FINAL SEMESTER ASSESSMENT (FSA) B.TECH. (CSE)
VI SEMESTER

UE18CS355 – OBJECT ORIENTED ANALYSIS AND DESIGN WITH SOFTWARE ENGINEERING LABORATORY

PROJECT REPORT

ON

PROJECT TITLE: Photo Editor

SUBMITTED BY

1) Hritik Saxena(PES2201800125)
2) Akshant Vedant(PES2201800132)
3) Prema Ramappa Hanchinal(PES2201800702)

JANUARY – MAY 2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RR CAMPUS,
BENGALURU – 560100, KARNATAKA, INDIA

TABLE OF CONTENTS

Photo Editor

ABSTRACT

The purpose of this Photo Editor Application is to capture, in natural language and at a functional level, the description and requirements of a photo editor. This is a functional description of those features required for a basic photo editor. This includes features such as cropping, rotating, zooming in/out, rotate, flip and differing contrast. The description for the requirements of the software can help improve the understanding of the system for a developer in a better fashion.

This system allows the user to take hard copy of the image using printer routines and allows the user to store screen image into the disk file using file format (JPG, PNG). Photo Editor is a general form pertains to the alteration and analysis of pictorial information. We find instances of photo editing occurring all the time in our daily lives. The objective of Photo Editor is to visually enhance or statistically evaluate some aspect of an image not readily apparent in its original form. There are various features provided by system to edit an existing image, which are as follows Image scaling includes zooming and shrinking images. We can use enlarging to zoom in on the art of an image for closer examination. Image shrinking is useful for saving disk space, fitting a large image into smaller display and pasting several images into one image of the same size. Image compression tool is an application,. The user will send images and according to the specification they will be modified. Image rotation tool is used to rotate the image by the specified angle. Resembling is used to increase the size of each pixel by a certain factor. We have used various filtering techniques like lightening, darkening,

We have developed a program, which can be used on compressed/uncompressed BMP, JPG, and PNG file formats to perform any of the above-mentioned functions. The product mimics some parts of the Google-owned application, Snapseed. As the intended application has only a handful of developers, a small set of functions based on priority of the Snapseed application are implemented.

Chapter 1

Software Requirements Specification
for
Photo Editor

1. Introduction

1.1 Purpose

The purpose of this document is to capture, in natural language and at a functional level, the description and requirements of a photo editor. This is a functional description of those features required for a basic photo editor. This includes features such as cropping, rotating, zooming in/out, adding text, color correction and differing contrast. The description for the requirements of the software can help improve the understanding of the system for a developer in a better fashion.

1.2 Intended Audience

This document is intended for design and project developers who are looking to work on the Photo Editing Application. Understanding the document in a chronological order will be very beneficial to the reader. The document starts with a brief introduction of the product and dives into the functionalities and features moving forward.

1.3 Product Scope

The system will be user-friendly and hence highly beneficial to the people working with a lot of pictures on a daily basis. They could be photographers, graphic designers, or even people who are highly active on social media. There are many problems that can be fixed or elements that can be enhanced with a photo editing software. The user will be able to modify the image in terms of size, angle and contrast. Functionalities such as adding text and color correction can also be done by the user.

1.4 Document Conventions

LI Load the image
CI Crop Image
RI Rotate Image
CRC Color Correction
CSC Contrast Correction
PNG Portable Network Graphics
OS Operating System
ZIA Zoom In/Out

1.5 References

1. IEEE 830 Template
2. The Photo Editor Application - Chapter 5
3. IEEE Recommended Practice for Software Requirements Specifications IEEE Computer Society, 1998

2. Overall Description

2.1 Product Perspective

An image is an artifact that depicts visual perception, such as a photograph or a two-dimensional picture, that resembles a subject—usually a physical object—and thus provides a depiction of it. A photo editor takes in an image and performs user-specific functions on it to transform it into how the user perceives it to be viewed. The product mimics some parts of the Google-owned application, Snapseed. As the intended application has only a handful of developers, a small set of functions based on priority of the Snapseed application are implemented.

2.2 Product Functions

There are essentially six functions(CSCI's) that the user can perform after loading the application :

1. Crop Image (**CI**): The image, after being loaded into the application, can be cropped, which essentially corresponds to changing the aspect ratio (dynamic change), done using the height and width bars provided on selecting the "Crop" option.

2. Rotate Image(**RI**) : The image, after being loaded into the application, can be rotated, around a fixed axis. The rotating function can be used by moving the vertical bar which changes the orientation of the image in question, on selecting the "Rotate" option.

3. Color Correction (**CRC**): The image, after being loaded into the application, can be color-corrected, meaning the image can be passed through filters, which in turn changes the theme/color of light, reflection and shadow light emission seen by the user.

4. Contrast Correction (**CSC**): The image, after being loaded into the application, can be contrast-corrected, wherein the image brightness can be enhanced or decreased based on the movement of the vertical bar provided to the user. White balancing is an integral part of this function.

5. Zoom in/out(**ZIA**): Zooming refers to enlarging or reducing into the pixels of an image. This can be done using the necessary buttons(zoom in, zoom out), and the necessary changes must be seen on the input image.

*The Use Case Diagram (**Figure 4.1**) summarizes all the features provided by the application.*

2.3 User Classes and Characteristics

No special knowledge or skills shall be assumed on the part of the users. Users shall not be expected to learn a set of commands in order to start using the application. Users shall not be expected to remember a list of commands while using the software – these shall be provided via menus, tool palettes. Users shall not be protected from data loss, and must download the image after the necessary edit operations are applied.

2.4 Operating Environment

A Java application / Android application is currently in the works, which will work on any JDK-installed/Android platform. Any type of image input file, can be loaded into the application and the output file that can be downloaded is of only one type, PNG.

2.5 Design and Implementation Constraints

There are currently two options that are considered for implementing the application :

1. Android Studio : Meet Android Studio | Android Developers
2. Java GUI : Java Platform Standard Edition 8 Documentation (oracle.com)

There are no specific constraints on the developer otherwise.

2.6 Assumptions and Dependencies

It is assumed that the requirements described in this document have the same levels of priority. Although no specific priorities have been documented, it is assumed that software developers may need to scrub priority requirements in the face of schedule and resource pressures. Requirements should only be scrubbed by agreement with all the developers, and only if it can be clearly demonstrated that there is no other high priority (non-scrubbed) requirement that depends upon the scrubbed requirements. No other special assumptions or dependencies have been identified.

3. External Interface Requirements

3.1 User Interfaces

The interaction with the user is through the GUI. The interaction is mainly through the use of button clicks and drags. The interface will have a blank slate before the user loads in the image and until the user decides to exit out of the application, the image shall remain in point of view, with the exception when the user removes/replaces the image that is currently being worked on. On downloading the image after editing, the blank slate returns for further images to be loaded and worked on, or until the user closes the application.

3.2 Software Interfaces

The application uses the file system residing in the **OS,** for looking after the downloaded image/loading the image that is residing in the file system to the application. No guarantees are given by the software for security assurances with regards to saving the loaded image during/after a security breach.

3.3 Communications Interfaces

The application will not interact with any other application/over the web transfers and will solely reside on the edge where it has been installed on. No other software/application interaction is a guarantee.

3.4 Hardware Interfaces

On the client side, a device that runs Android OS will be required. The device will contain the application installed on it for the usage.

4. Analysis Models

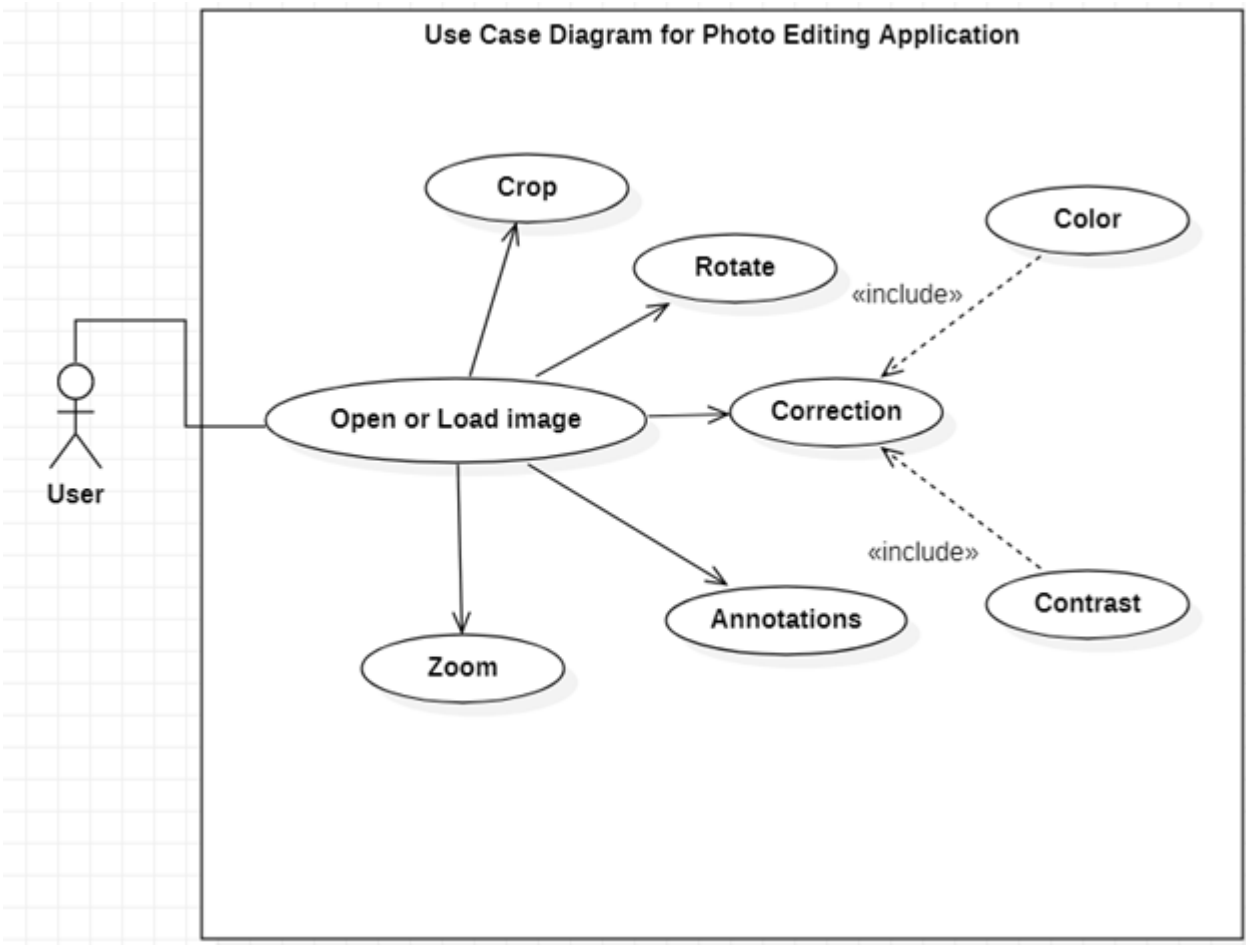


Figure 4.1. *Use Case Diagram*

5. System Features

The specifications of the system features with their priorities and sequences are mentioned below:

5.1 Load Image

Loading the image to the application.

5.1.1 Description and Priority

The first feature of the application is to get the required image from the user. This is done using button clicks to find the required file in the current system the application is running on and loading it on to the application. This is the most important feature of the application as all the other system features depend on this for their working, therefore this is a high priority feature.

Benefit (9) – Almost all of the features in the application depend on it.

Penalty (9)– Without the feature the application wouldn't run, therefore this will incur the highest penalty.

Cost (5) – A relatively low cost is assigned as it doesn't take much time to be implemented.

Risk (8) – There is a lot of risk in finishing the application without this feature.

5.1.2 Stimulus/Response Sequences

The user has only one button to activate this feature and on-click, a window pane with options to navigate through the file system to find the image to be worked on is seen, and the final conclusion of this sequence ends when the image is loaded into the application.

5.1.3 Functional Requirements

The software capabilities that are needed for this feature to work are: the underlying OS must be Windows/Mac or Android, a file system support, user interactive features such as GUI, a pointer that is useful for button clicks/touch input for Android.

- get_input(image_input): The image input must be taken in from the user. The input may or may not necessarily be an image. The handling of such errors which keeps the application in a valid state must be given priority.

