

---

# **TEST PLAN FOR PhotoN WEBSITE**

**Anudeep Rao Perala (CS21BTECH11043)**  
**Asli Nitej Reddy Busireddy (CS21BTECH11011)**  
**Narsupalli Sai Vamsi (CS21BTECH11038)**  
**Pranav Varma Pericherla (CS21BTECH11044)**

# Contents

<b>1</b>	<b>Test Overview</b>	<b>3</b>
<b>2</b>	<b>Test Details</b>	<b>7</b>
2.1	Unit Testing . . . . .	7
2.1.1	Appendix for format of inputs to the modules . . . . .	7
2.1.2	Input . . . . .	8
2.1.3	Function . . . . .	9
2.1.4	Output . . . . .	15
2.2	Integrated testing . . . . .	16
2.3	System Testing . . . . .	17
2.3.1	User Account Management . . . . .	17
2.3.2	Photo Management . . . . .	19
2.3.3	Folder Operations . . . . .	21
2.3.4	MarketPlace Interaction . . . . .	24
2.3.5	Album Operations . . . . .	25
2.3.6	Social Interaction . . . . .	27
2.3.7	Search Functionality . . . . .	29
2.4	Performance Testing . . . . .	30
<b>3</b>	<b>Test Analysis</b>	<b>31</b>

# 1 Test Overview

The following are the modules which we want to devise test plan for:

- After listing the modules which are of type **Atomic** are written **Unit testcases**, and for the **Non Atomic** modules we perform **Integrated testing**.

S.No.	Module Name	Module Type
1	Main	Non Atomic
2	Input	Non Atomic
3	HTML Input	Non Atomic
4	Get Text	Atomic
5	Get Password	Atomic
6	Get Tags List	Atomic
7	Get Image	Atomic
8	User Details	Non Atomic
9	Functions	Non atomic
10	User	Non atomic
11	Login	Non Atomic
12	Validate User	Atomic
13	Save Credentials	Atomic
14	Folder Operations	Non Atomic
15	Seeing photos from hidden folder	Non Atomic
16	Validate Password	Atomic
17	Get image IDs list	Atomic
18	Remove image from shared	Atomic

S.No.	Module Name	Module Type
19	Add to hidden folder	Atomic
20	Remove from hidden folder	Atomic
21	Add to favorites	Atomic
22	Remove from Bin	Atomic
23	Recovery from Bin	Atomic
24	Share Operations	Non Atomic
25	Album share	Non Atomic
26	Validate the u_ids from u_id.list (Album Share)	Atomic
27	Add entry to Sharing-Album	Atomic
28	Photo share	Non Atomic
29	Validate the u_ids from u_id.list (Photo Share)	Atomic
30	Add entry to Sharing-Photo	Atomic
31	Remove shared permission	Non Atomic
32	Process user ids	Atomic
33	Update on sharing image	Atomic
34	Update on sharing album	Atomic
35	Vault Operations	Non Atomic
36	Albums	Non Atomic
37	Remove from Album	Atomic
38	Add to Album	Atomic
39	Delete from Album	Atomic
40	Creating an Album	Non Atomic

S.No.	Module Name	Module Type
41	Create a new record in album_details	Atomic
42	Delete an album	Non Atomic
43	Removing an Album	Non Atomic
44	Remove the record in album_details	Atomic
45	Remove the(P_id,a_id) in album_library	Atomic
46	Search Operations	Non Atomic
47	Image Search	Non Atomic
48	Generate auto tags for entered text	Atomic
49	For some tags give me image_ids and ids list	Atomic
50	User Search	Atomic
51	Map	Non Atomic
52	Get the information of all the image locations	Atomic
53	API	Atomic
54	Photo Operations	Non Atomic
55	Global Feed Generation	Atomic
56	Saving a photo from Global feed or Shared folder	Atomic
57	Copying a photo to another album	Atomic
58	Moving between Albums	Atomic
59	Picture Uploading	Non Atomic
60	Generate tags	Atomic
61	Moving between albums	Atomic

