PetroBase


CakePHP


php


APACHE


jQuery
write less, do more.


Powered by
MySQL


WESTERN
WASHINGTON UNIVERSITY

Concerns For Implementation Team

- Talk to David Hirsh about the
    - User Interface - there are some missing features that are not shown in the User Interface.
    - Ask him about a current search parameters

- Color Issues

    There are a lot of problems wrt. color issues. some of these should be hashed out with Hirsch, however it is probable that color searching should be removed.

    The first color scoring method in the Scoring.pdf is computationally difficult, and hard to grasp and understand, It would be very difficult to implement, and many colors would be Null zones that do not represent any minerals.

    The second scoring method in the Scoring.pdf is simplified, however not very accurate. It requires a lot of premature computation, and will still not be easy to program.

    Even if there was a good means to compute color scoring the environment which you observe color can dramatically impact observed results. If you view a given color under an sodium, incandescent, or florescent bulb it will appear differently to the human eye.

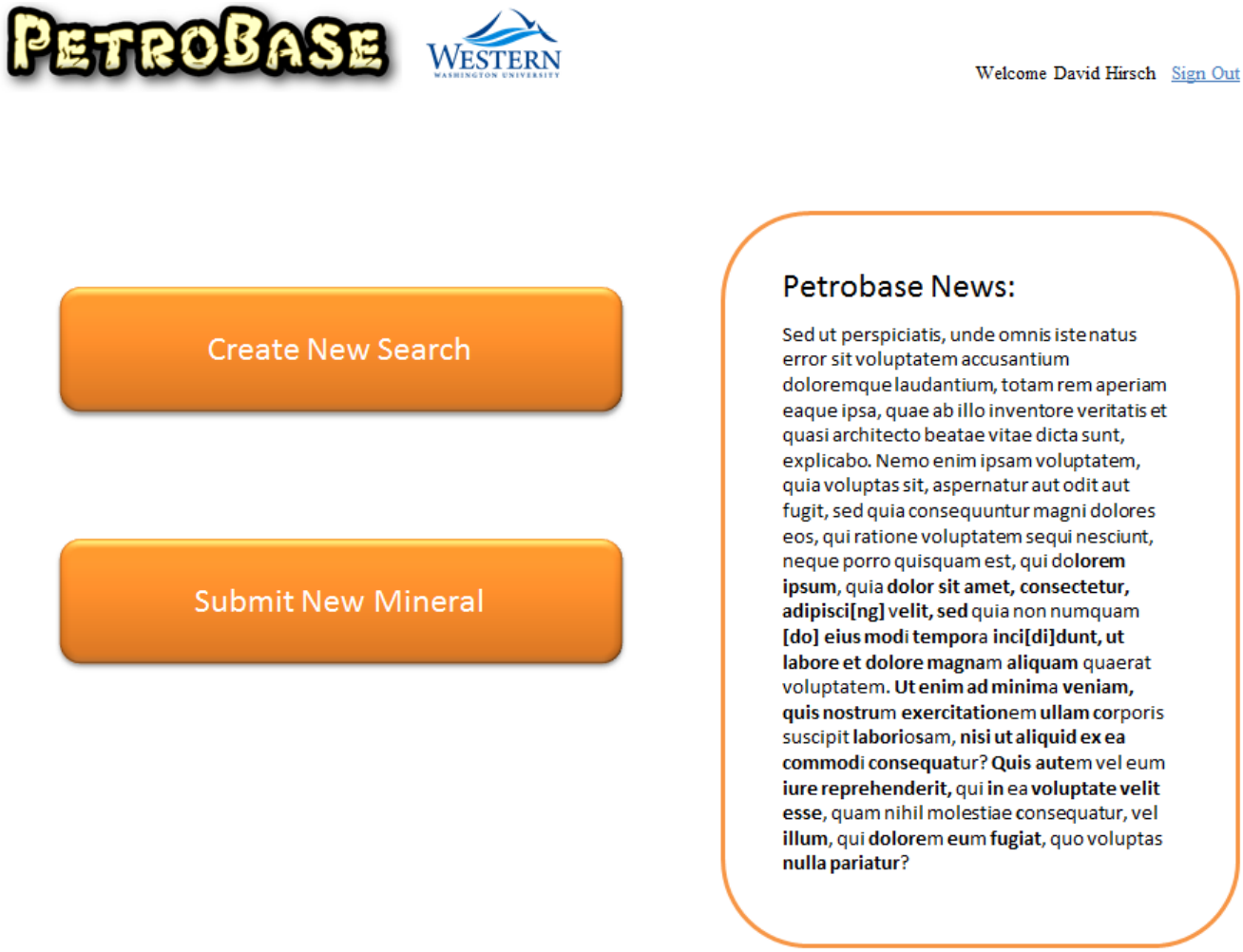
    Additionally, colors are represented differently on different brands and types of monitors.

    These issues should be known to the implementation team, not all of these have been discussed with Hirsch. These are legitimate concerns which if not addressed during implementation could result in a lot of wasted time.

