ArtPI

Team: put_me_to_REST
Roster: Jesse "McCree" Chen (PM), Kelvin Ng, Eric "Morty" Lau, David Xiedeng

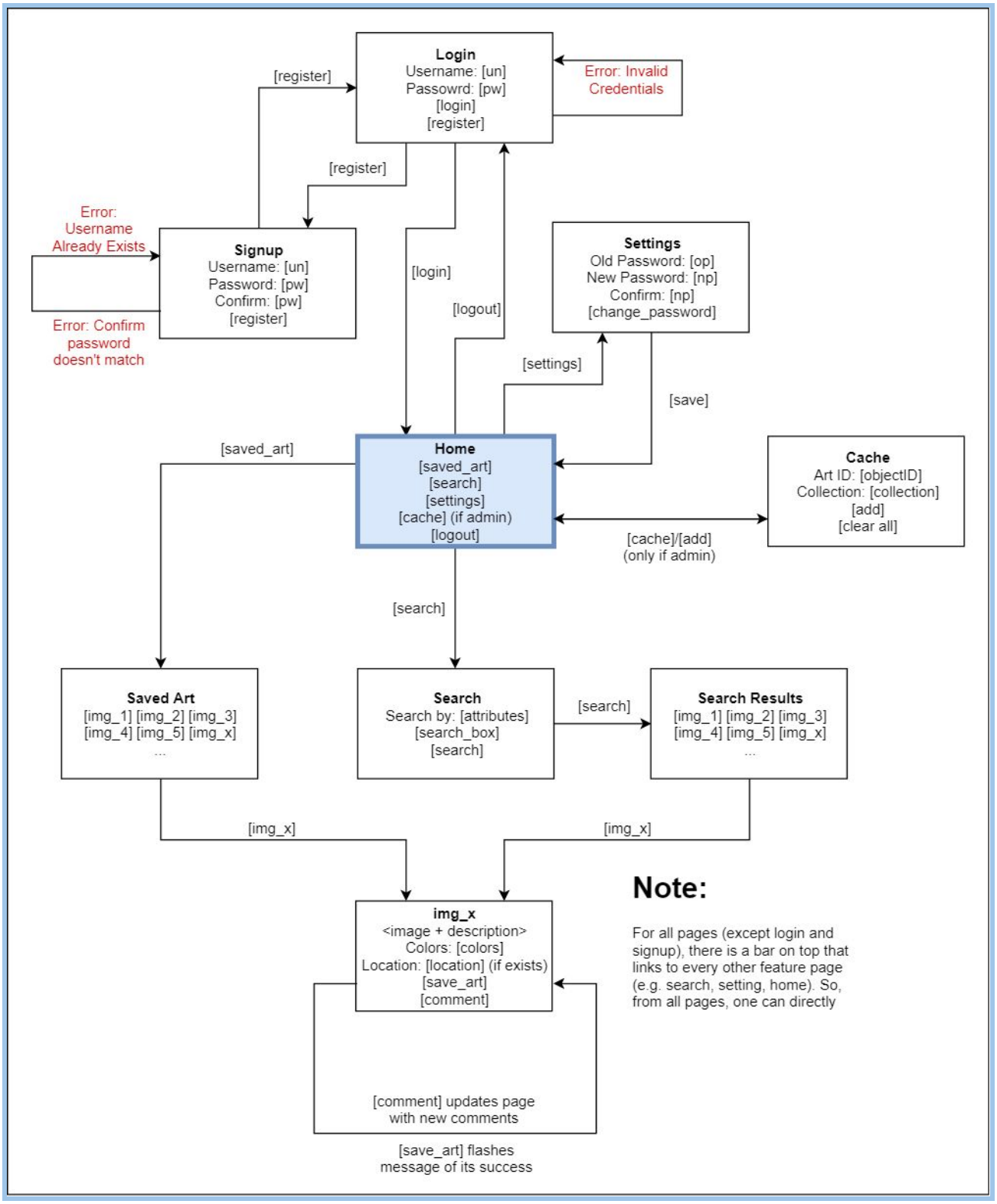
~~~~~~

Overview

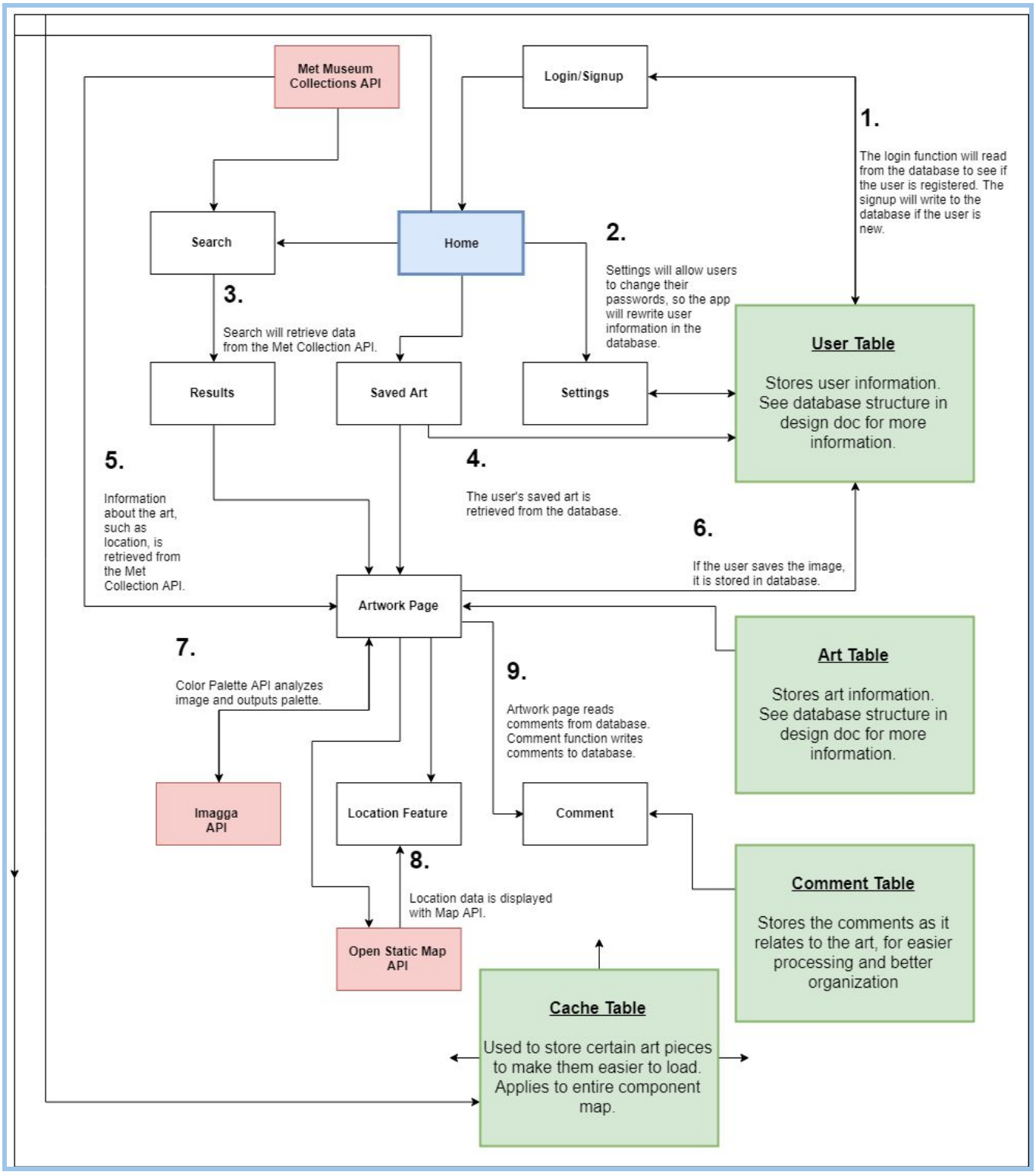
Our website provides a way for users to browse artworks from the Metropolitan Museum of Art's collection. Below are some of the features of our website:

1. Create Account
    a. Upon visiting the site, users can login or create an account to access the website's services
2. Search Gallery
    a. Type in keywords and the website will search through the Met's collections and display the results. Clicking on a piece will take the user to the art's description page.
3. View (Saved) Art
    a. By clicking on a piece of art from search results, the user will be taken to a page with descriptions of the art. In that page, the user will also be able to:
        i. Leave a comment
        ii. Save the art
        iii. Locate the "origin of artwork", if available
        iv. Determine the color palette
4. Comment
    a. Under the artwork and descriptions, users can see other users' comments and leave a comment.
5. Save Art
    a. By saving the art, the user can quickly find it again by going to their Saved Art page and viewing all their saved work.
6. Locate Origin
    a. The Met's API provides the creation place of their works, if known. By using this information and inputting it to our map API, we are able to generate a map of the location.
7. Color Palette
    a. By inputting the image file into a color palette identifying API (Imagga), we can provide information on the colors in the work. If the user is an artist or just studying art, it would be a helpful tool for them.
8. Settings
    a. Users can change their passwords.
~~~~~~

