Software Design

for

SkyBar Photo Management System

Group ID: R01

Abhyuday Choumal-CS20B1001,
Beerelly Srinitha-CS20B1004,
Kaustubh Kesharwani-CS19B1015,
Rakesh Kumbhkar-CS20B1017,
Sikander Kathat-CS20B1020,
Yugal Garg-CS20B1027.

Instructor: Manish Singh
Course: Software Engineering (CS210)
TA's: Suryamukhi K, Ashish Kishor
Date: 2nd April 2022

Contents

1.Overview

After detailed case study, following are the basic classes and actions that emerge out:-

Classes: (Basic building blocks of SkyBar)

S.N.	Class	Responsibility
1	Upload	Add photos to the database for further use of the user.
2	Dashboard	Show the particular items for that domain.
3	Download	Download the photo for the user in their system with a specific extension.
4	Security	It will manage full privacy and keep the users data safe and secure with us,rather than allowing others to access that.(No other users can access others data)
5	Sharing	Manage functionality of sharing data with multiple users.
6	Photo-Manager	Manage sorting, editing and settings.
7	GUI	Manage the Graphical User Interface.
8	Data Repository	Manage all the data of users. Work as a bridge in between the main modules and server.

Note: Other additional classes may get added during implementation.

Actions:

S.N.	Action
1	Add/Delete/Edit/Download Photo Dashboard/viewer
2	Select/Create/Delete/Share/Remove/Rename
3	Raise/Send/Delete Alerts
4	Validate user
5	Process/Save Image

2.Inheritance Structure:

There does not seem to be any inheritance structure because of the lack of commonality between the classes.

3.Aggregation :

The logical structure suggests the following aggregation between the classes.

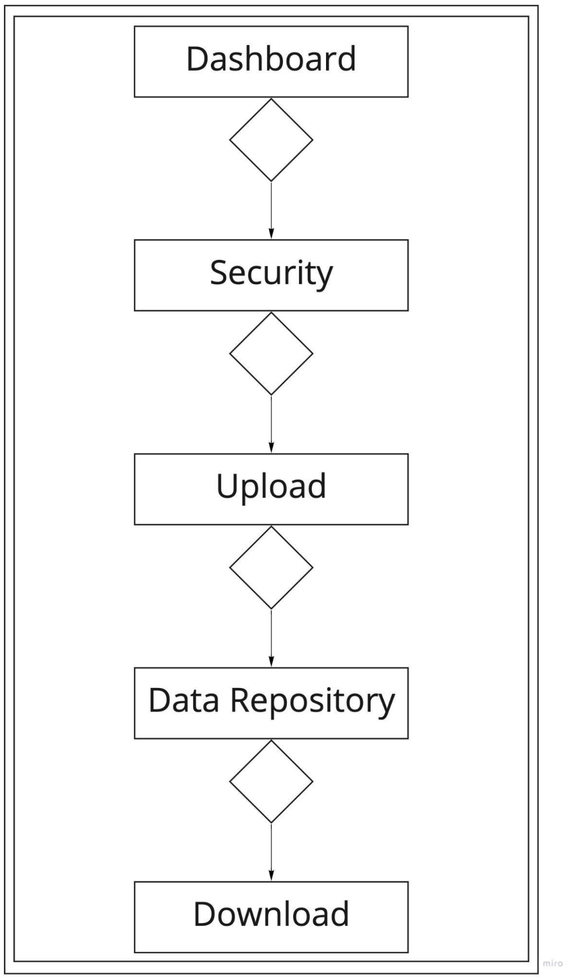
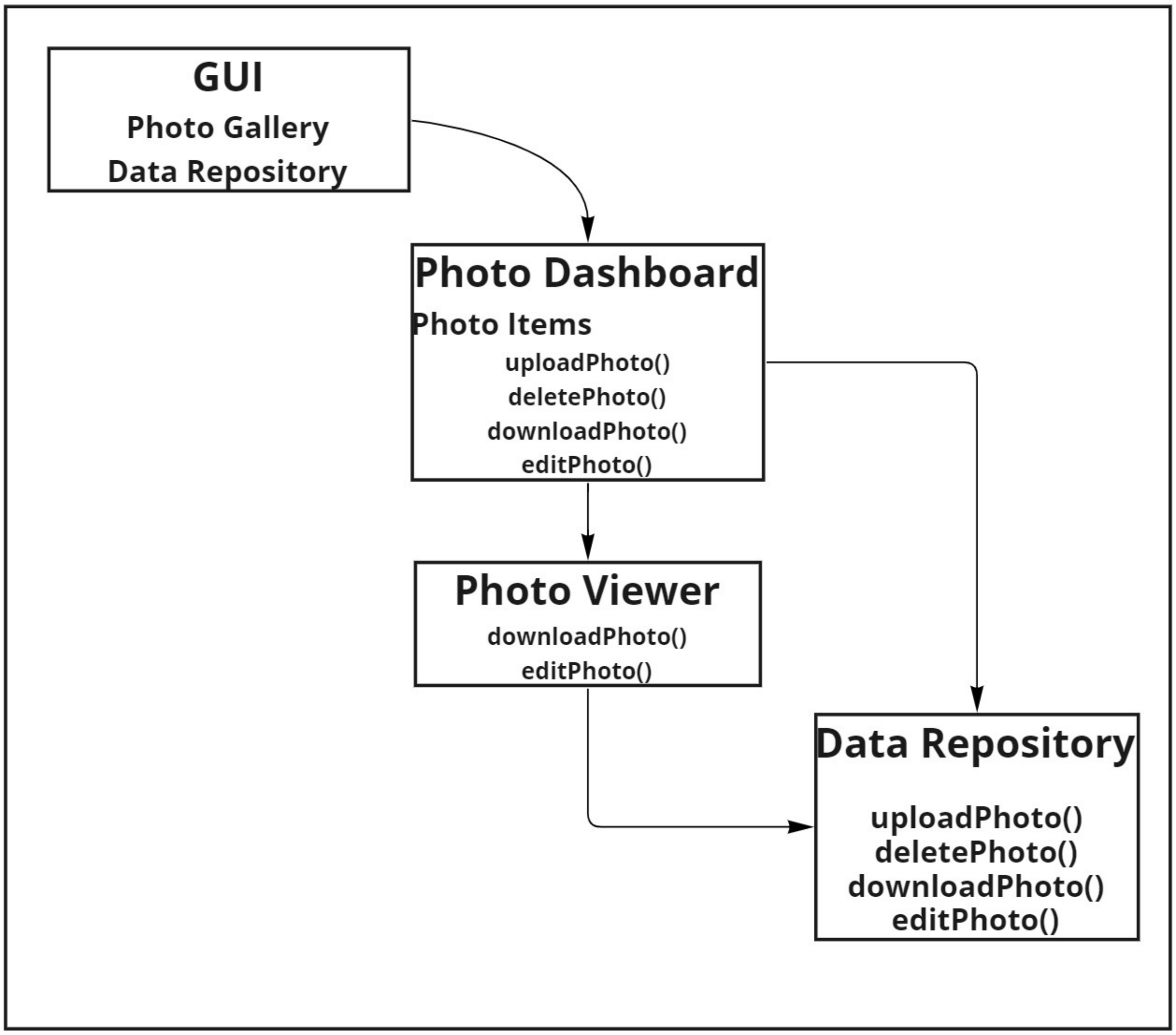