S.No.	Module Name	Module Type
62	Tag Operations	Non Atomic
63	Delivers final list of tags based on flag	Atomic
64	Add Tags	Non Atomic
65	Add tags to the user	Atomic
66	Add tags to photo	Atomic
67	Remove Tags	Non Atomic
68	Remove tags from user	Atomic
69	Remove tags from photo	Atomic
70	Outputs	Non Atomic
71	Show alerts according to return status	Atomic
72	Image card view	Atomic
73	Tile view of photos	Atomic
74	User's list view	Atomic
75	Profile view	Atomic
76	Map View	Atomic

## 2 Test Details

### 2.1 Unit Testing

#### 2.1.1 Appendix for format of inputs to the modules

##### Text

The Text can be any string of regex `(a-zA-Z| )?` and a can have a maximum of 1024 characters. Any text which does not suffice these conditions is classified as **invalid** and is called **inv-text**.

**text** - Valid text

**inv-text** - Invalid text

##### Username

The Username can be any string of regex `(0-9|a-zA-Z|_)?`. Any username which does not suffice these conditions is classified as **invalid** and is called **inv-username**.

**username** - Valid username

**inv-username** - Invalid username

##### Password

The Password can be any string of regex `(0-9|a-zA-Z|_|!|@||$|*|!)?`. Any password which does not suffice these conditions is classified as **invalid** and is called **inv-password**.

**password** - Valid password

**inv-password** - Invalid password

##### Tags

The Tags can be any string of the regex `(a-zA-Z)?`. Any tag which does not suffice these conditions is classified as **invalid** and is called **inv-tag**. And when we are looking into list of tags then we can have a maximum of 5 tags in tags.list otherwise it is considered **inv-tags.list**.

**tag** - Valid tag

**inv-tag** - Invalid tag

**tag.list** - Valid tag list

**inv-tag.list** - Invalid tag list

## Image

An Image can be a maximum of 5MB and it must be in the format of jpeg/ jpg/ png. Any image which does not suffice these conditions is classified as **invalid** and is called **inv-image**.

**image** - Valid image

**inv-image** - Invalid image

## Return status

We only have a limited number of return status i.e., we have range of return status. Any return status which does not suffice these conditions is classified as **invalid** and is called **inv-return**.

**return status** - Valid return status

**inv-status** - Invalid return status

## p\_id, a\_id, u\_id

All the photo ID(p\_id), album ID(a\_id) and user ID(u\_id) are positive integers. Any of p\_id, a\_id, u\_id which does not suffice these conditions is classified as **invalid** and is called **inv-pid, inv-aid, inv-uid**.

**x\_id** - Valid x\_id(x: p, a, u)

**inv-xid** - Invalid x\_id(x: p, a, u)

**x\_id.list** - Valid list of x ID's

**inv-x.list** - Invalid list of x ID's

## Location information

The location information is in the standard latitude and longitude information format. Any of location data which does not suffice these conditions is classified as **invalid** and is called **inv-location**.

**location** - Valid location

**inv-location** - Invalid location

### 2.1.2 Input

S.No.	Module Name	Input	Expected Output
1	Get Text	text	Data is sent to the next step in the flow
2		inv-text	Error Message
3	Get Password	password	Data is sent to the next step in the flow



S.No.	Module Name	Input	Expected Output
4		inv-password	Error Message
5	Get Tags List	tags.list	Data is sent to the next step in the flow
6		inv-tags.list	Error Message
7	Get Image	Image	Data is sent to the next step in the flow
8		inv-image	Error Message

### 2.1.3 Function

S.No.	Module Name	Input	Expected Output
9	Validate User	correct-username, correct-password	The user is logged in
10		correct-username, password	Wrong Password
11		inv-username, password	Error message
12		username, inv-password	Error message
13		inv-username, inv-password	Error message
14		username, password	Incorrect credentials
15	Save Credentials	username, password	Credentials Saved
16		inv-username, inv-password	Error message
17		inv-username, password	Error message
18		username, inv-password	Error message
19	Validate Password (for accessing locked folder)	password	Folder is unlocked and the locked photos are shown
20		inv-password	Error message
21	Get Image IDs list	u_id, v_id	p_id.list is returned
22		inv-uid, inv-vid	Error message