- error_on_invalid_entry(invalid_input): The application must respond to an invalid input after the window pane that is used to select the image is done with its job. The error can be a prompt or a message entry at one of the text boxes. The user has to understand that there is an error and must act on it to rectify it. Therefore, the message must be clear and loud and should take a substantial amount of space in the available frame.

- On_success(image_input): On a successful load of the image, necessary features open up to the user to act on editing the image. These features must not be available to the user before an image is selected. Any more movement in selecting the image must delay these features being shown on the frame.

5.2.3 Functional Requirements

The software capabilities that are needed for this feature to work are: the underlying OS must be Windows/Mac or Android, a file system support, user interactive features such as GUI, a pointer that is useful for button clicks/touch input for android.

- crop_image(height,width): The application must respond to the crop image option being picked and must concur with the necessary features mentioned in section 4.1.2.

- out_of_bounds(height,width): The height and the width inputs are limited to a specific size(TBD) and any out of bounds entry must be dealt with a prompt of the error message stating the same.

- on_complete(final_image): On a successful attempt of the required operation to be done on the image, the user now can select any other feature to make the necessary changes.

NOTE : All the other features(buttons) must be hidden when a feature is already selected. This ensures that the image is always in a valid state on jumping from one feature to another.

5.3 Rotate

Color correction with color grading on applied to an image.

5.3.1 Description and Priority

This feature is available to the user right after the load image feature gives a valid output. The rotate image option is given by two buttons (one for left, one for right). On selecting the “Rotate” option, the two buttons appear on the UI and the user can use them to flip the input images. (360 degree rotation).

Benefit (7) – An important feature therefore a high benefit value is specified.

Penalty (8)– A high penalty based on the benefit.

Cost (6) – A moderate cost as not a lot of time needs to be spent to implement this feature.

Risk (6) – There is a moderate amount of risk in implementing this feature.

5.3.2 Stimulus/Response Sequences

Rotation is an important aspect of augmentation and generally almost all the editors implement this feature. The user upon selecting the “Rotate” option, has two choices, either rotate left, or rotate right. These choices are made using button clicks and the necessary change is seen in the input image. An “exit” option is used to go back to the main list of features after using this feature.

5.3.3 Functional Requirements

The software capabilities that are needed for this feature to work are: the underlying OS must be Windows/Mac or Android, a file system support, user interactive features such as GUI, a pointer that is useful for button clicks/touch input for android.

- rotate(button_click): On selecting the rotate option, the user is provided with two buttons, which indicate their features through their labels.
- rotate_left(augment_image): Rotate the image 90 degrees to the left. Necessary changes must be seen in the window pane containing the input image.
- rotate_right(augment_image): Rotate the image 90 degrees to the right. Necessary changes must be seen in the window pane containing the input image.
- on_exit(out_of_rotate): On clicking this button, the input image holds the changes made and exits out of **RI**. All the other features will now be available.

NOTE : All the other features(buttons) must be hidden when a feature is already selected. This ensures that the image is always in a valid state on jumping from one feature to another.

5.4.3 Functional Requirements

The software capabilities that are needed for this feature to work are: the underlying OS must be Windows/Mac or Android, a file system support, user interactive features such as GUI, a pointer that is useful for button clicks/touch input for android.

- correction(button_click): On selecting this option, two different options can be seen by the user, “Color” and “Correction”.

- color_correction(coordinates): The color correction feature will have built in features and any necessary errors are handled by the API calls made. The three way color corrector is seen on selecting the color correction button.

- on_apply(changed_image): After the apply changes button is selected, the necessary changes are seen on the input image and all the set of features are once again available to the user. Note that after the changes are applied, the application would go back to a valid state and all of the options, correction -> color, must be selected again, for making more changes.

NOTE : All the other features(buttons) must be hidden when a feature is already selected. This ensures that the image is always in a valid state on jumping from one feature to another.

5.5 Contrast Correction

Contrast correction with white balancing on applied to an image

.

5.5.1 Description and Priority

This feature is available to the user right after the load image feature gives a valid output. The contrast correction option is generally used to white balance the input image. Currently, a two way color corrector (shades of white to black) is being considered to be an option. The drag option lets the user pick up the white balance.

Benefit (3) – A low benefit value is given to it, as it does not specially entail in a photo editor.

Penalty (3)– A moderate penalty based on the benefit.

Cost (7) – This has a high cost, as implementation will take a huge chuck of development time.

Risk (6) – There is an ample amount of risk in taking up this feature, as it may lead to a moderate amount of development time being wasted on it.

5.5.2 Stimulus/Response Sequences

White balancing is considered to be an important feature in the working of a photo editing application. The user will have to select the correction button and then would be given two options, namely, “Color” and “Contrast”. On picking the “Contrast” option, a window pane with the two way corrector window would appear and the user would make the necessary white balance adjustments. After the said adjustments are done, the “Apply” button is selected and the image would reflect the necessary changes that were made.

5.5.3 Functional Requirements

The software capabilities that are needed for this feature to work are: the underlying OS must be Windows/Mac or Android, a file system support, user interactive features such as GUI, a pointer that is useful for button clicks/touch input for android.

- correction(button_click): On selecting this option, two different options can be seen by the user, “Color” and “Correction”.

- contrast_correction(coordinates): The contrast correction feature will have built in features and any necessary errors are handled by the API calls made. The two way color corrector is seen on selecting the contrast correction button.

- on_apply(changed_image): After the apply changes button is selected, the necessary changes are seen on the input image and all the set of features are once again available to the user. Note that after the changes are applied, the application would go back to a valid state and all of the options, correction -> contrast, must be selected again, for making more changes.

NOTE : All the other features(buttons) must be hidden when a feature is already selected. This ensures that the image is always in a valid state on jumping from one feature to another.

.

.

Functional Requirements

The software capabilities that are needed for this feature to work are: the underlying OS must be Windows/Mac or Android, a file system support, user interactive features such as GUI, a pointer that is useful for button clicks/touch input for android.

- on_edit(change_text): After a button click is received from placing the box at the desired place, the user can now edit the text in the text box.

- on_click(window_pane): The final click will ensure that the text is placed on the image at the specified place. This moves the application into a valid state.

NOTE : All the other features(buttons) must be hidden when a feature is already selected. This ensures that the image is always in a valid state on jumping from one feature to another.

5.7 Zoom In/Out

Writing text onto the input image.

5.7.1 Description and Priority

This feature is available to the user right after the load image feature gives a valid output. The zoom feature helps in enlarging or reducing the size of the input image. This is done with the help of two buttons (**ZIA**), which are available for selection after the “Zoom” button is selected. Necessary labels indicate the functions of the two buttons.

Benefit (7) – A healthy benefit value is given as most editors in the market implement this feature.

Penalty (6)– A high penalty is incurred on failing to finish the feature.

Cost (3) – Zoom In/Out is a low cost, high priority feature.

Risk (5) – A moderate amount of risk is involved in implementation of this feature.

5.7.2 Stimulus/Response Sequences

The user will have to select the “Zoom” button and upon selecting this button, two new buttons appear on the UI. The two buttons are used for enlarging and reducing the input image size. A set limit is applied to both enlarge and reduce, where exceeding this limit, won’t change anything on the window pane. Note that there will be no prompt when the limit is exceeded. Note that to exit out of **ZIA**, there is an exit button which can be seen probably on the top right of the window pane.

5.7.3 Functional Requirements

The software capabilities that are needed for this feature to work are: the underlying OS must be Windows/Mac or Android, a file system support, user interactive features such as GUI, a pointer that is useful for button clicks/touch input for android.

- zoom(input_image): A button click will result in two new buttons appearing on the UI, which must be self-explanatory (based on the button label).

- on_click_enlarge(input_image_size): The image size must increase with this button click. A set limit is specified and on exceeding this limit the button won’t gauge any input from the user.

- on_click_reduce(input_image_size): The image size must decrease with this button click. A set limit is specified and on exceeding this limit the button won’t gauge any input from the user.

- on_exit(out_of_zoom): On clicking this button, the input image holds the changes made and exits out of **ZIA**. All the other features will now be available.

NOTE : All the other features(buttons) must be hidden when a feature is already selected. This ensures that the image is always in a valid state on jumping from one feature to another.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

An approximation can be made that the output image will be available in 30 seconds or under. Functionalities such as crop and rotate will require a lesser time period for processing as compared to the other functions. The system will be able to handle all images uploaded in any of the standard image formats (.jpeg, .png, .eps etc.).

6.2 Safety Requirements

No safety requirements have been identified.

6.3 Security Requirements

There are no security concerns with respect to the usage of the system as there will not be any record of the user or the image uploaded on the system side. The client will have an option to download the filtered image.

6.4 Software Quality Attributes

The final product will be an Android application and hence will hold all the generic features of an application.

- Reliability : The mentioned features need to be satisfied in the product so as to be of the utmost usage to any user.
- Usability : Shallow learning curve as it only involves dragging, dropping and selecting options from the given menu.
- Portability : Since it is an application, it can be installed on any mobile device anywhere.
- Maintainability : Application will not require any high maintenance.
- Testability : Minimum testing of all the functionalities is necessary.
- As mentioned, the application will be restricted to Android OS devices only.

6.5 Business Rules

The application will be open to all for usage. There are no limits on the types of users and since there are no security loopholes in the application, anyone can use it for any purpose at any point in time. Premium services for a set of functionalities can be launched after a period of time.

Knowledge in the following domains are necessary :

- Application Development (Android application in specific)
- Photography
- Minimum color sense

Appendix A: Glossary

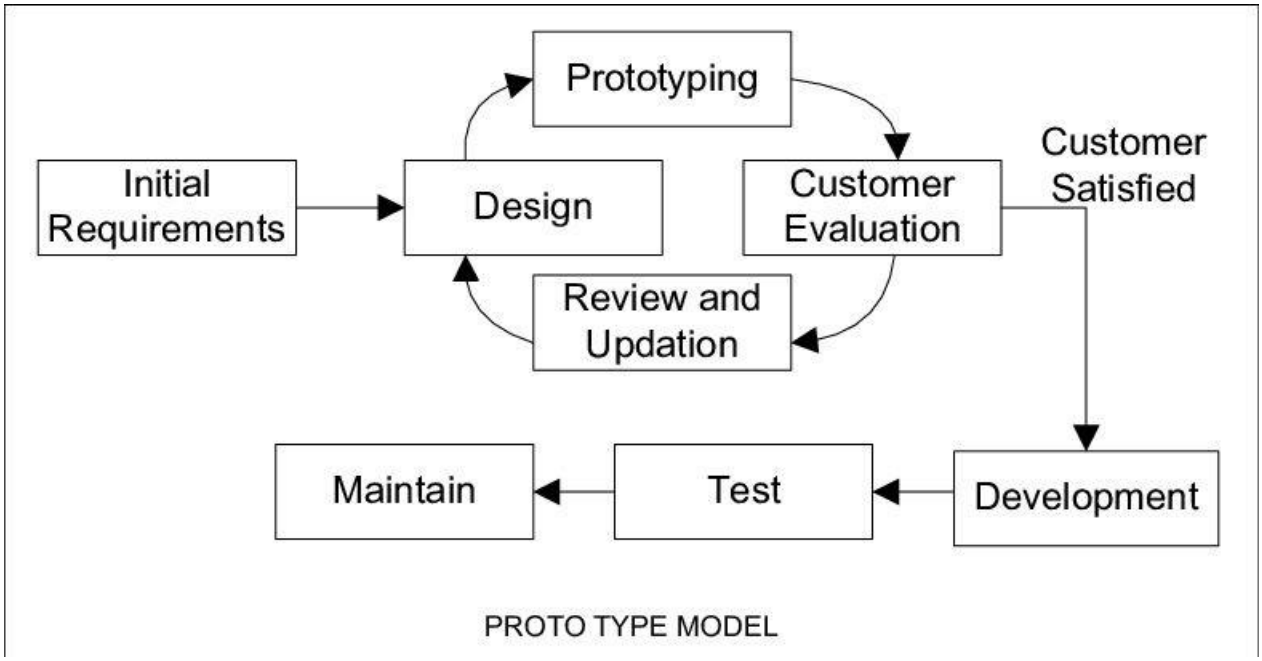
- GUI : Graphical User Interface
- OS : Operating System
- TBD : To Be Decided
- Android Studio : Unified environment to build applications for Android devices

Second Chapter

Project Plan

Lifecycle and Model

We propose a prototype model under the legacy Software Development Lifecycles (SDLC) for the photo editor project. This model is also known as a lightweight process model since it does not follow as much rigidity as the waterfall or the V-model. The flow diagram of a generalized prototype model can be seen below :



PROTO TYPE MODEL

The model involves gathering a few requirements at the beginning and forming a basic prototype with which the implementation can begin. After building a basic prototype, customers are allowed to evaluate, and their suggestions are taken to refine the requirements. This process is repeated until a final set of requirements is

listed down with which the final version of the product can be delivered.

