

PHOTOGRAPHY ASSISTANT

CHAI JAY JAY

SESSION 2021/2022

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY

MULTIMEDIA UNIVERSITY

MARCH 2021

PHOTOGRAPHY ASSISTANT

BY

CHAI JAY JAY

SESSION 2021/2022

THE PROJECT REPORT IS PREPARED FOR

**FACULTY OF INFORMATION SCIENCE & TECHNOLOGY
MULTIMEDIA UNIVERSITY
IN PARTIAL FULFILLMENT
FOR**

**BACHELOR OF INFORMATION TECHNOLOGY
B.I.T. (HONS) ARTIFICIAL INTELLIGENCE**

FACULTY OF INFORMATION SCIENCE & TECHNOLOGY

MULTIMEDIA UNIVERSITY

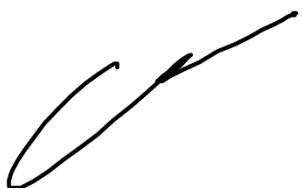
MARCH 2021

© 2022 Universiti Telekom Sdn. Bhd. ALL RIGHTS RESERVED

Copyright of this report belongs to Universiti Telekom Sdn. Bhd as qualified by Regulation 7.2 (c) of the Multimedia University Intellectual Property and Commercialization policy. No part of this publication may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Universiti Telekom Sdn. Bhd. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

DECLARATION

I hereby declare that the work have been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.



CHAI JAY JAY

Faculty of Information Science & Technology
Multimedia University

Date: 24/3/2022

ACKNOWLEDGEMENT

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my supervisor, Ts. Liew Tze Hui for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my parents and friends who had helped and given me encouragement along the way on my journey. They have tolerated and assisted me in many ways, which in my opinion is invaluable and indescribable. I am eternally grateful to have such close ones in my circle

ABSTRACT

In the digital age, any forms of media consumption, especially images and videos are in high demand. This is even so with smartphones becoming cheaper and more and more social media platform emerge unto the masses, this leads to an influx of photographers creating more and more content. Gone are the days where only dedicated cameras are able to produce high quality images, where even the low-end smart phones are able to create photos. However, this also leads to the problem of lacking knowledge to detect and manage the photos taken, causing lack of storage and abundance in poor quality photos. This phenomenon does not only occur in the everyday leisure photographer, but also in professional photographers. To professional photographers, time is precious and in shortage with the possibility of the client in need of images to be processed in an instant. Hence, this project is aim towards creating a photography management software. The main idea for this software is to develop a system that is able to sort the images based on their feature by measuring the contrast and blurriness level and categorize into 4 different folders. Furthermore, a photographer's life is hectic and always on the road, this leads to negligence in management of files as well as massive pillage of files in the user's storage device. The photography management software designs are also aimed towards helping with cleaning up said pillage by detecting files last modified date and give the ability if they are older than the user pre-set date and sort into newer and older folders.

TABLE OF CONTENTS

DECLARATION	III
ACKNOWLEDGEMENT	IV
ABSTRACT	V
TABLE OF CONTENTS	VI
LIST OF TABLES.....	VIII
LIST OF FIGURES.....	IX
LIST OF ABBREVIATIONS/ SYMBOLS	XII
LIST OF APPENDICES.....	XIII
CHAPTER 1 INTRODUCTION.....	1
1.1 Overview	1
1.2 Problem Statement.....	1
1.3 Project Objectives.....	1
1.4 Project Scope	2
1.5 Report Organisation.....	2
CHAPTER 2 LITERATURE REVIEW	4
2.1 Introduction	4
2.2 Amazon Photos.....	4
2.3 Google Photos	8
2.4 Microsoft Photos	11
2.5 Adobe Photoshop Lightroom Classic	13
2.6 FastStone Image Viewer.....	18
CHAPTER 3 METHODOLOGY	21
3.1 Rapid Application Development (RAD)	21
3.2 Requirements Planning.....	22
3.3 User Design	22
3.4 Rapid construction.....	22
3.5 Cutover	22
3.6 Tool, Techniques and Programming Languages	23

3.7	Gantt Chart	25
CHAPTER 4 IMPLEMENTATION PLAN / DESIGN		27
4.1	Proposed Solution Overview	27
4.2	Use Case Diagram	27
4.3	User Activity Diagram	28
4.4	Context Diagram	29
4.5	Data Flow Diagram	30
4.6	Interface Design.....	32
4.7	Algorithm	35
CHAPTER 5 IMPLEMENTATION PROCESS AND RESULT		37
5.1	Overview	37
5.2	Implementation Process.....	37
5.3	Result.....	51
5.4	Instruction Manual.....	56
CHAPTER 6 TESTING AND EVALUATION OF FINDINGS		65
6.1	Overview	65
6.2	Testing Outline	65
CHAPTER 7 CONCLUSION		67
7.1	Overview	67
7.2	Summary of work done	67
REFERENCES		68
APPENDICES.....		70

LIST OF TABLES

Table 2-1: Comparison between software.....	20
Table 6-1: Testing outline table.....	65

LIST OF FIGURES

Figure 2.1: Amazon Photos Main Menu.....	5
Figure 2.2: Amazon Photos Backup Menu.....	5
Figure 2.3: Amazon Photos Download Menu	6
Figure 2.4: Amazon Photos Import File menu	7
Figure 2.5: Amazon Photos Order Prints Menu.....	8
Figure 2.6: Google Photos Main menu	9
Figure 2.7: Google Photos Photo and Videos File Quality options	10
Figure 2.8: Google Photos Utilities menu	10
Figure 2.9: Microsoft photos main menu	11
Figure 2.10: Microsoft Photos People menu	12
Figure 2.11: Microsoft Photos Search menu	13
Figure 2.12: Adobe Lightroom Classic Main menu	14
Figure 2.13: Adobe Lightroom Classic Exporting as a “Catalog”.....	15
Figure 2.14: Adobe Lightroom Classic Catalogue Sample	15
Figure 2.15: Adobe Lightroom Classic Develop Menu	16
Figure 2.16: Adobe Lightroom Classic Book Menu	16
Figure 2.17: Adobe Lightroom Classic Slideshow menu	17
Figure 2.18: Adobe Lightroom Classic Print menu	17
Figure 2.19: FastStone Image Viewer Main Menu.....	18
Figure 3.1: Rapid Application Development.....	21
Figure 3.2: Gantt chart (Phase 1).....	26
Figure 4.1: Use Case Diagram	28
Figure 4.2: User Activity Diagram.....	29
Figure 4.3: Context Diagram.....	30
Figure 4.4: DFD Level-1 diagram	31
Figure 4.5: Photography Assistant Main Menu.....	32
Figure 4.6: Photography assistant main menu with filter option	33
Figure 4.7: Photography Assistant Scan menu filter option.....	33
Figure 4.8: Photography Assistant File Check menu with Date settings	34
Figure 4.9: Photography Assistant Date settings	34
Figure 4.10 : Contrast Detection implementation method	35
Figure 4.11: Sharpness calculation Implementation method	36

Figure 5.1: Main Window User interface source code	38
Figure 5.2: Age Filter User Interface Source code	39
Figure 5.3: Scan Feature dialog box user interface source code	40
Figure 5.4: Progress Bar user interface source code.....	41
Figure 5.5: Main Window function source code.....	41
Figure 5.6: Main Window initialization source code	42
Figure 5.7: “Select directory” button class source code	43
Figure 5.8: Source code used to display treeview	43
Figure 5.9: Source code for getting the path of each image.....	44
Figure 5.10: Source code for “get_feature_inputs” method.....	44
Figure 5.11: Source code for Scan Age and Scan Feature progress bar function	45
Figure 5.12: Source Code for “Scan Feature” button function.....	45
Figure 5.13: Source code for Sharpness detection algorithm.....	46
Figure 5.14: Source code for Contrast detection algorithm	46
Figure 5.15: Source Code for “Scan Age” button function	47
Figure 5.16: Source Code for error message.....	48
Figure 5.17: Source code for setting up Progress bar dialog window	48
Figure 5.18: “Scan Feature” Dialog box menu source code	49
Figure 5.19: “Scan Age” button function source code	50
Figure 5.20: Main Window User Interface	51
Figure 5.21: “Scan Feature” dialog box User interface	52
Figure 5.22: Output of “Scan Feature”	53
Figure 5.23: “Scan Age” dialog box user interface	54
Figure 5.24: “Scan Age” function output.....	54
Figure 5.25: Progress bar dialog box user interface	55
Figure 5.26: Run Menu	56
Figure 5.27: Command Prompt	56
Figure 5.28: main window	57
Figure 5.29: “Select directory” button location in Main Window	57
Figure 5.30: Output of “Select Directory” button.....	58
Figure 5.31: “Scan Feature” button.....	58
Figure 5.32: “Scan feature” dialog box	59
Figure 5.33: Notes for “Scan Feature”	59

Figure 5.34: Progress bar dialog box.....	60
Figure 5.35: output of “Scan Feature” function	60
Figure 5.36: Main Window	61
Figure 5.37: “Select Directory” button location in the Main Window	61
Figure 5.38: Output of “Select Directory” function	62
Figure 5.39: “Scan Age” button location.....	62
Figure 5.40: “Scan Age” dialog prompt	63
Figure 5.41: “Scan Age” dialog box.....	63
Figure 5.42: Output of “scan age” button function.....	63

LIST OF ABBREVIATIONS/ SYMBOLS

FYP	Final Year Project
MMU	Multimedia University
RAD	Rapid Application Design
OS	Operating System
GB	Gigabyte
CPU	Central Processing unit
RAM	Random Access memory
GHz	Gigahertz
HDD	Hard Disks Drive
SSD	Solid State Drive
RM	Malaysian Ringgit
USD	United State Dollar
iOS	IPhone Operating System
Mac	Macintosh
PC	Personal Computer
DFD	Data Flow Diagram
Adobe Photoshop Lightroom Classic	Adobe Lightroom Classic

LIST OF APPENDICES

Appendix A: Meeting Logs	70
Appendix B: Checklist for FYP Interim Report Submission	78

CHAPTER 1

INTRODUCTION

1.1 Overview

In this Final Year project report, the author has discussed the design and specification as well as documentation regarding the project “Photography Assistant”. The project entails multiple tasks, including problem statements, project objectives, literature review, methodology and proposed solution of the project itself, hardware requirements, software tools and user interface design.

1.2 Problem Statement

Modern younger generation Photographers, be it hobbyist or professional tends to have a busy schedule as well as lacking in formal photography education and experience. This scenario often leads photographer having a hard time in the effort to detect and identifying redundant images in a large array of image. Furthermore, with a busy schedule, Photographers often neglect in storage management, thus leading to forgotten images stored within the hard drive, this causes massive clutter and lack in storage space over time.

To design and implement an application namely “Photography Assistant” which are able to identify age of photos older than the date set by user, to be able to sort the scanned images into designated files as well as detecting redundant photos using languages and resources readily available in the author’s disposal.

1.3 Project Objectives

To design and implement an application namely “Photography Assistant”

The objectives are as shown below:

- To study the similar and different application developed

- To develop a system that assists photographers
- To develop a system that able to detect feature of photos
- To develop a system that are able detect image older than pre-set date

1.4 Project Scope

The target users are young adults (age 18-30) in Malaysia. These demographics is chosen because younger generations of photographer does not have enough experience or formal teachings towards selections of good images. Furthermore, younger generations tends to be more negligence towards storage managements as they often have the opinion that they are not important.

1.5 Report Organisation

For this report, it consists of five (5) chapters. The chapters included are the Introduction, Literature review, Methodology, System Design and finally Conclusion.

Chapter 1 - Introduction

Introduction mainly comprised of the general understanding of the project that includes overview, which show the overall report contents, problem statement which explains the problems leading to the creation of the project, project objectives, project scope as well as report organisation

Chapter 2 – Literature Review

Literature review or in this case a benchmark review mainly reviews other similar software to the project itself, describing their origin as well as their notable functions and features and finally have a general comparison between the reviewed software

Chapter 3 – Methodology

Methodology mainly describe the methods and which are required into developing the system itself. These methods includes the type of development model utilised that in this case will be RAD as well as list of tools and techniques needed develop the system.

Chapter 4 – System design

System design is the planning and design phase. Various charts, logical and graphical diagrams that are used to illustrate the design of the proposed system. The charts and diagrams include use case diagram, user activity diagram, context diagram and Level-1 data flow diagram.

Chapter 5 – Implementation Process and Result

Implementation process and result chapter describe the codes and steps necessary to produce the output program. Diagram and source codes images has been used to illustrate the output of the system as well as a explanation. Furthermore an instruction manual has been added as well for assistance in using the software

Chapter 6 - Conclusion

The summary of works concluded in the whole Final Year Project phase 1 throughout the semester that was given.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

After searching the internet for existing applications that qualifies to are similar to the proposed “Photography Assistant” , the author has come up with several applications that are developed by major companies such as “Google” , “Microsoft”, all of the applications are able to view information about the various photos imported into the system.

2.2 Amazon Photos

In 2011, Amazon Releases their cloud services known as Amazon Drive, together with this release they also release a service known as Amazon Photos, which is a cloud based photos and video sharing service application. In order to access the application , users may choose to download the software applications which are released to several different Operating System platform which includes Android , iOS , Windows , Linux as well as Mac OS. Alternatively, users may choose to use their online platform which users may be access through their favourite browsers after Signing into Amazon Photos.

Amazon Photos primarily is used as a cloud storage system for Photos and videos. Within this application, users may conduct several activities related to storage of media. The Application features several menus, which includes the settings, the home menu, the backup menu, as well as the download menu. In the main menu of the “Amazon Photos’ applications after signing in, users will be greeted with blue and turquoise themed user interface, On the left side of the menu, users will see a side navigation bar with several buttons which includes, “home”, “Backup”, ”Download” as well as “Settings”. At the top bar, users will see a “Pause all activity” button as well as a drop down menu which users may change or view their personal preference of their profile as well as the applications. In the “Home” menu itself, users may see instructions as to “Drag & Drop” functions to upload their photos or videos.

Furthermore, the users may see the data limit for their plans, the numbers of photos and videos backed up, and the total number of photos uploaded according to year. Users may also view their recent uploads in the bottom of the menu. As for the “Backup” and “Download” menu , the user interfaces designs were simple where users may view the folders they have uploaded as well as adding more folders to be uploaded. In the “Download” menu, users may choose to download the folders or album that they have uploaded.

