

Registration:

User registers their “main” account details. This remains constant and is always retained in the database.

Registration generates internal “account number” also to be used for image naming on upload.

As I am moving to a user account based system, I will need a “user name” field with password/pass verification field, and an email verification field.

Once user registers, they have access immediately, but they are also sent an email confirmation that they signed up and now have an account.

The registration form includes the following fields and components:

- Name ***: Two input fields for First and Last names.
- Studio Name**: A single input field, highlighted with a dashed blue border.
- Website**: A single input field.
- Address ***: A series of input fields for Street Address, Address Line 2, City, State / Province / Region (with a dropdown menu currently showing 'United States'), and Postal / Zip Code.
- Phone (US)**: Three input fields for area code, prefix, and number, with labels (###), ###, and #### below them.
- Phone (INT'L)**: A single input field.
- Email ***: A single input field.

user name, self explanatory

purchase deadline and upload deadline should be continuously counting down to deadline, hours/minutes/seconds

editable contact info - self explanatory

credits purchase - there are varying bonuses at 10, 20 and 30 credits that need to be automatically credited to the user's acct (but not charged) when they order in those quantities (2 bonus for 10-19, 5 bonus for 20-29 and 10 bonus for 30+) - full integration with google checkout api a must (<http://code.google.com/apis/checkout/developer/index.html>) - user acct is credited automatically if payment successful. if not, goes into pending (credits not available) until payment is successful - pending credits are automatically dropped after an admin-specified time period.

image upload - user has as many image upload fields as they do confirmed (not pending) paid credits. script checks against image dimension requirements and alerts to change before upload is allowed to begin. also, categories must be chosen for image upload to begin. images must be JPG for upload to begin. user must be able to upload only a portion of their images (ie, if they have 10 credits and only want to upload 5 at the time, they can, and their credits available is reduced by only the amount they used). upload progress indicators are a must, and once upload is complete, uploads must be reflected in the “current image uploads” section to assure user upload was successful. images to be stored on a folder on the server as well as emailed to a specific email address for redundancy. on upload, script changes user filenames to “(system generated user acct # from registration)(category)(sequential number).jpg”. script checks for previous image uploads to guard against conflicting file names (ie - if user has already uploaded (acct)(some_category)1.jpg, then the next time they upload under the same category, it should be the next sequential number.

current image uploads - self explanatory

support/message center - simply a user/admin postable message system.

The user dashboard layout includes the following sections:

- Welcome, (User Name)**: A header section with a purchase deadline countdown and an entry deadline countdown.
- Contact Info**: A section for contact information from registration data, which is editable.
- Credits Status**: A section showing Credits Available (# of paid/verified credits) and Credits Pending (awaiting successful payment).
- Image Uploads**: A section for purchasing credits, including a quantity field and a link to pay for credits via Google Checkout XML API.
- Image Upload Line**: A section for uploading images, with one line for each credit available to the user.
- Current Image Uploads**: A section showing three image upload fields (Image 1, Image 2, Image 3) with browse file upload fields and category drop-downs, along with an upload button.
- Message Center**: A section for support tickets and messages, including a user text entry field with a submit button and a transcript of communications with DATE, USER (user ID or administrator), and MESSAGE (showing newest messages first, oldest last).

Notes regarding user end as seen above:

Forgotten password retrieval a must, using email address. Only one account allowed per email address! All data retrieved from the initial registration form should be in its own permanent database - never purging this info. All data retrieved from the account snapshot page needs to be able to be backed up and purged twice a year, credits, account transactions, images, support transcripts, etc are all purged twice a year.

I need to be able to export all fields to a comma delimited document.

Back End/Admin Needs:

I need the ability to:

Add/edit/delete users or specific info as needed.

Manually apply credits or move credits from pending to confirmed/paid.

Manually remove credits if need be.

Manage image uploads, able to delete if necessary specific images.

See all users in list view and see all pertinent stats/fields.

Be able to narrow list to all users with credits remaining and send a batch email to those users.

Send messages to everyone's support/message center at once, and ideally, be able to select users to do so as well.

Have the image purchase become unavailable when the deadline is reached (admin specified).

Have the image upload become unavailable when the deadline is reached (admin specified).

Have the ability to back up the database at any time.

Have the ability to purge all but the user account/contact info from the database completely.