

**Dharmsinh Desai University, Nadiad**

**Faculty of Technology**

**Department of Computer Engineering**

**B. Tech. CE Semester – VI**

Subject: (CE – 620) **Object Oriented Software Engineering**

Project Title**: *Pixby***

**Submitted by**:

Name: **Vyom Pathak**

Roll No: **CE099**

ID: **17CEUON038**

Guided By:

**Prof. Prashant M. Jadav**

Associate Professor, CE Dept.,

Dharmsinh Desai University



Dharmsinh Desai University, Nadiad

Faculty of Technology, Department of Computer Engineering

**CERTIFICATE**

This is to certify that Object Oriented Software Engineering’s project entitled “**Pixby**” is the bonafied report of work carried out by

1. **Pathak Vyom (17CEUON038)**

Of Department of Computer Engineering, Semester VI, academic year 2019-20,

under our supervision and guidance.

|  |  |
| --- | --- |
| Guide | HOD |
|  |  |
| **Prof. Prashant M. Jadav** | **Dr. C. K. Bhensdadia** |
| Associate Professor of | Head of the Department of |
| Department of Computer | Department of Computer |
| Engineering, Dharmsinh Desai University, Nadiad. | Engineering, Dharmsinh Desai University, Nadiad. |

|  |  |  |
| --- | --- | --- |
| **No.** | **Content** | **Page No.** |
| **1** | Introduction | **5** |
| **2** | Software Requirement Specifications | **6** |
| **3** | Design Documents | **12** |
| **4** | Implementation Details | **22** |
| **5** | Functionality Prototype | **23** |
| **6** | Testing Details | **35** |
| **7** | Conclusion | **39** |
| **8** | Limitations and Future Extensions | **40** |
| **9** | Bibliography | **41** |

**Overview**

This application is used for editing photos and saving them and downloading them.

User can upload image and can edit the images and than download or save according to their sign in status.

**Introduction**

This application is used for editing photos. User can upload the photo to the system and edit according to their needs. The software provides functionalities like image RGB value modification, image styling, and image flipping along the axis, image rotation as well as applying some filters on the images like Sepia, Greyscaling, and Inverting etc. User can than save the edited image in his/her account on the server or can download the images. User can also see their saved images as well as re-edit them. User can also perform undo and redo operation on their images to remove or add the features. Admin can remove user and can add new filters.

**Technology Used:**

* Windows Form Application to create application.
* Microsoft Sql Server as database. [MSSQL]
* UMLET for Designing Different UML Diagram.

**Tools Used:**

* I used Github as a version control method and to manage the project.
* I used Visual Studio for development of the code.

**Software Requirement Specification**

**Pixby**

**FUNCTIONAL REQUIREMENTS:**

→ Admin

**R1. Manage User:**

DESCRIPTION: Admin can view all users ,remove users according to his/her

desire.

**R1.1 Remove User:**

INPUT: Admin selects the user.

OUTPUT: Selected user will be removed.

**R1.2 View Users:**

INPUT: Admin selects view option.

OUTPUT: User list will be shown

**R2 Manage Filters:**

DESCRIPTION: Admin can add and delete any filter he/she desires.

**R2.1 Add Filter:**

Input: Filter code.

Output: Confirmation Status Message will be shown.

**R2.2 Remove Filter:**

Input: Select Filter Code.

Output: Confirmation Status Message will be shown.

→ Client

**R1. Manage Images:**

Description: This functionality allows user to upload images on the editor using the local file manager. It also allows user to save the uploaded image in the server after all the processing has been completed. It also allows user to remove image from the editor. User can also download his/her images from his/her account.

**R1.1 Upload Image:**

INPUT: User selection from file manager for his/her desired image.

OUTPUT: Confirmation message.

PROCESS: The selected image will be validated i.e. checked if the image

uploaded is of the correct format or not.

NEXT: The user will make his/her desired changes on the image.

EXCEPTION FLOW: If the selected image is not in correct format error will be

thrown.

**R1.2 Save Image:**

INPUT: User selects the save option.

OUTPUT: Confirmation message.

PROCESS: The specified the new image will be validated.

EXCEPTION FLOW: If the image is not valid error will be thrown.

**R1.3 Remove Image:**

DESCRIPTION: User selected image will be removed.

INPUT: User selection.

OUTPUT: Confirmation message.

**R1.4 Download Image:**

DESCRIPTION: User selected image will be downloaded.

INPUT: User selection.

OUTPUT: Confirmation message.

EXCEPTION FLOW: If the file path is not valid error will be thrown.

**R2. Manage Filters:**

DESCRIPTION: This functionality will allow user to use various type of filter according to his/her desire. It includes: image cropping, brightness adjustment, apply different styles, image flipping, image rotation and image RGB value modification.

**R2.1 Manage Crop Filter:**

INPUT: User selects the area to be cropped.

OUTPUT: Image transformation will be shown.

**R2.2 Manage Rotation Filter:**

DESCRIPTION: User selects which type of rotation to perform and selects the

value of rotation.

INPUT: User gives degree of rotation and type of rotation.

OUTPUT: Image transformation will be shown.

**R2.3 Manage Flipping Filter:**

DESCRIPTION: User will select which type of flip to perform horizontal or vertical.

INPUT: User selects type of flip.

OUTPUT: Image transformation will be shown.

**R2.4 Manage Color Filter:**

DESCRIPTION: User will select the appropriate color value for red, green and blue

from range 0-255.

INPUT: User type of color and give the appropriate color level.

OUTPUT: Image transformation will be shown.

**R2.5 Manage Brightness Filter:**

INPUT: User selects appropriate level of brightness.

OUTPUT: Image transformation will be shown.