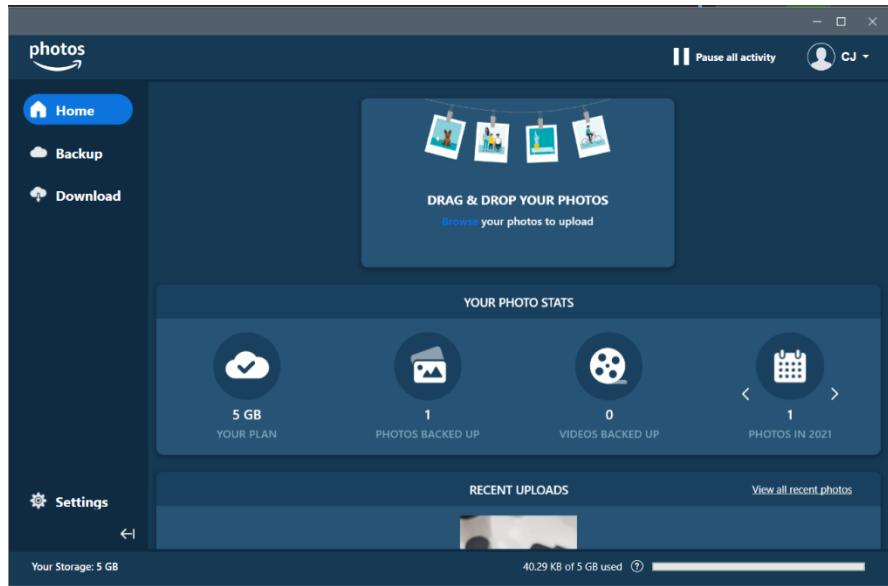


Figure 2.1: Amazon Photos Main Menu

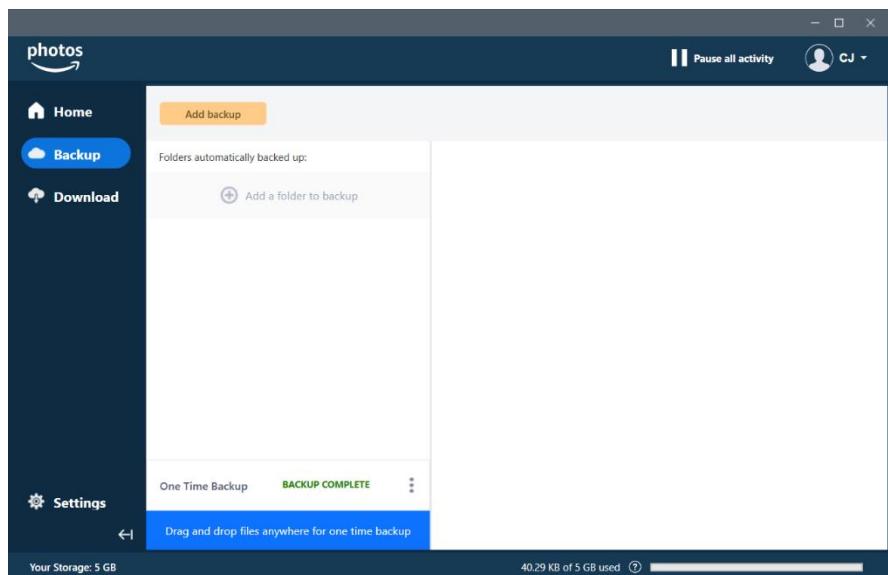


Figure 2.2: Amazon Photos Backup Menu

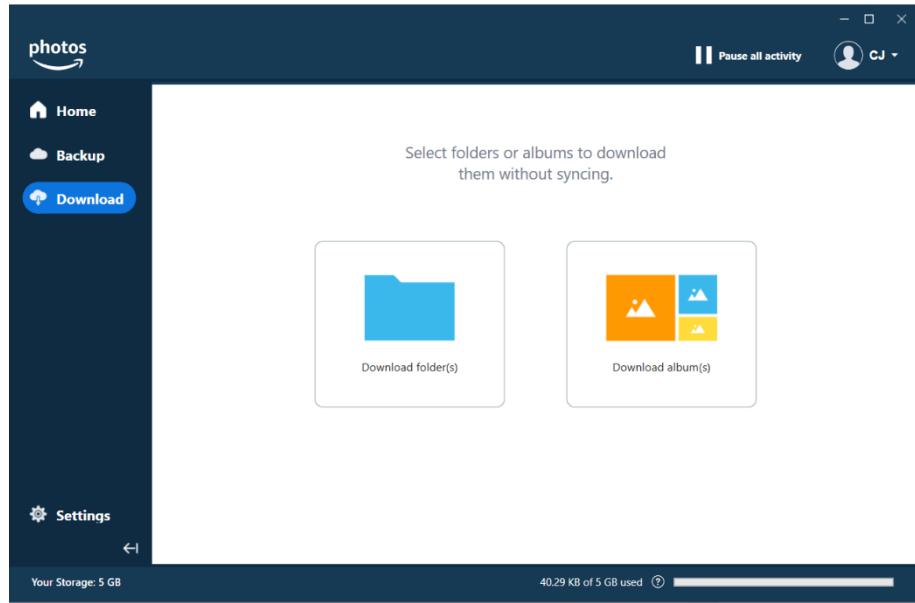


Figure 2.3: Amazon Photos Download Menu

In terms of general functionality of the application and its system, there are several that are noteworthy, one of which are the ability to “Drag & Drop”, this function speeds up the process of storage as the user no longer require to search through their directory and reselect their desired photos to be stored.

Next we have the ability to create folders in the application which allows for users to organize their storage in a neater manner. Next we have the ability to view the usage details of the storage, the details includes the type of files that have been uploaded, the sizes of the categories of files, and the percentage of space these file categories occupy.

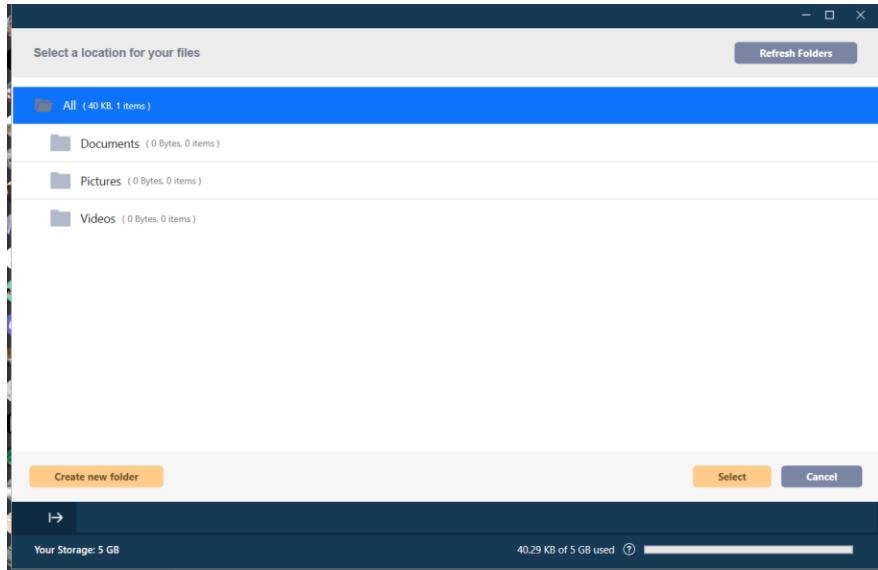


Figure 2.4: Amazon Photos Import File menu

Furthermore, we can see that Amazon Photos has another function of which it enables the user to order prints for their Photos via Amazon services, this saves time for users as they no longer require searching for a printing services externally. Furthermore, thanks to Amazon Photos linking to Amazon Web Services, Users are able to share their photos, videos, and even albums via email, links, contacts, and invite post through Facebook, shared viewers may add comments and share the photos that they fancy.

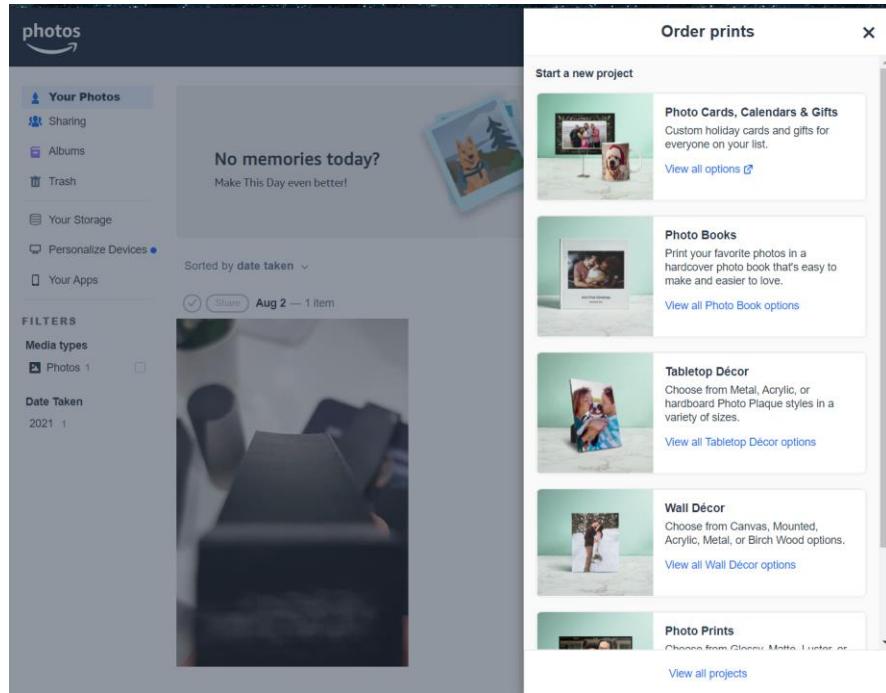


Figure 2.5: Amazon Photos Order Prints Menu

Furthermore, Amazon Photos enables the user to synchronise their system with Amazon cloud services. Amazon Photos also enables the users to adjust their bandwidth settings as well as proxy settings for auto or manual.

Amazon Photos is a subscription-based service. However, they do have free of charge services with the only limitation being the storage size limited to 5GB. Amazon Photos subscription plans are as the following, 100GB for 1.99 USD per month, 1TB for 6.99 USD per month, 2TB for 11.99 USD per month

2.3 Google Photos

Google first announced Google Photos in May 2015 on Android, iOS and web as a service for photo sharing and storage services. Currently Google Photos are available free with limitation placed on its storage sizes and is sync with the user's Google Account.

Google Photos are a web-based storage system, thus requiring users to have a browser to use the service. As soon as the user logs onto their Gmail account, they will

be greeted with intuitive menu system which are similar to another service provided by Google which is Google Drive. An important detail that users must understand is that the storage of Google Photos are synced with Google Drive hence the storage are affected between the 2 service. In the main menu, the user will be greeted with a sidebar on the left and the top bar menu as well as a blank page with “drag & drop” instruction on it.

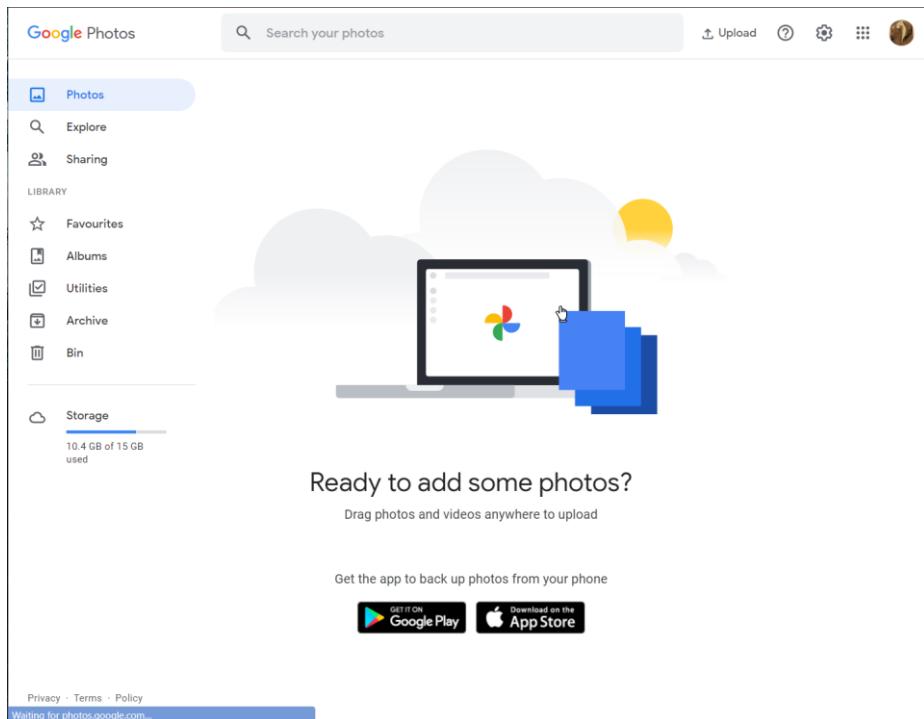


Figure 2.6: Google Photos Main menu

On the left sidebar, users will be able to access various menus such as photos, explore, sharing, favourites, albums, utilities, archive, bin as well as an indicator for the amount of storage using Gigabyte Format. On the top navigation bar, users will be greeted with the Google Photos logo, a search bar, and upload button, help & feedback button, settings, Google Apps, and Google logo account button.

In terms of noteworthy functions of Google Photos, one of which is the ability to drag & Drop the photos, videos and files that users would like to be stored. Another noteworthy function is the ability to set the quality of the photo that wished to be stored; the choices are original quality and reduced quality. Furthermore, Google

Photos shows the progress of upload by showing a progress bar and the name of the file that the user uploaded.

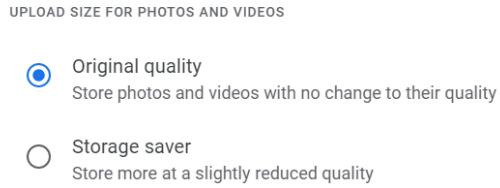


Figure 2.7: Google Photos Photo and Videos File Quality options

The next noteworthy function to be taken note of is the ability to synchronize with not only the PC of the user but also the mobile devices which the user has installed the app and logged into using the same account as the PC counterpart.

Furthermore, Google Photos has a library section that provides several unique functions such as creating a Movie, animations and Collages from the stored images and video files. The library section also provides a unique function that is adding a partner account which enables you to share the library.

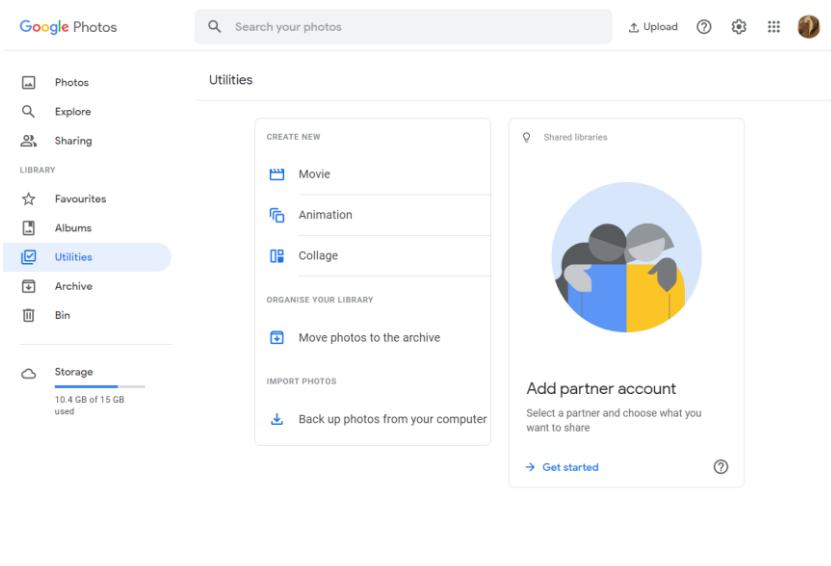


Figure 2.8: Google Photos Utilities menu

Finally yet importantly, Google Photos is a subscription-based service, which has its rate a RM 8.49/month for 100GB of storage, RM 11.99/month for 200GB and RM 42.99/month for 2TB of storage.