S.No.	Module Name	Input	Expected Output
23		inv-uid, v_id	Error message
24		u_id, inv-vid	Error message
25	Remove image from shared	u_id, p_id	Image is removed from shared
26		inv-uid, inv-pid	Error message
27		inv-uid, p_id	Error message
28		u_id, pid	Error message
29	Add to hidden folder	p_id	Photo added to hidden folder
30		inv-pid	Error Message
31	Remove from hidden folder	p_id	Photo removed from hidden folder
32		inv-pid	Error Message
33	Add to favorites	p_id	Photo added to favorites
34		inv-pid	Error Message
35	Remove from Bin	p_id	Photo is deleted from Bin
36		inv-pid	Error Message
37	Recovery from Bin	p_id	Photo is recovered from bin and is added to All Photos
38		inv-pid	Error Message
39	Validate the u_ids from u_id.list (Photo Share)	u_id.list	User IDs are validated for sharing
40		inv-uid.list	Error Message
41	Add entry to sharing Image	p_id, u_id.list	Entry is added to Sharing Image
42		inv-pid, inv-uid.list	Error message
43		inv-pid, uid.list	Error message

S.No.	Module Name	Input	Expected Output
44		pid, inv-uid.list	Error message
45	Validate the u_ids from u_id.list (Album Share)	u_id.list	User IDs are validated for sharing
46		inv-uid.list	Error Message
47	Add entry to Sharing Album	p_id, u_id.list	Entry is added to Sharing Album
48		inv-pid, inv-uid.list	Error message
49		inv-pid, uid.list	Error message
50		pid, inv-uid.list	Error message
51	Process user ID's	u_id.list	User IDs are validated for removing shared permission
52		inv-uid.list	Error Message
53	Update on sharing image	u_id.list	User IDs are validated for updating the permissions to new list of users for sharing the photo
54		inv-uid.list	Error Message
55	Update on sharing album	u_id.list	User IDs are validated for updating the permissions to new list of users for sharing the album
56		inv-uid.list	Error Message
57	Remove from album	p_id, a_id	The photo p_id is removed from the album a_id
58		inv-pid, inv-aid	Error message
59		inv-pid, aid	Error message
60		pid, inv-aid	Error message
61	Add to album	p_id, a_id	The photo p_id is added to the album a_id

S.No.	Module Name	Input	Expected Output
62		inv-pid, inv-aid	Error message
63		inv-pid, aid	Error message
64		pid, inv-aid	Error message
65	Delete from Album	u_id	p_id is validated for removing the p_id from album
66		inv-pid	Error Message
67	Create a new record in album_details	u_id, a_id, album_name	New album is created with album ID a_id and name album_name
68		If atleast one of u_id/ a_id/ album_name is <b>inv</b>	Error message
69	Delete an Album	a_id	a_id is validated for deleting the album
70		inv-aid	Error Message
71	Remove record in album details	a_id	a_id is validated for deleting the album from album_details
72		inv-aid	Error Message
73	Remove the(p_id.list,a_id) in album library	p_id.list, a_id	The record is deleted in album library
74		inv-pid.list, a_id	Error message
75		inv-pid.list, inv-aid	Error message
76		p_id.list, inv-aid	Error message
77	Generate auto tags for entered text	text	Text is sent for tag generation
		inv-text	Error message
78	For some tags give me image ids and ids list	tags.list	tags.list is sent search and the images with same tags are returned
		inv-tags.list	Error message

S.No.	Module Name	Input	Expected Output
79	User search	username	User with the username is returned
		inv-username	Error message
80	Get the information of all the image locations	u_id, a_id	All the locations of images in a_id of u_id are returned
81		inv-uid, inv-aid	Error message
82		inv-uid, a_id	Error message
83		u_id, inv-aid	Error message
84	API	location	Processed longitude and latitude information of the inputted data is returned
85		inv-location	Error message
86	Global Feed Generation	tags.list	After validation of the tags photos with tags as that of inputted list are returned
87		inv-tags.list	Error message
88	Saving a photo from Global feed or Shared folder	p_id	Photo is saved to All Photos
89		inv-pid	Error message
90	Copying a photo to another album	image, p_id, a_id	Photo is copied to the destination album
91		If atleast one of image/ p_id/ a_id is <b>inv</b>	Error message
92	Moving between Albums	p_id, a_id1, a_id2	Photo is moved from a_id1 to a_id2
93		If atleast one of p_id/ a_id1/ a_id2 is <b>inv</b>	Error message
94	Check image validity	Correct image format (e.g., JPEG)	Image is valid and ready for upload.