Sitemap


Login
Username: [un]
Passowrd: [pw]
[login]
[register]
[register]
Error: Invalid Credentials
[register]
Error: Username Already Exists
Signup
Username: [un]
Password: [pw]
Confirm: [pw]
[register]
Error: Confirm password doesn't match
[login]
[logout]
Settings
Old Password: [op]
New Password: [np]
Confirm: [np]
[change_password]
[settings]
[save]
[saved_art]
Home
[saved_art]
[search]
[settings]
[cache] (if admin)
[logout]
Cache
Art ID: [objectID]
Collection: [collection]
[add]
[clear all]
[cache]/[add]
(only if admin)
[search]
Saved Art
[img_1] [img_2] [img_3]
[img_4] [img_5] [img_x]
...
Search
Search by: [attributes]
[search_box]
[search]
[search]
Search Results
[img_1] [img_2] [img_3]
[img_4] [img_5] [img_x]
...
[img_x]
[img_x]
img_x
<image + description>
Colors: [colors]
Location: [location] (if exists)
[save_art]
[comment]
[comment] updates page with new comments
[save_art] flashes message of its success
Note:
For all pages (except login and signup), there is a bar on top that links to every other feature page (e.g. search, setting, home). So, from all pages, one can directly

Component Map



APIs:

- Met Museum Collections
    - Used to access the Met's collection of artworks. This is the main API of our website. All other features are interactions of the Met's artworks.
- Imagga
    - This API determines an image's color palette, which is a feature on our website. The idea is to both decorate the page and to help artists identify colors if they want to replicate the piece.
- Open Static Map API
    - This API uses the Met's given location of an artwork (only some artworks have known locations) and provides a static map that we can display. This is another feature that we used to make the information on in the Met database more interactive or user-friendly.

Database Structure

1. User Table

This stores information about user accounts and their private information.

userID	**user**	**password**
1	testUser1	pword1
2	testUser2	pword2
3	trollUser	shrek
…	…	…
#integer	#string	#string

2. Art Table

This stores artwork, their information, and comments.

commentID	**objectID***	**comment**	**user**	**timestamp**
1	208218	Such a nice vase!	1	11--16--2019:10:05:31
2	41293	Hate the sunflower ;(	3	11--17--2019:21:43:02
3	208218	@testUser0, disagree. It looks too murky.	2	11--17--2019:22:00:17

	…	…	…	…
#integer	#integer	#string	#integer	#import datetime

* Conveniently, the Met database has an “objectID” for every item in their collection. So, instead of referring or saving them with their names, we can just use a short number to get the item.

3. Saved Art

This stores artwork, their information, and comments.

userID	**objectID**
1	208218
2	41293
3	41293
…	…
#integer	#integer

4. Cache

This stores data of the artworks for our homepage so we don’t have to constantly call the API.

objectID	**Era**	**Title**	**Artist**	**Image Link**
201926	Medieval and Gothic	Choir screen from the Cathedral of Valladolid	Rafal Amezua	https://images.metmuseum.org/CRDImages/es/original/ES4661.jpg
…	…	…	…	…
#integer	#string	#string	#string	#string

Role Assignments

Jesse "McCree" Chen (PM): Design document, project flow, art-curating, bug-finding
Kelvin Ng: Back-end database and front-end implementation of Met API
Eric "Morty" Lau: Full-stack, CSS, Bootstrap, database, Implementation of Imagga API
David Xiedeng: Back-end database and front-end implementation of Map Static API

Towards the end of our projects, when each individual piece is done, we just worked on small bits and features to polish the website. So, this role assignment is really only reflective of the first stages of the project. It worked out well and we have a working website to our satisfaction.

Revisions since v0

Since v0, we have made several design choices that polished our final results. Here are the major feature changes:

- Instead of making separate pages for commenting, viewing location, etc, we made them fit in a single page with multiple functionalities. This is more consistent with the "modern" design of websites and is less work and less clunky.
- A top bar is added, like a directory list, to make navigating the website much easier.
- The homepage is changed from Saved Art (now separate) to a curated list of "featured" art, sorted by era.
- A cache table is added, especially for the homepage, because the Met API is very slow. It's still slow, but not as bad.
- Added an admin account, for administration purposes. Admins can manually cache artworks, while having all other features as well.
- Team flag has been made!
- API keys moved to a separate json file and added an error page if API keys are not detected.
- Plus much more smaller fixes to make things smoother