- We attempted to standardize the language used over all documents with intentions of clarity.
- The scoring.pdf document expresses the preferred methods for addressing scoring and justifications for rejected methods. This is to better inform the implementation team so they can make more informed changes.

User Interface

- Welcome Page



Petrobase News:

A simple description of what petroBase is and what is going with related materials. The should be an echo of a admin message. Above the news, if the user is not logged on there will be the word log on where sign out is on this example.

- Navigation

Navigation tabs are of each of the categories in the database except for occurrence which is taken care of in the results page. On mouse-over a transparent black box will show a small description of the attribute. Also the tab bar will show which page is selected.

- Dynamic variables

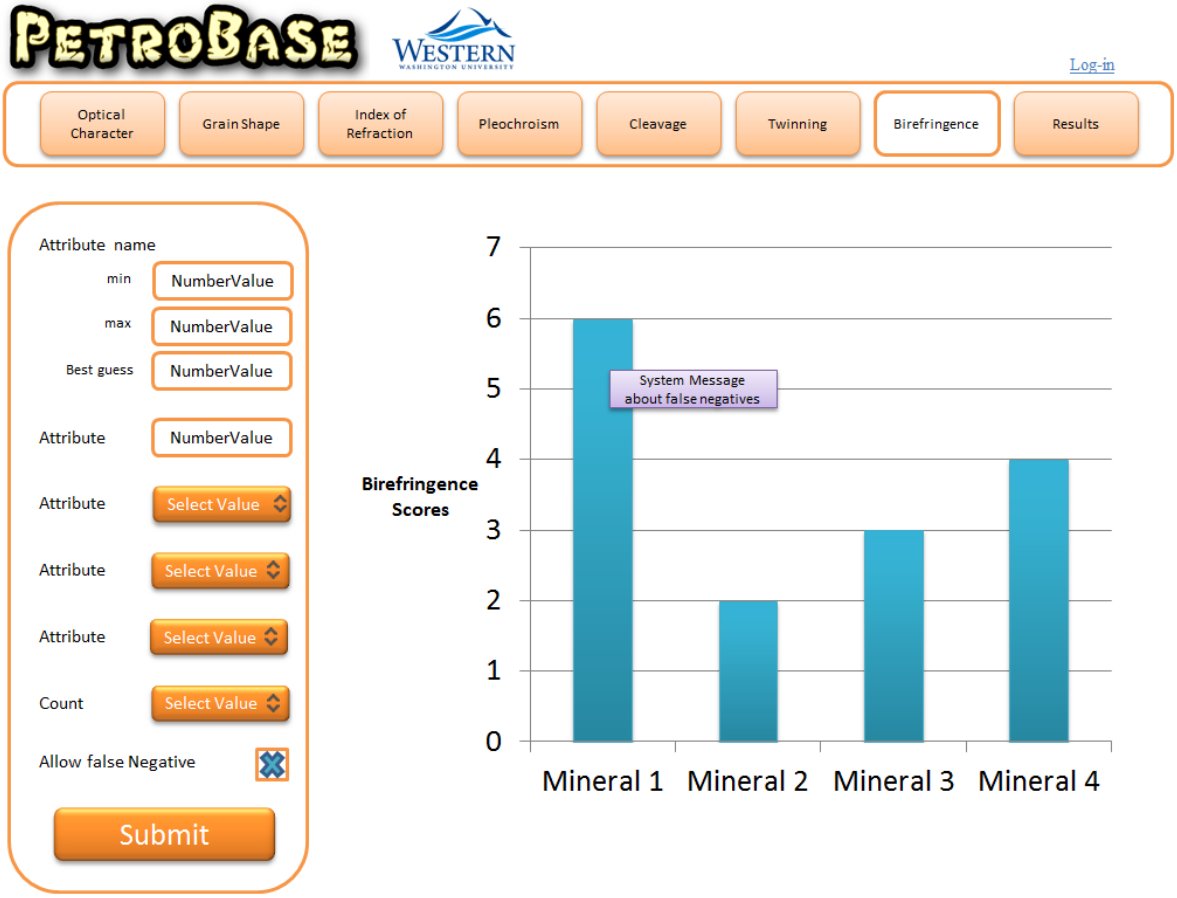
There are 2 variable that are present on all categories and the results page.

1) Count - How many of the top minerals in order of that certain category that is displayed in the graph.

2) Allow False Negative - A mineral that falls into an attribute type that is not selected by the user but based on input parameters are possible mineral matches.

A user should be able to dynamically select whether false negatives are displays.
For more detail on False Negatives look at Flow Chart in scoring.pdf.

- Category User Interface



Attribute Section

Each category has a set of attributes. Depending on the attribute the user can do three thing:

1) Choose from a selection of options which is done using a drop-down list.
2) The user puts in a min and max of the attribute and there best guess.
3) Enter in just a best guess answer

1) If the attribute is an enumerated type(refer to scoring.pdf on what enumerating types are)then the user has the ability to select one of the option.

2) The min and max are not a min and max of a given mineral they are what the user believes what the min and max are and then they place their best guess in between the min and max.