S.No.	Module Name	Input	Expected Output
95		Unsupported image format (e.g., TIFF)	Error message
		Image size exceeds limit (e.g., over 5MB)	Error message
		Below minimum resolution is	Error message
		Corrupted image file	Error message
		Image with incorrect aspect ratio	Error message
96	Generate tags	image	tags for the image are returned
97		inv-image	Error message
98	Delivers final list of tags based on flag	tags.list	tags are processed and then returned for photo tags updation
99		inv-tags.list	Error message
100	Add tags to user	tags.list, uid	New tags are added to user profile
101		inv-tags.list, uid	Error message
102		inv-tags.list, inv-uid	Error message
103		tags.list, inv-uid	Error message
104	Add tags to photo	tags.list, pid	New tags are added to the photo
105		inv-tags.list, pid	Error message
106		inv-tags.list, inv-pid	Error message
107		tags.list, inv-pid	Error message
108	Remove tags from user	tags.list, uid	Input Tags are removed from the user profile
109		inv-tags.list, uid	Error message
110		inv-tags.list, inv-uid	Error message
111		tags.list, inv-uid	Error message

S.No.	Module Name	Input	Expected Output
112	Remove tags from photo	tags.list, pid	Input Tags are removed from the photo
113		inv-tags.list, pid	Error message
114		inv-tags.list, inv-pid	Error message
115		tags.list, inv-pid	Error message

#### 2.1.4 Output

S.No.	Module Name	Input	Expected Output
116	Show alerts according to return status	return status	Corresponding alert is displayed
117		inv-status	Error message
118	Image card view	p_id	Corresponding image is displayed in the image card view
119		inv-pid	Error message
120	Tile view of photos	list.p_id	Corresponding images are displayed in tile view
121		inv-pid.list	Error message
122	Users list view	u_id.list	List view of the users is displayed
123		inv-uid.list	Error message
124	Profile view	u_id	Profile view of the user is displayed
125		inv-uid	Error message
126	Map view	location	Map view of the inputted location's is displayed
127		inv-location	Error message

## 2.2 Integrated testing

S.No.	Module	Test Description	Expected Outcome
1	User	User provides correct credentials to Login module; upon validation, Login module saves user_id token.	User is successfully logged in and a user_id token is saved and directed to Homepage
2	Seeing photos from hidden folder	User provides correct password to validate user access and retrieve image IDs list from the locked folder.	The images are displayed from correct image IDs that are fetched in the locked folder page.
3	Album share	User provides a list of user IDs to share an album with; system validates the IDs and shares the album.	Album is shared, and a success status is returned.
4	Photo share	User provides a list of user IDs to share a photo with; system validates the IDs and shares the photo.	Photo is shared, and a success status is returned.
5	Remove shared permission	User inputs user IDs to remove sharing permissions from a selected photo or album.	Permissions are removed, and a success status is returned.
6	Creating an Album	User inputs album details to create an album	The new album is created and displayed alongside the list of existing albums.
7	Delete an Album	User selects album to delete	Moves all photos (from every album ) to bin and displays rest of albums



S.No.	Module	Test Description	Expected Outcome
8	Removing an Album	User selects album to remove	data related to album are deleted, photos will have no effect and displays rest of albums
9	Image Search	User inputs text description	images are filtered from generated tags and are displayed to User
10	Map	User selects View on Map inside Album Page	gathers location data for all images in current album, and display locations on World-map using API
11	Picture Uploading	User selects image from local device to Upload	Success status will be displayed for valid image else Error Message
12	Add Tags	User inputs list of tags for Picture or UserDetails to update	Displays return status after Updation and validation
13	Remove Tags	User inputs list of tags for Picture or UserDetails to remove	Displays return status after Removal and validation

## 2.3 System Testing

In this section we would like to test all the features that we have stated in the SRS of the project.