2.4 Microsoft Photos

Microsoft Photos is an application that enables the user to organize, view, and light edit their photos and videos on their machine. First developed and released in 2012 by Microsoft, Microsoft photos went through a significant amount of name changes as well as improvement with the latest iteration released in 2019 for Windows 10.

Microsoft photos is a software primarily designed for users as a photo storage and viewing software. Microsoft Photos unlike the priory-reviewed software has an interesting UI design whereby it enables dark mode for aesthetics. When users first launch the app, Users will be greeted with recently added photos under the collection tab. Microsoft photos has a top navigation tab, which consists of collections, Albums, People, Folders, Video Editor, Search bar, Import options, account options and settings.

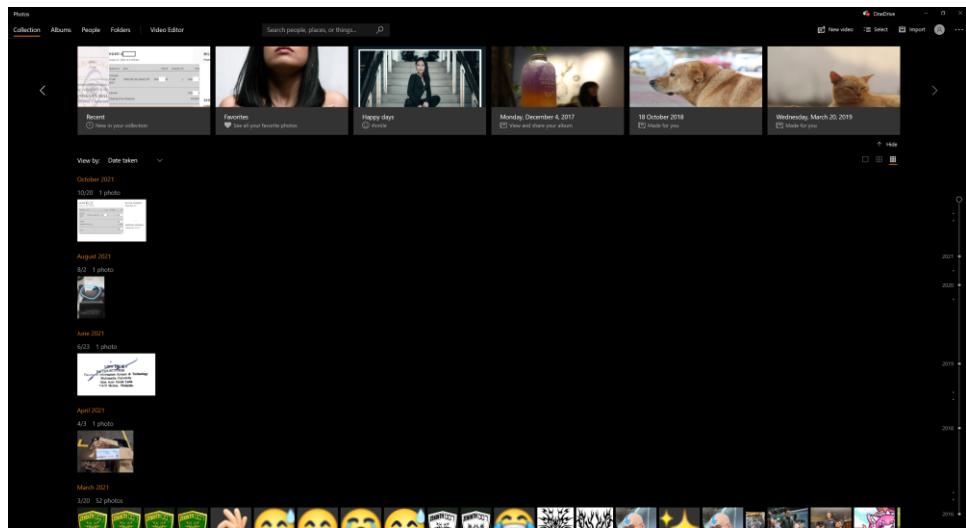


Figure 2.9: Microsoft photos main menu

Microsoft Windows provides several main functions which includes Photo management, Photo Editing, Video editing, Photo and Video Import. Firstly, in photo management, any photographs that are viewed prior with Microsoft photos will be added automatically into Microsoft Photos. There are several noteworthy functions in Microsoft Photos, First of which is the ability to change the viewing options with choices such as Date taken and Recently Added, furthermore in terms of viewing options such as grid view with date labels on top of the album and timeline of the scrolling bar. Furthermore, Microsoft Photos are able to detect faces and compile into folders under the “People” tab.

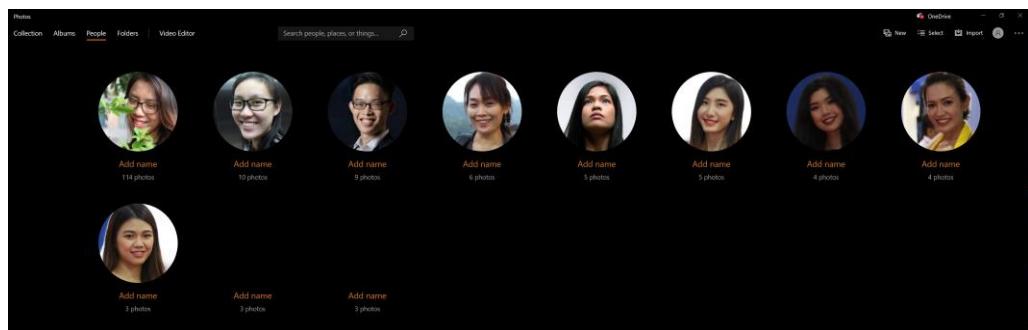


Figure 2.10: Microsoft Photos People menu

Furthermore, Microsoft Photos has several unique options when it comes to their search function which includes “People” which are photos whom faces had been categorized, location where the image or file was taken, “Things” which are tags that users have placed whenever they import images or files, highlighted which are favourites , images with text and recent images.



Figure 2.11: Microsoft Photos Search menu

2.5 Adobe Photoshop Lightroom Classic

Adobe Photoshop Lightroom Classic otherwise known as Adobe Lightroom is a creative photo manipulation and storage software developed by Adobe incorporated. First released in 2007, the software transformed the world of professional and hobby photography. This is due to the fact of its streamlined method of storage management and photo manipulation system enabling mass manipulation at once.

Adobe Lightroom Classic is considered to be a professional grade all-in-one software where users are able to edit, organise and sort their photo images in depth. Furthermore, Adobe Lightroom also incorporates many other functions that are needed by professionals in the photography industry. Aside from editing the images themselves, users will also be able to create slideshows, tag the photos, and create photobooks as well as high quality format printing. When users first launched the software, users will be greeted with a grey page with a dark coloured user interface. In Adobe Lightroom, there are multiple pages with different functions. By default, users will first enter the “Library” page which can be considered the main menu, there are side panels on all sides. On the top panel, the left side of the panel will showcases the logo and name of the software as well as the username of the user, on the right side of the top panel is the tab selection sections. The tabs that are included are “Library”, “Develop”, “Map”, “Book”, “Slideshow”, “Print” and “Web”. On the left

navigation panel, it displays the selected images, the catalogues as well as the folders within the user's machine. For the bottom navigation panel, it displays a carousel of images imported into the software with a number of miscellaneous utilities. The side navigation panels changes according to the functions of each tab.

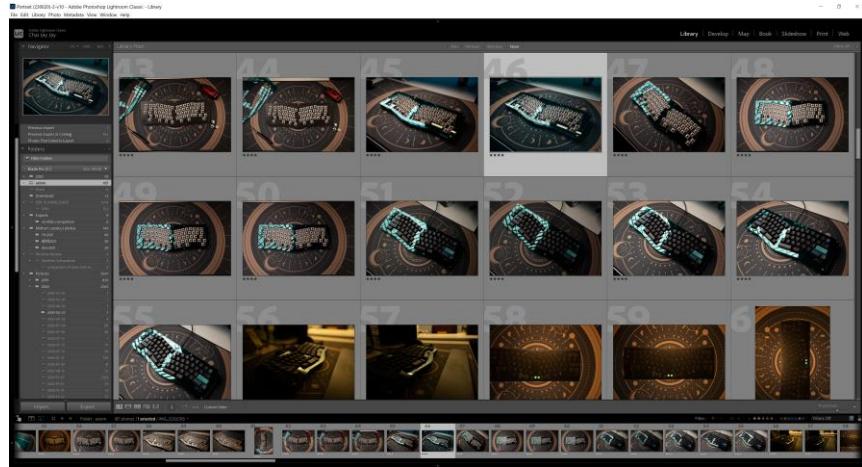


Figure 2.12: Adobe Lightroom Classic Main menu

In Adobe Photoshop Lightroom Classic, there are several noteworthy functions that make the software unique as a organizer. The first noteworthy functions comes from its ability to sync with cloud, Adobe incorporated implements a cloud system which the users of the software may upload their work and able to sync with any devices that are able to utilize their software. The uniqueness of this function is that if the user decided to upload their photo with edited settings to the cloud, they will be able to access that photo and its settings via any other device that are able to utilize Adobe Photoshop Lightroom Classic.

The next noteworthy Functions is the ability to catalogue imported images. The images that the users have imported no matter the quantity of the photo will be catalogued into a file, which consists of all the metadata linking to the imported images. This catalogue function is unique as it allows users to transfer the catalogue file into another machine along with the images and they will be able to continue working on the manipulation of images with the altered settings from the original machine.

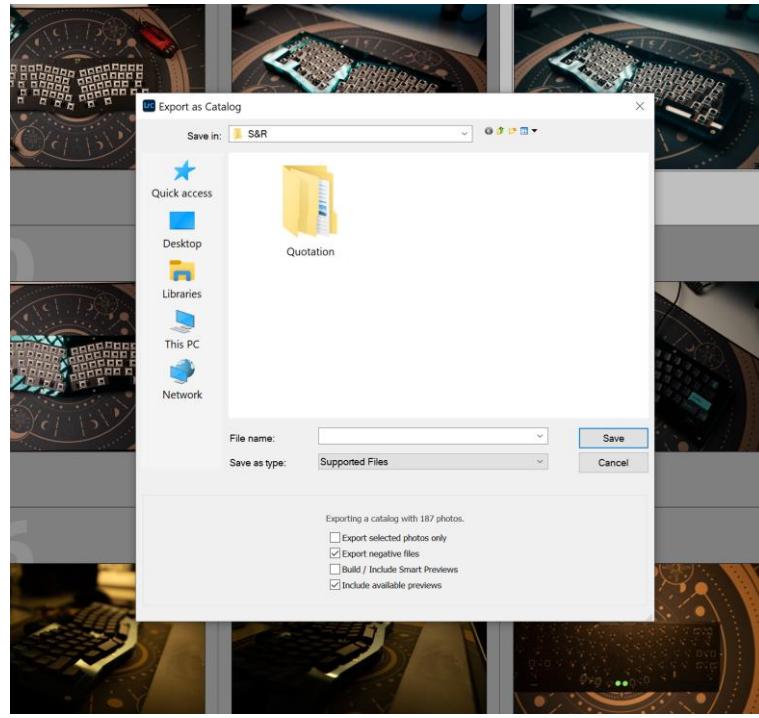


Figure 2.13: Adobe Lightroom Classic Exporting as a “Catalog”

Name	Date modified	Type	Size
axiom	10/25/2021 12:42 ...	File folder	
Test_catalogue Previews.lrdata	10/25/2021 12:42 ...	File folder	
Test_catalogue	10/25/2021 12:42 ...	Adobe Lightroom ...	24,264 KB

Figure 2.14: Adobe Lightroom Classic Catalogue Sample

Another Noteworthy function of Adobe Lightroom Classic is the ability to manipulate large quantities of photos at once. Adobe Lightroom Classic is extremely popular with professional photographers due to its flexibility and functionality, the ability to manipulate large quantities of photo at the same time did not exists at the time, photographers were forced to import a single photo and edit it and export it using alternative software. Adobe Lightroom Classic is able to import large quantities of photos, manipulate the images with a simplified User interface and the ability to export all or specific images at one time. Furthermore, in this software, users are able to create slide shows, photo books and prints with detailed settings that user may manipulate.



Figure 2.15: Adobe Lightroom Classic Develop Menu

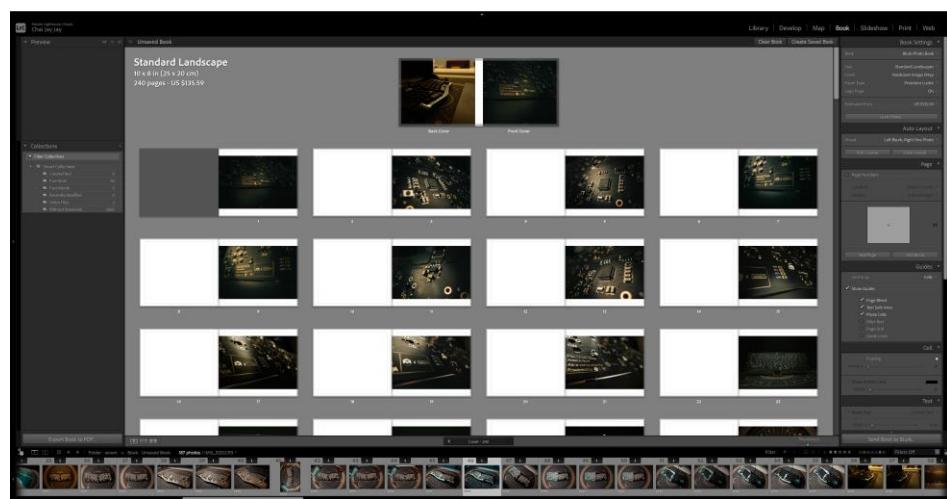


Figure 2.16: Adobe Lightroom Classic Book Menu



Figure 2.17: Adobe Lightroom Classic Slideshow menu

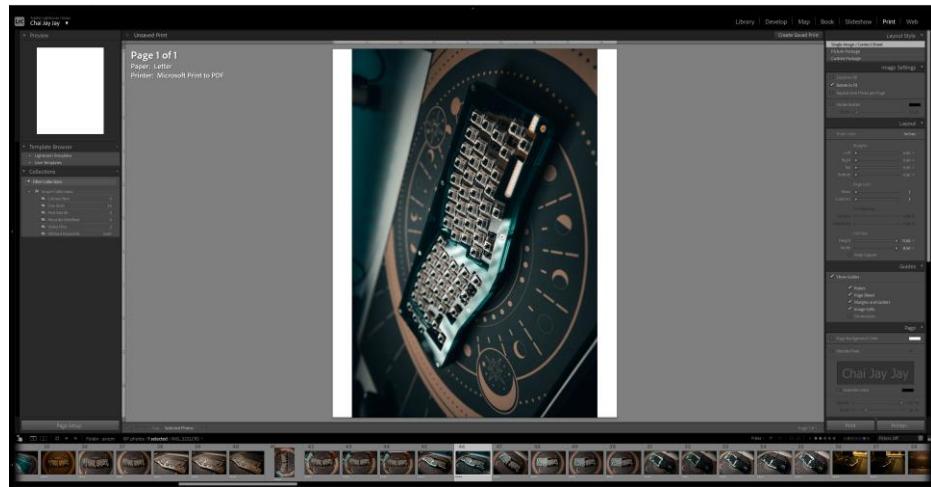


Figure 2.18: Adobe Lightroom Classic Print menu

Adobe Photoshop Lightroom Classic is a subscription based software to Adobe Creative Cloud. When users subscribed to Adobe Creative Cloud, there are several packages offered, the first being the Photography Package that also offers several packages. The first package offer is the Photoshop Lightroom Classic package; it comes with Adobe Photoshop Lightroom, Adobe Photoshop, and Adobe Photoshop Lightroom Classic with 20 GB of cloud storage at 9.99 US dollars per month. The next package will be the same as the first package with the differences being it is offered at 1 TB of cloud storage at 19.99 US dollars per month, whereas if users wish to subscribe to the standalone software, the Adobe Photoshop Lightroom with 1TB of cloud storage is offered at 9.99 US dollars per month.