**R2.6: Manage Style Filter:**

DESCRIPTION: User can select from a range of different styles like greyscale,

negative effect and Arctic effect.

INPUT: User selects appropriate styling for the image.

OUTPUT: Image transformation will be shown.

**R3. Error Handling:**

DESCRIPTION: If during any process error occurs appropriate error message will be displayed. For upload image,saving image,download image or applying filter error handling is done.

**R3.1 Upload Error:**

DESCRIPTION: If any type of error occurs during upload user will be prompted with a dialog-box.

INPUT: User selects an image from his/her local machine.

OUTPUT: Message will be prompted to the user.

**R3.2 Saving Error:**

DESCRIPTION: If any kind of error occurs during the saving stage of an image, error will be shown to the user.

INPUT: User selects the save option.

OUTPUT: Message will be prompted to the user.

**R3.3 Downloading Error:**

DESCRIPTION: If any kind of error occurs during the downloading stage of an image, error will be shown to the user.

INPUT: User selects the dowload option.

OUTPUT: Message will be prompted to the user.

**R4. Manage User:**

DESCRIPTION: User can create new account , login to existing account and remove

his/her account.

**R4.1 Sign-Up User:**

DESCRIPTION: User enters his/her details and will be validated from the database.

INPUT: User details.

OUTPUT: Confirmation Message.

**R4.2 Login User:**

DESCRIPTION: User will enter his/her details and will be saved in the database.

INPUT: User details.

OUTPUT: Confirmation Message.

**R4.3 Remove User:**

DESCRIPTION: User can delete his/her account from the application using the

remove option.

INPUT: User selects the remove option.

OUTPUT: Confirmation Message.