Fig-3

4. Association:

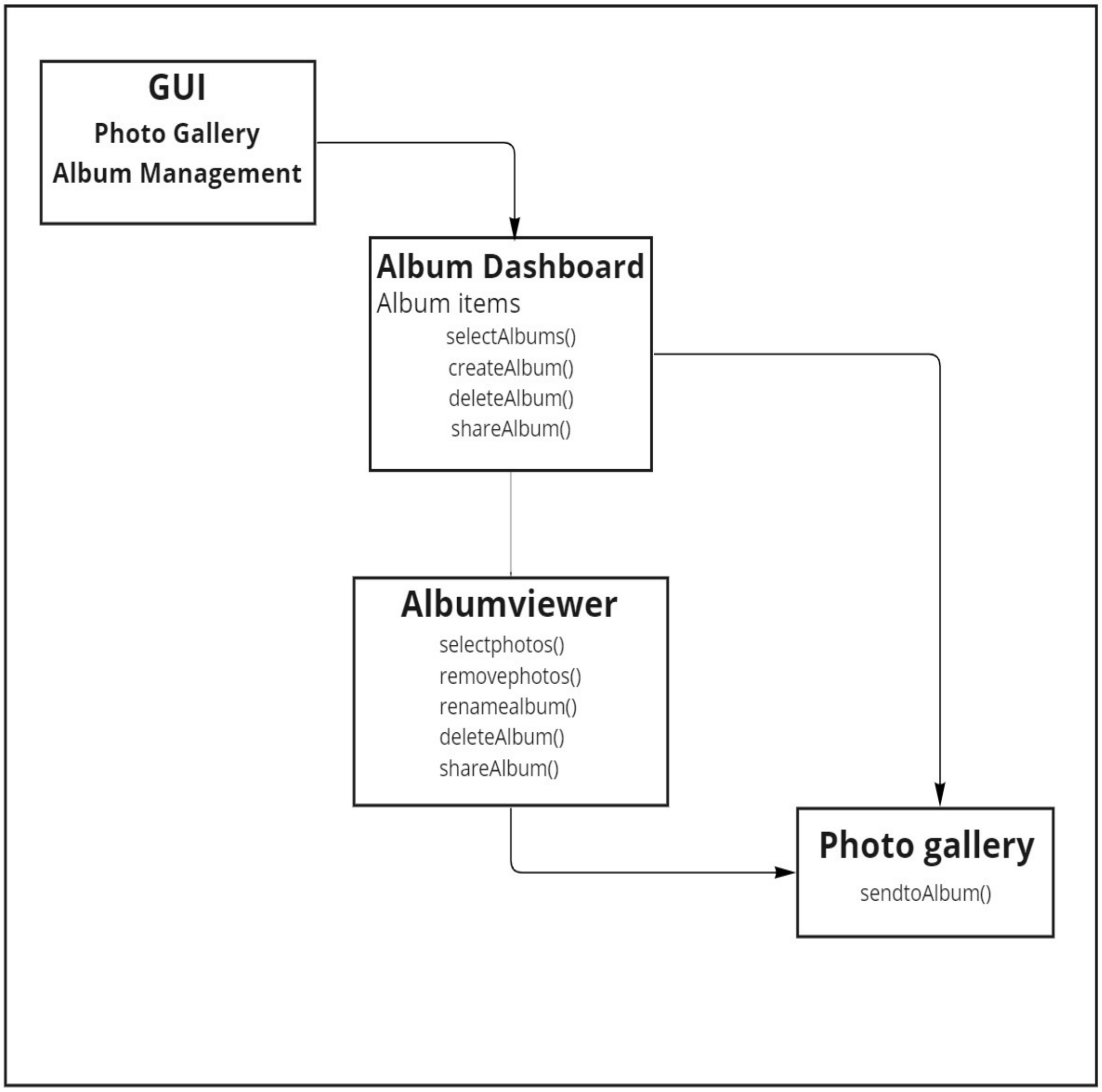
4.1 Principal Action

Association for action add/Delete/Edit/Download Photo Dashboard/Viewer



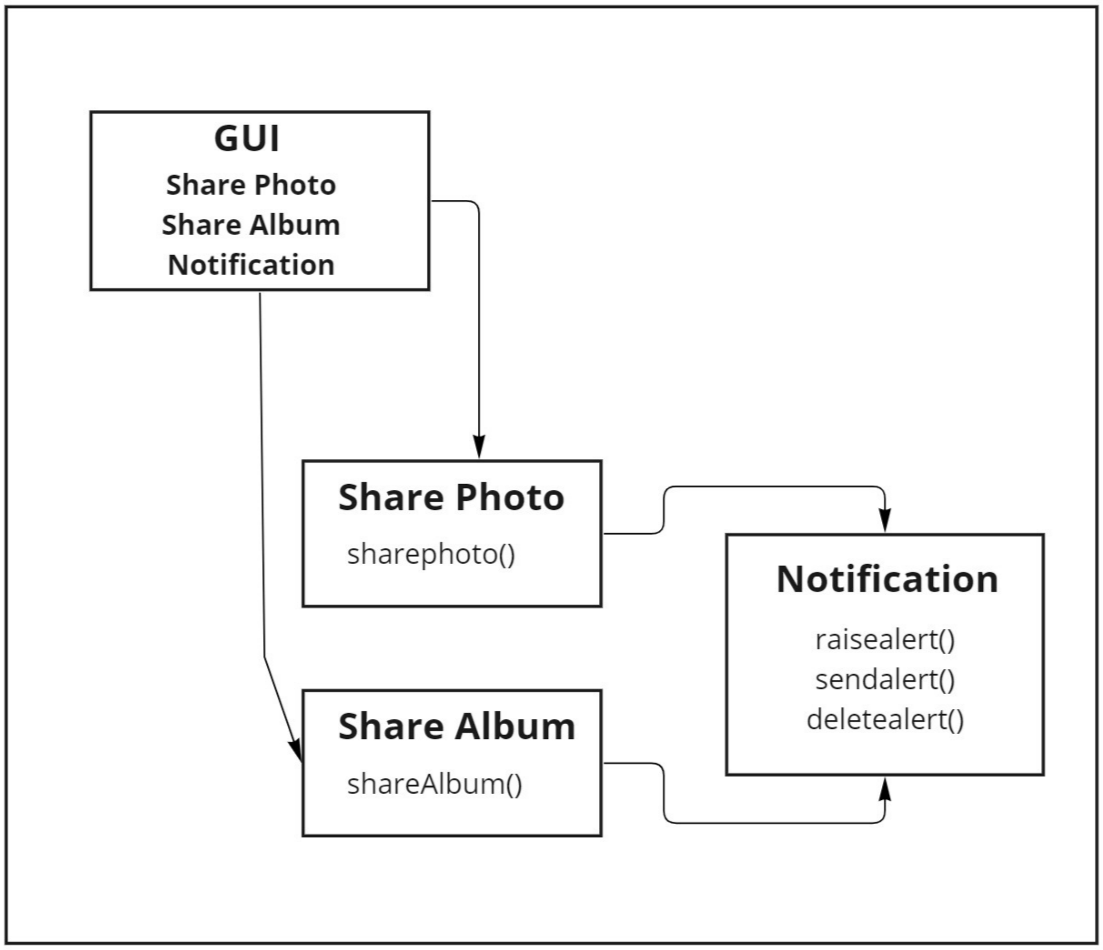
4.2 Principal Action

Association for action Select/Create/Delete/Share/Remove/Rename



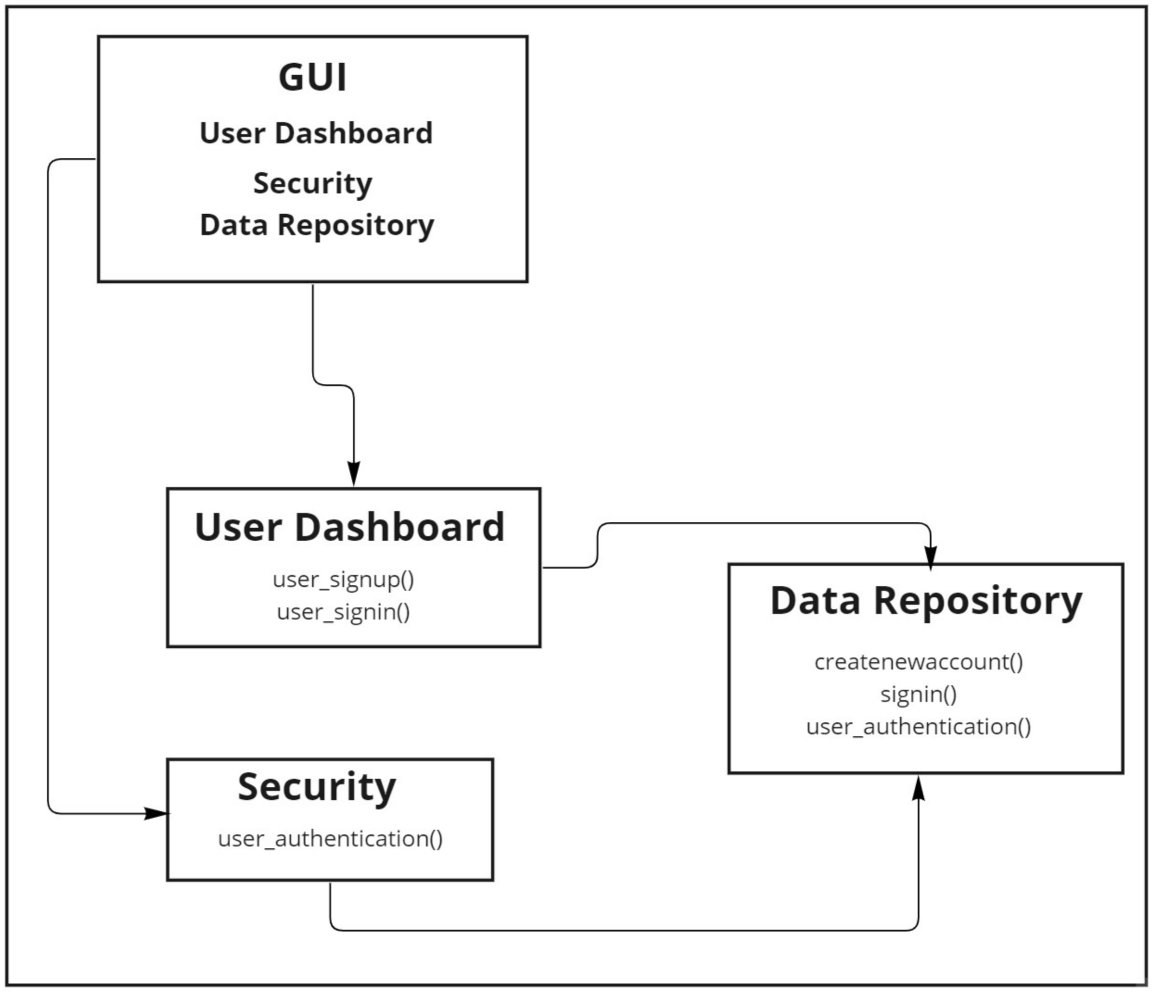
4.3 Principal Action

Association for action Raise/Send/Delete Alerts



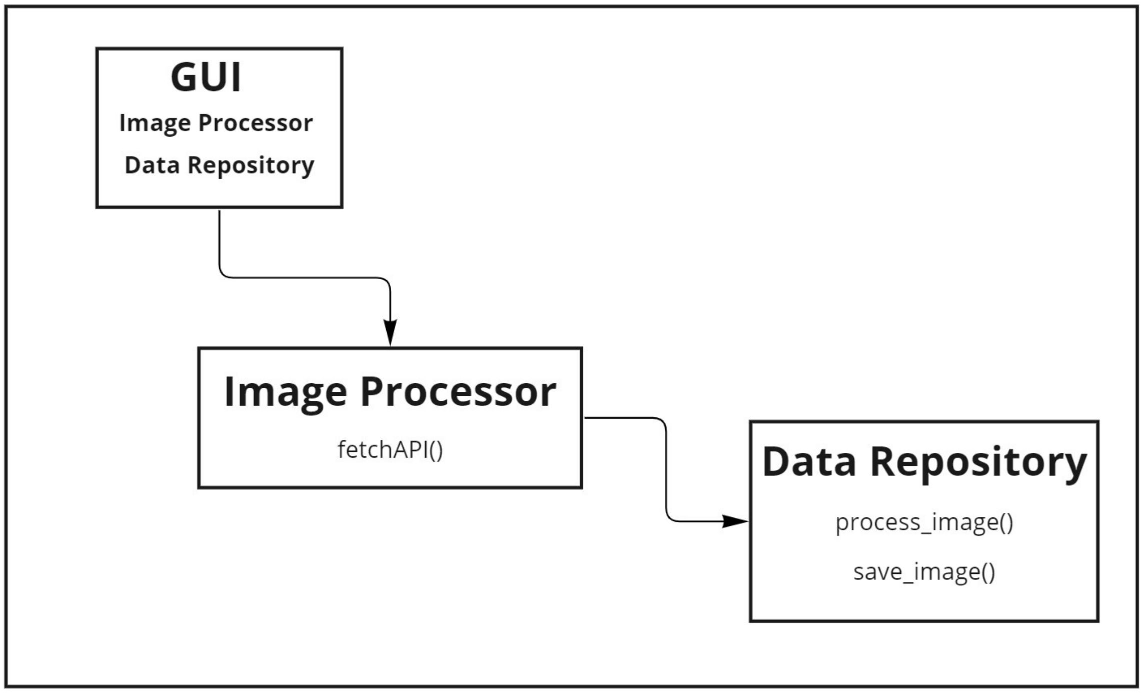
4.4 Principal Action

Association for action Validate user



4.5 Principal Action

Association for action Process/save Image



5. Detail Design Specification:

It consists of a list of main classes and their attributes and methods with proper comments.

5.1 Class Upload

//method//

GET //using the HTML form.

5.2 Class Dashboard

//attributes//