2.6 FastStone Image Viewer

Adobe once FastStone Image Viewer is an image viewer, browser, manager and lightweight editor. FastStone Image Viewer is free for any personal and educational usage; this makes it one of the most accessible Image organizer/viewer. FastStone Image Viewer is free on Microsoft Windows OS only.

FastStone Image Viewer offers a Windows Explorer-like user interface. When users first launch the software, they will be able to see a simple yet complicated user interface. On the left side of the screen, they will be able to see a directory navigation bar where they will be able to select the folder containing images that users wish to organize. On the top of the screen, users will see 2 separate rows of navigation bar filled with icons which are buttons with various usage.

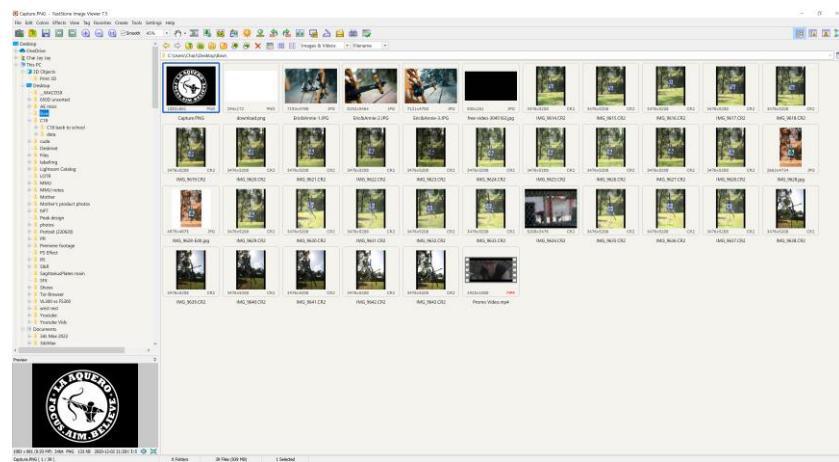


Figure 2.19: FastStone Image Viewer Main Menu

FastStone image viewer are able to read in this software, users will be able to do a number of task. First of which is ability to manipulate images. The functions to manipulate image includes blurring selected areas of the image, resizing, cropping, rotating, sharpening, brightening, adjusting colours, red eye removal or reduction, drawing objects on images. Furthermore, in terms of image manipulations, FastStone Image Viewer also includes special and colour effects such as grey scale, negative, adding drop shadow etc.

FastStone image viewer also offers features in organization of the images. One of these features are batch processing where users will be able to convert or rename large collections of images at will. Furthermore, the software also features file-tagging

and drag-and-drop functions to organize and sort through the images. To assist in the process of organization of images, the software includes a small preview screen in the bottom left of the UI for easier viewing, the software also support a dual-monitor configurations and image compare functions for easier viewing and comparison of the images.

Furthermore, FastStone Image Viewer has other functions that can be notable for users. One of which includes the ability to create Slideshows builder, Contact sheet builder, Multi-page file builder, Multi-page file splitter. All of these functions are usable when It comes to documentation as well as presentation.

Overall FastStone image viewer is an easy to use and very accessible software for beginners thanks to its simple design and abundant of functions.

Table 2-1: Comparison between software

Functions	Comparison					
	Amazon Photos	Google Photos	Microsoft Photos	Adobe Photoshop Lightroom Classic	FastStone Image Viewer	Photography Assistant
Photo Storage	Able to store up to subscribed storage size or only 5GB by default	Able to store up to subscribed storage size or only 5GB by default	Able to store up to the limit of user own storage device	Not able to store, but able to export as a catalogue with manipulated settings	Not able to store	Not able to store
Scan for images older than set date	No	No	No	No	No	Yes
Redundant photo detection	Yes	Yes	No	No	No	Yes
Photo manipulation	No	No	Yes, simple manipulation	Yes , in detail	Yes	No
Printing	Yes	Yes	Yes	Yes	Yes	No
Subscription Service	Yes	Yes	No	Yes	No	No
Free Option	Yes but limited to 5GB storage	Yes but limited to 5GB storage	Yes	Yes, 30 days trial only	Yes	Yes
Availability in Malaysia	Yes	Yes	Yes	Yes	Yes	Yes

CHAPTER 3

Methodology

3.1 Rapid Application Development (RAD)

For this project, the author has decided to use Rapid Application Development otherwise known as RAD. RAD is a methodology created by taking inspiration from Systems Development Life Cycle in the process for creating a system or software. RAD emphasize on rapid prototyping and receiving feedback from end users, this allows for constant update and frequent creation of new iteration, indirectly allows for flexibility in the development process in changes and lowering the costs of planning.

The figure 3.1 below display the flow and describes the flow and process of the Rapid Application Development Methodology.

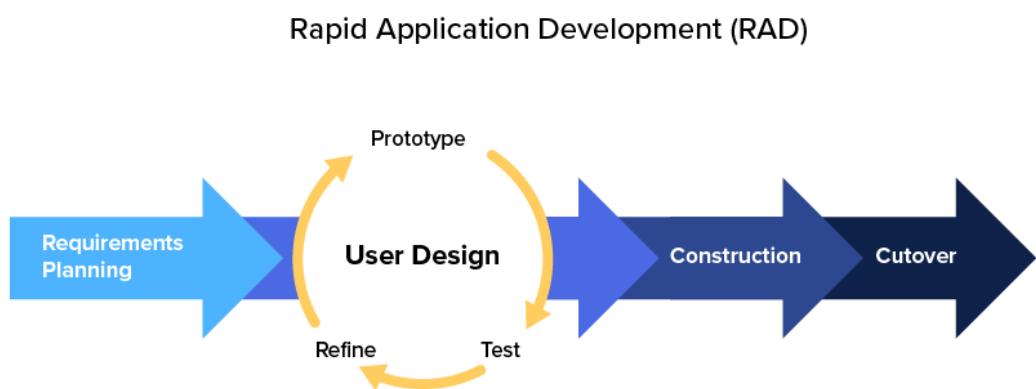


Figure 3.1: Rapid Application Development

3.2 Requirements Planning

In this stage, developers are required to conduct preparation and meetings with the lecturer in order to discuss the necessary requirement for the project, the requirements include the language. Once those have been discussed, a project overview, scope, problem statements, and objectives will be defined in detail. Once those are complete, the developer must conduct a benchmark analysis will be conducted where the various functions and system designs are analysed and discussed and compared with the developers design. Communications between the developer and the customer that in this case is the lecturer must be continuous to ensure changes can be conducted if needed.

3.3 User Design

In the User design phase, once all the details in requirement planning stages has been confirmed and finalised, the developer will start to develop prototypes where users will constantly test it to find any error or bugs. A constant and continuous communications between developer and end user is key in this stage as it is one of the critical phases in the development. The reason it is a critical phase is because changes are constantly done in this phase, once the developer has created a prototype , it will then be passed on to the end user to test and search for bugs to be fixed or lack of functionality to be added.

3.4 Rapid construction

In the rapid construction phase, developers will take the designs and prototypes and convert them into working models. This process is primarily focused the programming, coding, testing and implementation of the system. Furthermore, the customer will also request for changes based on their experience and findings until the final working model meets the expectations and desire.

3.5 Cutover

The cutover stage is the final stage of the RAD methodology. In this phase, the developers will implement the final products. This completes by finalising the

functions, features, and user interfaces of the product meets the requirements of the project scope. Furthermore, maintenance and bug search process are also continued in this phase while the implementation process is underway to ensure a satisfying final product.

3.6 Tool, Techniques and Programming Languages

3.6.1.1 Hardware Requirement

- Central Processing Unit

With the use of algorithm in the system as well as image processing and analysis, it is recommended for users to use at least a quad core 64-bit CPU with clock speed of at least 2.4 GHz for maximum efficiency in the processing process

- Memory

With the intensity that is expected from the usage of the CPU, it is recommended for users to have a minimum of 8GB for their RAM storage in order to maximise the efficiency of their CPU performance

- Storage

With the software main function scanning the files within the storage, a Hard Disk Drive are not recommended as the waiting time will be impractical in a real life usage, hence a Solid State Drive is recommended for users to fully maximize the efficiency of the process as a SSD is exponentially faster than the HDD counterpart.

3.6.1.2 Software Tools

- Visual Studio Code

Visual Studio Code otherwise known as VS code is a source code editor with plenty of useful functions. These functions are used to ease the developers in their coding process with functions such as auto-indentations, highlights, and bracket matching. VS code has a large number of third party extension, which the user can install to ease their coding process.

- Qt Designer

Qt Designer is a Qt tool primarily used to design and build graphical user interface using Qt Widgets. The method of usage is simplified as it consists of drag and drop of majority of graphical parts of the program.

3.6.1.3 Programming Languages

- Python

Python is a high level programming language that is widely used in many applications and many fields. These applications includes Graphical user interface developments, data analysis and game development. Python is widely used and well supported with many libraries to assist with any requirements needed

3.6.1.4 Library/Module

- OpenCV

OpenCV otherwise known as Open Source Computer Vision Library is an open-sourced free-to-use library of programming functions that is mainly used for computer vision. One of the main applications and functions is the ability to use GPU acceleration for real-time operations as well as object detection

- OS

OS is a module that provides a way to use the functionalities which are dependent on the operating system of the user. The limitation is limited to the design of the built-in function of the operating system.

- PyQt5

PyQt5 is comprehensive set of Python bindings for Qt which is also a cross-platform GUI library used to develop python interface. PyQt5 is free and is developed by Riverbank Computing from Britain. From

within this library, the system will extract several module which are QtGui, QtCore, and QtWidgets

- datetime

Datetime is a module built in python which allows the system to work with dates as date objects.

- sys

sys is a module that allows access to variables used or maintained by the interpreter as well as functions connected to the interpreter.

- SciPy

SciPy is a free and open-source Python library that is mainly used for Scientific and technical computing. Some of the functions that makes SciPy needed is its ability for reading images, image segmentation, face detection as well as feature extraction.

- Numpy

Numpy is an open sourced library used to work with arrays. Its functions work with linear algebra and matrices which will be used.

3.7 Gantt Chart

The Gantt Charts Illustrates the process for Phase 1. For the first phase that is requirement analysis, it is conducted for a span of 3 weeks starting with discussion with supervisor in regards for a project title, proceeded with proposal submission and approval confirmation. In the 2nd week, the author defined the project overview, followed by defining problem statement, objectives and scope continuing in the 3rd week, The author also conducts benchmark review in the 3rd week as well. In the 4th week, the author proceeds with system design phase, where the author began with preparation of a Gantt chart, followed by context diagram, level-1 data flow diagram, use case diagram and user activity diagram for 5 weeks in total, last the author

proceeded to design the user interface of the prototype for the rest of the available weeks.

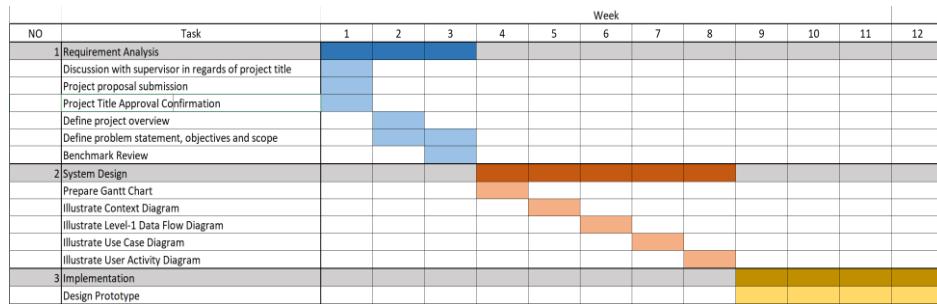
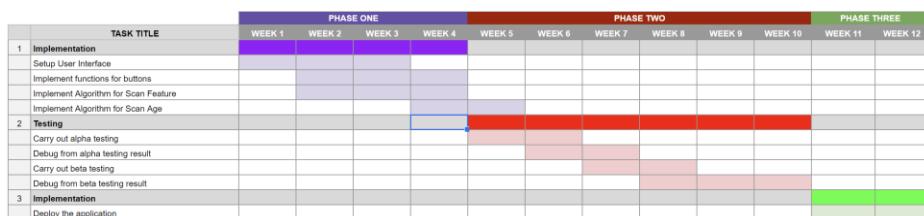


Figure 3.2: Gantt chart (Phase 1)

The Gantt Charts Illustrates the process for Phase 2. There are 2 main stages in this phase, with them being Implementation stage and Testing stage. In the first stage, the first activity is setting up the user interface for a span of 3 weeks overlapping other activity. The second activity is implementing functions for buttons which overlaps with other activity as well. The third and fourth stage implements algorithms into Scan Feature button and Scan Age button which overlaps as conducted for 3 weeks and 2 weeks respectively. In the 2nd stage which is testing, the author starts with alpha testing followed by debugging overlapping each other for 1 week and conducted for 2 weeks respectively. Once Alpha testing and debugging is complete, the author proceeded with beta testing and debugging for 2 weeks and 3 weeks respectively, the 2 activity overlaps each other for 1 week. Lastly is deployment of application for last 2 final weeks.



CHAPTER 4

IMPLEMENTATION PLAN / DESIGN

4.1 Proposed Solution Overview

Photography Assistant is a software application introduced to solve the project. The application is designed primarily as a photography management software for professional photographers who frequently utilize the burst function of their cameras. The results in such actions are large quantities of images with numbers being anywhere from 20 to 300 images in a time span of 2 to 3 seconds of utilizing the function. With the software, users will be able to find the best photos based on a number of criteria including contrasts and blurriness. Furthermore often more than never, photographers tends to store large quantities of images and occasionally forget about the existence of said photos after a certain period of time, the software is able will be able to detect files that was last accessed which the time period may be set by user.

4.2 Use Case Diagram

Use case diagram will display the primary uses and interaction between the necessary parties with the software in order to achieve their goals. There will be a single user interacting with the software in order to achieve the user's goals. As per displayed in the User Case Diagram below, the user will be able to interact in a few manner. The first of which is that the user will be able to import images into the software, then the user will then set threshold level for the algorithm. Once the threshold is set, user will be able to instruct the software to go through each image and scan for their feature using the given algorithm, once the scanning process is completed, the program will then sort them into 4 different folders. Furthermore, the user will be able to check for old files, the user will be able to set a date, which the software will scan for files older than the set date and sort and categorize the folders into 2 different folders being older and newer.