**NON FUNCTIONAL REQUIREMENTS:**

SOFTWARE: Visual Studio, Github Desktop

OS: Windows

**Design Documents**

1. Use Case Diagram:

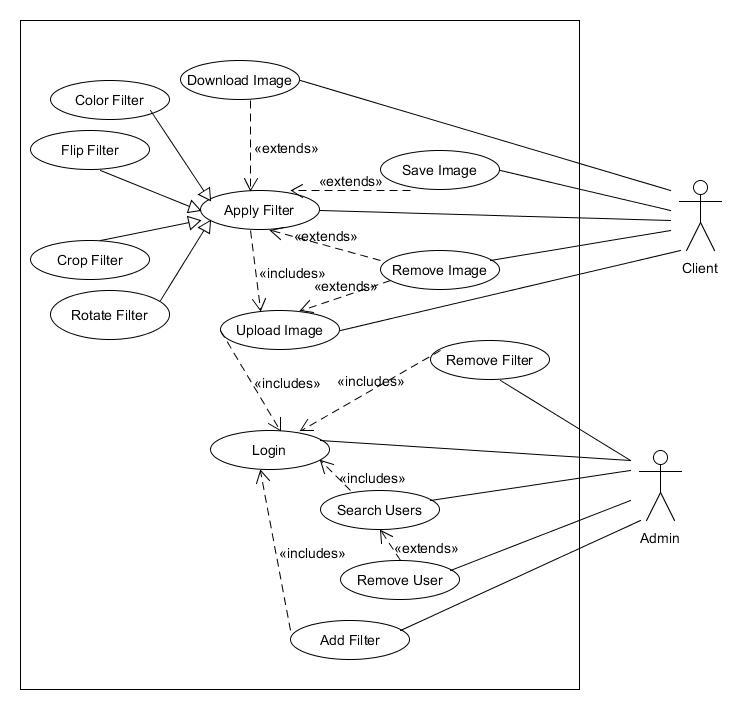


Figure 1 : Use Case Diagram

1. Class Diagram:

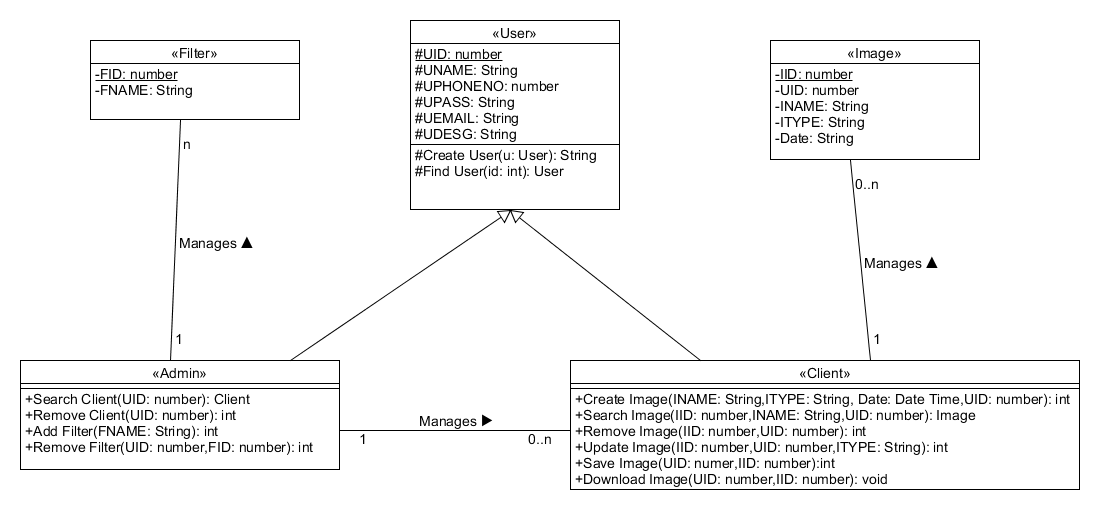


Figure 2 : Class Diagram

1. Sequence Diagram:

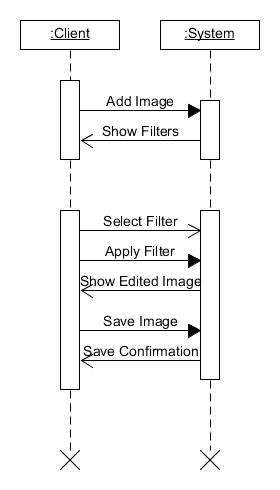


Figure 3 : Sequence Diagram

Apply Filter Scenario

1. Activity Diagrams :

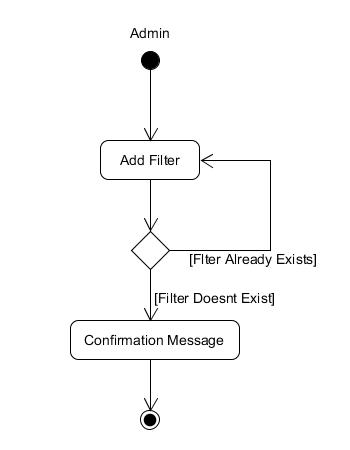


Figure 4 : Activity Diagram

Add Filter Activity

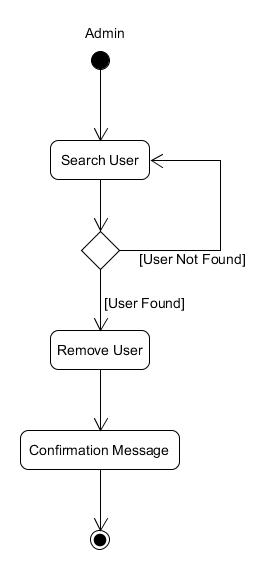
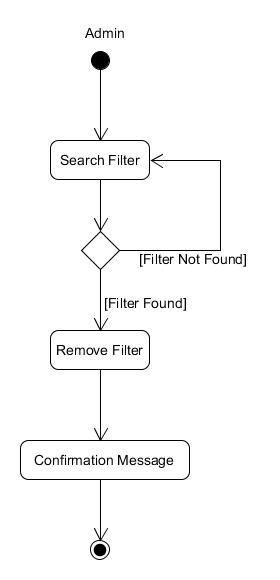


