integer.

username Login name. Stored as varchar(50).

password User password. Encrypted and salted. Stored as a varchar(255).

email User email. Stored as varchar(100).

activated User activation flag. Stored as tinyint(1).

banned User banned flag. Supersedes activation flag. Stored as tinyint(1).

ban\_reason Short description of why user was banned. Stored as text.

new\_password\_key

When user requests a password reset the key that is emailed to them is stored here. Stored as a varchar(50).

new\_password\_requested

This is a datetime column that stores the date and time a new password was requested.

new\_email If the email is changed, I think.

new\_email\_key

Part of the authentication. Key is associated with email that is sent out to validate user.

last\_ip Last IP address user logged in from.

last\_login Date and time of last successful login.

created Date and time user was created.

modified Date and time user credentials were modified. Really useless since last\_login modifies record each time.

group\_id Foreign key defining authority of user. By default all new users are assigned “300” which is the Guest ID.

A visual showing the relationship of the tables is on the next page. This is generally out of date.