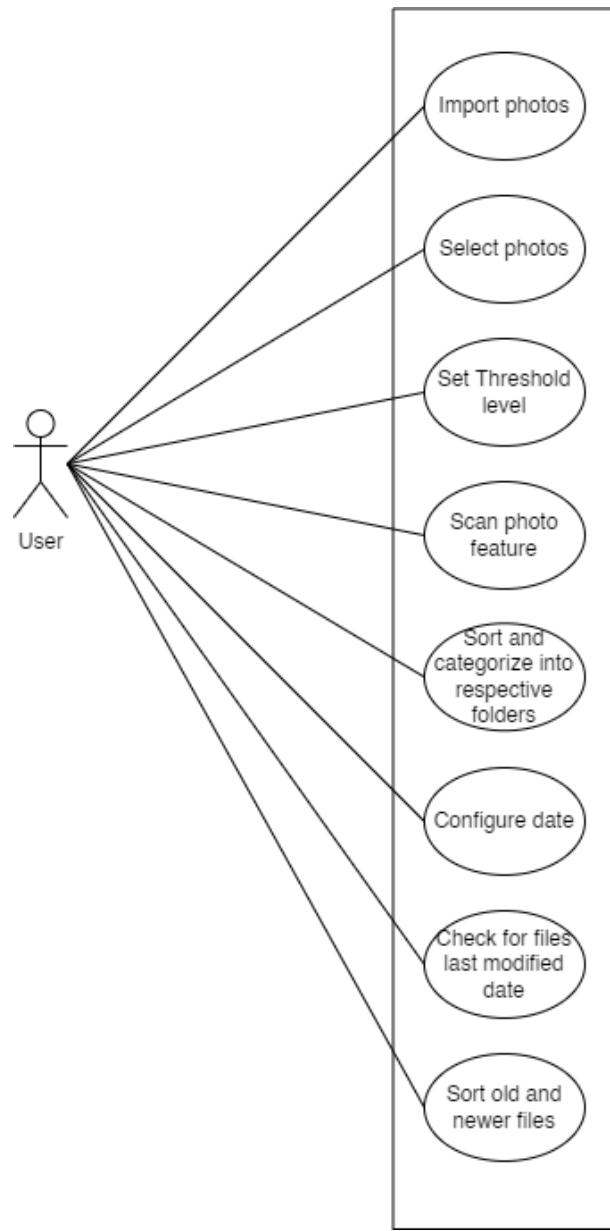


Figure 4.1: Use Case Diagram

4.3 User Activity Diagram

User Activity diagram is a diagram that displays the coordination and flow of activities and actions that can be taken by the users. According to figure 4.2 shown below, when users launch the software for the first time, users will be greeted by the software's main window where users will be able to perform 2 functions. In the main menu user may import a directory of the user's choosing consisting of photos, then the user may select the images they wished to be analysed and set the threshold level for the algorithm, then the software will scan the images selected and sort the images into

respective folders. Next, the other functions are Scan Age. This function allow users to set a date and initiate the software-scanning feature to detect files older than the set date and sort the files into respective folders.

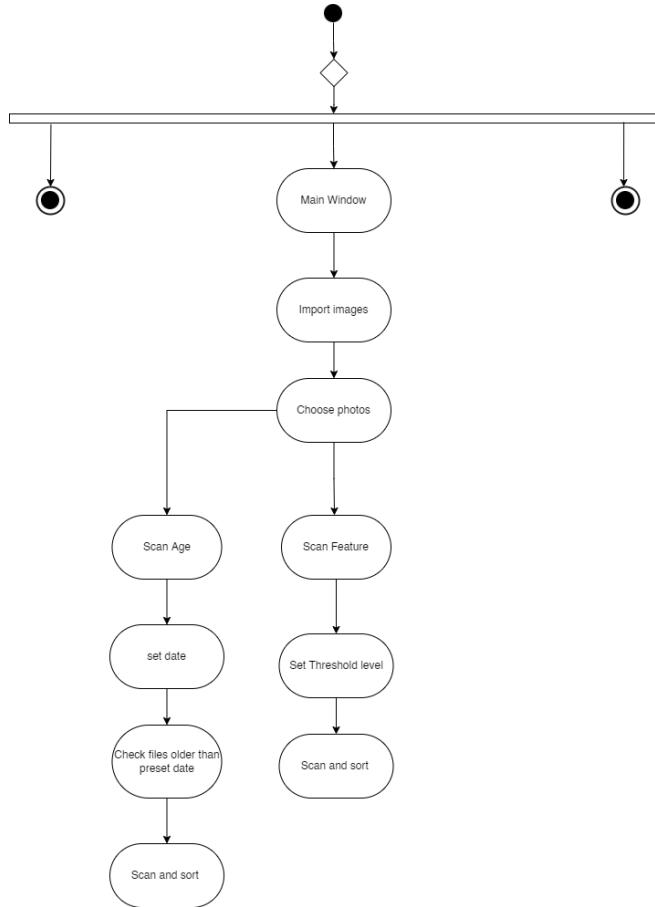


Figure 4.2: User Activity Diagram

4.4 Context Diagram

Context Diagram is a diagram that displays the relationship of the entire system with external entities. In this context diagram show in Figure 4.3, the author has showed that the primary interaction between the system and the external entity that in this case is users of the system is that users will provide images or photos in folders to the system that in return the system will analyse and sort the images into respective folders. Furthermore, users will provide the selection of photos as well as desired threshold level and date for the software system to process and return sorted images into respective folders

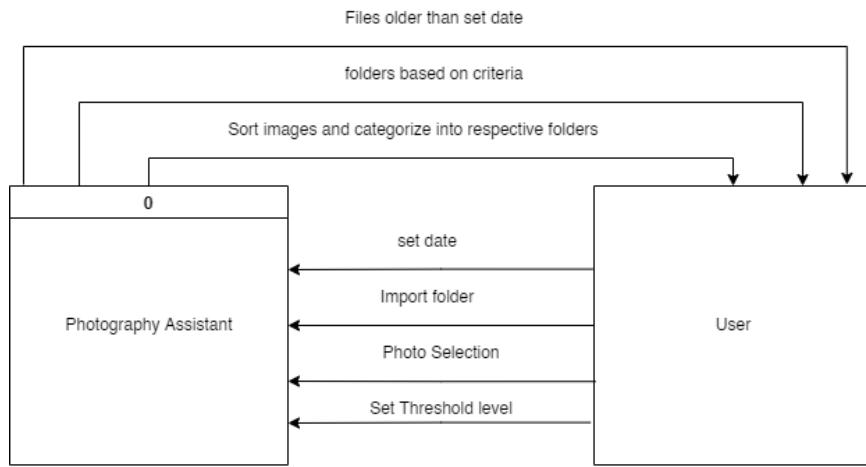


Figure 4.3: Context Diagram

4.5 Data Flow Diagram

Data Flow Diagram is a diagram that displays path of information flow through a system. In figure 4.4 shown below, it is shown that when users first launch the software, they will be directed into the main window. In the main menu, users may be able import existing directory which are folders containing images which are to be processed. Once this is completed the user will have 2 choices, the first of which scan feature of the selected images. Once the users have selected their desired images and selected the scan feature button, they will be able to input the threshold value for processing, once the processing is complete, the software will then sort and categorize the images into 4 different folders based on their features which are contrast and blurriness. The alternative function is scan age. Once users have selected their desired images and click on the button, a menu will prompt similar to scan feature button. In this dialog box, users will input the desired date for processing, the system will scan the image's last modified date and sort them into 2 different folders based on the date given.

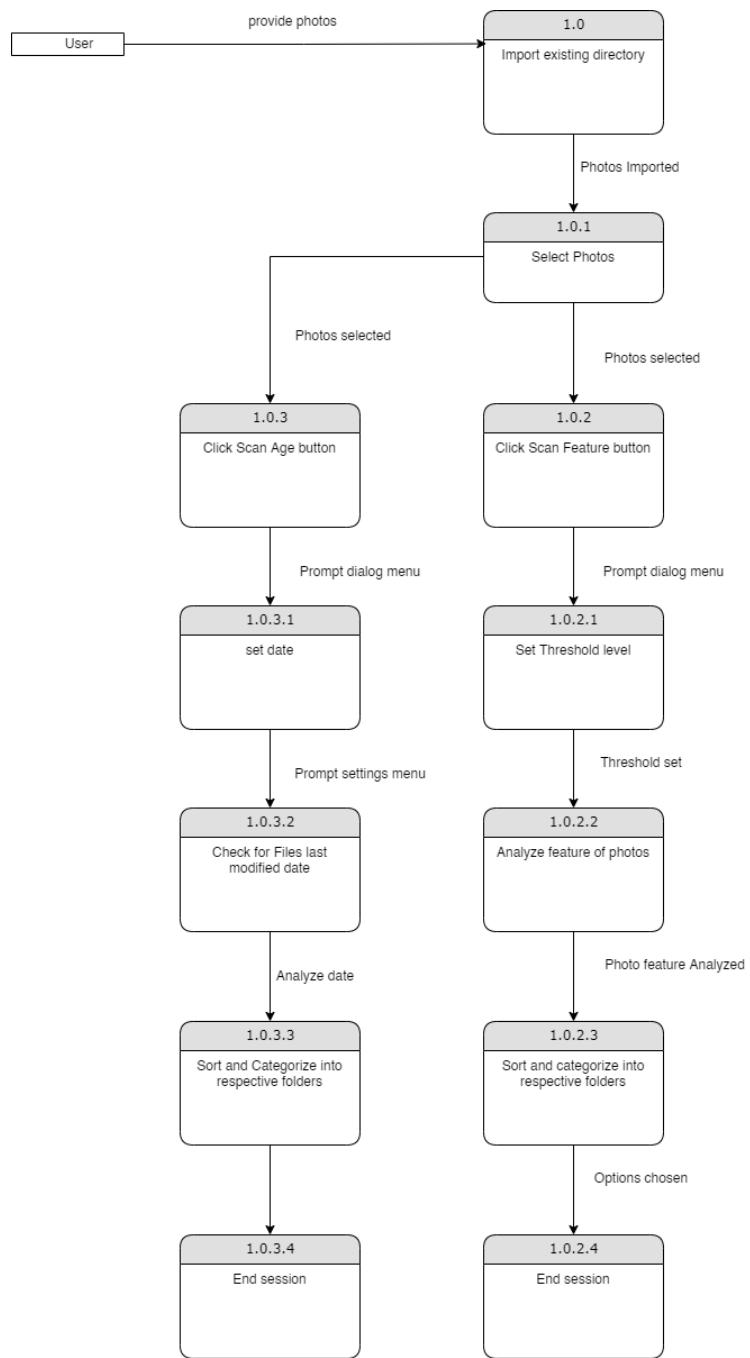


Figure 4.4: DFD Level-1 diagram

4.6 Interface Design

In this subsection of the report, it showcase the idea for the design of the software, the main idea for the designs is to be straightforward and simple, with clear icons and text to show and indicate functions their functions.

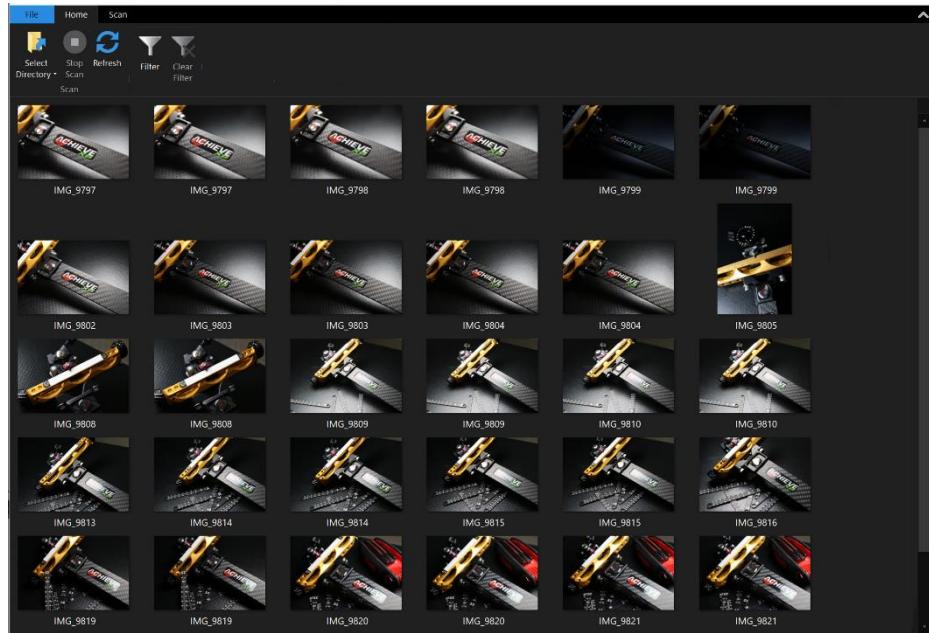


Figure 4.5: Photography Assistant Main Menu

The Home menu doubles as a menu for Best image scanning menu. In this menu, there will be a top navigation bar for the user to interact. In the top navigation bar, the users will be able to select directories to import the photos they wish to import using the select directory button. Furthermore, users will be able to refresh the system to clear the imported images using the “Refresh” button. Furthermore, users will be able to input their scanning criteria in the “Filter” menu. When the user select the “Filter” button, a small menu will prompt out for users to set their criteria, the criteria include Contrast level and Blurriness level.

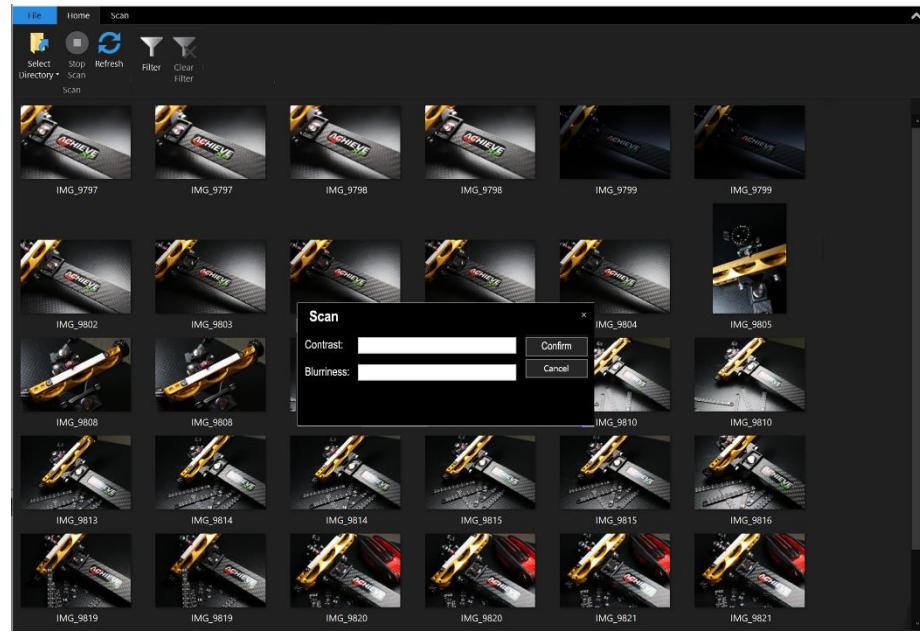


Figure 4.6: Photography assistant main menu with filter option



Figure 4.7: Photography Assistant Scan menu filter option

For the Scan menu, there will also be a top navigation bar. The top navigation bar is similar to the Home menu with “Select Directory” button, “Start scan” and “Stop scan” button, “Refresh” button. However, with the Scan menu, there will be additional “Date Settings” and “Clear Date” button. The function of these button are as the name suggests, when users select the “Date Settings” button a small menu will be prompted where users will be able to select the date for filtering, the “Clear Date” button will allow the user to clear the date previously set by users