Figure 5 : Activity Diagram Figure 6 : Activity Diagram

Remove Filter Activity Remove User Activity

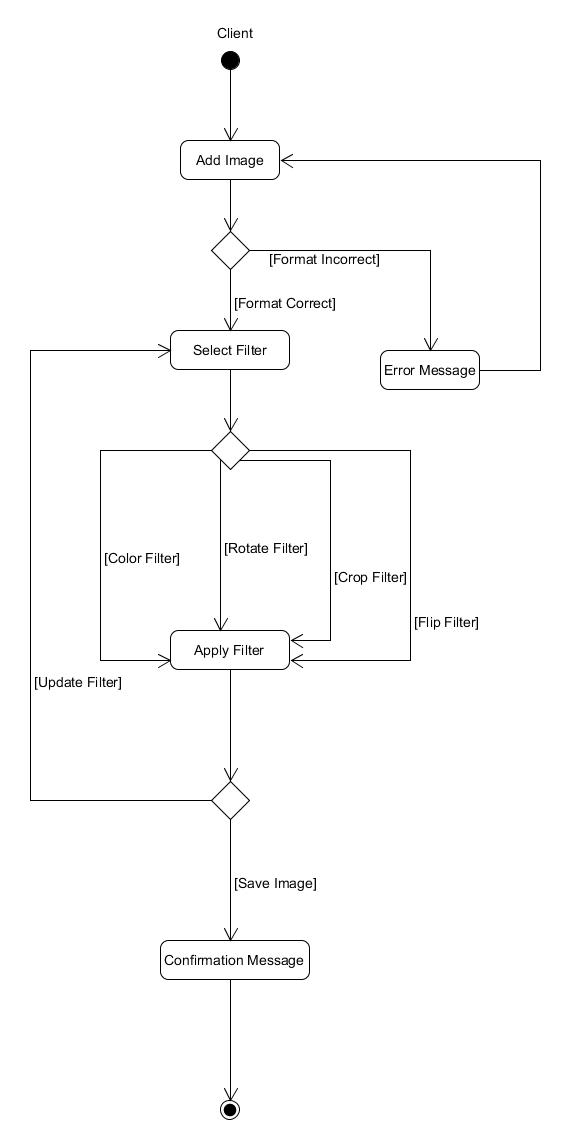


Figure 7 : Activity Diagram

Apply Filter Activity

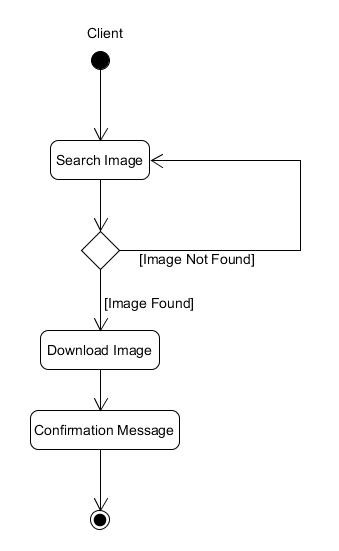


Figure 8 : Activity Diagram

Download Image Activity

1. State Diagram:

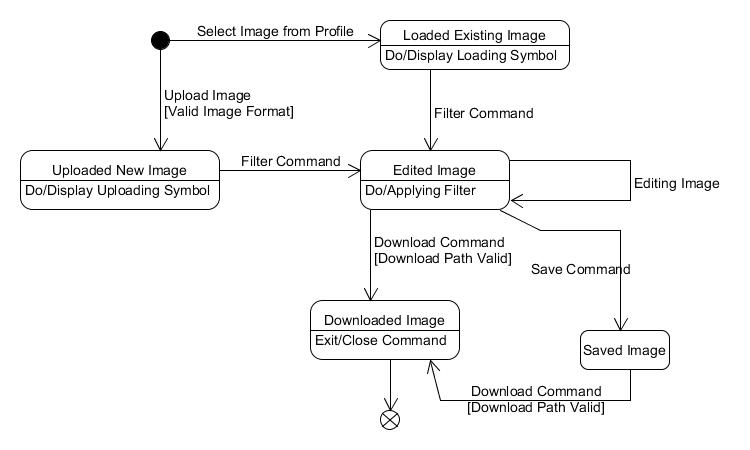


Figure 9 : State Diagram

Image State Diagram

1. E-R Diagram:

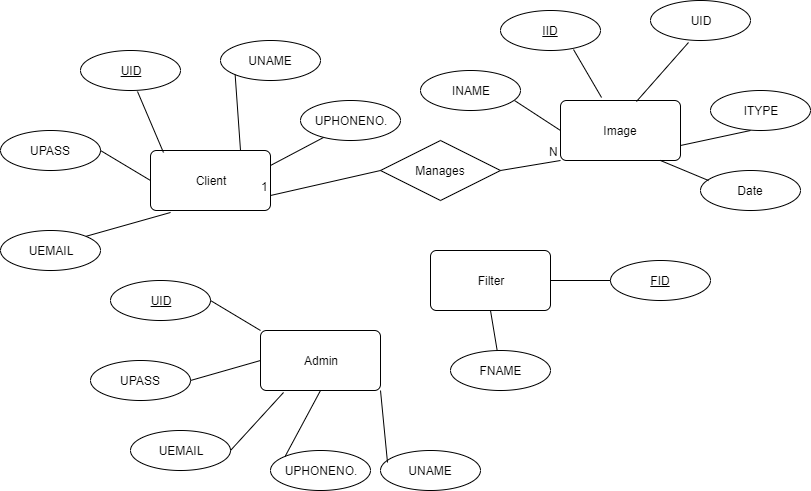


Figure 10 : E-R Diagram

7. Data Dictionary:

|  |  |  |  |
| --- | --- | --- | --- |
|  | User |  |  |
| Attributes | Datatype | Size | Feature |
| UID | int | 20 | AUTO\_INC |
| UNAME | varchar | 30 |  |
| UDESG | varchar | 20 |  |
| UEMAIL | varchar | 20 |  |
| UPASS | varchar | 20 |  |
| UPHONENO. | number | 10 |  |

Table 1 : Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
|  | Filter |  |  |
| Attributes | Datatype | Size | Feature |
| FID | varchar | 20 | AUTO\_INC |
| FNAME | varchar | 30 |  |

Table 2 : Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
|  | Image |  |  |
| Attributes | Datatype | Size | Feature |
| UID | int | 20 | FK |
| IID | int | 20 | AUTO\_INC |
| INAME | varchar | 20 |  |
| ITYPE | varchar | 20 |  |
| Date | Date Time |  |  |

Table 3 : Data Dictionary

**Implementation Details**

**Sign In And Sign Up Module:**

I used a static variable to maintain the session of the user which has signed in. During sign in process, the user data is verified by the database and if the user is not in the database it throws an error. User can also register him/her-self using the registration form.

**Editor Module:**

I used the Image Processing Library of the C# to process the images that are uploaded by the user or are selected from his/her profile. I created the copy of Image Object by using the Clone Method of the Image Object thus using the prototype design pattern. User can select different variety of apply from like inversion of image, rotate image horizontally, greycaling the image, changing the contrast as well as the brightness of the image. User can also Undo and Redo the filters applied on the images. For this functionality I used the memento Design pattern to store the previous state of the Image Object.

**Image Handling Module:**

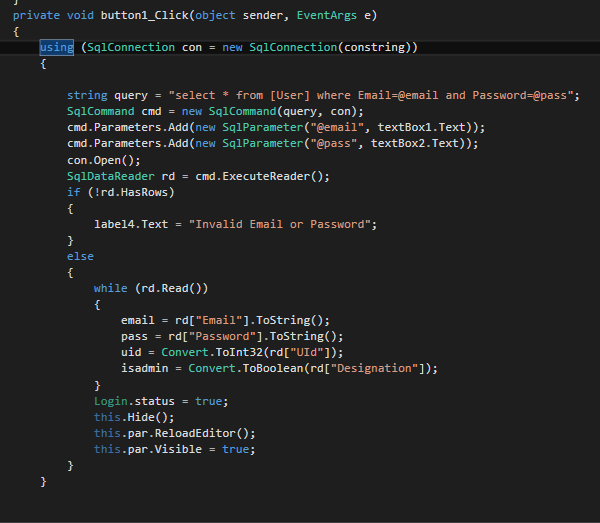
User can either upload image that he/she wants to edit using the upload image button. User can download the edited image using the download image button. User can save the image to his/her profile if he/she is logged in.