3) The best guess answer is used for when the user only need to put a single answer in for a given attribute. For example in the Cleavage category enter how

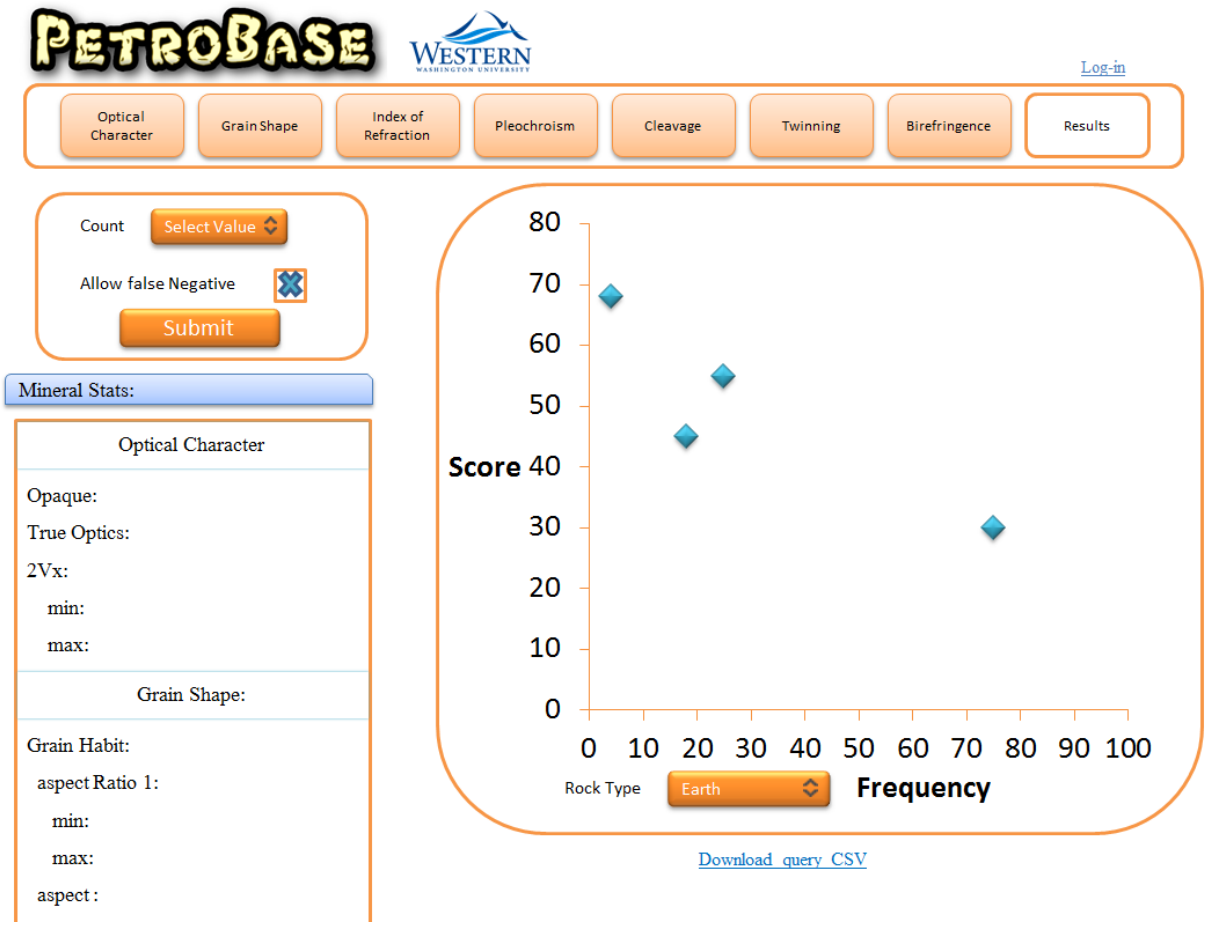
many cleavage planes there are.

Graph

The graph should be done with jQuery (http://jquery.com/) and the addon jqPlot (http://www.jqplot.com/). The graph will need to use the Count and Allow False Negatives (described up above in dynamic variables) to create the graph. Plot the top minerals of the category you are currently in while only showing how many minerals the user selected for Count. The same goes with Allow False Negatives except you show the top minerals which will include the False Negative minerals. For Categorical graph is represented as a bar graph.

When you mouse over one of the bars it will show if the mineral is a false negative if you have Allow False Negatives is checked.

- Results User Interface



Mineral Statistics

Mineral Stats. will be a div were will grow with any overflow with the Categories there will not be and scroll bar. It will contain all of the minerals(that the user selects from the graph) information which comes from the database. The attributes will be displayed in there proper categories. The Categories names can be click to expand or collapse so if the user doesn't care about seeing a categories attribute.

Rock Type

Is a drop-down list that contains the different occurrences minerals are found on the earth with the default being abundance in the database. This is a Pruning Attribute (refer to scoring.pdf for more information on Pruning Attributes) which can dramatically change the outcome of what is being shown on the graph.