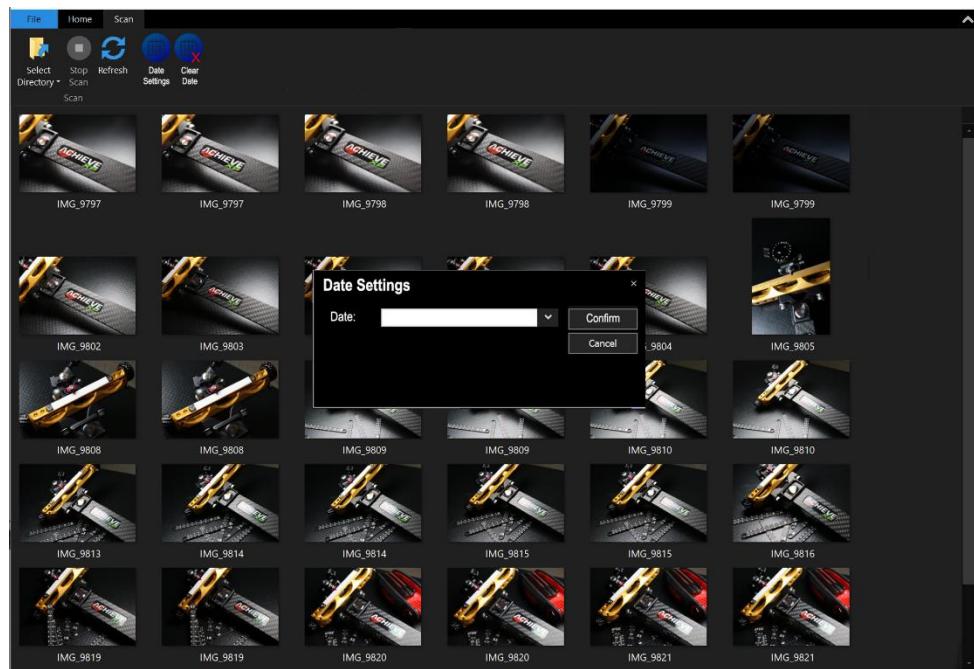


Figure 4.8: Photography Assistant File Check menu with Date settings



Figure 4.9: Photography Assistant Date settings

4.7 Algorithm

4.7.1 Contrast detection

In order to measure contrast there are multiple method and ideas in order to do so such as measuring the difference between the brightest and darkest pixel of an image. The author has decided to calculate the standard deviation of the pixel of the image away from mean. In layman terms, it measures how “spread out” are the pixels of the image across the histogram. The method to implement this function is rather direct and straightforward. Utilizing the built in function of the operating system, we will be able to read the date of the image and calculate the standard deviation directly using the implementation showcased in figure. However do note that in order to simplify the process, conversion of image into grayscale is necessary.

```
# Load image
im = Image.open('image.png')

# Calculate statistics
stats = ImageStat.Stat(im)

for band, name in enumerate(im.getbands()):
    print(f'Band: {name}, min/max: {stats.extrema[band]}, stddev: {stats.stddev[band]}')
```

Figure 4.10 : Contrast Detection implementation method

4.7.2 Blurriness detection

In order to detect blurriness of an image, we must first understand the concept of blurriness. Blurriness refers to lack of sharpness in vision or in the case of this article is image. Thus the lower the sharpness of an image, the more blurry the image is. Hence in order to calculate the sharpness of an image, the author decide to use a simple method which is estimation by the average gradient magnitude. The algorithm reads an image and iteratively smooths the image gradually to present the reduction in sharpness. By calculating the average gradient magnitude of the image using the implementation method showcased in figure 4.11, the author is able to quantify the blurriness of the image. However do note that in order to simplify the process, conversion of image into grayscale is necessary.

```
im = Image.open(filename).convert('L') # to grayscale
array = np.asarray(im, dtype=np.int32)

gy, gx = np.gradient(array)
gnorm = np.sqrt(gx**2 + gy**2)
sharpness = np.average(gnorm)
```

Figure 4.11: Sharpness calculation Implementation method

CHAPTER 5

IMPLEMENTATION PROCESS AND RESULT

5.1 Overview

This chapter will be discussing the implementation steps and procedure taken to complete the project. This Chapter will be divided into 2 main subtopics, first of which will cover the codes used to create User interface and functionality of the software application. In another subtopic, we will be discussing the output of the previously discussed codes. The subtopics will include screenshots of the codes for better understanding.

5.2 Implementation Process

In the Implementation Process, it is primarily separated into UI design using PyQt5 as well as functionality programming. The UI includes 4 prompt dialog boxes including a progress bar dialog box, the functionality requires the program to be able to scan the selected images based on the contrast and blurriness and separate them into 4 different folders. Furthermore the program are required to be able to scan the last modified date and categorize the selected images based on said date into 2 files which are before and after files.

5.2.1 Main window user interface

```
from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(819, 555)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.verticalLayout = QtWidgets.QVBoxLayout(self.centralwidget)
        self.verticalLayout.setObjectName("verticalLayout")
        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.horizontalLayout.setObjectname("horizontalLayout")
        self.selectDirBtn = QtWidgets.QToolButton(self.centralwidget)
        self.selectDirBtn.setMinimumSize(QtCore.QSize(100, 70))
        self.selectDirBtn.setObjectName("selectDirBtn")
        self.horizontalLayout.addWidget(self.selectDirBtn)
        self.scanFeatureBtn = QtWidgets.QToolButton(self.centralwidget)
        self.scanFeatureBtn.setMinimumSize(QtCore.QSize(100, 70))
        self.scanFeatureBtn.setObjectName("scanFeatureBtn")
        self.horizontalLayout.addWidget(self.scanFeatureBtn)
        self.scanAgeBtn = QtWidgets.QToolButton(self.centralwidget)
        self.scanAgeBtn.setMinimumSize(QtCore.QSize(100, 70))
        self.scanAgeBtn.setObjectName("scanAgeBtn")
        self.horizontalLayout.addWidget(self.scanAgeBtn)
        spacerItem = QtWidgets.QSpacerItem(40, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)
        self.horizontalLayout.addItem(spacerItem)
        self.verticalLayout.addLayout(self.horizontalLayout)
        self.treeView = QtWidgets.QTreeView(self.centralwidget)
        self.treeView.setSelectionMode(QtWidgets.QAbstractItemView.ExtendedSelection)
        self.treeView.setObjectName("treeView")
        self.verticalLayout.addWidget(self.treeView)
        MainWindow.setCentralWidget(self.centralwidget)

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "Photographer Assistant"))
        self.selectDirBtn.setText(_translate("MainWindow", "Select Directory"))
        self.scanFeatureBtn.setText(_translate("MainWindow", "Scan Feature"))
        self.scanAgeBtn.setText(_translate("MainWindow", "Scan Age"))
```

Figure 5.1: Main Window User interface source code

In the above diagram, you may be able to see the source required to design the layout as well as placement of the required buttons and treeview panel in the Main menu user interface. The modules used from PyQt5 library includes QtCore, QtGui and QtWidgets. The buttons are set up using QToolButton from QtWidgets module. The buttons that includes “Select Directory”, “Scan Feature” and “Scan Age”. Using setWindowTitle function, the author has also set the title of the window of the menu, as well as labelling the buttons with their prospective names

5.2.2 Age Filter dialog box user interface

```
from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_Dialog(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Dialog")
        Dialog.resize(379, 67)
        self.verticalLayout = QtWidgets.QVBoxLayout(Dialog)
        self.verticalLayout.setObjectName("verticalLayout")
        self.dateTimeEdit = QtWidgets.QDateTimeEdit(Dialog)
        self.dateTimeEdit.setObjectName("dateTimeEdit")
        self.verticalLayout.addWidget(self.dateTimeEdit)
        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.confirmBtn = QtWidgets.QPushButton(Dialog)
        self.confirmBtn.setObjectName("confirmBtn")
        self.horizontalLayout.addWidget(self.confirmBtn)
        self.cancelBtn = QtWidgets.QPushButton(Dialog)
        self.cancelBtn.setObjectName("cancelBtn")
        self.horizontalLayout.addWidget(self.cancelBtn)
        self.verticalLayout.addLayout(self.horizontalLayout)

        self.retranslateUi(Dialog)
        QtCore.QMetaObject.connectSlotsByName(Dialog)

    def retranslateUi(self, Dialog):
        _translate = QtCore.QCoreApplication.translate
        Dialog.setWindowTitle(_translate("Dialog", "Dialog"))
        self.confirmBtn.setText(_translate("Dialog", "Confirm"))
        self.cancelBtn.setText(_translate("Dialog", "Cancel"))
```

Figure 5.2: Age Filter User Interface Source code

The above diagram showcase the source code for setting up the User interface for Age Filter prompt menu. The module used in PyQt5 library includes QtCore, QtGui and QtWidgets. There are several buttons included in the prompt menu which includes Confirm button and the Cancel button. The author has also set the name of the menu as well as labelling text for the buttons

5.2.3 Feature Filter dialog box user interface

```
from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_Dialog(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Dialog")
        Dialog.resize(379, 172)
        self.verticalLayout = QtWidgets.QVBoxLayout(Dialog)
        self.verticalLayout.setObjectName("verticalLayout")
        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.contrastSpinBox = QtWidgets.QDoubleSpinBox(Dialog)
        self.contrastSpinBox.setObjectName("contrastSpinBox")
        self.contrastSpinBox.setMinimumSize(QtCore.QSize(300, 0))
        self.contrastSpinBox.setMinimum(1.0)
        self.contrastSpinBox.setMaximum(100.0)
        self.contrastSpinBox.setObjectName("contrastSpinBox")
        self.horizontalLayout.addWidget(self.contrastSpinBox)
        self.verticalLayout.addLayout(self.horizontalLayout)
        self.horizontalLayout_2 = QtWidgets.QHBoxLayout()
        self.horizontalLayout_2.setObjectName("horizontalLayout_2")
        self.sharpnessSpinBox = QtWidgets.QDoubleSpinBox(Dialog)
        self.sharpnessSpinBox.setObjectName("sharpnessSpinBox")
        self.sharpnessSpinBox.setMinimumSize(QtCore.QSize(300, 0))
        self.sharpnessSpinBox.setMinimum(1.0)
        self.sharpnessSpinBox.setMaximum(10.0)
        self.sharpnessSpinBox.setObjectName("sharpnessSpinBox")
        self.horizontalLayout_2.addWidget(self.sharpnessSpinBox)
        self.verticalLayout.addLayout(self.horizontalLayout_2)
        self.noteLabel = QtWidgets.QLabel(Dialog)
        self.noteLabel.setObjectName("noteLabel")
        self.noteLabel.setText(_translate("Dialog", "Note: The lower the blurriness value, the blurrier the photo is."))
        self.label_2 = QtWidgets.QLabel(Dialog)
        self.label_2.setObjectName("label_2")
        self.label_2.setText(_translate("Dialog", "Blurriness: "))
        self.label_3 = QtWidgets.QLabel(Dialog)
        self.label_3.setObjectName("label_3")
        self.label_3.setText(_translate("Dialog", "1 ≤ Contrast ≤ 100"))
        self.label_4 = QtWidgets.QLabel(Dialog)
        self.label_4.setObjectName("label_4")
        self.label_4.setText(_translate("Dialog", "1 ≤ Blurriness ≤ 10"))
        spacerItem = QtWidgets.QSpacerItem(20, 10, QtWidgets.QSizePolicy.Minimum, QtWidgets.QSizePolicy.Fixed)
        self.verticalLayout.addItem(spacerItem)
        self.horizontalLayout_3 = QtWidgets.QHBoxLayout()
        self.horizontalLayout_3.setObjectName("horizontalLayout_3")
        self.confirmBtn = QtWidgets.QPushButton(Dialog)
        self.confirmBtn.setObjectName("confirmBtn")
        self.horizontalLayout_3.addWidget(self.confirmBtn)
        self.cancelBtn = QtWidgets.QPushButton(Dialog)
        self.cancelBtn.setObjectName("cancelBtn")
        self.horizontalLayout_3.addWidget(self.cancelBtn)
        self.verticalLayout.addLayout(self.horizontalLayout_3)
        self.retranslateUi(Dialog)
        QtCore.QMetaObject.connectSlotsByName(Dialog)

    def retranslateUi(self, Dialog):
        _translate = QtCore.QCoreApplication.translate
        Dialog.setWindowTitle(_translate("Dialog", "Dialog"))
        self.label.setText(_translate("Dialog", "Contrast: "))
        self.label_2.setText(_translate("Dialog", "Blurriness: "))
        self.noteLabel.setText(_translate("Dialog", "Note: The lower the blurriness value, the blurrier the photo is."))
        self.label_3.setText(_translate("Dialog", "1 ≤ Contrast ≤ 100"))
        self.label_4.setText(_translate("Dialog", "1 ≤ Blurriness ≤ 10"))
        self.confirmBtn.setText(_translate("Dialog", "Confirm"))
        self.cancelBtn.setText(_translate("Dialog", "Cancel"))
```

Figure 5.3: Scan Feature dialog box user interface source code

The diagram above showcase the source code for creating the user interface within Feature Filter prompt menu. The main library used was PyQt5 where QtCore, QtGui and QtWidgets modules was used. Primary widgets used within the user interface itself are 2 spinboxes for input which are contrastSpinBox and sharpnessSpinBox. There are also 2 buttons which are Confirm and Cancel as well as several labels which are used to label with notes for user to take note of during input stage.

5.2.4 Progress Bar dialog box user interface

```
from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_Dialog(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Dialog")
        Dialog.resize(78, 59)
        self.verticalLayout = QtWidgets.QVBoxLayout(Dialog)
        self.verticalLayout.setObjectName("verticalLayout")
        self.label = QtWidgets.QLabel(Dialog)
        self.label.setObjectName("label")
        self.verticalLayout.addWidget(self.label)
        self.progressBar = QtWidgets.QProgressBar(Dialog)
        self.progressBar.setProperty("value", 0)
        self.progressBar.setObjectName("progressBar")
        self.verticalLayout.addWidget(self.progressBar)

        self.retranslateUi(Dialog)
        QtCore.QMetaObject.connectSlotsByName(Dialog)

    def retranslateUi(self, Dialog):
        _translate = QtCore.QCoreApplication.translate
        Dialog.setWindowTitle(_translate("Dialog", "Dialog"))
        self.label.setText(_translate("Dialog", "Scanning in progress"))
```

Figure 5.4: Progress Bar user interface source code

The diagram above showcase the source code for creating the user interface within Progress Bar Prompt menu. The Main library used was PyQt5, the modules imported includes QtCore, QtGui, QtWidgets. The author utilised the built in function to setup a progress bar within the menu from QtWidgets.