The advantage of this mechanism is that the customers are involved throughout the process and hence a more user-friendly system can be built easily. This model also favours the developers in terms of freezing the requirements early and hence has an upper hand over the other models.

Since the GUI is an important aspect for a photo editor, this model will allow us to revisit the prototype from time to time which can help improve the entire system.

Tools Required

Planning Tool:

Visual Paradigm,Git/Gitlab

Planning is done in two stages, here, we've incorporated VP into Git, first by analyzing the requirements in VP and implementing these tasks by creating milestones and objectives through the use of issues in Git/Gitlab.

Design Tool:

StarUML, TeamGantt, Visual Paradigm

Star UML was used for constructing the Use Case Diagram and is going to be used for future UML diagrams that are needed for this project.

Version Control:

Git/Gitlab

A version control tool is used in this project due to its dependency on CI/CD as it deals with a prototype model.

Development Tool:

JavaScript-(NodeJS, Angular)

A web based application with OOP is one of the requirements stated for implementation. NodeJS and Angular, both offer easy integration for creating the system required.

Bug Tracking:

Git/Gitlab

Git provides an easy way of communication through issues and open for comments sections for bug tracking and error reporting.

Testing Tool:

Selenium

Currently, the testing tool that is considered to be ideal for this project is Selenium.

Deliverables

List :

- ***Software Requirement Specification***
- Design Document
- Project Plan
- Project Design
- Management Plan
- Testing
- Final Build
- Final Report

Build Components:

- Frontend UI : A skeleton structure is analyzed and built from scratch. References shall be specified wherever necessary.

- SRS/Design Document/Plans : All the necessary documents are written by the team members from scratch. Diagrams and Flowcharts that are drawn are on the basis of the given template.

- Code Integration : Integration and merging of code is done using a version control tool to maintain a smooth workflow and easy transfer of code.

- Scheduling/Milestones/Objectives/Deadlines : All the important sidelines with respect to the project are considered and implemented from scratch by the team members.

Documentation/Report :

A flowing document is passed around to be filled up by the team members which is used to maintain the overall documentation ofthe code written , difficulties faced and solutions to the same, and steps to be taken on encountering errors must be jotted down. A final report, that is user-friendly, is done on the project that includes the workflow of the whole application built.

Reuse Components:

- Customer Requirements : The list of the requirements such as cropping, rotating etc. of a customer with respect to a photo editor app can be reused.

- Proof of Concept : This system can be categorized as a doable one since there are many applications such as Snapseed, InShot, Prisma etc. in the market that serve as a proof of concept.

- Testing : Testing is done using a testing tool by feeding in the final deliverable, that is the application, into the testing software. Custom code is written wherever necessary.

- APIs: Integrated APIs are used for building up the different use cases wherever libraries are defined.

Reusable components mentioned use existing code, APIs whenever feasible and meet the requirements.

Work Breakdown Structure

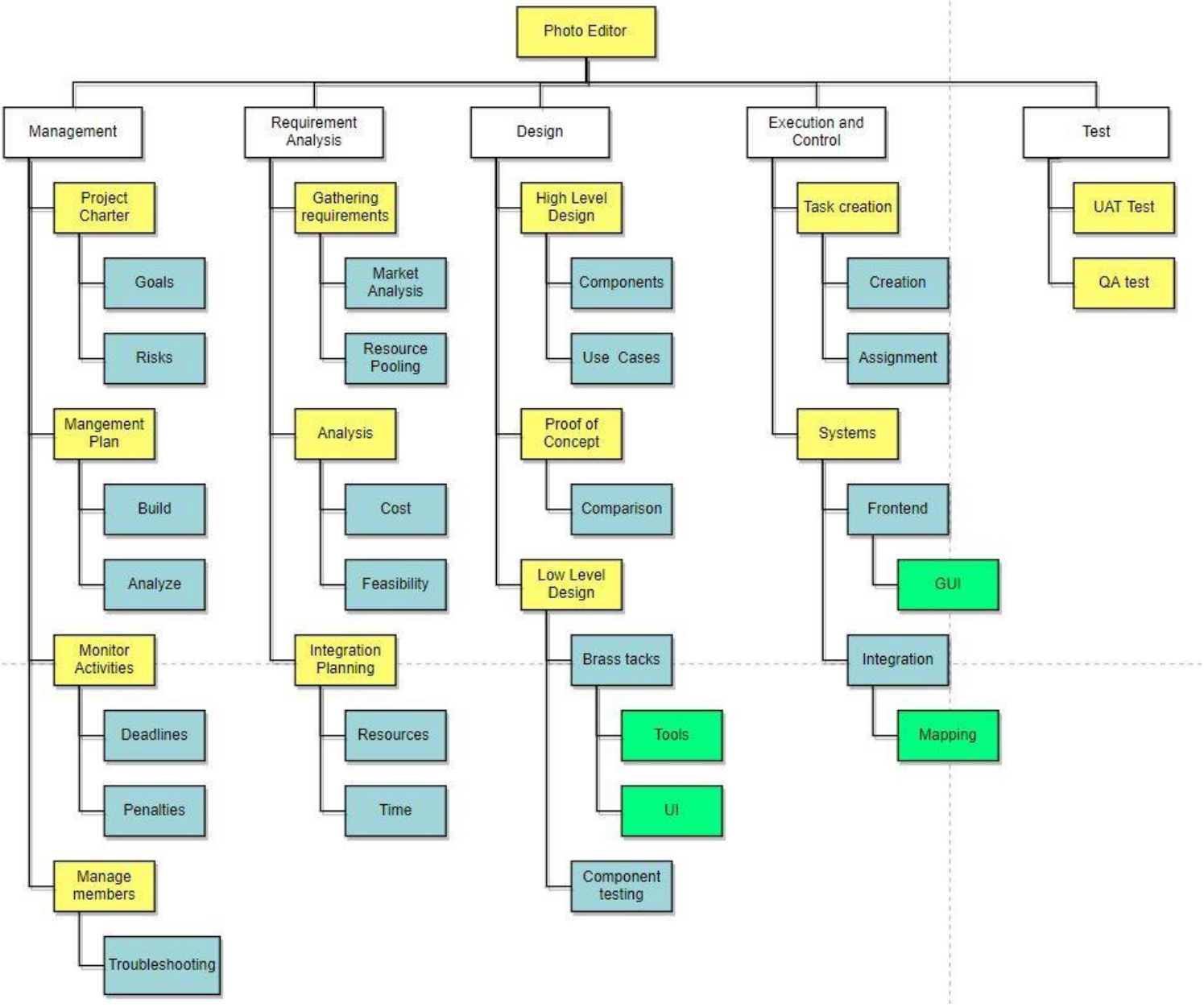


Photo Editor

1. Management

Project Charter

Goals

The goals of the project, the initial outline with customer fulfilment in mind, is put into action.

Risk

Initial risk with respect to cost, both in person and time, must be analysed before giving the go-ahead.

Management Plan

Build

Set up a management team to fulfil the goals set. Availability of members and time constraints must be accounted for. Note that the selection of members is not done here and just the availability is checked.

1.2.2 Analyse

In the analysis phase, final deductions about the constraints that are set are analysed and decisions to go-ahead are taken.

Monitor Activities

Deadlines

Soft deadlines are set and the management must assign project leads to start working on the application. Leads in turn assign members to work on different parts of the project, giving them various hard deadlines to meet.

1.3.2 Penalties

Any deadlines that are not met must be dealt with, by scrutinization/removal of members/lead off the project.

Monitor Activities

1.4.1 Troubleshooting

Any member or lead that faces problems in completing their assignment must be given sufficient time and help to fix/solve their problems.

2. Requirement Analysis

Gathering Requirements

Market Analysis

Analysis of market trends on the product being made, with future

demand and cost constraints
included.

Resource Pooling

Scouring of resources needed for finishing MVP (minimum viable product) and final integration.

Analysis

Cost

Cost analysis of the pooled resources and cost in terms of manpower and time must be calculated.

Feasibility

Feasibility must include further analysis of other constraints that are not included in cost requirement analysis. This may include scheduling, funding and other off-stream constraints.

Integration Planning

Resource

Resource integration involves combining effective resources at the right time. This may involve scheduling different integrations at different points in time.

Time

Time sharing of different planned schedules to meet various deadlines must be taken into account for smooth functioning of the project in a version control environment.

3. Design

High Level Design

Component

Component diagrams are structural diagrams that are used to model the static implementation view of a system. This involves modelling the physical aspects of the system.

Use Case

To understand the behavioral part of our system, a use case diagram is used to depict the various relationships between all the use cases, actors and systems in play.

Proof of Concept

Comparison

This involves a basic understanding of our system done in model analysis and comparing it with a similar product on the market for further changes to be made.

Low Level Design

Brass Tacks

Tools

All types of tools required to start working on the project is mentioned here. The most important tools that are necessary for the system to work are : planning tool, design tool, version control, development tool, bug tracking, testing tool.

UI

A skeleton structure must be done for a rough visualization of what the user will see and work with. This template will serve as a reference for the UI/UX designers.

Component Testing

Component testing must deal with feasibility of working of different components created. This indirectly refers to modulating the features of UI components in our project.

4. Execution and Control

Task Creation

Creation

Create tasks to meet deadlines. Usage of version control tools and timeline/deadline milestones to communicate the same to all the team members.

Assignment

Assign tasks to team members, which involves issue creation/role assignment with deadlines/milestones set.

Systems

Frontend

GUI

The GUI framework with the right elements and their workings must be made.
Developments must be made from the skeleton structure drawn in the Design Phase.

Integration

Mapping

Mapping of functionalities to UI components after coding up the features. Features are individually coded up and then integrated after testing their working.

5. Test

5.1 UAT Test

The goals formulated at the start of the project must be satisfied. As an evolutionary prototype model is used, customer evaluation is key for continuous updation of the initial and subsequent prototypes. UAT is done multiple times until the conditions set in the charter are met and the final deliverable is verifiable.

5.1 QA Test

The testing tools and customer feedback give out important results which determine the steps taken by the

prototype model to drive the system. Scrutinization is key at this stage as it forces changes to be made in response, which helps in building up the next best prototype.

Effort Required

The Constructive Cost Model aka Cocomo is a procedural cost effective model that is widely used for the approximation of effort and time required for building a project. The model wholly depends on the number of lines of code as its input.

We will use the basic Cocomo model for the calculations. Since the project involves a small team and has already been worked on previously as per literature, we can use the Organic system under Cocomo.

The calculation of effort for a basic organic Cocomo is given by :

$$E = a(KLOC)^b \quad \ldots \text{ eq. 1}$$

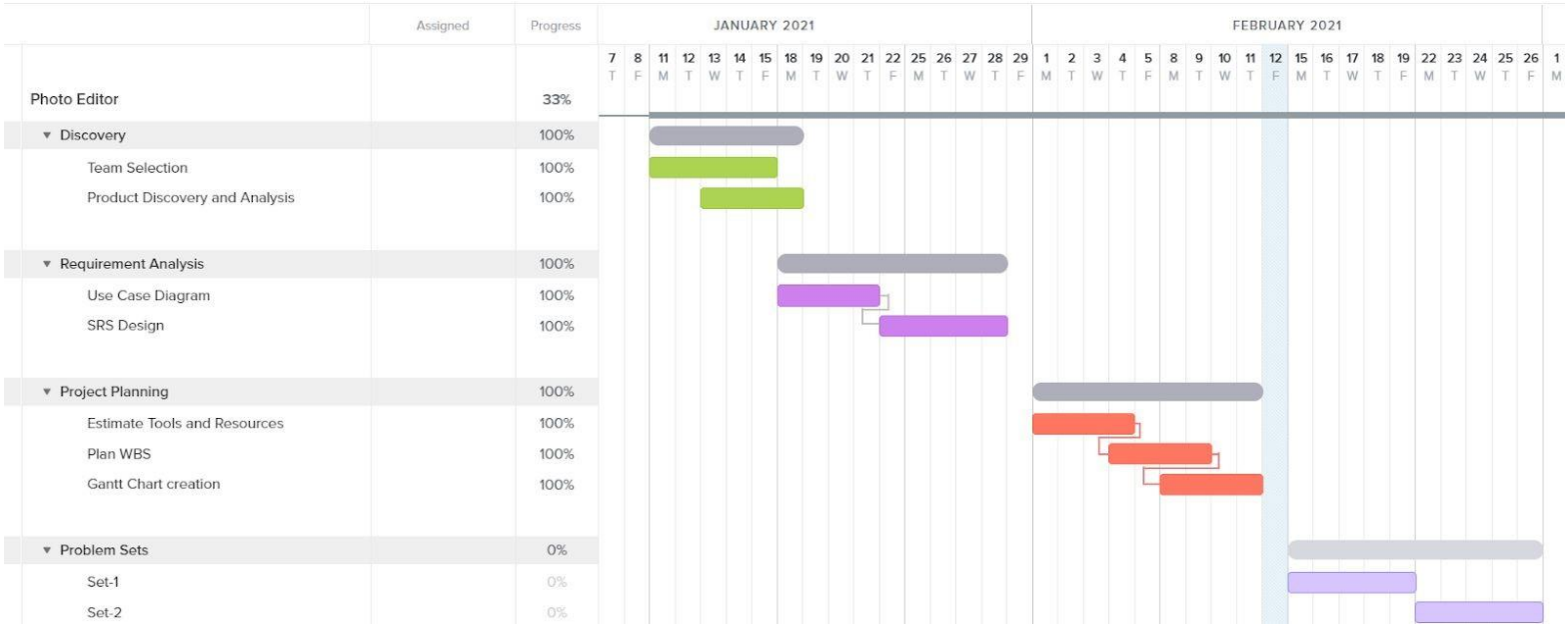
where KLOC is the number of lines of code in the scale 1 unit = 1000 lines and the values of a and b for a basic organic model are 2.4 and 1.05 respectively.