### 2.3.1 User Account Management

S.No.	Functionality	Scenario	Expected Outcome
-------	---------------	----------	------------------

S.No.	Functionality	Scenario	Expected Outcome
1	Signup	<ul style="list-style-type: none"> <li>• User navigates to the signup page on PhotoN</li> <li>• User enters the required credentials: username, password</li> <li>• User submits the form to create an account</li> </ul>	<ul style="list-style-type: none"> <li>• System validates the input data</li> <li>• The system redirects the user to the login page with a message Account successfully created</li> </ul>
2	Login	<ul style="list-style-type: none"> <li>• User navigates to the login page on PhotoN</li> <li>• User enters login credentials: username and password</li> <li>• User submits the form to log in</li> </ul>	<ul style="list-style-type: none"> <li>• System checks the credentials</li> <li>• If the credentials are correct, the system logs the user in and redirects to their profile page</li> <li>• If the credentials are incorrect, the system shows an error message Invalid email or password</li> </ul>
3	Logout	<ul style="list-style-type: none"> <li>• User clicks on the logout button in the navigation menu</li> </ul>	<ul style="list-style-type: none"> <li>• System logs the user out</li> </ul>

S.No.	Functionality	Scenario	Expected Outcome
4	Profile Management	<ul style="list-style-type: none"> <li>• User navigates to the profile settings</li> <li>• User edits their personal information and submits the changes</li> </ul>	<ul style="list-style-type: none"> <li>• System updates the user's profile with the new information</li> </ul>

### 2.3.2 Photo Management

S.No.	Functionality	Scenario	Expected Outcome
1	Upload Photos	<ul style="list-style-type: none"> <li>• User navigates to the upload interface</li> <li>• User selects photos to upload from their device</li> </ul>	<ul style="list-style-type: none"> <li>• The system uploads the selected photos</li> </ul>
2	Move to bin	<ul style="list-style-type: none"> <li>• User selects the photo to delete</li> <li>• User initiates the delete action, which moves the photo to the bin</li> </ul>	<ul style="list-style-type: none"> <li>• The system moves the selected photo to the bin</li> </ul>

S.No.	Functionality	Scenario	Expected Outcome
3	Edit Photos	<ul style="list-style-type: none"> <li>• User selects a photo to edit from their gallery</li> <li>• User modifies the attributes of the photo</li> </ul>	<ul style="list-style-type: none"> <li>• The system applies the changes to the photo</li> <li>• The system updates the photo in the gallery with the new attributes</li> </ul>
4	Edit Photos	<ul style="list-style-type: none"> <li>• User selects a photo to edit from their gallery</li> <li>• User modifies the attributes of the photo</li> </ul>	<ul style="list-style-type: none"> <li>• The system applies the changes to the photo</li> <li>• The system updates the photo in the gallery with the new attributes</li> </ul>
5	Save Photo	<ul style="list-style-type: none"> <li>• User selects a photo from the global feed</li> <li>• User saves the photo to their collection of All Photos</li> </ul>	<ul style="list-style-type: none"> <li>• The system adds the selected photo to the user's All Photos collection</li> </ul>

S.No.	Functionality	Scenario	Expected Outcome
6	Download Photo	<ul style="list-style-type: none"> <li>• User selects a photo to download from their gallery or the global feed</li> <li>• User initiates the download action</li> </ul>	<ul style="list-style-type: none"> <li>• The system processes the download</li> <li>• The photo is saved to the user's local device</li> </ul>
7	PhotoMap	<ul style="list-style-type: none"> <li>• User accesses the PhotoMap feature</li> <li>• User views photos as points on a map based on the geolocation data associated with the photos</li> </ul>	<ul style="list-style-type: none"> <li>• The system displays a map with points representing the location of the user's photos</li> </ul>

### 2.3.3 Folder Operations

S.No.	Functionality	Scenario	Expected Outcome
1	Favorite a photo	<ul style="list-style-type: none"> <li>• User views a photo in their gallery or on the global feed</li> <li>• User selects the option to mark the photo as a favorite</li> </ul>	<ul style="list-style-type: none"> <li>• The system adds the photo to the Favorites folder</li> </ul>