**Database Module:**

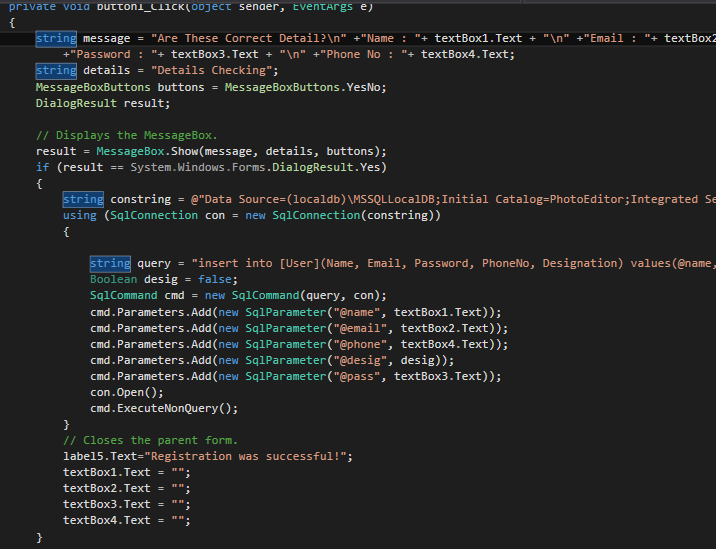
Whenever a user registers his/her self to the system the database of the users are updated. Whenever a user tries to login database is checked if the user is present in the system. Whenever Admin removes a user from the system, the database of the user is updated.

