Daily changes log:

* local\_login\_create\_user()
  + Changed some error message text to make them more clear
  + Added an error message if the validationkey field is empty
  + Almost added a check for an empty password but decided to leave it if you want to use an empty password to make testing easy.
* get\_user\_settings()
  + Empty list throws an error. Checking to see if an item was found fixed it.
* db\_set\_global\_settings()
  + Fixed some errors relating to nulls.
* get\_limited\_search()
  + Bug in GroupID. No longer vulnerable to sql injection as well.
* update\_course\_settings()
  + Did not return properly on success.

These are some changes I would like to make but did not. Most of these took longer to explain than they will to implement and are totally safe. But, I said the db was locked, so no changes without permission.

* 15 minutes safe to add and test. **add\_setting\_to\_course(),** add\_**user\_setting\_to\_user()** . These would be different from current settings and exactly like the settings that are currently in usergroups. There are times when this might be more useful than the existing permission based primary setting structure I created for you. The course settings would be visible to anyone in the course (shown only in get\_my\_..\_course). The user setting would probably only be visibile to the user. In order to ensure this change affects nothing else, I don’t add it to the create course and create user functions. Instead add two separate procedures listed above to set the setting after the course and user are created. It is null by default and set null to delete. Lastly, add a key for the setting to the json return object in some of the functions that get a course and get a user. This should be a safe and easy change. The reason for this change is that it will make creating things like user favorites or course announcements much simpler and more predictable. I think it is also more intuitive for you.
* 2 minutes to add and test. **local\_login\_change\_password()** – this will allow a user to change their password. Their current username and password will be required. This system is only intended for testing, so no admins able to reset passwords or anything like that. This function requires no changes from other functions nor the data from any tables other than LocalLogin. Extremely safe.
* 15 minutes to add and test. **update\_user\_information()** – this function would only affect the user table and only the fields: EmailAddr, FirstName,LastName. These fields serve no function in the database other that handing to the UI, so no changes anywhere else. This should be safe.