S.No.	Functionality	Scenario	Expected Outcome
2	UnFavorite a photo	<ul style="list-style-type: none"> <li>• User navigates to their Favorites folder</li> <li>• User selects the option to remove the photo from Favorites</li> </ul>	<ul style="list-style-type: none"> <li>• The system removes the photo from the Favorites folder</li> </ul>
3	Like a photo	<ul style="list-style-type: none"> <li>• User browses photos in the global feed</li> <li>• User selects the option to like a photo</li> </ul>	<ul style="list-style-type: none"> <li>• The system marks the photo as liked and adds it to the Global favorites folder</li> </ul>
4	Unlike a photo	<ul style="list-style-type: none"> <li>• User views their Global favorites folder</li> <li>• User selects the option to unlike a photo</li> </ul>	<ul style="list-style-type: none"> <li>• The system removes the photo from the Global favorites folder</li> </ul>

S.No.	Functionality	Scenario	Expected Outcome
5	Add to Locked Folder	<ul style="list-style-type: none"> <li>• User selects a photo from their gallery</li> <li>• User chooses to add the photo to the Locked folder for additional privacy</li> </ul>	<ul style="list-style-type: none"> <li>• The system adds the photo to the Locked folder</li> </ul>
6	Remove from Locked Folder	<ul style="list-style-type: none"> <li>• User accesses their Locked folder</li> <li>• User selects the photo to remove from the Locked folder</li> </ul>	<ul style="list-style-type: none"> <li>• The system removes the photo from the Locked folder</li> </ul>
7	Restore from Bin	<ul style="list-style-type: none"> <li>• User navigates to the Bin where deleted photos are temporarily stored</li> <li>• User selects the photo to restore</li> </ul>	<ul style="list-style-type: none"> <li>• The system restores the photo to the All Photos collection</li> </ul>

S.No.	Functionality	Scenario	Expected Outcome
8	Delete from Bin	<ul style="list-style-type: none"> <li>• User opens the Bin</li> <li>• User selects the photo to permanently delete</li> </ul>	<ul style="list-style-type: none"> <li>• The system permanently deletes the photo from the Bin and the PhotoN application</li> </ul>
9	Remove from Shared Folder	<ul style="list-style-type: none"> <li>• User views the contents of the Shared folder</li> <li>• User selects the photo or album to remove</li> </ul>	<ul style="list-style-type: none"> <li>• The system removes the photo or album from the Shared folder</li> </ul>
10	Remove from Global Photos	<ul style="list-style-type: none"> <li>• User views their contributions to the Global Photos section or Marketplace</li> <li>• User selects the photo to remove</li> </ul>	<ul style="list-style-type: none"> <li>• The system removes the photo from the Global Photos section</li> </ul>

### 2.3.4 MarketPlace Interaction

S.No.	Functionality	Scenario	Expected Outcome
-------	---------------	----------	------------------



S.No.	Functionality	Scenario	Expected Outcome
1	Add a photo to global feed	<ul style="list-style-type: none"> <li>• User selects a photo from their private collection</li> <li>• User initiates the action to add the photo to the global feed</li> </ul>	<ul style="list-style-type: none"> <li>• The system adds the photo from the user's private collection to the global feed</li> </ul>

### 2.3.5 Album Operations

S.No.	Functionality	Scenario	Expected Outcome
1	Create Album	<ul style="list-style-type: none"> <li>• User navigates to the album creation interface in the PhotoN application</li> <li>• User provides an album name for creation</li> </ul>	<ul style="list-style-type: none"> <li>• The system creates a new album</li> </ul>
2	Remove Album	<ul style="list-style-type: none"> <li>• User selects an existing album</li> <li>• User initiates the removal of the album without deleting the photos it contains from All Photos folder</li> </ul>	<ul style="list-style-type: none"> <li>• The system removes the album from the user's albums list</li> <li>• The photos that were in the album remain in the user's collection of All Photos</li> </ul>

S.No.	Functionality	Scenario	Expected Outcome
3	Delete Album	<ul style="list-style-type: none"> <li>• User selects an existing album</li> <li>• User chooses to delete the album, which moves the photos to the bin</li> </ul>	<ul style="list-style-type: none"> <li>• The system deletes the album and moves the photos it contained to the bin</li> </ul>
4	Add Photo to Album	<ul style="list-style-type: none"> <li>• User navigates to an existing album</li> <li>• User selects photo from All Photos to add to the album</li> <li>• User initiates the action to add the photo to the global feed</li> </ul>	<ul style="list-style-type: none"> <li>• The system adds the photo from the user's private collection to the global feed</li> </ul>
5	Copy Photo to Album	<ul style="list-style-type: none"> <li>• User navigates to the vault where some photos are stored</li> <li>• User selects a photo to be copied to an album</li> </ul>	<ul style="list-style-type: none"> <li>• The system copies the photo from the vault to the specified album</li> </ul>