Dashboard Objects //JSON of all items on dashboard
Username // Current session user stats

//methods//

displayItems() //display the required items from DOM
getUser() //get user data from current session

5.3 Class Download

//methods//

POST // post using app.post included in nodejs

5.4 Class Security

//attributes//

String username: //username of the application user is stored
String Password: //the respective password of the application user

//methods//

boolean UserValidation(String username, String Password);
//user validation for personal security of individual users
boolean changePassword(String oldPassword, String newPassword,String RetypePassword);
//to change or modify the password of the validated users.

5.5 Class Sharing

//attributes//

Sender //owner of the picture
Receiver // the person to whom we share the image

//methods//

userMange() //Send sender image to receiver data segment

5.6 Class Photo-Manager

//methods//

sortPhoto() //Sorting of the photo
add_a_task() //add a image
delete_a_task() //delete a image
extension_changer() //Change the image extension using API

5.7 Class GUI

//attributes//

Dashboard //Object of the dashboard class
PhotoManager // Object of Photo manager class
DataRepository // Object of the DataRepository Class
SecureLogin // Object of the Security Class

//methods//

void createGUI() //creates the Graphical User Interface

5.8 Class Data Repository

//methods//

getData() // Get data from server
sendData() // Send data to the server
save_image() // Save data received from the server
find_data() //find data requested by the server