**Function Prototype:**

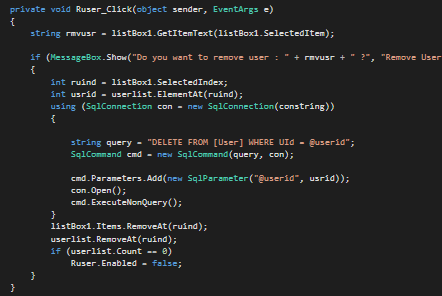
**User Login Functionality:**

****

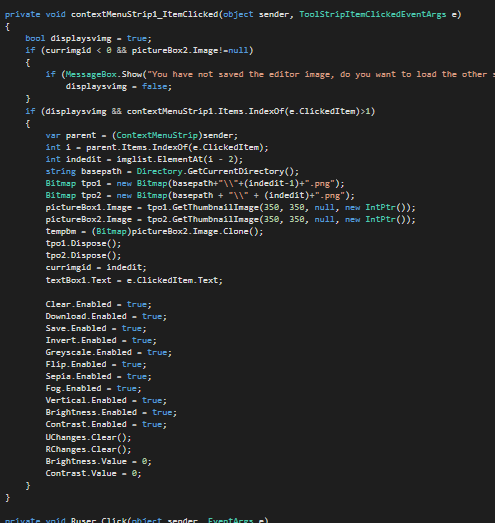
**User Registration Functionality :**

****

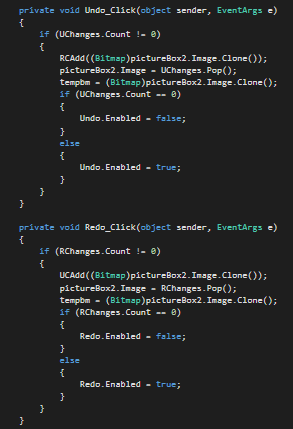
**Remove User Functionality :**

****

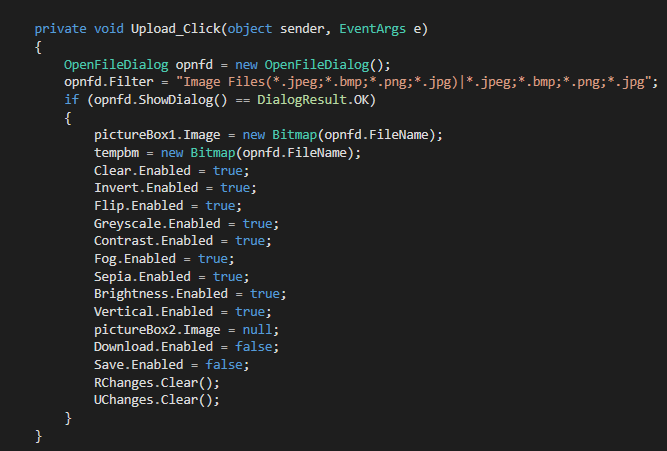
**Display Saved Image :**

****

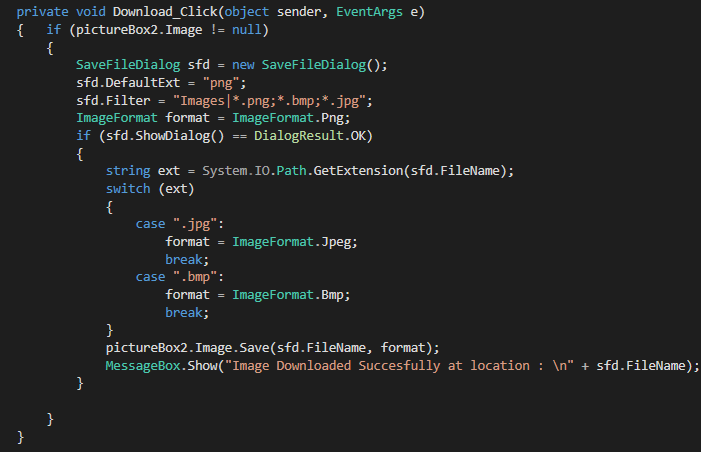
**Undo and Redo Functionality :**

****

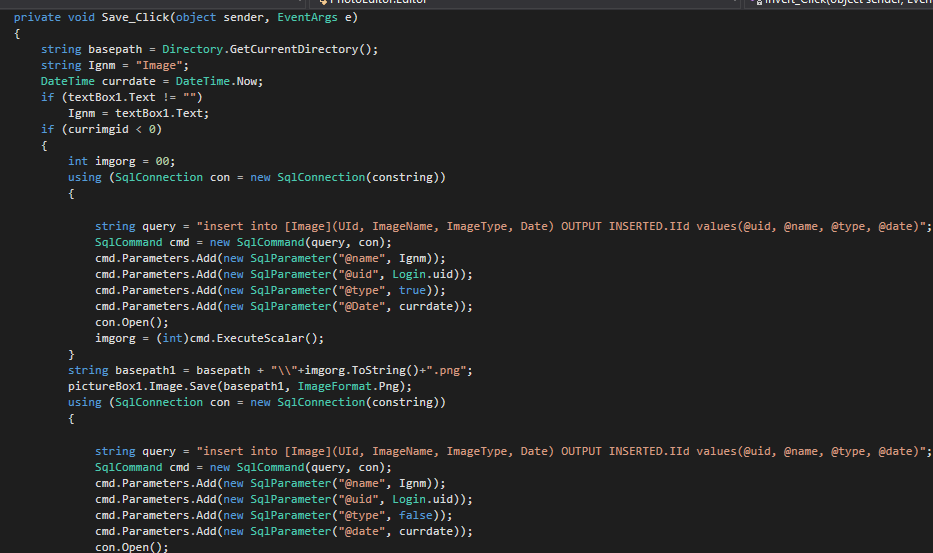
**Upload Image Functionality :**

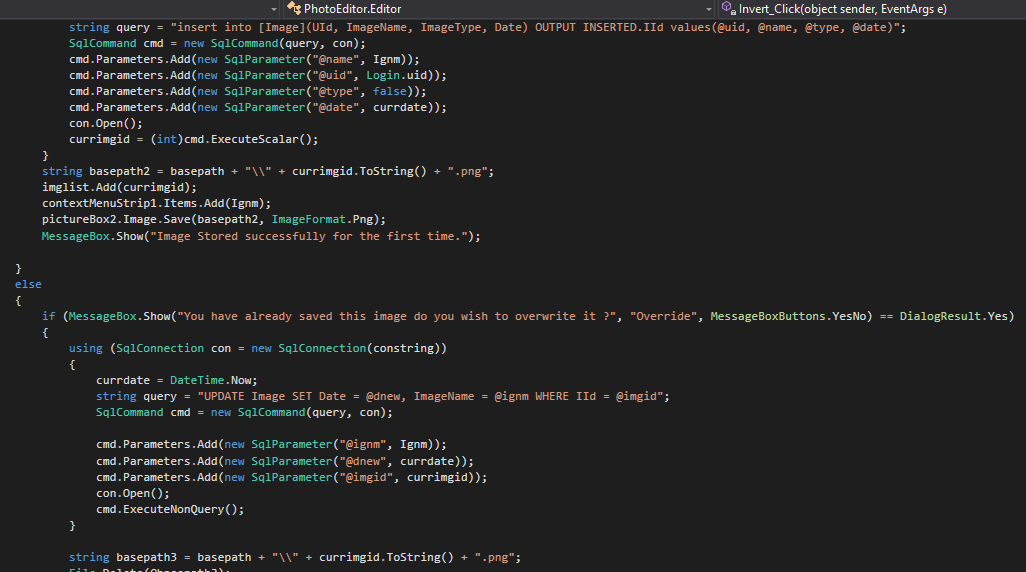