S.No.	Functionality	Scenario	Expected Outcome
6	Move Photo Between Albums	<ul style="list-style-type: none"> <li>• User opens an album and selects a photo to move</li> <li>• User chooses a destination album for the selected photo</li> </ul>	<ul style="list-style-type: none"> <li>• The system moves the photo from the original album to the destination album</li> </ul>
7	Remove Photo from Album	<ul style="list-style-type: none"> <li>• User navigates to an album and selects a photo to remove</li> </ul>	<ul style="list-style-type: none"> <li>• The system removes the photo from the album</li> </ul>

### 2.3.6 Social Interaction

S.No.	Functionality	Scenario	Expected Outcome
1	Follow User	<ul style="list-style-type: none"> <li>• User navigates to another user's profile page on PhotoN</li> <li>• User clicks on the Follow button on that profile</li> </ul>	<ul style="list-style-type: none"> <li>• The system updates the following list for the user</li> </ul>

S.No.	Functionality	Scenario	Expected Outcome
2	Unfollow User	<ul style="list-style-type: none"> <li>• User goes to the profile page of a user they are currently following</li> <li>• User clicks on the Unfollow button</li> </ul>	<ul style="list-style-type: none"> <li>• The system updates the following list and removes the unfollowed user</li> </ul>
3	List of Following Users	<ul style="list-style-type: none"> <li>• User navigates to their profile or a dedicated following list page</li> <li>• User views the list of users they are following</li> </ul>	<ul style="list-style-type: none"> <li>• The system displays all users that the current user is following</li> </ul>
4	Photo Sharing	<ul style="list-style-type: none"> <li>• User selects a private photo from their collection</li> <li>• User initiates the share action and selects another user or multiple users to share the photo with</li> </ul>	<ul style="list-style-type: none"> <li>• The system shares the selected photo with the chosen users</li> </ul>

S.No.	Functionality	Scenario	Expected Outcome
5	Album Sharing	<ul style="list-style-type: none"> <li>• User selects a private album from their collection</li> <li>• User chooses the share option and selects another user or a group to share the album with</li> </ul>	<ul style="list-style-type: none"> <li>• The system shares the selected album with the chosen users</li> </ul>

### 2.3.7 Search Functionality

S.No.	Functionality	Scenario	Expected Outcome
1	Photo Search	<ul style="list-style-type: none"> <li>• User enters a search term related to tags or descriptions in the photo search bar</li> <li>• User initiates the search action</li> </ul>	<ul style="list-style-type: none"> <li>• The system retrieves and displays photos matching the entered tags or descriptions</li> </ul>
2	Album Sharing	<ul style="list-style-type: none"> <li>• User types another user's username into the user search bar</li> <li>• User executes the search action</li> </ul>	<ul style="list-style-type: none"> <li>• The system lists user profiles that match the search username</li> </ul>

## 2.4 Performance Testing

S.No.	Testing Type	Scenario	Expected Outcome
1	Load Testing/ Response time	Simulating around 10 users and then testing the load performance of the website for tasks like login ,photo upload, Album creation, search, etc	The system should respond within time below 4 sec under peak load
2	Scalability Testing	Doubling the user activities like uploads, searching, etc	The system should scale to accommodate the increased activities with no more than a 10% increase in response time
3	Stability Testing	Sustaining a load of around 10 users for 5 hours	No system crashes or significant performance should degradation occur
4	Concurrency Testing	Test the system's handling of simultaneous actions	All the concurrency actions should be done so that all the race conditions are eliminated
5	Resource Usage	Monitor CPU, memory, and storage use under different loads	CPU usage peaks at 60%, memory at 75%, and storage growth is linear with content addition
6	Database Performance	Evaluating the efficiency of database operations under load	Database transactions complete within 2 seconds for 95% of operations during peak load