Graph

The graph should be done with jQuery (http://jquery.com/) and the addon jqPlot (http://www.jqplot.com/). The graph will need to use the Count and Allow False Negatives (described up above in dynamic variables) to create the graph. Plot the top minerals of the category you are currently in while only showing how many minerals the user selected for Count. The same goes with Allow False Negatives except you show the top minerals which will include the False Negative minerals.

The result page graph is represented as a scatter plot. Where the top minerals from the scoring algorithm are shown with respect to the Count variable. Also when you mouse over a mineral in the scatter plot you will see what mineral it and if you click that mineral then Mineral Stats. will display the mineral information.

Download query CVS

A list of results in a CVS format.

Admin User Interface

- PetroBase Log-in

A simple log-on. Use it.

If the user want to create an account the will click on Not a user Click Here to create the account. All they need to do is register an email address and a password. Once they fill out the registration form a verification email will be sent to their email address. Once the verify the account the user will then have editor rights.

- ChangeSet Submission

If a User wants to create and submit a changeset. The attributes will be the attributes for the given category that is selected in the navigation tab. The mineral name and photo are static throughout all the navigation tab. The user does not have to have all fields in each category filled in. Upon a submission the changeset is packaged up with cakePHP and stored to the database for an admin to review later.

- Navigation

1. **Home**
   Talked about later.

2. **View History**
   Shows all changesets that have been made to the database and has the ability to revert changes back to a previous version in the database.

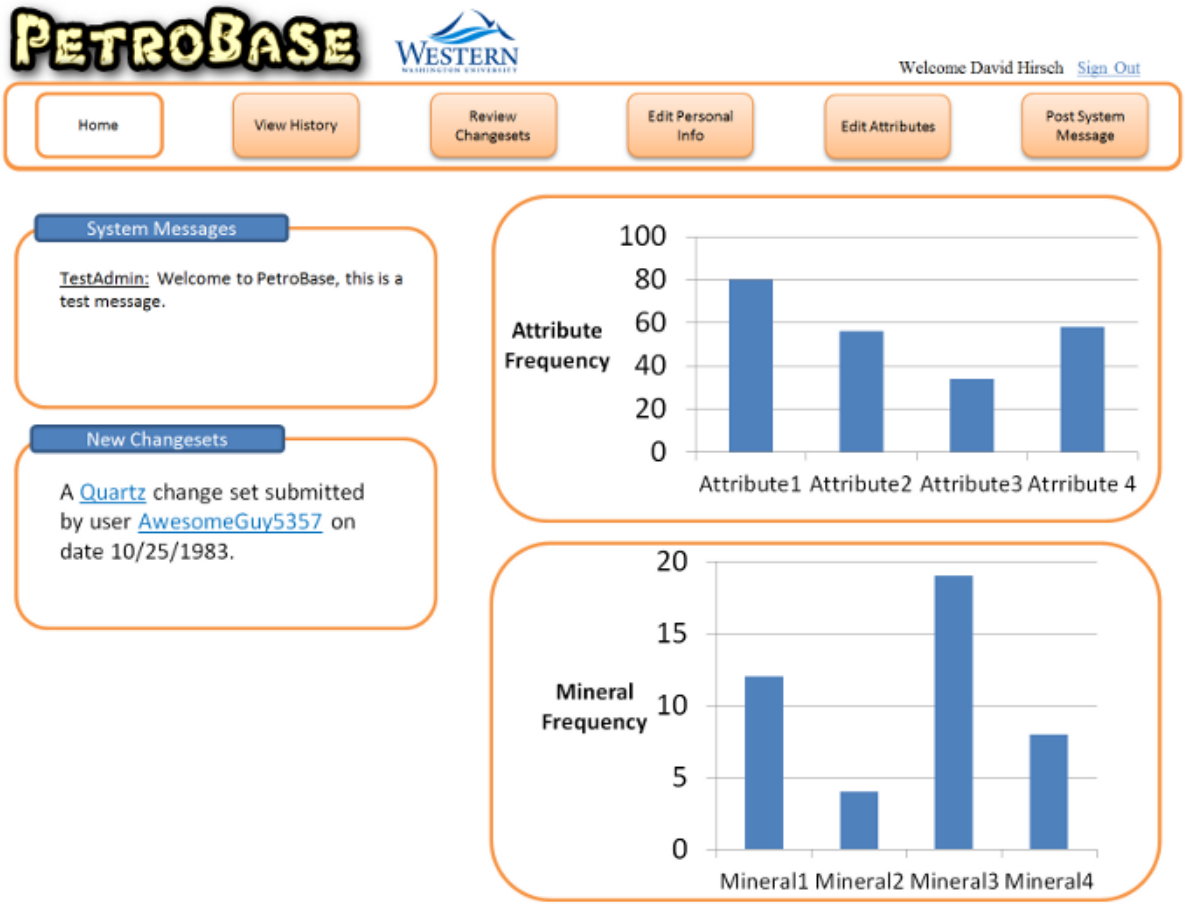
3. **Review Changesets**
   Explained later in Review ChangeSets

4. **Edit Personal Info**
   Figure this one out for yourself.

5. **Edit Attributes**
   Only Admin can use this feature. It allows the admin to change an attributes characteristic number.