5.2.5 Main Window Function Source Code

```
from PyQt5 import QtGui, QtCore, QtWidgets
from datetime import datetime
import numpy as np
import cv2
import sys
import os
import main_ui, progress_bar_ui, filter_feature_ui, filter_age_ui

> class MainWindow(QtWidgets.QMainWindow, main_ui.Ui_MainWindow):...

> class ProgressBarDialog(QtWidgets.QDialog, progress_bar_ui.Ui_Dialog):...

> class FilterFeatureDialog(QtWidgets.QDialog, filter_feature_ui.Ui_Dialog):...

> class FilterAgeDialog(QtWidgets.QDialog, filter_age_ui.Ui_Dialog):...

> if __name__ == '__main__': ...
```

Figure 5.5: Main Window function source code

The diagram above showcase the source code for setting up the functions for the software application. The main libraries used within the system are, PyQt5,

dateime, numpy, cv2, sys, os. However there are several modules imported from the listed libraries which include QtGui, QtCore, QtWidgets and datetime. The reason for extracting the datetime from datetime library is due to the design of the library itself. There are several classes created for the functions of the system. The classes are the following, “MainWindow” class, “ProgressBarDialog” class, “FilterFeatureDialog” class and “FilterAgeDialog” class.

5.2.5.1 Main Window initialization setup source code

```
class MainWindow(QtWidgets.QMainWindow, main_ui.Ui_MainWindow):
    def __init__(self):
        # Obligatory main setup
        super().__init__()
        self.setupUi(self)
        self.retranslateUi(self)

        # Variables initialization
        self.file_paths = []
        self.sharpness = 0
        self.contrast = 0
        self.dt = datetime.now()

        # Widget-related initialization
        self.feature_pbar_dialog = ProgressBarDialog(window_title='Scan Feature')
        self.age_pbar_dialog = ProgressBarDialog(window_title='Scan Age')
        self.filter_feature_dialog = None
        self.filter_age_dialog = None
        self.model = None

        # Widget configurations
        self.feature_pbar_dialog.hide()
        self.age_pbar_dialog.hide()
        self.selectDirBtn.setToolButtonStyle(QtCore.Qt.ToolButtonTextUnderIcon)
        self.selectDirBtn.setIcon(QtGui.QIcon('icons/select_folder.png'))
        self.selectDirBtn.setIconSize(QtCore.QSize(36, 36))
        self.scanFeatureBtn.setToolButtonStyle(QtCore.Qt.ToolButtonTextUnderIcon)
        self.scanFeatureBtn.setIcon(QtGui.QIcon('icons/features.png'))
        self.scanFeatureBtn.setIconSize(QtCore.QSize(36, 36))
        self.scanAgeBtn.setToolButtonStyle(QtCore.Qt.ToolButtonTextUnderIcon)
        self.scanAgeBtn.setIcon(QtGui.QIcon('icons/calendar.png'))
        self.scanAgeBtn.setIconSize(QtCore.QSize(36, 36))

        # Button function assignments
        self.selectDirBtn.clicked.connect(self.select_directory)
        self.scanFeatureBtn.clicked.connect(self.start_feature_scan)
        self.scanAgeBtn.clicked.connect(self.start_age_scan)

        # Show main window
        self.show()
```

Figure 5.6: Main Window initialization source code

The first class which is the “MainWindow” class has several defined functions which contributes to several functions of the system itself. The first defined function is “`_init_`”, It is mainly used to initialize variables, Widgets and connecting function class to the button in other User Interface Files.

5.2.5.2 Select Directory button class definition

```
def select_directory(self):
    # Open Window's directory selector
    dir_path = QtWidgets.QFileDialog.getExistingDirectory(self, 'Select Directory')

    # Display the tree-view
    if dir_path:
        self.display_treeview(dir_path)
```

Figure 5.7: “Select directory” button class source code

The 2nd function which can be found in the “MainWindow” class is the “select_directory” functions. Here the author programmed the functions to open existing directory or folders by calling “getExistingDirectory” method from PyQt5 and append to “dir_path”, which if “dir_path” returns true, the system will display

5.2.5.3 Treeview display class

```
def display_treeview(self, path):
    # Create the model to be applied to the tree-view
    self.model = QtWidgets.QFileSystemModel()
    self.model.setRootPath(path)

    # Get the index of the path to be displayed on the tree-view
    path_index = self.model.index(path)

    # Set the model of the tree-view and update the root index
    self.treeView.setModel(self.model)
    self.treeView.setRootIndex(path_index)
```

Figure 5.8: Source code used to display treeview

The 3rd function which can be found in the “MainWindow” class is the “display_treeview” function. In this function, following the rules of PyQt5 which requires the system to create an empty QFileSystemModel, the author set the rootpath of the model to the directory which the user will select. In order for the treeview to display the path of selected directory, index retrieved from the model of the path is required.

5.2.5.4 Get image path class

```
def get_image_paths(self):
    # Empty the file path list
    self.file_paths.clear()

    # Get the tree-view indexes of the selected images
    indexes = self.treeView.selectedIndexes()

    # Get the file path for each of the indexes and append to the file path list
    for index in indexes:
        file_path = self.model.filePath(index)
        if file_path not in self.file_paths:
            self.file_paths.append(self.model.filePath(index))
```

Figure 5.9: Source code for getting the path of each image

The 4th function which can be found in the “MainWindow” class is the “get_image_paths” function. In this function, the previously selected file has to be cleared in order to clear the selections for “ScanFeature” button function. Once user has selected images from the treeview, the system will then append the indexes of the selected images to “indexes” object which will then run a for-loop in order to the the filepath of the selected indexes.

5.2.5.5 Feature inputs dialog box class

```
def get_feature_inputs(self):
    # Prompt the dialog box
    sharpness, _ = QtWidgets.QInputDialog.getDouble(self, 'Set Filter Value', 'Sharpness: ', value=self.sharpness)
    contrast, _ = QtWidgets.QInputDialog.getDouble(self, 'Set Filter Value', 'Contrast: ', value=self.contrast)

    # Assign the sharpness and contrast value
    self.sharpness = sharpness
    self.contrast = contrast
```

Figure 5.10: Source code for “get_feature_inputs” method

The 4th function which can be found in the “MainWindow” class is the “get_image_paths” function. In this function, the previously selected file has to be cleared in order to clear the selections for “ScanFeature” button function. Once user has selected images from the treeview, the system will then append the indexes of the selected images to “indexes” object which will then run a for-loop in order to the the filepath of the selected indexes.

5.2.5.6 Progress bar function class

```

def update_feature_pbar_progress(self, current_index):
    self.feature_pbar_dialog.progressBar.setValue(int(current_index/len(self.file_paths)*100))

def update_age_pbar_progress(self, current_index):
    self.age_pbar_dialog.progressBar.setValue(int(current_index/len(self.file_paths)*100))

```

Figure 5.11: Source code for Scan Age and Scan Feature progress bar function

The 5th and 6th function which is defined in the “MainWindow” class are used to set up the function of the progress bar dialog box. The function is setup by using the formula: [(the current index) / length of total path selected] multiplied by 100. This formula is primarily used for the progress indicator as well as for the animation.

5.2.5.7 Scan Feature button function

```

def start_feature_scan(self):
    # Get the selected images' paths from the tree-view
    self.get_image_paths()

    # Get the user input for sharpness and contrast
    confirmed = FilterFeatureDialog(parent=self).exec_()

    # If the user clicks "confirm"
    if confirmed:
        if self.file_paths:
            # Display the progress bar dialog
            self.feature_pbar_dialog.show()

            # Create the separate folders to sort the images
            folder_names = [
                f'{self.chosen_dir}/blurriness_LT_{self.sharpness}_contrast_LT_{self.contrast}',
                f'{self.chosen_dir}/blurriness_MT_{self.sharpness}_contrast_MT_{self.contrast}',
                f'{self.chosen_dir}/blurriness_LT_{self.sharpness}_contrast_MT_{self.contrast}',
                f'{self.chosen_dir}/blurriness_MT_{self.sharpness}_contrast_LT_{self.contrast}'
            ]
            if not os.path.exists(folder_names[0]):
                os.mkdir(folder_names[0])
            if not os.path.exists(folder_names[1]):
                os.mkdir(folder_names[1])
            if not os.path.exists(folder_names[2]):
                os.mkdir(folder_names[2])
            if not os.path.exists(folder_names[3]):
                os.mkdir(folder_names[3])

            # Go through every image
            for i, file_path in enumerate(self.file_paths):
                # Skip if it is a folder
                if os.path.isdir(file_path):
                    continue

                # Get the file name
                file_name = file_path.split('/')[-1]

                # Open the image and convert it to gray-scale
                image = cv2.imread(file_path)
                image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

                # Calculate the image's sharpness
                sharpness = self.get_image_sharpness(image_gray)

                # Calculate the image's contrast
                contrast = self.get_image_contrast(image_gray)

                # Sort the images
                if sharpness < self.sharpness and contrast < self.contrast:
                    cv2.imwrite(f'{folder_names[0]}\\{file_name}', image)
                elif sharpness > self.sharpness and contrast > self.contrast:
                    cv2.imwrite(f'{folder_names[1]}\\{file_name}', image)
                elif sharpness < self.sharpness and contrast > self.contrast:
                    cv2.imwrite(f'{folder_names[2]}\\{file_name}', image)
                elif sharpness > self.sharpness and contrast < self.contrast:
                    cv2.imwrite(f'{folder_names[3]}\\{file_name}', image)

                # Update the progress bar value
                self.update_feature_pbar_progress(current_index=i)

                # Call this method to update the GUI thus showing the flowing effect of the progress bar
                QtWidgets.QApplication.processEvents()

            else:
                self.display_nfs_message()

            # Hide and reset the progress bar
            self.feature_pbar_dialog.hide()
            self.feature_pbar_dialog.progressBar.setValue(0)

            # Clear the selection
            self.treeView.clearSelection()

```

Figure 5.12: Source Code for “Scan Feature” button function

The above diagram showcase the function “start_feature_scan”, this function is ran after user pressed confirm their setting for feature scanning. The function start by running “get_image_paths” function to get the path of selected images. We then proceed to if the dialog’s confirm button is pressed it will return a 1 to execution function. We then run an If else statement, if the user confirmed and if the images is selected they will show progress bar, then the system will proceeded to create 4 folders in the selected directory if they do not exist. Once the folders has been completed the system will then run through each image selected and run the algorithm and sort them matching the criteria set by user. Once the process is complete, the progress bar will reset and hide, and the selection in the treeview will be cleared.

5.2.5.8 Image Blurriness detection method

```
def get_image_sharpness(self, image):
    # Convert the image object to a numpy array to perform calculations
    image_array = np.asarray(image, dtype=np.int32)

    # Calculate the image's sharpness
    gy, gx = np.gradient(image_array)
    g_norm = np.sqrt(gx ** 2 + gy ** 2)
    sharpness = np.average(g_norm)

    return sharpness
```

Figure 5.13: Source code for Sharpness detection algorithm

The above diagram showcase the source code for the function “get_image_sharpness”, this function primarily consists of the algorithm used to obtain the sharpness which will be used to measure the blurriness of the image, as the lower the sharpness of the image is equivalent to higher blurriness.

5.2.5.9 Image Contrast detection method

```
def get_image_contrast(self, image):
    # Return image's contrast
    return image.std()
```

Figure 5.14: Source code for Contrast detection algorithm

The above diagram showcase the source code for the function “get_image_contrast”, this function primarily consists of the algorithm used to calculate the contrast of each images.

5.2.5.10 Age Scan button function

```
def start_age_scan(self):
    # Get the selected images' paths from the tree-view
    self.get_image_paths()

    # Get user input for the datetime
    confirmed = FilterAgeDialog(parent=self).exec_()

    # If the user clicks "confirm"
    if confirmed:
        if self.file_paths:
            # Display the progress bar dialog
            self.age_pbar_dialog.show()

            # Create the separate folders to sort the images
            folder_names = [
                f'{self.chosen_dir}/before_{self.dt.strftime("%d-%m-%Y-%H-%M-%S")}',
                f'{self.chosen_dir}/after_{self.dt.strftime("%d-%m-%Y-%H-%M-%S")}'
            ]
            if not os.path.exists(folder_names[0]):
                os.mkdir(folder_names[0])
            if not os.path.exists(folder_names[1]):
                os.mkdir(folder_names[1])

            # Go through every image
            for i, file_path in enumerate(self.file_paths):
                # Skip if it is a folder
                if os.path.isdir(file_path):
                    continue

                # Get the file name
                file_name = file_path.split('/')[-1]

                # Open the image |
                image = cv2.imread(file_path)

                # Get the image's last-modified date
                timestamp = os.path.getmtime(file_path)
                dt = datetime.fromtimestamp(timestamp)

                # Sort the images
                if dt < self.dt:
                    cv2.imwrite(f'{self.chosen_dir}\\{folder_names[0]}\\{file_name}', image)
                elif dt > self.dt:
                    cv2.imwrite(f'{self.chosen_dir}\\{folder_names[1]}\\{file_name}', image)

                # Update the progress bar value
                self.update_age_pbar_progress(current_index=i)

                # Call this method to update the GUI thus showing the flowing effect of the progress bar
                QtWidgets.QApplication.processEvents()
            else:
                self.display_nfs_message()

            # Hide and reset the progress bar
            self.age_pbar_dialog.hide()
            self.age_pbar_dialog.progressBar.setValue(0)

            # Clear the selection
            self.treeView.clearSelection()
```

Figure 5.15: Source Code for “Scan Age” button function

The above diagram showcase the function “start_age_scan”, this function is ran after user pressed confirm their setting for age scanning. The function start by running “get_image_paths” function to get the path of selected images. We then proceed to show to the progress bar dialog box and create separate folders to sort the images which for age scanning is 2 different folders of before and after the specified date set by users. We then go through every image’s last modified date and getting the time stamp, we then sort the images into respective folders and hide the progress bar and reset it and clear the selection of images in the treeview. In the event that there are no images selected, the system will run “display_nfs_message” which will display a error message.

5.2.5.11 Error message

```
def display_nfs_message(self):
    QtWidgets.QMessageBox.about(self, 'No Image Selected', 'You have not selected an image to be processed')
```

Figure 5.16: Source Code for error message

The above diagram showcase the source code for the function “display_nfs_message”, this function primarily used for display an error message box stating “ You have not selected an image to be processed” with “No image selected” as the prompt menu title. This function is primarily used in the events that there no images selected by users.

5.2.6 Progress Bar dialog menu

```
class ProgressBarDialog(QtWidgets.QDialog, progress_bar.ui.Ui_Dialog):
    def __init__(self, window_title=''):
        # Obligatory main setup
        super().__init__()
        self.setupUi(self)
        self.retranslateUi(self)

        # Set the title of the window
        self.setWindowTitle(window_title)

        # Prevent interaction on the parent window
        self.setWindowModality(QtCore.Qt.WindowModality.ApplicationModal)

        # Show the dialog
        self.show()
```

Figure 5.17: Source code for setting up Progress bar dialog window

The above diagram showcase the source code for the class “ProgressBarDialog”, this function primarily used to set up the progress bar dialog box. The source code are similar with any other setup for any menu with the exception of setting the modality of the window to be unable to interact with parent window. This is done so to prevent any accidental exit during running of program.