****

**Download Image Functionality :**

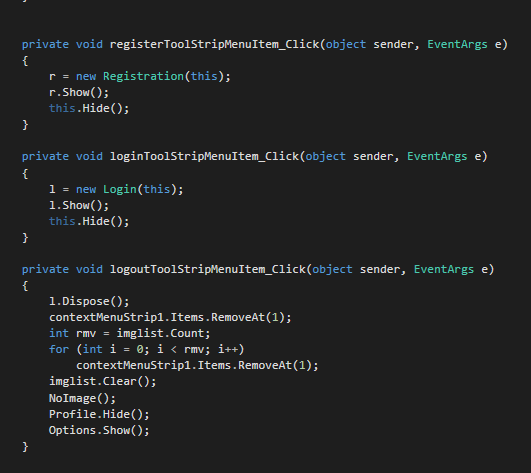
****

**Save Image Functionality :**

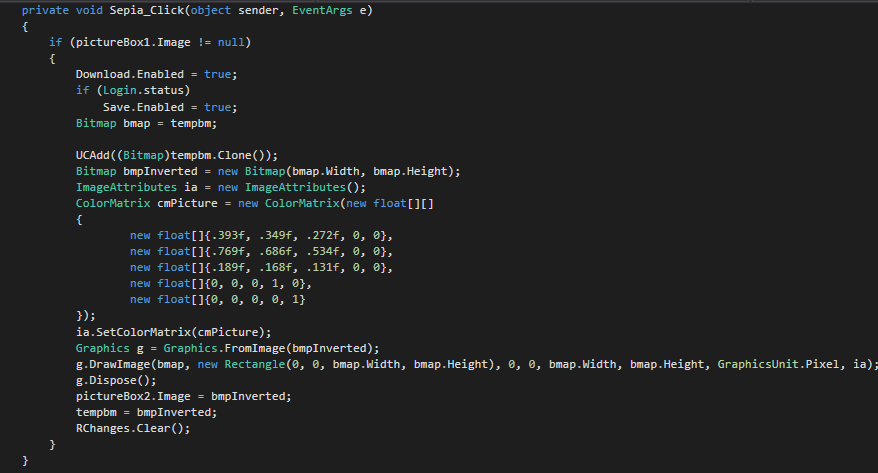
****

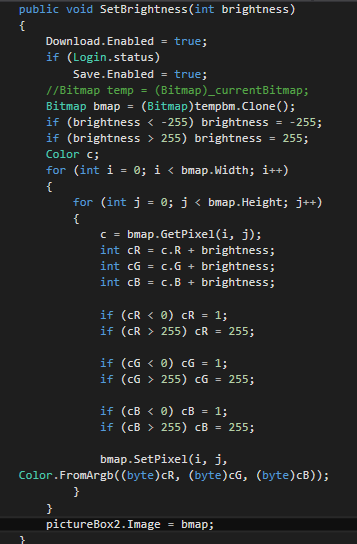
****

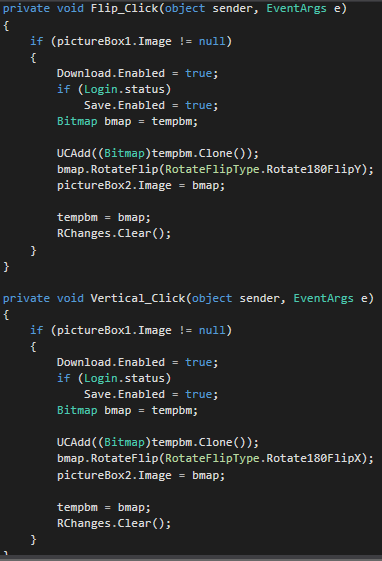
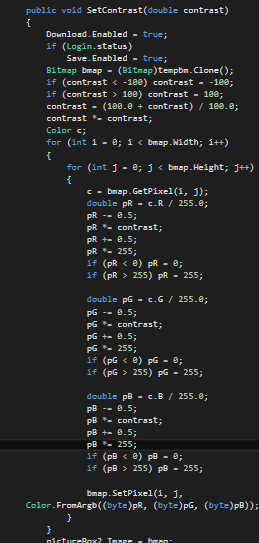
**Login , Registration and Logout Functionality for editor :**

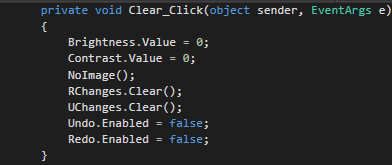
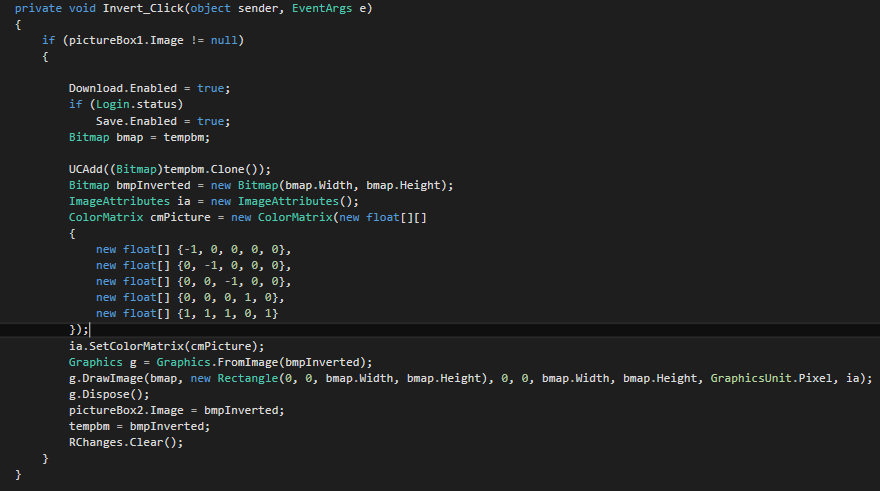
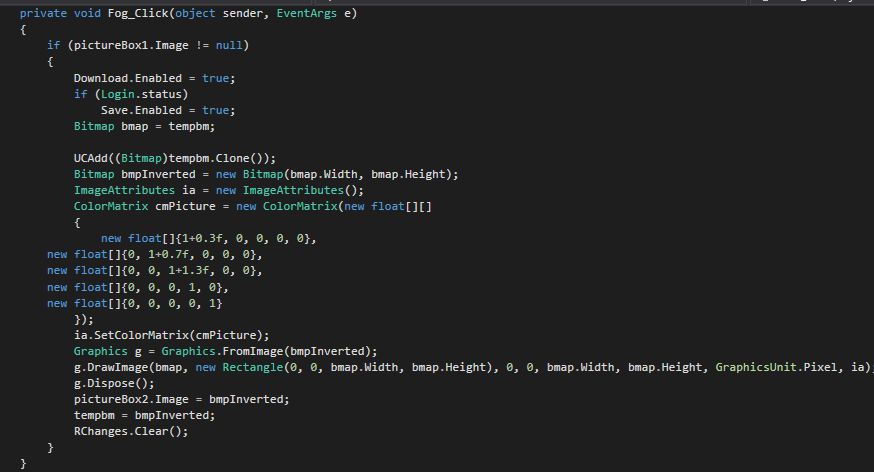
****