6. **Post System Message**
   The admin can send out messages to other admins or to change the Petrobase News (Explained earlier).

- Admin Home page



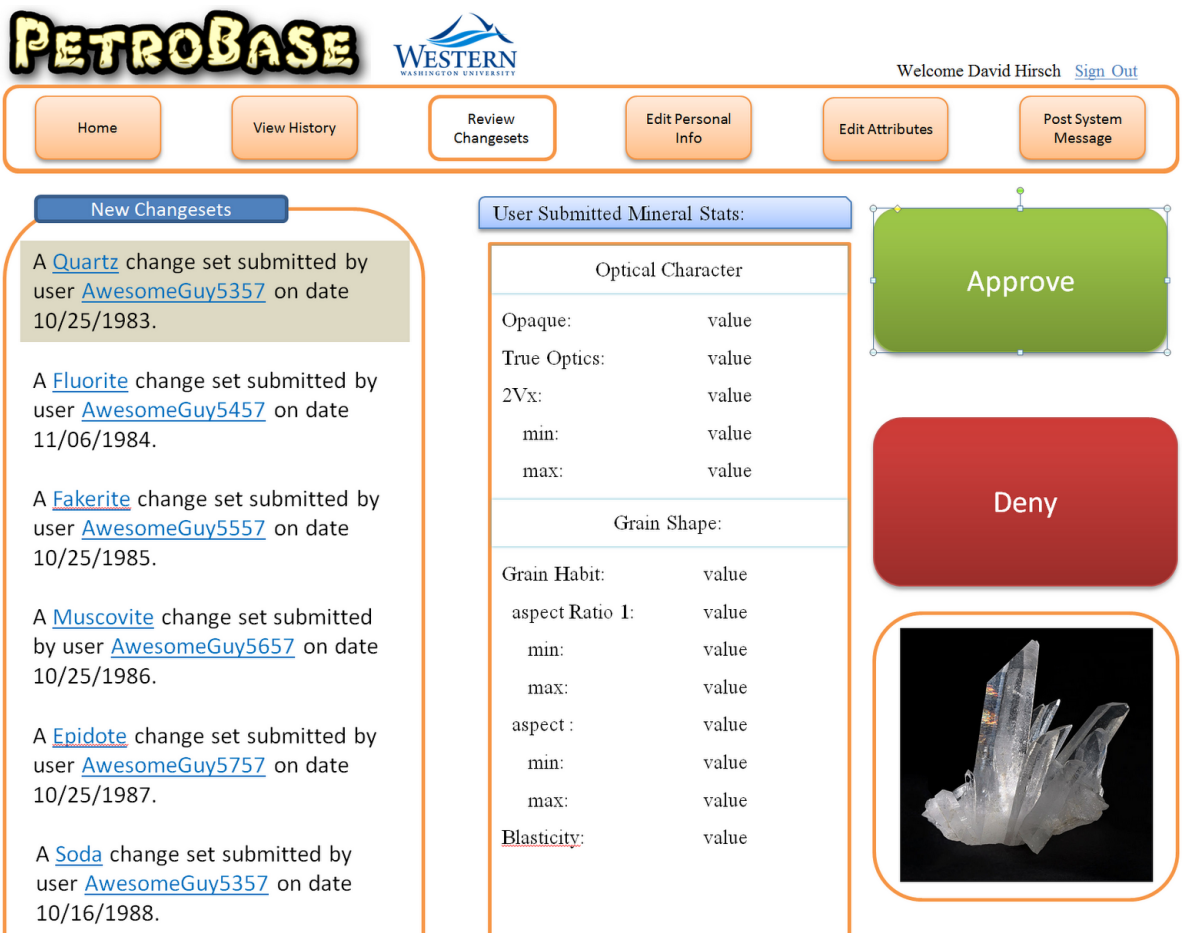
1. System Messages
   Any new messages that other admins have left for either just that one admin or all admins

2. New Changesets
   List of a new changesets that are needing to be reviewed and if you can click on the mineral name it will load that changesets into the review changesets window.