### 3 Test Analysis

S.No.	Module Name	Technique for Testing	Reason for choosing the technique
1	Get Text	Boundary Value Analysis	Text inputs have boundary conditions that need thorough testing, especially with text length
2	Get Password	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
3	Get Tags List	Boundary Value Analysis	Checking that only a fixed number of tags are taken as input
4	Get Image	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
5	Validate User	Pair Wise testing	Checking for all class of possible combinations for the input variables
6	Save Credentials	Pair wise testing	Checking for all class of possible combinations for the input variables
7	Validate Password (for accessing locked folder)	Equivalence Class partitioning	Checking if the input provided belongs to valid class
8	Get Image IDs list	Pair Wise testing	Checking for different class of combinations of user ID, vault ID
9	Remove image from shared	Pair Wise testing	Checking for different class of combinations of user ID, photo ID
10	Add to hidden folder	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
11	Remove from hidden folder	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
12	Add to favorites	Equivalence Class Partitioning	Checking if the input provided belongs to valid class

<b>S.No.</b>	<b>Module Name</b>	<b>Technique for Testing</b>	<b>Reason for choosing the technique</b>
13	Remove from Bin	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
14	Recovery from Bin	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
15	Validate the u ids from u id.list (Photo Share)	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
16	Add entry to Sharing Photo	Pair wise testing	Checking for all class of possible combinations for the input variables
17	Validate the u ids from u id.list (Album Share)	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
18	Add entry to Sharing Album	Pair wise testing	Checking for all class of possible combinations for the input variables
19	Process user ID's	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
20	Update on sharing image	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
21	Update on sharing album	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
22	Remove from album	Pair wise testing	Checking for all class of possible combinations for the input variables
23	Add to album	Pair wise testing	Checking for all class of possible combinations for the input variables
24	Delete from Album	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
25	Create a new record in album_details	Pair wise testing	Checking for all class of possible combinations for the input variables
26	Delete an Album	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
27	Remove the record in album details	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
28	Remove the(P_id.list,a_id) in album library	Pair wise testing	Checking for all class of possible combinations for the input variables



S.No.	Module Name	Technique for Testing	Reason for choosing the technique
29	Generate auto tags for entered text	Boundary Value Analysis	Text inputs have boundary conditions that need thorough testing, especially with text length
30	For some tags give me image ids and ids list	Equivalence Class Partitioning	Not allowing the class of tags which belong to particular class
31	User search	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
32	Get the information of all the image locations	Pair wise testing	Checking for all class of possible combinations for the input variables
33	API	Equivalence Class Partitioning	Checking if the list provided has valid information for location
34	Global Feed Generation	Equivalence Class Partitioning	Checking if the tags list provided belongs to valid class
35	Saving a photo from Global feed or Shared folder	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
36	Copying a photo to another album	Pair wise testing	Checking for all class of possible combinations for the input variables
37	Moving between Albums	Pair wise testing	Checking for all class of possible combinations for the input variables
38	Generate Tags	Equivalence Class Partitioning	Checking if the input provided belongs to valid class
39	Delivers final list of tags based on flag	Equivalence Class Partitioning	Checking if the tags list provided belongs to valid class
40	Add tags to user	Pair wise testing	Checking for all class of possible combinations for the input variables
41	Add tags to photo	Pair wise testing	Checking for all class of possible combinations for the input variables
42	Remove tags from user	Pair wise testing	Checking for all class of possible combinations for the input variables
43	Remove tags from Photo	Pair wise testing	Checking for all class of possible combinations for the input variables

<b>S.No.</b>	<b>Module Name</b>	<b>Technique for Testing</b>	<b>Reason for choosing the technique</b>
44	Show alerts according to return status	Boundary Value analysis	Checking if the input return status is in the allowed range of values
45	Image card view	Equivalence Class Partitioning	Checking if all the photo IDs provided belongs to valid class
46	Tile view of photos	Equivalence Class Partitioning	Checking if all the photo IDs provided belongs to valid class
47	User's list view	Equivalence Class Partitioning	Checking if all the user IDs provided belongs to valid class
48	Profile view	Equivalence Class Partitioning	Checking if the user ID provided belongs to valid class
49	Map View	Equivalence Class Partitioning	Checking if the mappings list provided belongs to valid class