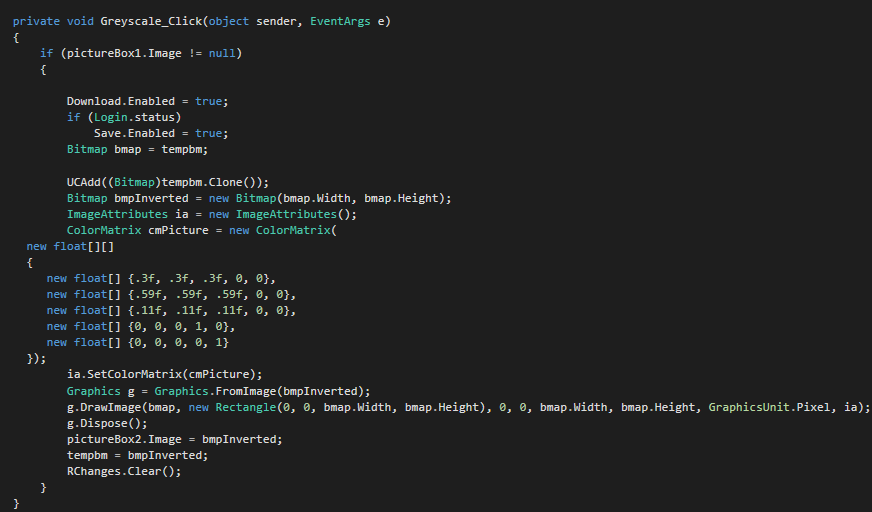
**Different Filters :**

****

****

****

****

****

**Testing Details**

**Testing Methods:**

**Sign In Testing :**

Username : abc Username : abc

Password : 123 Password : abc

Output : Invalid Output : Valid

**Registration Testing :**

Name : abcd

\*Email : [abc@gmail.com](mailto:abc@gmail.com)

\*Password : 123abc

\*Phone No. : 1234567890

Output : Registration Successful.

If any of the \* fields is empty :

Output : Registration Failed.

**Upload Testing :**

Input : Valid Image File Path

Output : Picture Displayed in the PictureBox

**Download Testing :**

Input : Valid Destination Path with Valid Image format

Output : Picture saved successfully and MessageBox displayed.

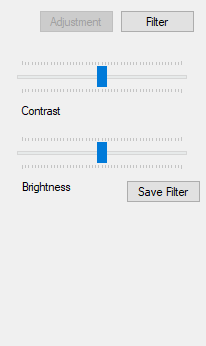
**Save Testing :**

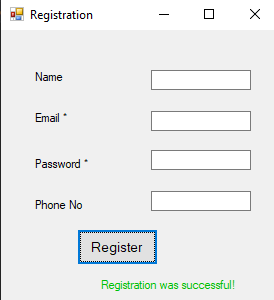
Input : New Image in the PictureBox

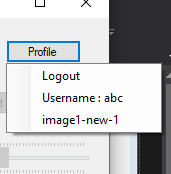
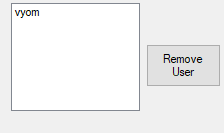
Output : Image stored in the server as well as the path stored in the database and MessageBox displayed.

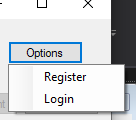
Input : Overriding already saved image in the PictureBox

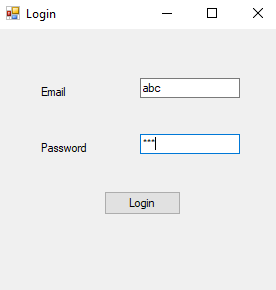
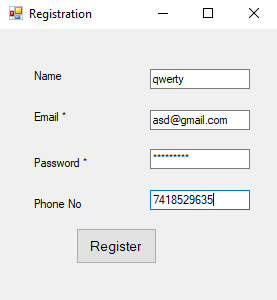
Output : Image name overridden if changed in the database as well as time of the image updated and MessageBox displayed.

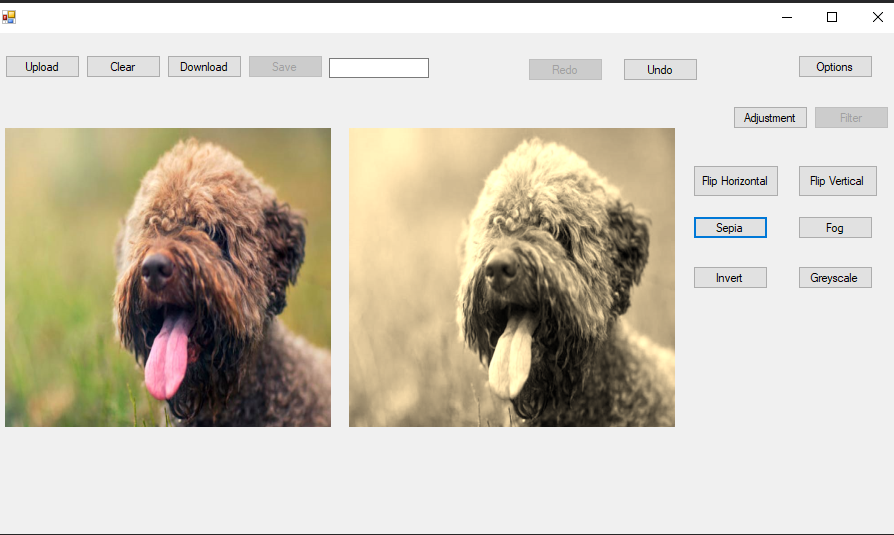
**Screen-Shots**

****

****

****

****

****

**Conclusion**

Isuccessfully implemented the user registration functionality as well as the user login functionality. I also successfully implemented the image upload functionality as well as image download functionality. I also successfully implemented the image editing functionality using the C# in-built Image Processing Library. I was also able to implement the Image saved functionality that saved images can be re-edited for a particular image. Admin side functionality to remove user was successfully implemented.

**Limitation and Future Extension**

**Limitations:**

* Undo and redo operation can only be done 5 times.
* User cannot download the saved images from her profile.
* User Images cannot be deleted from his profile.

**Future Extension:**

* More filters can be added to increase the image manipulation done by the user.
* Some styling/color can be added to the whole application looks too simple.
* User can remove his/her image from the profile.

**Bibliography**

**References:**

<https://stackoverflow.com/>

<https://www.youtube.com/>

<https://www.geeksforgeeks.org/>

<https://docs.microsoft.com/en-us/dotnet/>

<https://www.codeproject.com/>

<https://www.c-sharpcorner.com/>