3. Graph
   i. Attribute
      A graph of the top attributes that are used in the search algorithm.

   ii. Mineral
      The top minerals that returned from the search algorith

- Review ChangeSets



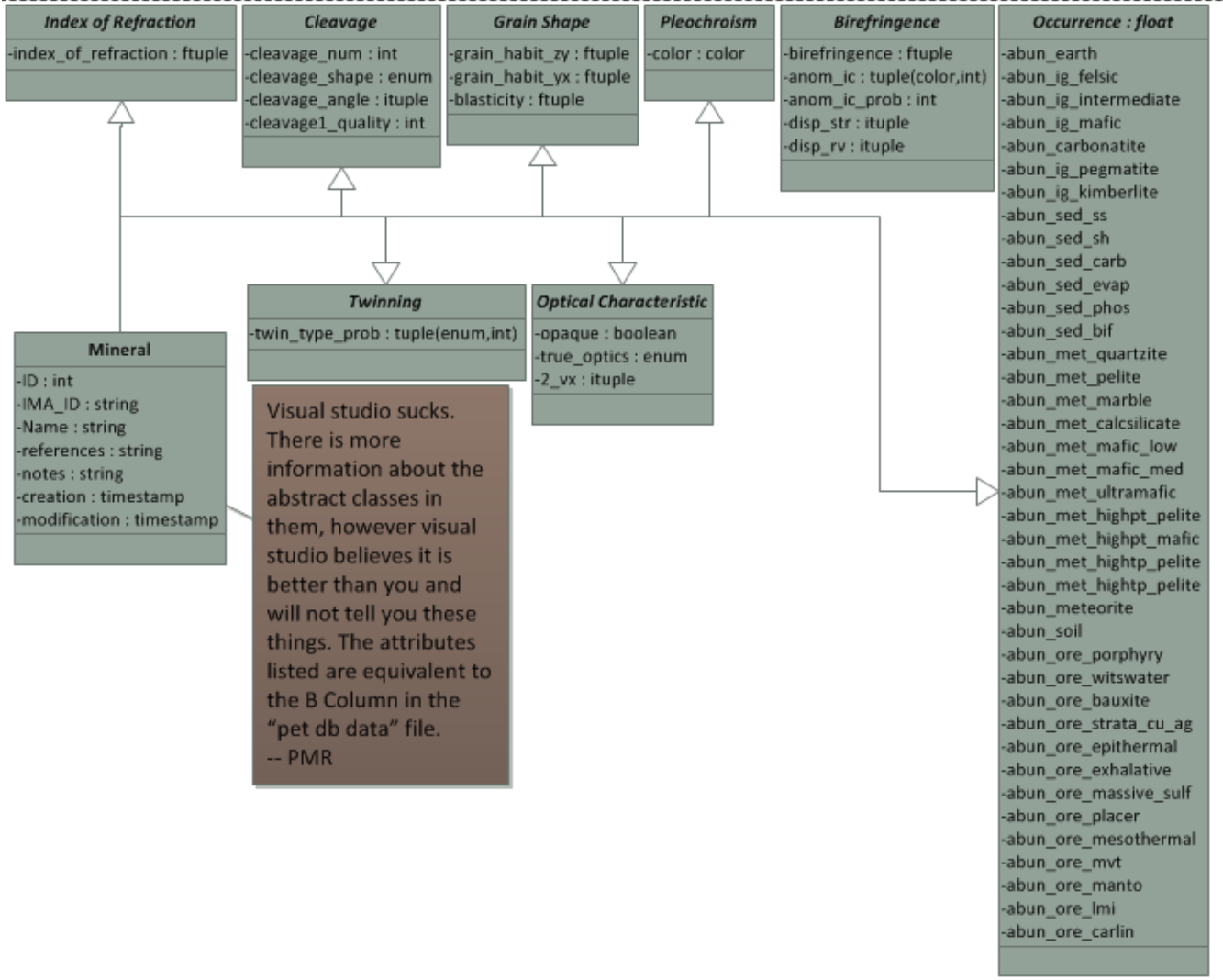
New Changesets and User Submitted Mineral Stats both dynamically grow and do not use scroll bars.

1. New ChangeSets is a list of all outstanding and upon clicking on one of the changesets it will highlight the selected mineral under review. In addition, it will put up the mineral stats that the user has submitted.
2. User Submitted Mineral Stats is a static column that contains all of the proposed mineral stats. As with the Mineral Stats on the Results page the categories can expand or collapse once the name of the category is click on.

Upon an approval or deny the changesets is removed from the list of New Changesets.

