

# Capstone Project Technical Design

Ethan Shilo-Draper

## File Names

- All pages will be named descriptively, in a single word if possible, with the first letter capitalized and subsequent words in camel case. All codeBehind files will have names identical to the page they run behind.
  - o Examples: Login, Dashboard
- Master pages will keep the default names of Site.Master and Site.Mobile.Master.
- Model classes will follow the same naming conventions as pages.
  - o Example: Account
- Images used across the site will follow the same naming conventions as pages. They will be placed in the Site\_Images folder.
  - o Example: Placeholder, DefaultAvatar
- Images uploaded by users will be autoformatted to begin with the user's username, followed by an underscore, followed by the file name of the uploaded file.
  - o Example: ethansd\_Phantom
- Avatar images uploaded by users will follow the same conventions as regular uploads.

## Naming Conventions

Names for anything besides pages are in camel case with a lowercase first letter, with some exceptions described below.

### Page elements

- The IDs of text input fields will begin with "txt\_" followed by a word describing what is to be entered.
  - o Examples: txt\_Title, txt\_Medium
- The IDs of label elements will begin with "lbl\_" followed by a word describing what text will appear there.
  - o Examples: lbl\_uploadError, lbl\_loginError
- The IDs of buttons will begin with "btn" followed by a word describing the action of the button.
  - o Examples: btn
- Wrappers will be <div> tags, with IDs beginning with a descriptor of their contents, followed by "Wrapper".
  - o Examples: headerWrapper, homeWrapper
  - o Wrappers for subareas of flex boxes will instead be the descriptor followed by "Left" or "Right" as appropriate
    - Examples: uploadLeft, uploadRight
- Forms will be encased in ASP panels, with IDs being a descriptor of the form followed by "Form".
  - o Examples: loginForm, uploadForm
- Any elements requiring an ID that do not fit into these categories will be named descriptively.
  - o Examples: imageUpload, dashboardImageGrid

## Variables

- All variables will be camel case and begin with a lowercase letter.
- A few recurring variables will be used:
  - o errors: used to store a series of error messages. Messages are added based on necessity.
  - o Status: used to store either errors or a success message. In some cases, this success message contains information to be stored to a variable, such as a user ID.

## Functions

- Default functions will, by necessity, keep a capital first letter.
- All functions written specifically for this project will retain the formatting of camel case and lowercase first letter.
- Functions to run when a button is clicked will be named with the name of the button followed by “\_Click”.

## Database

- The table containing user login and profile information will be titled Capstone\_Login.
- The table containing user upload information will be title Capstone\_Uploads.

## Organization

### Folders

- All web pages will be kept in the root folder of the project.
- The main CSS file will be kept in the Content folder, as is the default for ASP.Net projects.
- The Account class (described below) will be kept in the Models folder and will use the namespace Capstone\_Project.Models.
- Images used in the layout of the site (including placeholders, most of which will be removed later in development) will be kept in the Site\_Images folder.
- Images uploaded by users as part of their gallery will be kept in the Uploads folder.
- Images uploaded by users to be used as their avatars / icons will be kept in the Avatars folder.

### Functions

- The Account class will represent a user account, containing variables and validation methods for all fields within the login database.
- The Account class will also contain all functions using SQL statements, both query and nonquery. This will include registration and login, as well as upload.
- All other functions will remain in the codeBehind file for their respective pages.