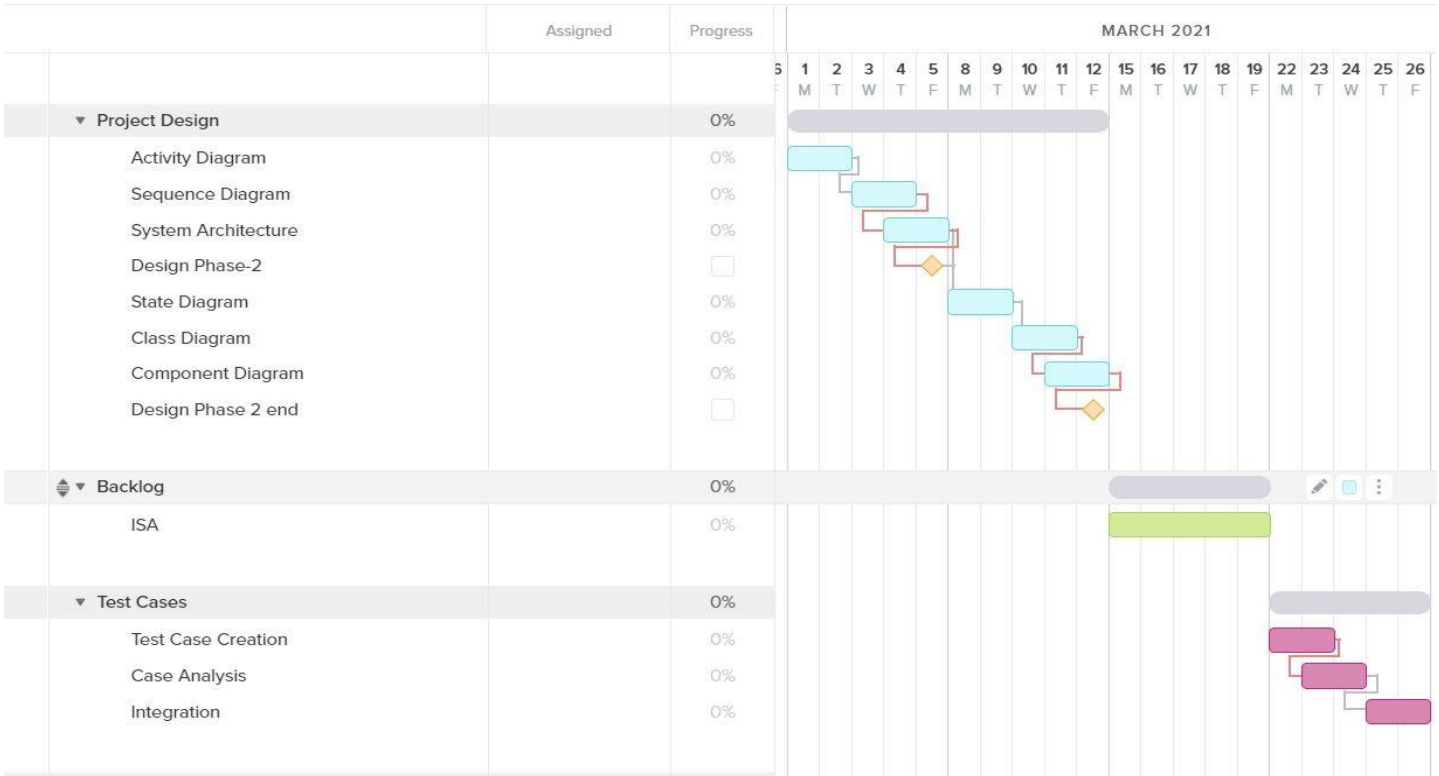
An approximation is made that this project will take 1200 lines of code. Substituting all the values in eq. 1, we get,

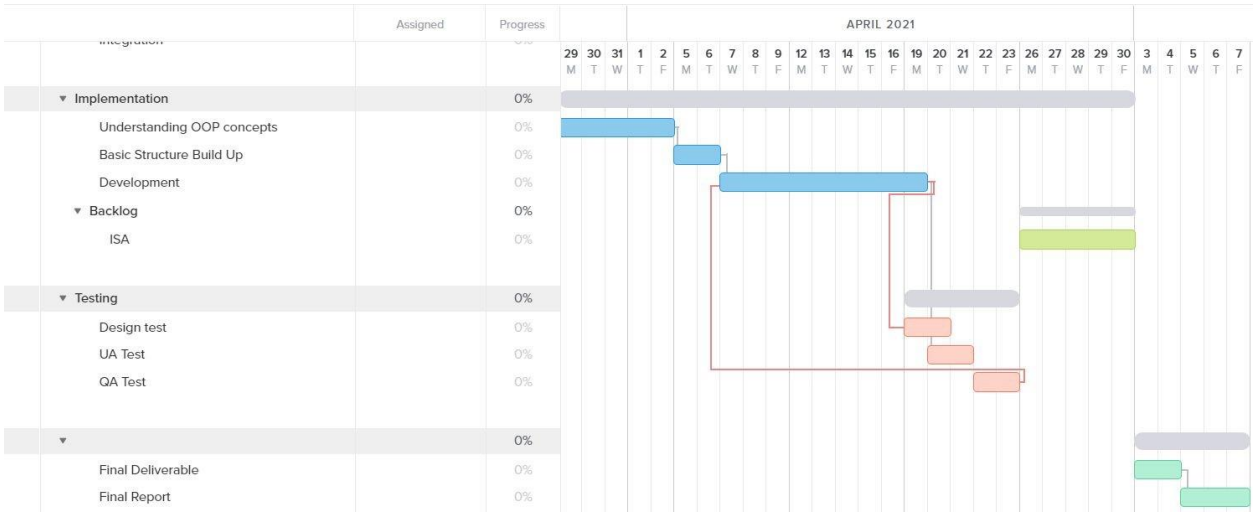
E = 2.90 Person-Months

Hence, the effort required is 2.90 PM

Gantt Chart for Scheduling


Assigned
Progress
JANUARY 2021
FEBRUARY 2021
Photo Editor 33%
Discovery 100%
Team Selection 100%
Product Discovery and Analysis 100%
Requirement Analysis 100%
Use Case Diagram 100%
SRS Design 100%
Project Planning 100%
Estimate Tools and Resources 100%
Plan WBS 100%
Gantt Chart creation 100%
Problem Sets 0%
Set-1 0%
Set-2 0%


Assigned
Progress
MARCH 2021
Project Design 0%
Activity Diagram 0%
Sequence Diagram 0%
System Architecture 0%
Design Phase-2
State Diagram 0%
Class Diagram 0%
Component Diagram 0%
Design Phase 2 end
Backlog 0%
ISA 0%
Test Cases 0%
Test Case Creation 0%
Case Analysis 0%
Integration 0%

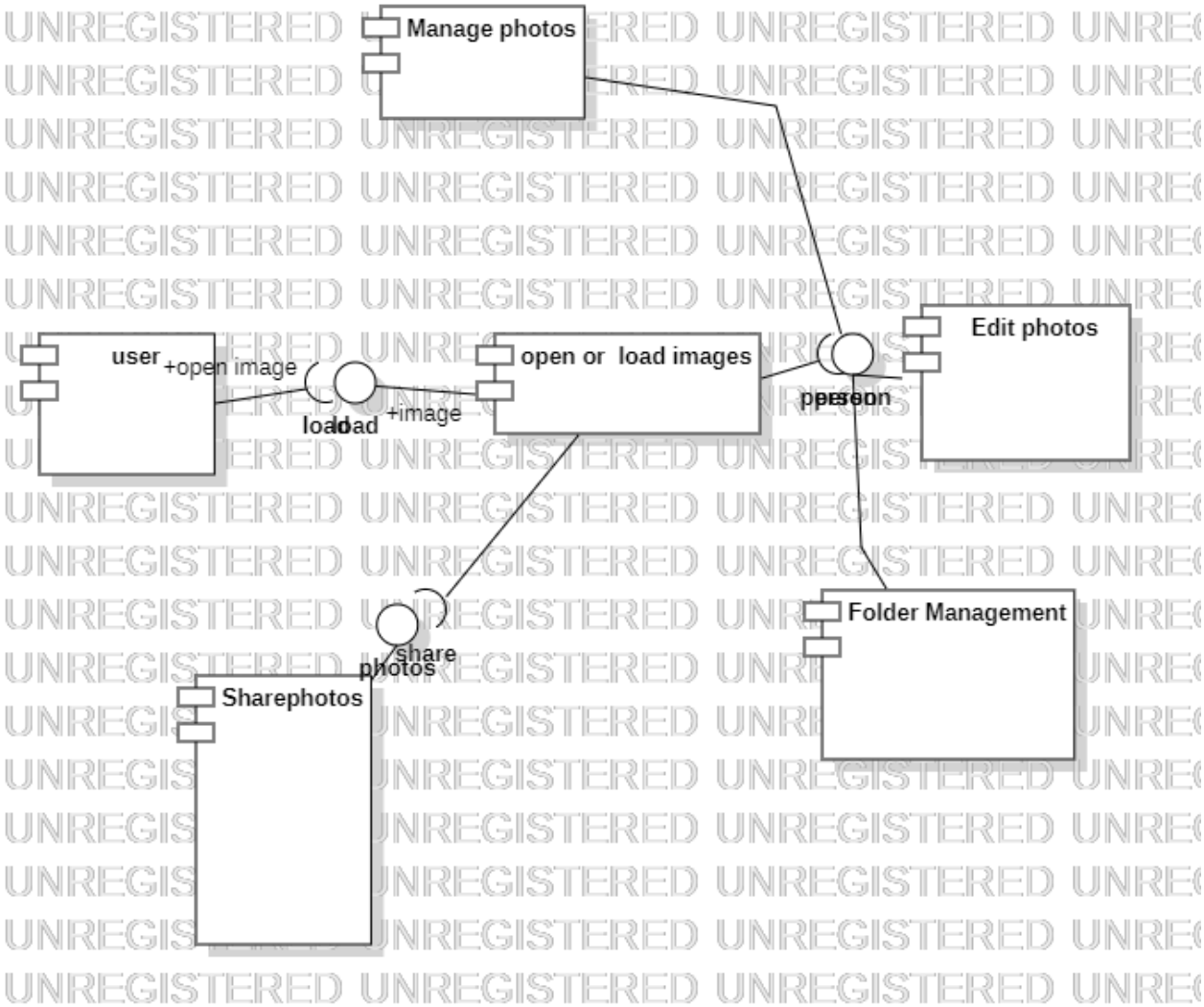


Third Chapter

Design Diagrams

Component Diagram for Photo Editing

Component diagram are often drawn to help model implementation details and double check that every aspect of the system's required function is covered by planned development.
In the diagram Component diagram of Photo editor below, each component is enclosed in a small box.
Which shows components, provided and required interface, ports, and relationship between the manage photos, edit photos, share and folder management