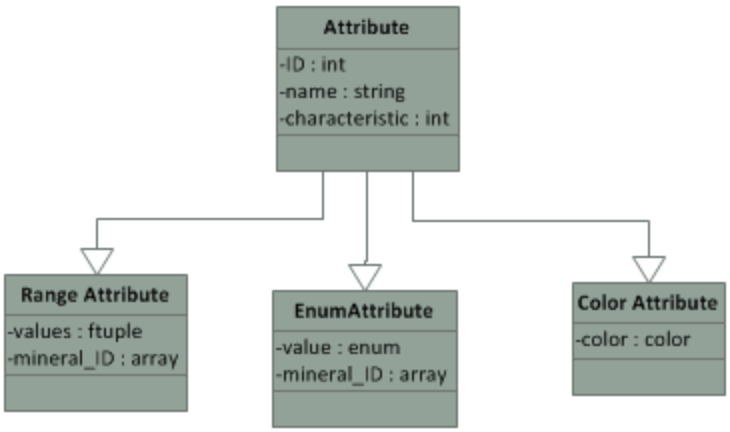
Class Architecture

- Class Diagrams



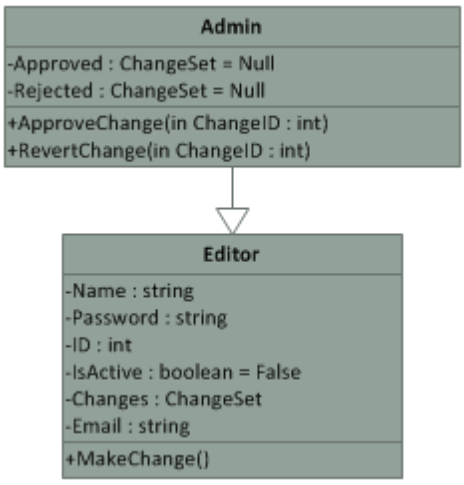
This is the class diagram for the mineral. All of the inherited classes are abstract. The use of abstract classes is to separate the mineral attributes into categories. Since the spec for this project states that we should use PHP. The tuple class needs to be created.There is no tuple in PHP. This tuple class should implement the ArrayAccess interface. Please refer to the petroDB appendix for more information about the individual attribute types. Also although not displayed every mineral might have a picture associate with it. Pictures should be stored as flat files and there path will be stored in the mineral object. The mineral should also have a changeset id to link the mineral to it’s changeset.

- Attribute Class Diagram



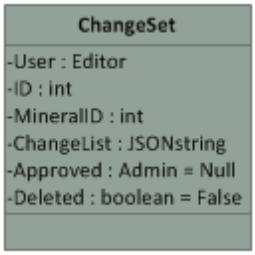
The main attribute class is abstract because all attributes are either a Range, Enum, or Color. Once again for more detail on the attributes is in the appendix.

- Admin Class Diagrams



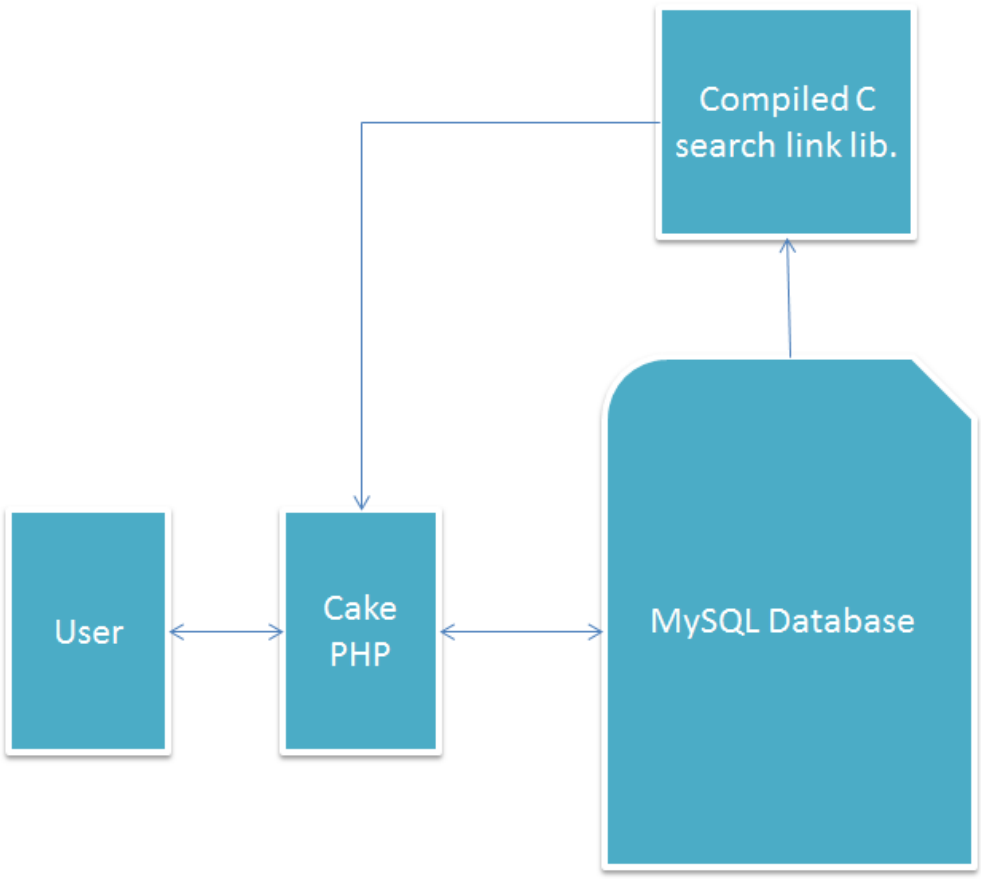
There are two user classes, the editor and the admin. The editor has simple privileges that anyone can get. All they need to do is register an email address and a password. Once they fill out the registration form a verification email will be sent to their email address. Once the verify the account the user will then have editor rights. Editors may submit new changesets to be reviewed. The idea behind the editor account is that a user may make real world observations on a mineral and then submit those observation to be reviewed. If those observations are accepted by an administrator then the change set will be added to the set of search-able minerals. An administrator has access to the admin section of the website. The admin can review submitted changesets, post system messages, change mineral attributes, revert old changesets, and edit their personal information(name, phone number, email, etc...).

