ArtPI

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

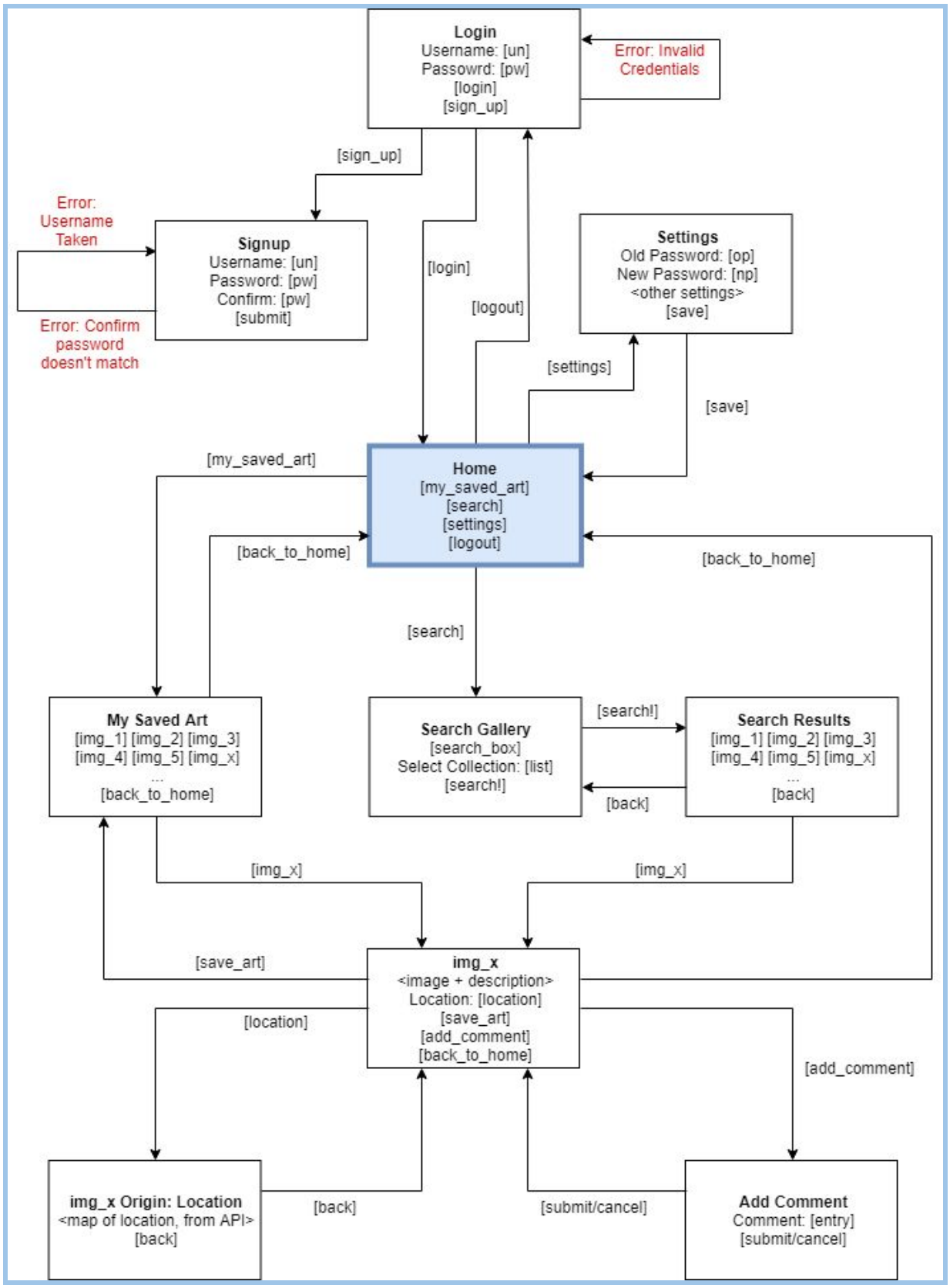
~~~~~~

Overview

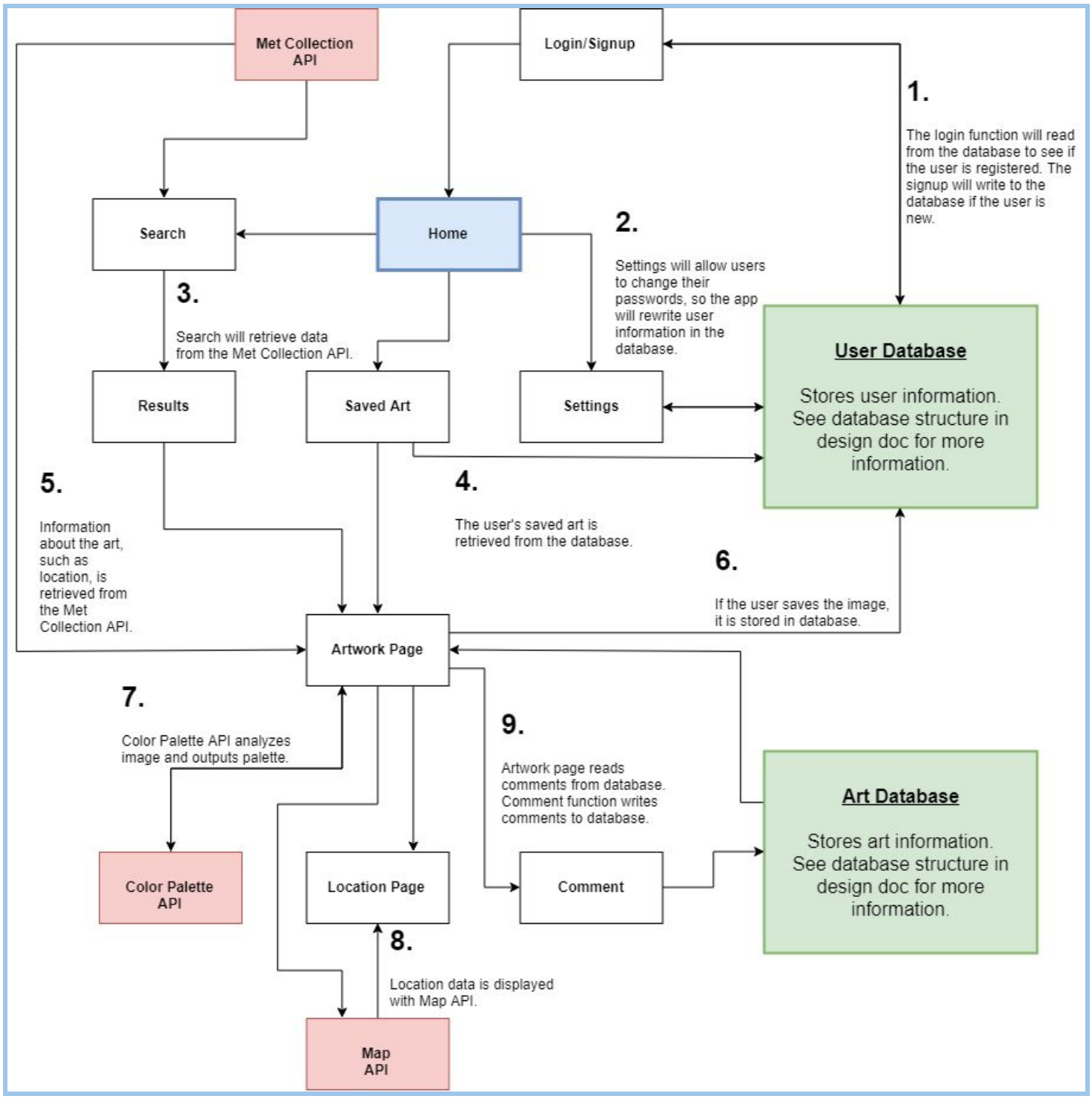
Our website provides a way for users to browse artworks from the Metropolitan Museum of Art's collection. Below are some of the features of our website:

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 "birthplace", 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 homepage 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]
[sign_up]
Error: Invalid Credentials
[sign_up]
Error: Username Taken
Signup
Username: [un]
Password: [pw]
Confirm: [pw]
[submit]
Error: Confirm password doesn't match
[login]
[logout]
Settings
Old Password: [op]
New Password: [np]
<other settings>
[save]
[settings]
[save]
[my_saved_art]
Home
[my_saved_art]
[search]
[settings]
[logout]
[back_to_home]
[back_to_home]
[search]
My Saved Art
[img_1] [img_2] [img_3]
[img_4] [img_5] [img_x]
...
[back_to_home]
Search Gallery
[search_box]
Select Collection: [list]
[search!]
[search!]
[back]
Search Results
[img_1] [img_2] [img_3]
[img_4] [img_5] [img_x]
...
[back]
[img_x]
[img_x]
[save_art]
img_x
<image + description>
Location: [location]
[save_art]
[add_comment]
[back_to_home]
[location]
[add_comment]
img_x Origin: Location
<map of location, from API>
[back]
[back]
[submit/cancel]
Add Comment
Comment: [entry]
[submit/cancel]

Component Map



APIs:

- Met Museum Collections
- Imagga
- Open Static Map API
- OpenCage Geocoder API

Database Structure

1. User Table

This stores information about user accounts and their private information.

```
(userID INTEGER, user TEXT, password TEXT)
```

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.

```
(artID INTEGER, comment TEXT, user TEXT, timestamp BLOB)
```

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

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

Role Assignments

Jesse “McCree” Chen (PM): Design document, project flow, coding here and there
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