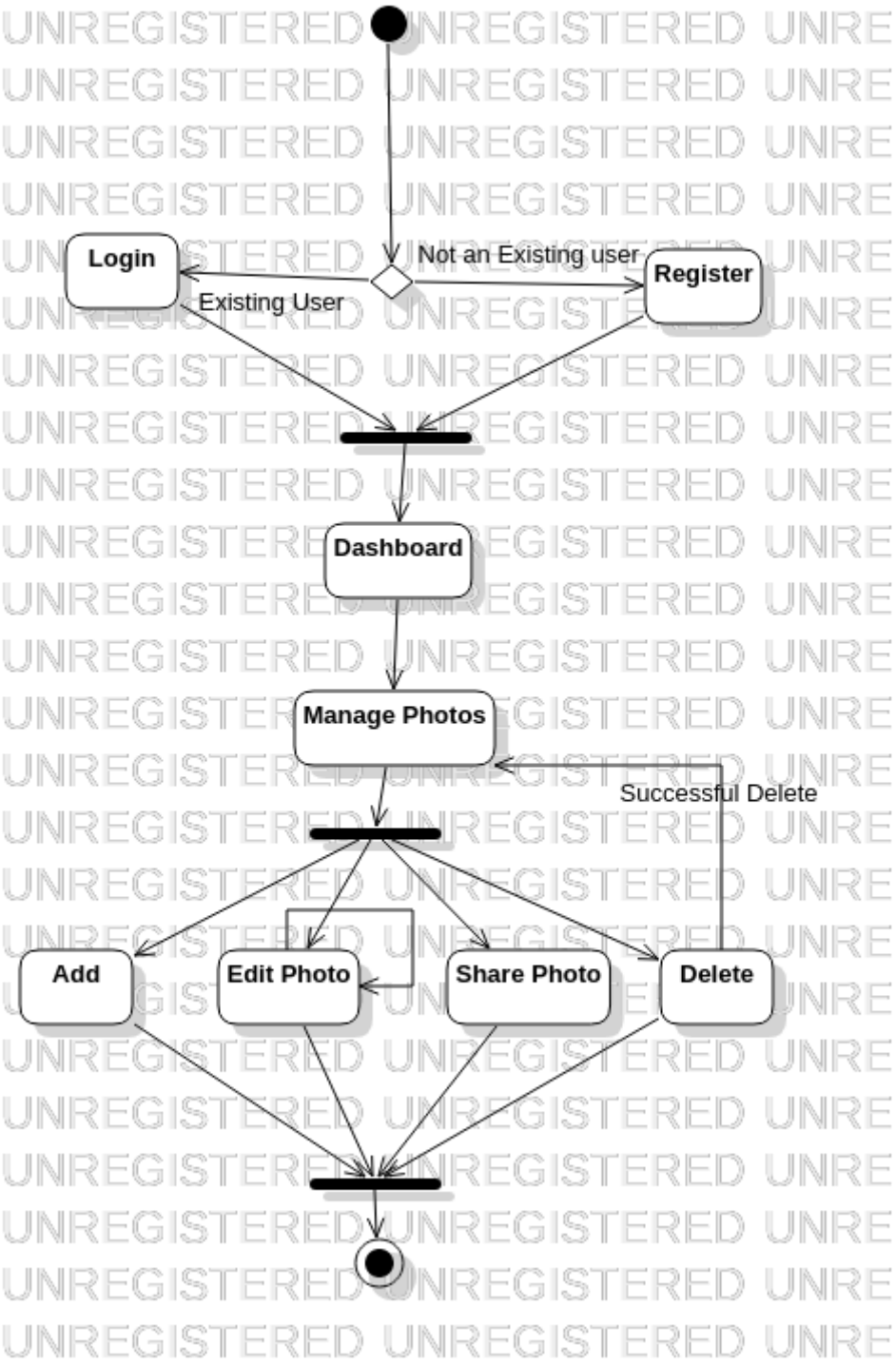


Activity Diagram

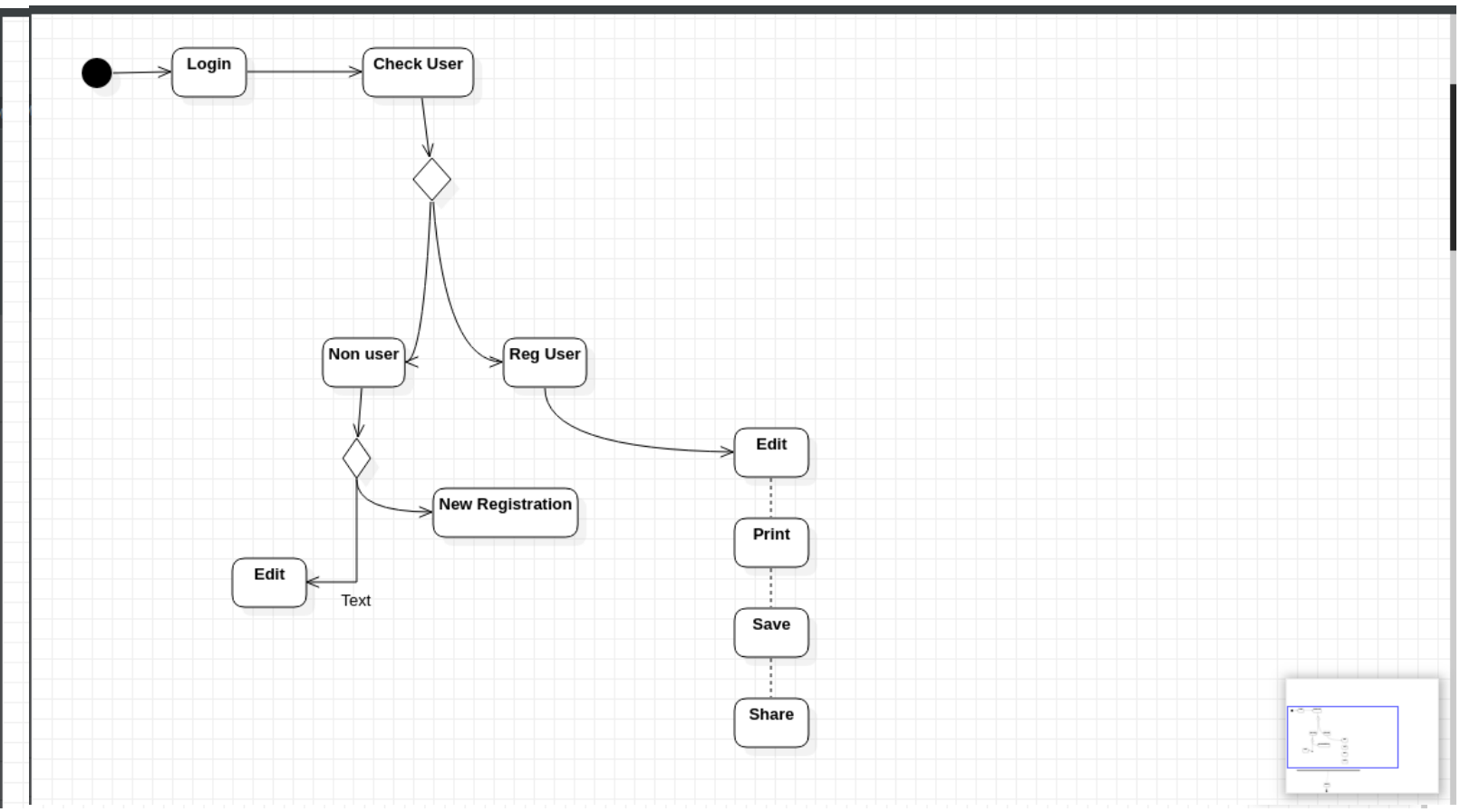
An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. Here possible activity states are, manage photos, edit, add, delete and Login. We begin with initial state

Login in

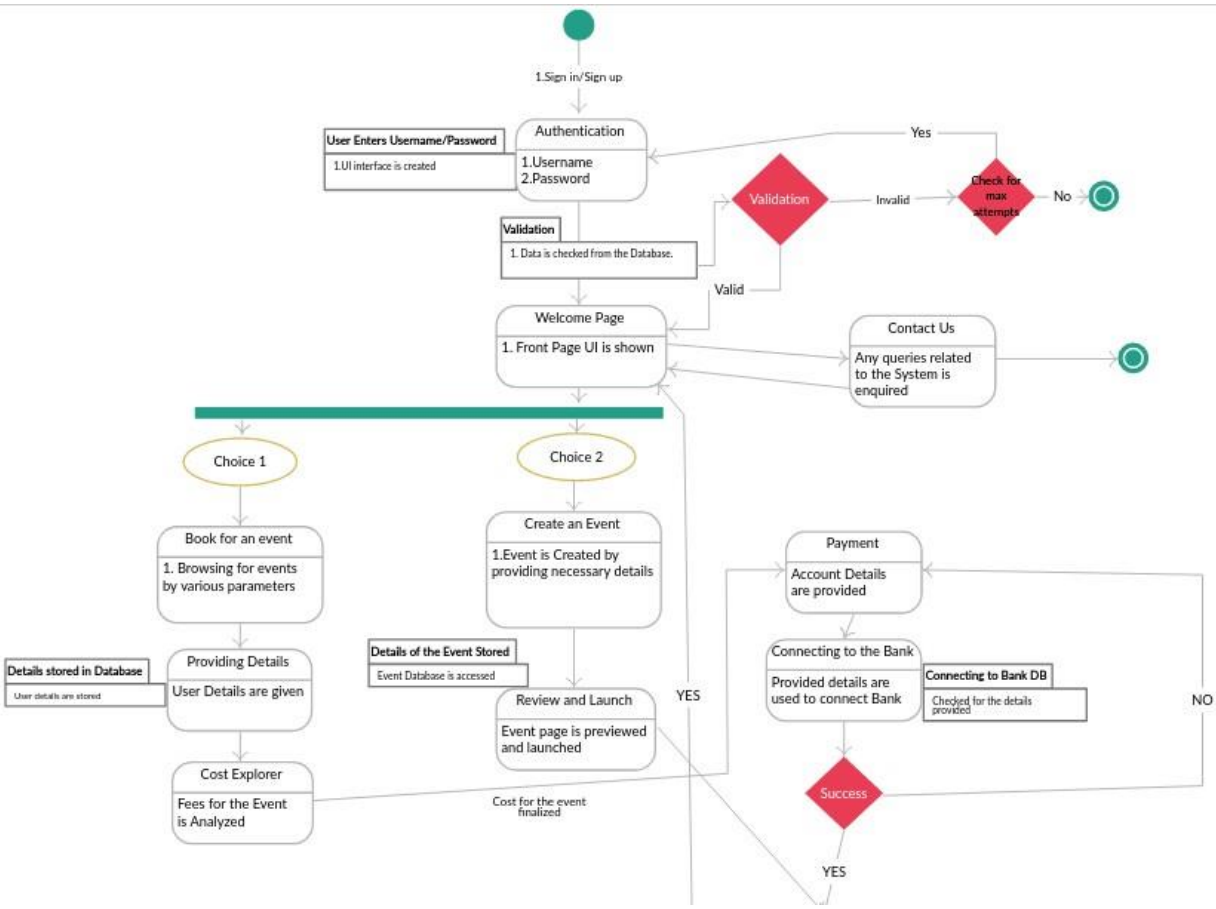
Users can add image, edit, delete, share photo