- ChangeSet

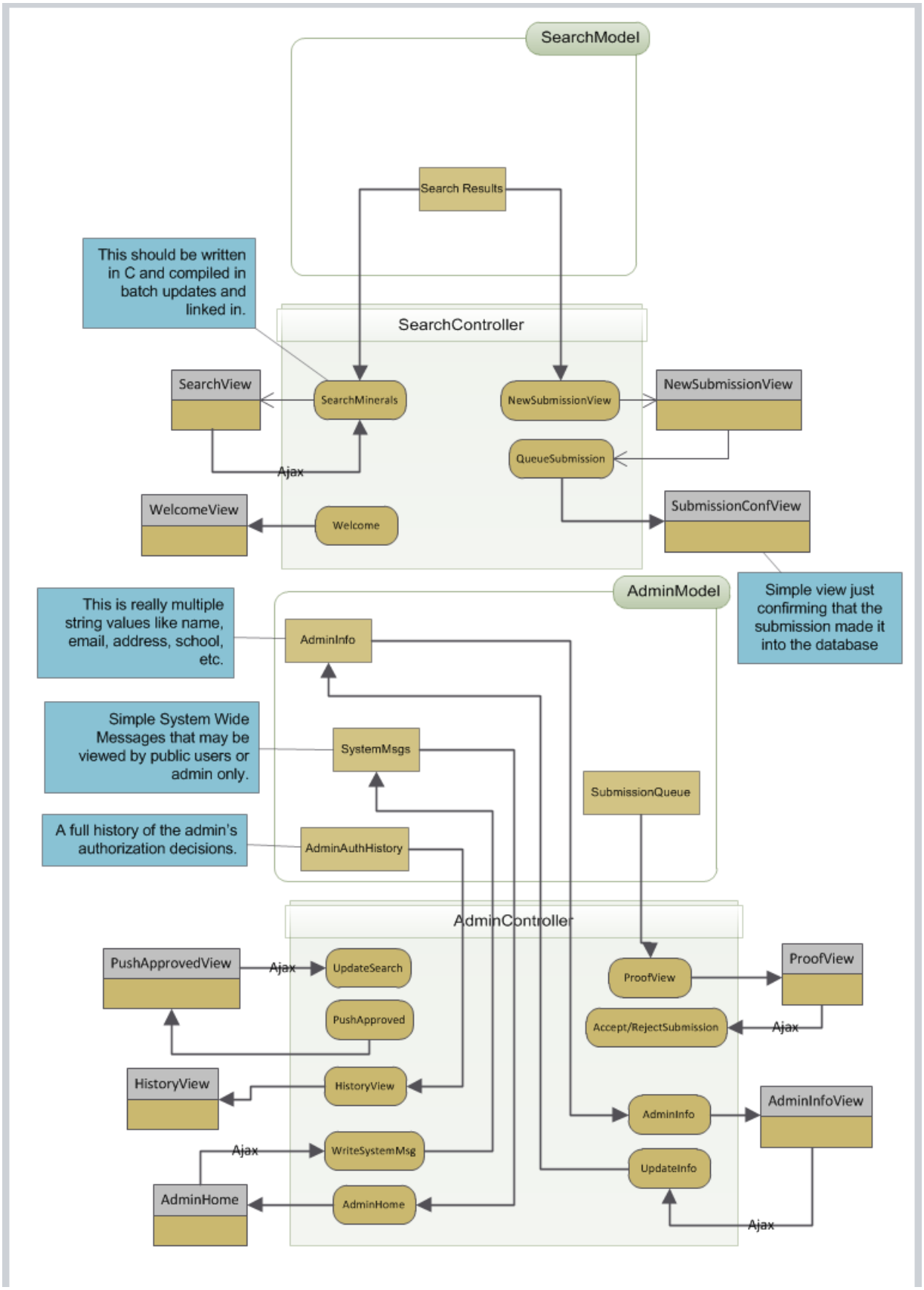


The changeset is a linking class. It links the id numbers of minerals with who submitted the mineral and if the changeset has been approved. Although as a policy nothing should ever been deleted.

Website Architecture



The website will be handled by cakePHP. Cake will also handle interaction between classes and the database. Cake has ability to package up classes and save them in the database. You should read more about cake here. The search algorithm will be written in C, compiled, and linked into the search controller. The C file will contain a static dump of the mineral database. This will decrease search time because no connection will have to be made to the database. Also since the dump will be static, new data structures will not have to be built at run time.


SearchModel
Search Results
This should be written in C and compiled in batch updates and linked in.
SearchController
SearchView
SearchMinerals
NewSubmissionView
NewSubmissionView
QueueSubmission
Ajax
WelcomeView
Welcome
SubmissionConfView
Simple view just confirming that the submission made it into the database
AdminModel
This is really multiple string values like name, email, address, school, etc.
AdminInfo
Simple System Wide Messages that may be viewed by public users or admin only.
SystemMsgs
SubmissionQueue
A full history of the admin's authorization decisions.
AdminAuthHistory
AdminController
PushApprovedView
Ajax
UpdateSearch
ProofView
ProofView
PushApproved
Accept/RejectSubmission
Ajax
HistoryView
HistoryView
AdminInfo
AdminInfoView
Ajax
WriteSystemMsg
UpdateInfo
AdminHome
AdminHome
Ajax

The previous image is of the model view controller and not all views are shown. The only views shown are not the simple views, that have non-trivial controllers.