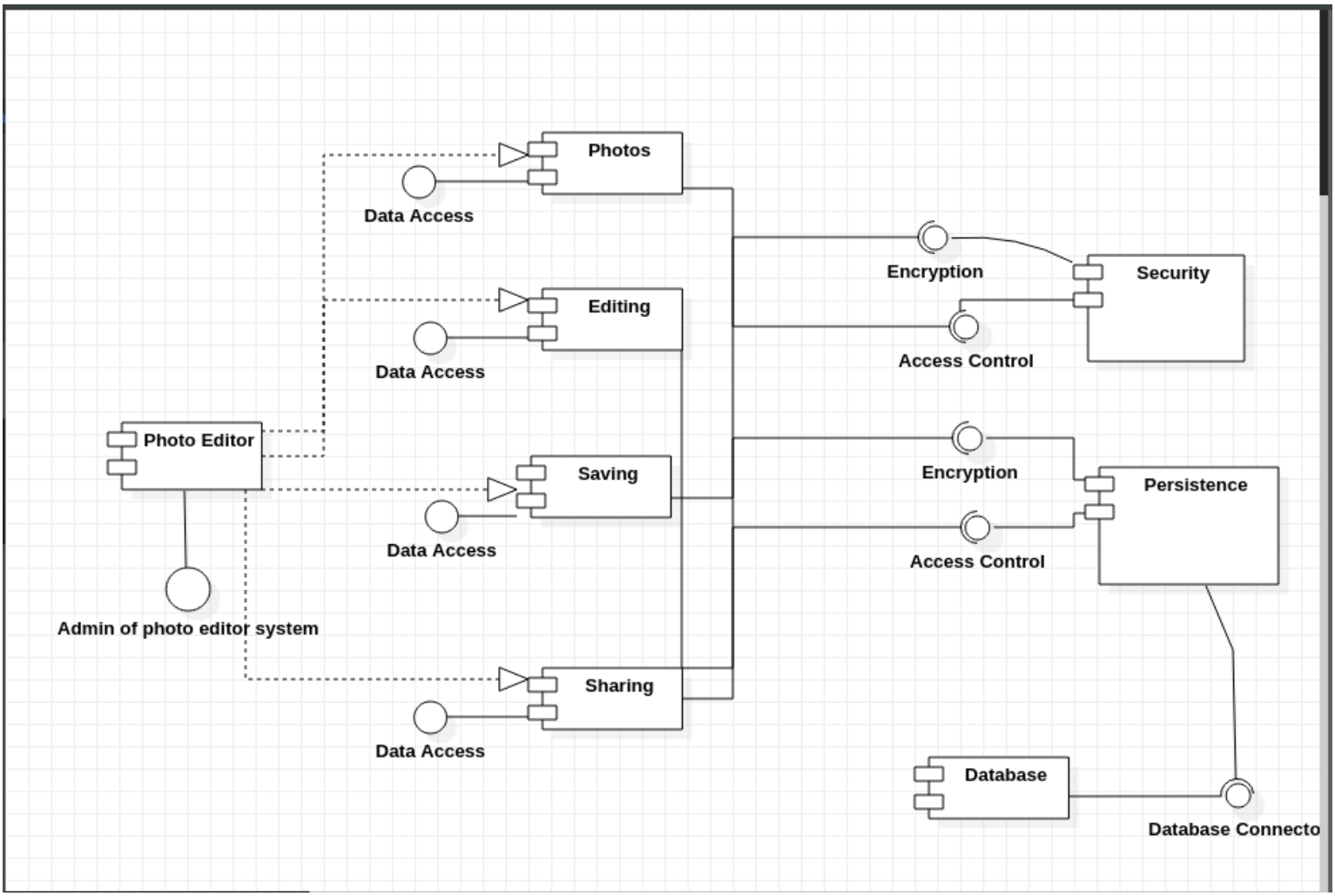


STATE Diagrams

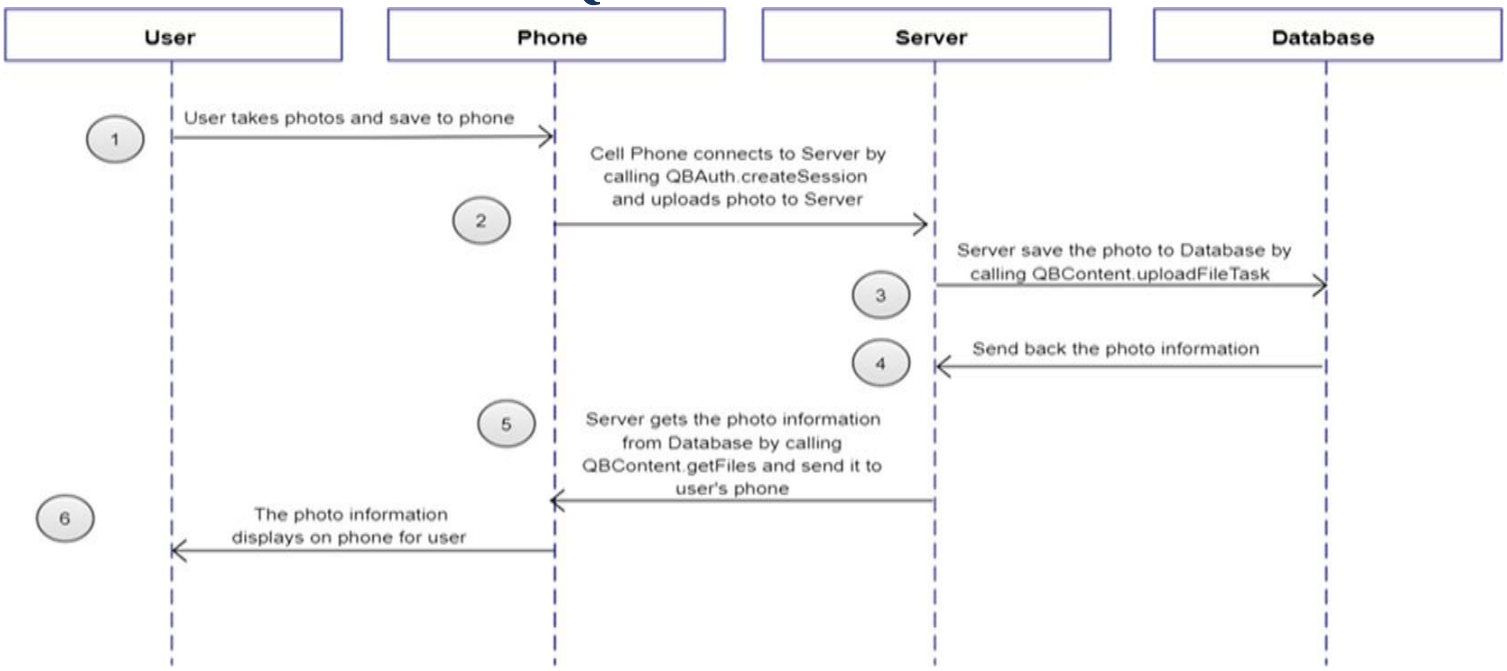


System Architecture

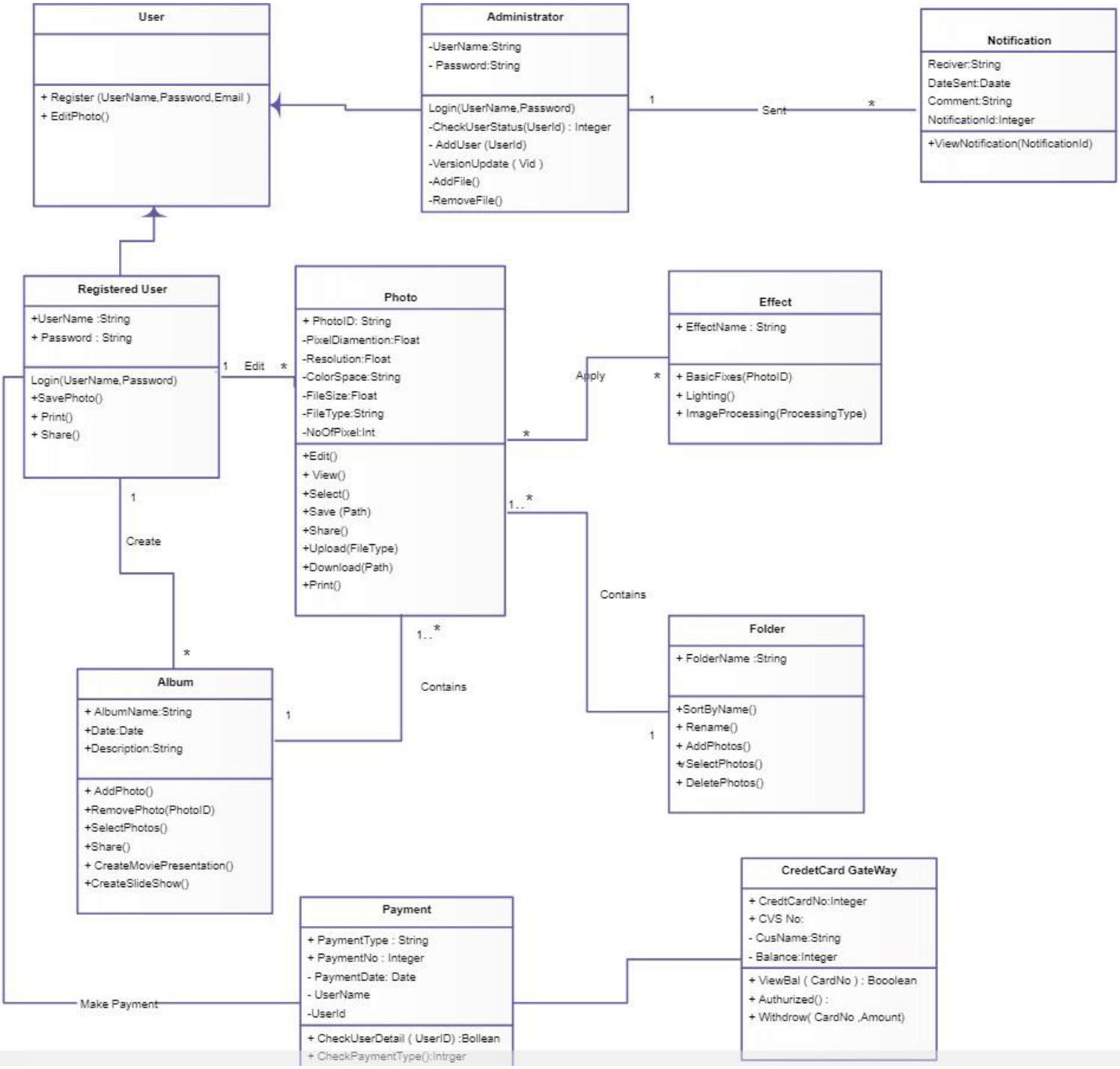

1.Sign in/Sign up
Authentication
1.Username
2.Password
User Enters Username/Password
1.UI interface is created
Validation
1. Data is checked from the Database.
Validation
Yes
Invalid
Check for max attempts
No
Valid
Welcome Page
1. Front Page UI is shown
Contact Us
Any queries related to the System is enquired
Choice 1
Choice 2
Book for an event
1. Browsing for events by various parameters
Create an Event
1.Event is Created by providing necessary details
Payment
Account Details are provided
Details stored in Database
User details are stored
Providing Details
User Details are given
Details of the Event Stored
Event Database is accessed
Review and Launch
Event page is previewed and launched
YES
Connecting to the Bank
Provided details are used to connect Bank
Connecting to Bank DB
Checked for the details provided
NO
Cost Explorer
Fees for the Event is Analyzed
Cost for the event finalized
Success
YES



SEQUENCE DIAGRAM



CLASS DIAGRAM



Module Description

Load Image

Loading the image to the application.

The first feature of the application is to get the required image from the user. This is done using button clicks to find the required file in the current system the application is running on and loading it on to the application. This is the most important feature of the application as all the other system features depend on this for their working, therefore this is a high priority feature.

Benefit Almost all of the features in the application depend on it.

Penalty – Without the feature the application wouldn't run, therefore this will incur the highest penalty.

Cost – A relatively low cost is assigned as it doesn't take much time to be implemented.

Risk – There is a lot of risk in finishing the application without this feature.

- get_input(image_input): The image input must be taken in from the user. The input may or may not necessarily be an image. The handling of such errors which keeps the application in a valid state must be given priority.

- error_on_invalid_entry(invalid_input): The application must respond to an invalid input after the window pane that is used to select the image is done with its job. The error can be a prompt or a message entry at one of the text boxes. The user has to understand that there is an error and must act on it to rectify it. Therefore, the message must be clear and loud and should take a substantial amount of space in the available frame.

- On_success(image_input): On a successful load of the image, necessary features open up to the user to act on editing the image. These features must not be available to the user before an image is selected. Any more movement in selecting the image must delay these features being shown on the frame.

- crop_image(height,width): The application must respond to the crop image option being picked and must concur with the necessary features mentioned in section 4.1.2.

- out_of_bounds(height,width): The height and the width inputs are limited to a specific size(TBD) and any out of bounds entry must be dealt with a prompt of the error message stating the same.

- on_complete(final_image): On a successful attempt of the required operation to be done on the image, the user now can select any other feature to make the necessary changes.

Rotation is an important aspect of augmentation and generally almost all the editors implement this feature. The user upon selecting the “Rotate” option, has two choices, either rotate left, or rotate right. These choices are made using button clicks and the necessary change is seen in the input image. An “exit” option is used to go back to the main list of features after using this feature.

5.3.3 Functional Requirements

The software capabilities that are needed for this feature to work are: the underlying OS must be Windows/Mac or Android, a file system support, user interactive features such as GUI, a pointer that is useful for button clicks/touch input for android.

- rotate(button_click): On selecting the rotate option, the user is provided with two buttons, which indicate their features through their labels.
- rotate_left(augment_image): Rotate the image 90 degrees to the left. Necessary changes must be seen in the window pane containing the input image.
- rotate_right(augment_image): Rotate the image 90 degrees to the right. Necessary changes must be seen in the window pane containing the input image.
- on_exit(out_of_rotate): On clicking this button, the input image holds the changes made and exits out of **RI**. All the other features will now be available.

Contrast Correction

Contrast correction with white balancing on applied to an image

.

This feature is available to the user right after the load image feature gives a valid output. The contrast correction option is generally used to white balance the input image. Currently, a two way color corrector (shades of white to black) is being considered to be an option. The drag option lets the user pick up the white balance.

Benefit - A low benefit value is given to it, as it does not specially entail in a photo editor.

Penalty – A moderate penalty based on the benefit.

Cost – This has a high cost, as implementation will take a huge chuck of development time.

Risk – There is an ample amount of risk in taking up this feature, as it may lead to a moderate amount of development time being wasted on it.

White balancing is considered to be an important feature in the working of a photo editing application. The user will have to select the correction button and then would be given two options, namely, “Color” and “Contrast”. On picking the “Contrast” option, a window pane with the two way corrector window would appear and the user would make the necessary white balance adjustments. After the said adjustments are done, the “Apply” button is selected and the image would reflect the necessary changes that were made.

- correction(button_click): On selecting this option, two different options can be seen by the user, “Color” and “Correction”.

- contrast_correction(coordinates): The contrast correction feature will have built in features and any necessary errors are handled by the API calls made. The two way color corrector is seen on selecting the contrast correction button.

- on_apply(changed_image): After the apply changes button is selected, the necessary changes are seen on the input image and all the set of features are once again available to the user. Note that after the changes are applied, the application would go back to a valid state and all of the options, correction -> contrast, must be selected again, for making more changes.

- on_edit(change_text): After a button click is received from placing the box at the desired place, the user can now edit the text in the text box.

- on_click(window_pane): The final click will ensure that the text is placed on the image at the specified place. This moves the application into a valid state.

Zoom In/Out

This feature is available to the user right after the load image feature gives a valid output. The zoom feature helps in enlarging or reducing the size of the input image. This is done with the help of two buttons (**ZIA**), which are available for selection after the “Zoom” button is selected. Necessary labels indicate the functions of the two buttons.

Benefit – A healthy benefit value is given as most editors in the market implement this feature.

Penalty – A high penalty is incurred on failing to finish the feature.

Cost – Zoom In/Out is a low cost, high priority feature.

Risk – A moderate amount of risk is involved in implementation of this feature.

The user will have to select the “Zoom” button and upon selecting this button, two new buttons appear on the UI. The two buttons are used for enlarging and reducing the input image size. A set limit is applied to both enlarge and reduce, where exceeding this limit, won’t change anything on the window pane. Note that there will be no prompt when the limit is exceeded. Note that to exit out of **ZIA**, there is an exit button which can be seen probably on the top right of the window pane.

- zoom(input_image): A button click will result in two new buttons appearing on the UI, which must be self-explanatory (based on the button label).

- on_click_enlarge(input_image_size): The image size must increase with this button click. A set limit is specified and on exceeding this limit the button won’t gauge any input from the user.

- on_click_reduce(input_image_size): The image size must decrease with this button click. A set limit is specified and on exceeding this limit the button won’t gauge any input from the user.

- on_exit(out_of_zoom): On clicking this button, the input image holds the changes made and exits out of **ZIA**. All the other features will now be available.

Chapter Five

Functionality Lists: -

1. Manage Photos
2. Edit Photos

3. Share Photos

4. Folder Management

1. Manage Photos

Test Case ID	Name of Module	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Result	Actual Result	Test Result
UT-01	Manage Photos	Select local photos to upload	User should run photo editor	1.Compile the editor.py 2.Click the open image button	Photos Should be Available	After successful uploading, the uploaded photos are displayed	After successful uploading, the uploaded photos are displayed	Photo uploading function is working well
UT-02	Manage Photos	Picking JPG,PNG,JP EG Image format	User should run photo editor	1.Compile the editor.py	Images formats are JPG,PNG,JPE G	Successfully uploaded the image	Successfully uploaded the image	PASS
UT-03	Manage Photos	Picking Image Size upto 1 MB size	User should run photo editor	1.Compile the editor.py	Images of 1 mb size must be available	Successfully uploaded the image	After successful uploading, the uploaded photos are displayed in the page	PASS
UT-04	Manage Photos	View the Image before Editing	User should run photo editor	Compile and run the .py file	Image is getting modifie D	Displaying the image	Image is Displayed	PASS
UT-05	Manage Photos	Editing Image Size up to 5 MB size	User should run photo editor	Compile and run the .py file	Image of 5 mb should be available	Successfully uploaded the image	Successfully uploaded the image	PASS
UT-06	Manage Photos	Viewing the Image after editing	Editing the image	Selecting any buttons to edit like rotate ,flip etc	Photo is availabl E			PASS

UT-07	Save image	Saved the edited image	Open a image	Selecting any buttons to edit like rotate ,flip etc	Photo is getting saved	Image is getting saved	New image is saved successfully	PASS
UT-08	Save photo	Any path can be selected to save the image	Open a image		New photo is getting saved to any path	New image is saved to any selected path successfully	New image is saved to selected path Successfully	PASS
UT-09	Revert to Original	Reset back to original image	Edit the image	1.Compile the editor.py 2.click the revert to original button	Original image is displayed	Original image is displayed	Original Image is displayed Successfully	PASS
UT-10	QUIT	Exit from photo editor	User should run photo editor	1.Compile the editor.py 2.click the QUIT button	Photo editor will be closed	Photo editor is getting closed	Photo editor is getting closed successfully	PASS

2. Edit Photos

Test Case ID	Name of Module	Test Case Description	Pre-Conditions	Test Steps	Test Data	Expected Results	Actual Result	Test Result
UT-01	Rotate left	Rotate the image left by 90 degree	User should run photo editor	1. Compile the editor.p y 2. click the Rotate left button	Image is getting rotated	Image should rotate left by 90 degree	Image is rotated left by 90 degree successfully	PASS
UT-02	Rotate Right	Rotate the image right by 90 degree	User should run photo editor	1. Compile the editor.p y 2. click the Rotate left button	Image is getting rotated	Image should rotate right by 90 degree	Image is rotated right by 90 degree successfully	PASS
UT-03	FLIP HORIZONT AL	Image is Flipped Horizontally	User should run photo editor	1. Compile the editor.p y 2. click the Flip Horizontal button	Image is getting Flipped by horizontal	Image should FLIP Horizontal	Image is Flipped Horizontally	PASS
UT-04	FLIP Vertical	Image is Flipped Vertical	User should run photo editor	1. Compile the editor.p y 2. click the Flip Vertical button	Image is getting Flipped by Vertical	Image should FLIP Vertical	Image is Flipped Vertically	PASS

UT-05	Brightness UP	Brightness is increased	User should run photo editor	1. Compile the editor.p y 2. click the Brightness UP button	Image brightness is increased	Image brightness should be increased	Image brightness should be increased successfully	PASS
UT-06	Brightness Down	Brightness is Decreased	User should run photo editor	1. Compile the editor.p y 2. click the Brightness Down button	Image brightness is Decreased	Image brightness should be Decreased	Image brightness should be Decreased successfully	PASS
UT-07	Contrast up	Contrast is Getting increase	User should run photo editor	1. Compile the editor.p y 2. click the Brightness Contrast up	Increase d the separation between dark and bright	Should Increase the separation between dark and bright	Increased the separation between dark and bright successfully	PASS
UT-08	Contrast Down	Contrast is Getting Decreased	User should run photo editor	1. Compile the editor.p y 2. click the Brightness Contrast up	Increased the separation between dark and bright	Should Increase the separation between dark and bright	Increased the separation between dark and bright Successfully	PASS

3. Share Photos

Test Case ID	Name of Module	Test Case Description	Pre - Condition	Test Steps	Test Data	Expected Result	Actual Result	Test Result
UT-1	Share Photos	Selecting photos to be shared	The photos should be stored in my system.	Opening the folder in which the photos are stored and then selecting them.	Photos should be there.	The photos that we want to send should be selected.	The photos that we want to send should be selected successfully	PASS
UT-2	Share Photos	Selecting the format of the photos(Jpeg,Png ,JPG	The photos should be stored in my system.	Opening the folder in which the photos are stored and then selecting them.	Photos should be there	The format in which we want to send should be selected.	The format in which we want to send should be selected successfully.	PASS
UT-3	Share Photos	Security	Photos should only be available to the root user.	Only the root user can access the photos to be sent.	Photos are in the encrypted folder.	Only the person who has the access can send the photos.	Only the person who has the access can send the photos.	PASS
UT-4	Share Photos	Mode of Transfer(Through USB)	The photos should be stored in my system and the system should have	Selecting the photos and copying them to the USB drive.	Photos in the USB.	Photos have been copied in the USB and can be given to	Photos have been copied in the USB and can	PASS

			a working USB port.			the receiver	be given to the receiver successfully	
UT-5	Share Photos	Mode of Transfer(Through Email)	The photos should be stored in my system and Receiver and sender should have a working email.	Selecting the photos and uploading them and sending them to the receivers E-mail.	Photos that have been uploaded.	The receiver should receive the photos through their Email.	The receiver should receive the photos through their Email successfully..	PASS
UT-6	Share Photos	Mode of TransferThrough Social media(Facebook,Instagram, Whatsapp,Messenger)	The photos should be stored in the system and Receiver and sender should have a valid social media handle.	Uploading the selected photos to the receivers social media chat box and then sending them.	Photos that have been uploaded on the receivers handle.	The receiver should receive the photos through their Social media	The receiver should receive the photos through their Social media successfully.	PASS
UT-7	Share Photos	Mode of Transfer (Bluetooth)	The photos should be stored in the system and Receiver and sender should have active Bluetooth pairing.	Selecting the photos and sending them over Bluetooth.	Photos that have been selected for Bluetooth transfer	The receiver should receive the photos over Bluetooth.	The receiver should receive the photos over Bluetooth successfully.	PASS

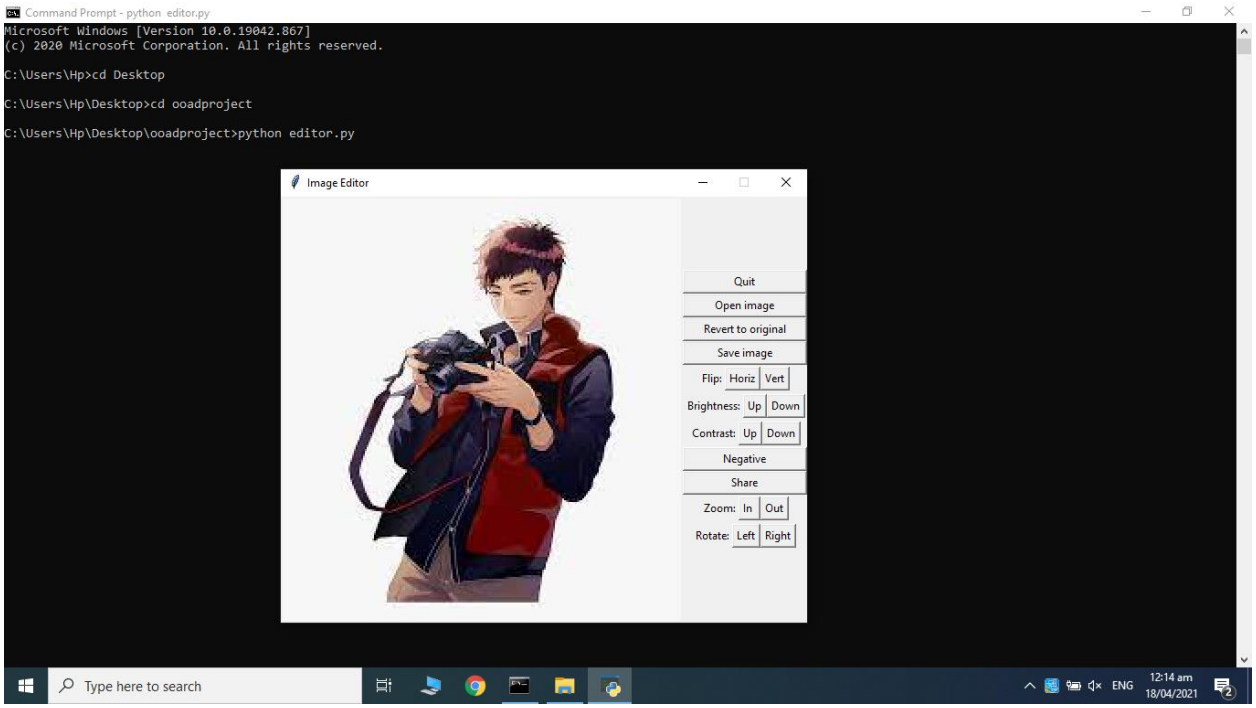
UT-8	Share Photos	Sharing Completed	The receiver should have received all the photos.	All the photos have been sent through different platforms.	The photos that have been sent	All the photos shared with the receiver.	All the photos shared with the receiver successfully.	PASS

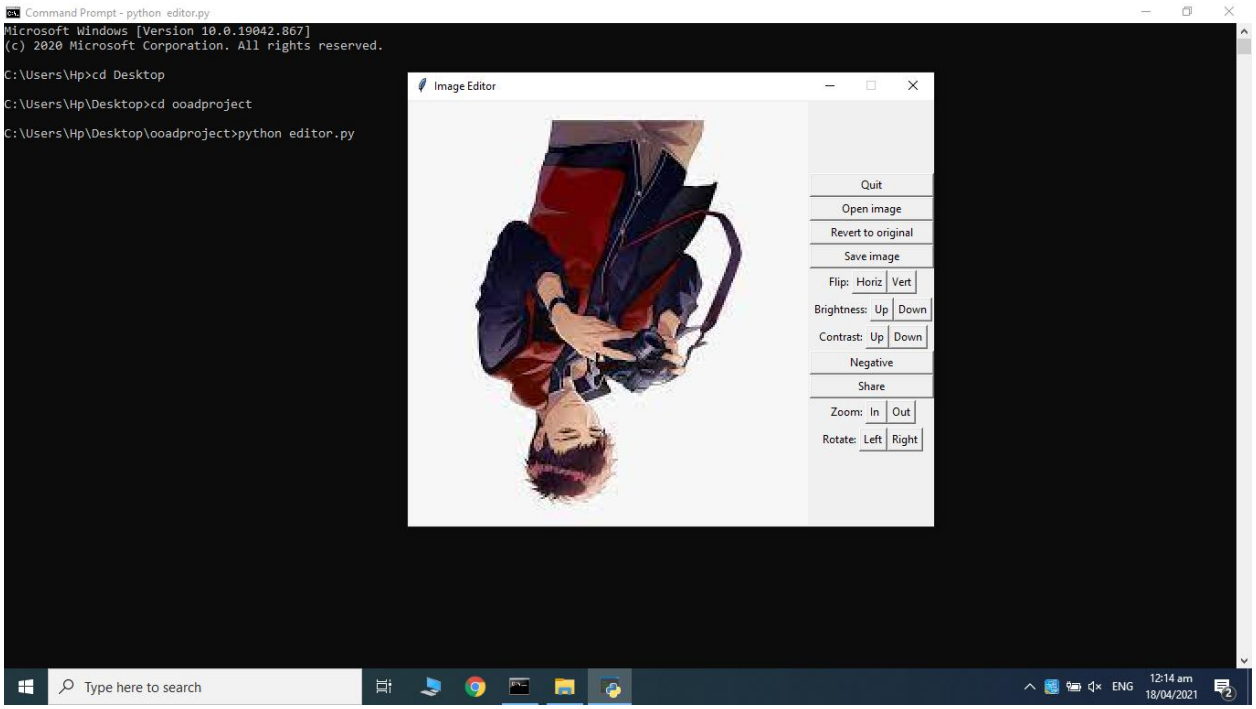
4. Folder Management

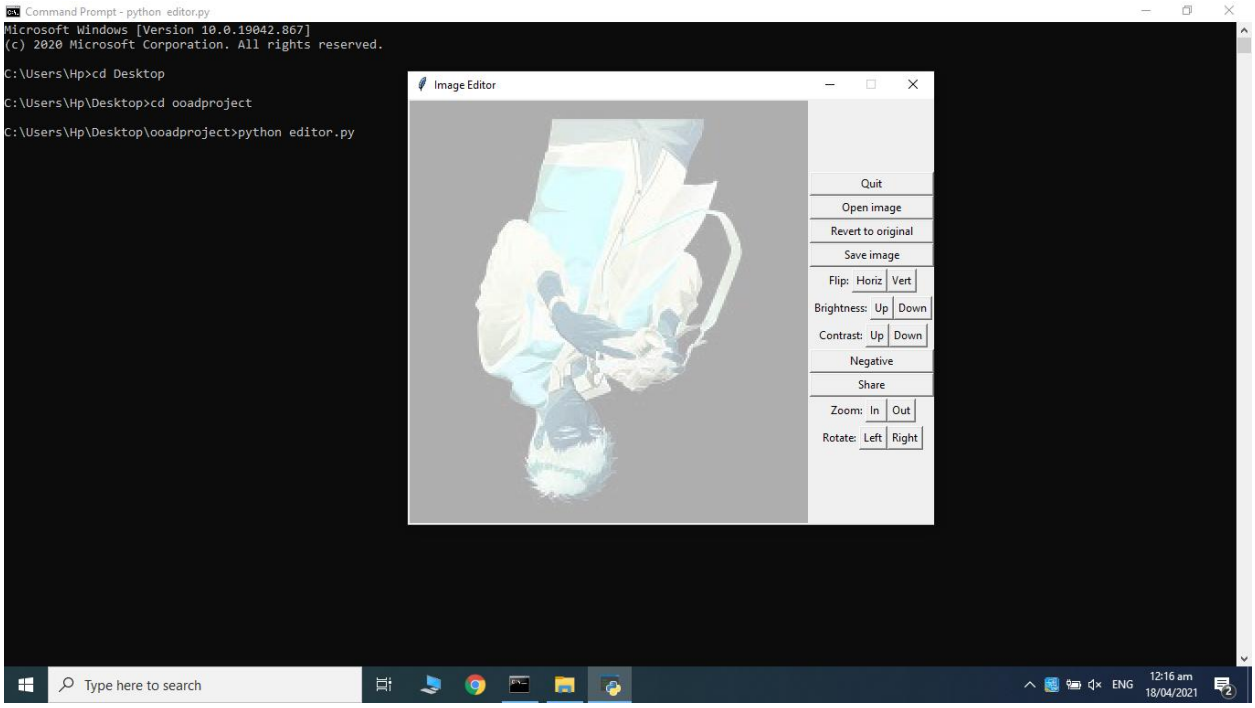
Test Case ID	Name of Module	Test Case Description	Pre - Condition	Test Steps	Test Data	ExpecteResult	Actual Result	Test Result
UT-1	Save a photo to a folder	Upload any edited photo to the folder	User should run photo editor	Select an edited photo	New photo is saved to the folder	New photo is saved to the folder	New photo is saved to the folder successfully	PASS
UT-2	Delete a photo from the folder	Delete any edited photo from the folder	User should run photo editor	Select an edited photo	Photo is deleted from the folder	Photo is deleted from the folder	Photo is deleted successfully	PASS
UT-3	Moving photos	Select photos and move to another album folder	User should run photo editor	Select an edited photo	The current page no longer displays the photo, and views the target album, including the photo	The current page no longer displays the photo, and views the target album, including the photo	The photo is moved successfully to a different folder	PASS

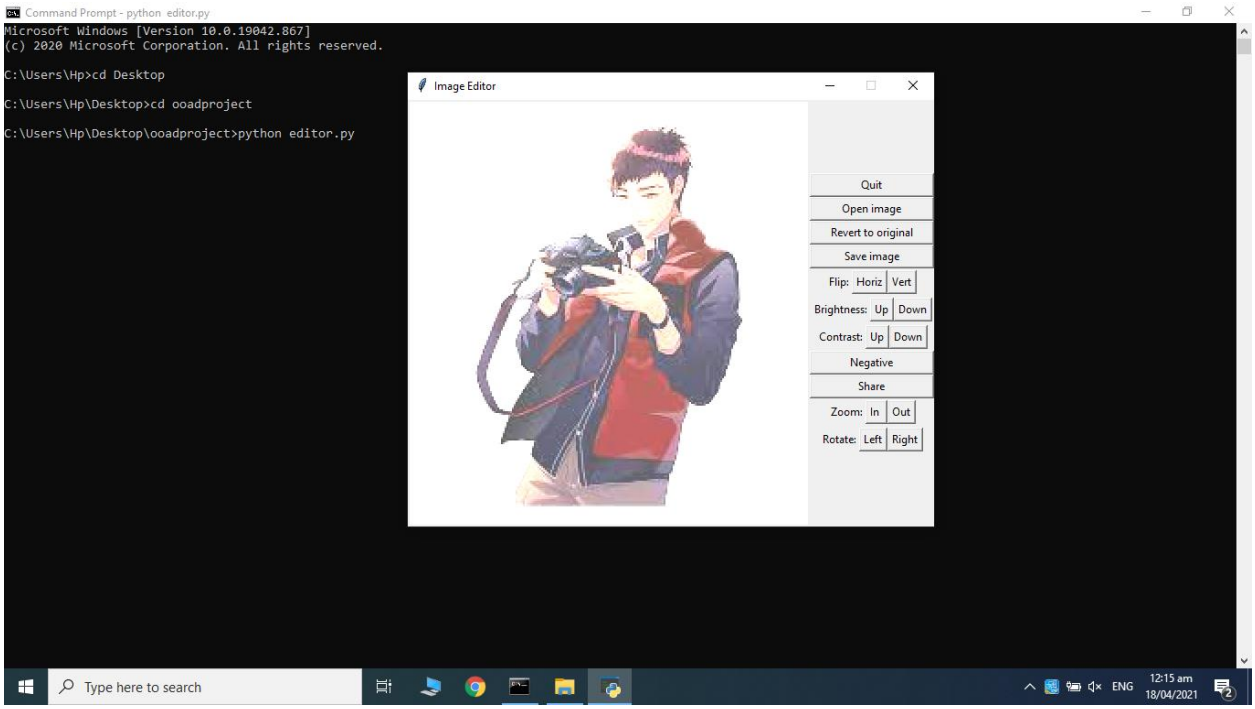
UT-4	Access a collection	Select the collection item	User should run the photo editor		Jump to the corresponding target interface	Jump to the corresponding target interface	Target Interface Opened Successfully
UT-5	Delete a collection	Select the collection item to delete	User should run the photo editor		Favorite page does not display this collection target anymore	Favorite page does not display this collection target anymore	Function of Deleting Collection item is Working Well
UT-6	Create a new album	Create a new album,enter the album name	User should run the photo editor	Open the folder	Show named album	Show named album	Function of creating a new album is normal
UT-7	Select sorting methods	Select the photo sorting method in the works display page, with sorting according to the upload time	User should run the photo editor	Open the folder	All sorting according to the relevant demands	All sorting according to the relevant demands	Function of Photo sorting is Working Well
UT-8	Share Folder	Share the selected folder	The receiver should have received the folder	The folder has been sent through different platforms.	The folder that has been sent	The folder shared with the receiver.	The format in which we want to Send should be Selected

Output Screen Shots.






Command Prompt - python editor.py
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\Hp>cd Desktop
C:\Users\Hp\Desktop>cd ooadproject
C:\Users\Hp\Desktop\ooadproject>python editor.py
Image Editor
Quit
Open image
Revert to original
Save image
Flip: Horiz Vert
Brightness: Up Down
Contrast: Up Down
Negative
Share
Zoom: In Out
Rotate: Left Right
Type here to search
ENG
12:16 am
18/04/2021


Command Prompt - python editor.py
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\Hp>cd Desktop
C:\Users\Hp\Desktop>cd ooadproject
C:\Users\Hp\Desktop\ooadproject>python editor.py
Image Editor
Quit
Open image
Revert to original
Save image
Flip: Horiz Vert
Brightness: Up Down
Contrast: Up Down
Negative
Share
Zoom: In Out
Rotate: Left Right
Type here to search
ENG
12:15 am
18/04/2021