5.2.7 Feature Filter Dialog menu

```
class FilterFeatureDialog(QtWidgets.QDialog, filter_feature.Ui_Dialog):
    def __init__(self, parent=None):
        # Obligatory main setup
        super().__init__()
        self.setupUi(self)
        self.retranslateUi(self)

        # Set the title of the window
        self.setWindowTitle('Scan Feature')

        # Variables initialization
        self.parent = parent

        # Button function assignments
        self.cancelBtn.clicked.connect(self.reject)
        self.confirmBtn.clicked.connect(self.return_values_to_parent)

        # Widget configurations
        self.initialize_spinbox_value()

    def initialize_spinbox_value(self):
        # Initialize the values of the spinbox
        if self.parent:
            self.sharpnessSpinbox.setValue(self.parent.sharpness)
            self.contrastSpinbox.setValue(self.parent.contrast)

    def return_values_to_parent(self):
        # Return the contrast and sharpness to the parent (MainWindow)
        if self.parent:
            # Get the sharpness and contrast from the spin-boxes
            sharpness = self.sharpnessSpinbox.value()
            contrast = self.contrastSpinbox.value()

            # Pass the values to the parent (MainWindow)
            self.parent.sharpness = sharpness
            self.parent.contrast = contrast

            # Close the dialog
            self.accept()
```

Figure 5.18: “Scan Feature” Dialog box menu source code

The above diagram showcase the “FilterfeatureDialog” class which is the function class for Filter Feature dialog. In this source code, the dialog menu was first initialized then set up with their window title “Scan Feature”. Then initialize the parent variable. Once that has been completed, Confirm and cancel buttons were assignment with their respective functions. We then move onto initializing the spinbox which is a widget’s value. We set up the function “initialize_spinbox_value” by passing the initialize contrast and sharpness variables from “MainWindow” class into the spinboxes, this is done to preserve the value that has been inputted by users previously. Once that has been completed, we move onto “return_values_to_parent” class where we will pass the contrast and sharpness input onto parent classes for running of the algorithm later on.

5.2.8 Age Filter Dialog menu

```
class FilterAgeDialog(QtWidgets.QDialog, filter_age.ui.Ui_Dialog):
    def __init__(self, parent=None):
        # Obligatory main setup
        super().__init__()
        self.setupUi(self)
        self.retranslateUi(self)

        # Set the title of the window
        self.setWindowTitle('Scan Age')

        # Variables initialization
        self.parent = parent

        # Button function assignments
        self.cancelBtn.clicked.connect(self.reject)
        self.confirmBtn.clicked.connect(self.return_values_to_parent)

        # Widget configurations
        self.initialize_dateedit_value()

    def initialize_dateedit_value(self):
        # Initialize the values of the date-edit
        if self.parent:
            qdate = QtCore.QDate(self.parent.dt.year, self.parent.dt.month, self.parent.dt.day)
            qtime = QtCore.QTime(self.parent.dt.hour, self.parent.dt.minute, self.parent.dt.second)
            qdatetime = QtCore.QDateTime(qdate, qtime)
            self.dateTimeEdit.setDateTime(qdatetime)

    def return_values_to_parent(self):
        # Return the datetime to the parent (MainWindow)
        if self.parent:
            # Get the datetime object from the datetime edit widget
            dt = self.dateTimeEdit.dateTime().toPyDateTime()

            # Pass the datetime to the parent (MainWindow)
            self.parent.dt = dt

            # Close the dialog
            self.accept()
```

Figure 5.19: “Scan Age” button function source code

The above diagram showcase the “FilterAgeDialog” class which is the function class for Age filter dialog box. In this source code, the dialog menu was first initialized then set up with their window title “Scan Age”. Then initialize the parent variable. Once that has been completed, Confirm and cancel buttons were assignment with their respective functions. We then move onto defining a function which is “initializing_dateedit_value”. In order to setup the value for QDateTime function, we start by setting up the qdate object and qtime object, where we pass the year, month and day into qdate, the hour, minute and second onto qtime. Once these are completed and qdate and qtime is passed into QDateTime, then and only then will the system be able to setup the value for setDate. The following function “return_values_to_parent” which are the values returned to parent variable in “MainWindow” class will be used for age filtering. The system convert the datetimeEdit object value into python date time format and pass back to parent in “MainWindow” class

5.3 Result

5.3.1 Main Window

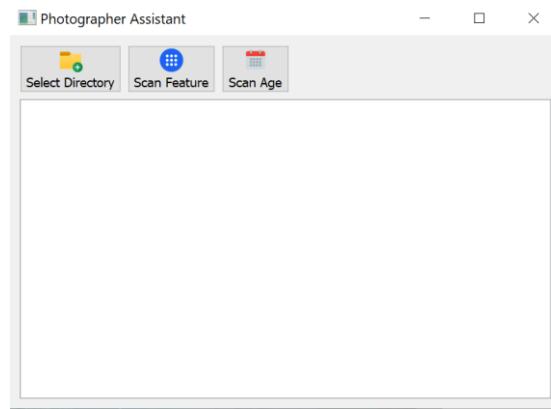


Figure 5.20: Main Window User Interface

The above diagram showcase the user interface design for Main Window. The main title of the Main Window is the name of the software itself which is “Main Window”, In the window there are 3 buttons on the top of the window namely: “Select Directory”, “Scan Feature” and “Scan Age”. Below the set of buttons is a empty treeview field, which will be showcase the directory the user have chosen from using the “Select Directory” button.

5.3.2 Scan Feature Dialog box

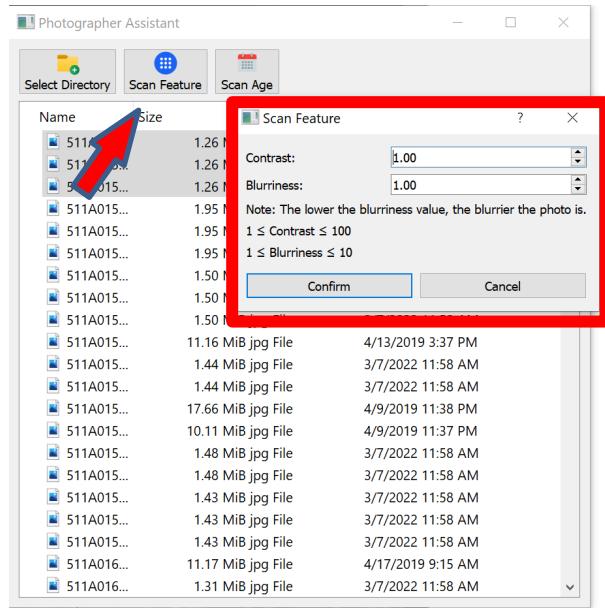


Figure 5.21: “Scan Feature” dialog box User interface

The above diagram showcase the user interface design for “Scan Feature” dialog prompt box. This dialog box prompts out after the user have clicked on the “Scan Feature” button. In this dialog box there are 2 spinboxes accompanied by a footnote beneath it. The user will be able to input their desire value for Contrast and Blurriness with a range of value indicated in the footnote in doubles. Do not that Blurriness is measured using sharpness algorithm thus the lower the value of the Blurriness, the more less sharp the image is.

5.3.3 Feature Filter output

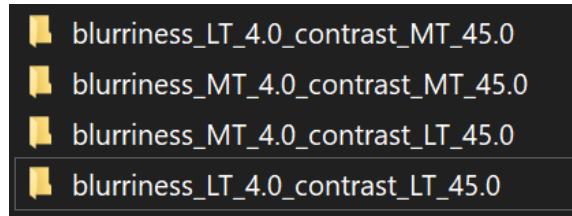


Figure 5.22: Output of “Scan Feature”

The above diagram showcase the output of the Scan Feature button function. Once the system has run the algorithm and sort the image, It will create 4 folders with the criteria being:

1. blurriness less than X , contrast more than Y
2. blurriness more than X , contrast more than Y
3. blurriness more than X , contrast less than Y
4. blurriness less than X , contrast less than Y

Once these files has been created, the system will sort and copy the scanned image into the each folder fitting of the feature of the image themselves.

5.3.4 Age Filter Dialog menu

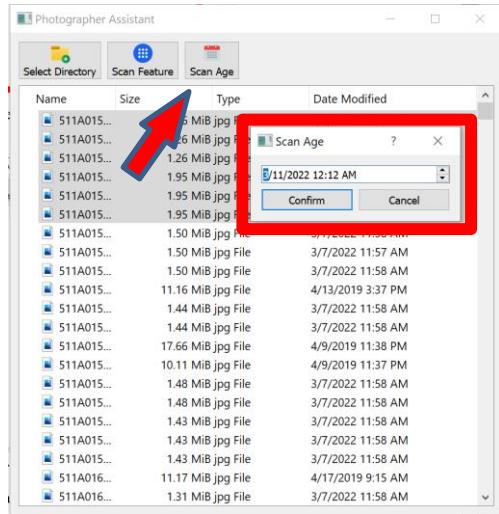


Figure 5.23: “Scan Age” dialog box user interface

The above diagram showcase the user interface design for “Scan Age” dialog prompt box. This dialog box is prompted after the user has clicked on the “Scan Age” button on the top of main window. In this dialog box, there is a spinbox for date time selection for user to input their desired date for filtering of the image’

5.3.5 Age Filter output

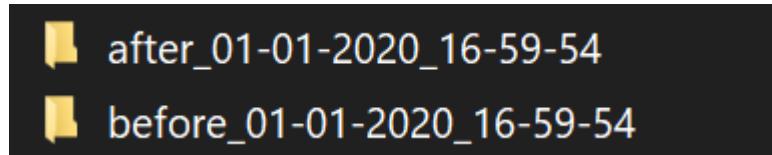


Figure 5.24: “Scan Age” function output

The above diagram showcase the output for “scan Age” button function. The system will create 2 new folders with their criteria being before and after the input date. The system will then go through each image’s last modified date and sort and copy them into the folders.

5.3.6 Progress Bar

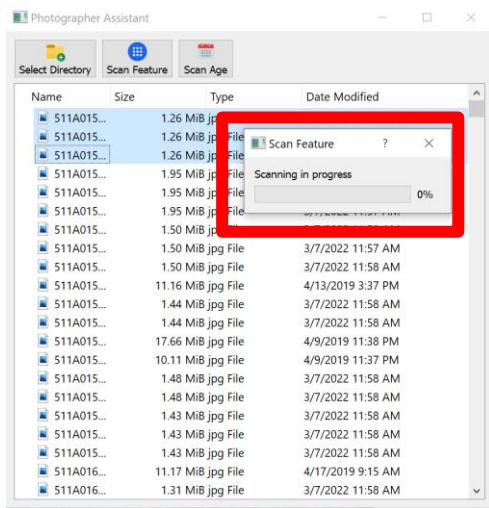


Figure 5.25: Progress bar dialog box user interface

The above diagram showcase the user interface design for Progress bar dialog prompt box. This window are hidden initially when the software is first launched, once the values has been confirmed in the “Scan Feature” dialog box or the “Scan Age” dialog box. Once it has been confirmed and the system is going through each image the percentage will be calculated by the index of current image divided by the total image and multiplied by 100.

5.4 Instruction Manual

5.4.1 System Requirement Installation

In order to start getting your system in order to run the system you must first setup your system.

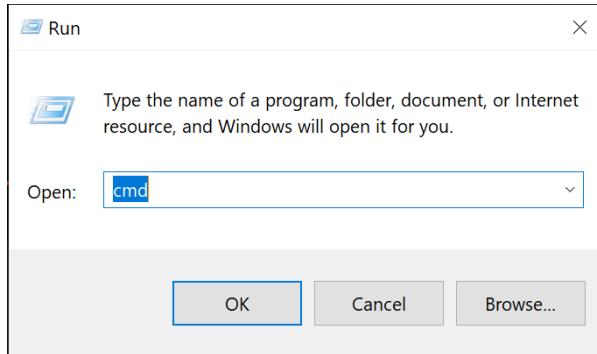


Figure 5.26: Run Menu

1. You may right click on your windows icon on the bottom left screen or press Win + X keys to initiate “run” menu as you can see in Figure 5.27.



Figure 5.27: Command Prompt

2. Once this is complete, enter “cmd” which and press ok, this will launch your command prompt
3. Once Command Prompt is launched, change your directory to the folder where the program is located using “cd” command
4. Once you have changed your directory use “pip install -r requirements.txt” and let it install the necessary modules.

5.4.2 Steps of operation

5.4.2.1 Launching Program

1. Launch your command prompt, and change your directory using “cd” command followed by your directory URL.
2. Once you have changed your directory, Use “Python Main.py” to launch the program

5.4.2.2 Sorting images based on blurriness and contrast

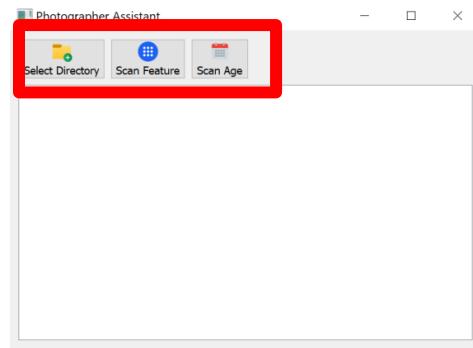


Figure 5.28: main window

1. When you first launch the program you will encounter a screen with 3 buttons on the top left corner and an empty menu below as shown in figure 5.28.

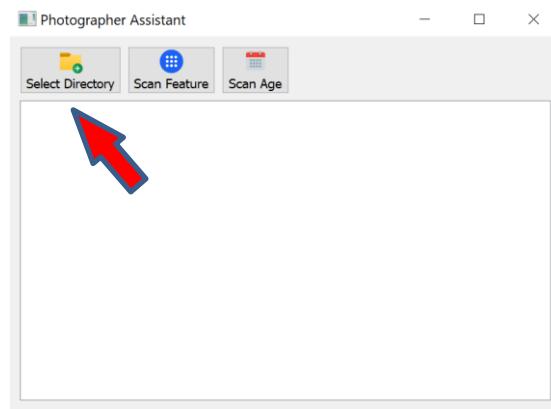


Figure 5.29: “Select directory” button location in Main Window

2. Press “select directory” button to choose the specific folders containing your Images as shown in figure 5.29.

Name	Size	Type	Date Modified
511A015...	1.26 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.26 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.26 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.95 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.95 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.95 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.50 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.50 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.50 MiB	jpg File	3/7/2022 11:58 AM
511A015...	11.16 MiB	jpg File	4/13/2019 3:37 PM
511A015...	1.44 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.44 MiB	jpg File	3/7/2022 11:58 AM
511A015...	17.66 MiB	jpg File	4/9/2019 11:38 PM
511A015...	10.11 MiB	jpg File	4/9/2019 11:37 PM
511A015...	1.48 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.48 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.43 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.43 MiB	jpg File	3/7/2022 11:58 AM
511A016...	11.17 MiB	jpg File	4/17/2019 9:15 AM
511A016...	1.31 MiB	jpg File	3/7/2022 11:58 AM

Figure 5.30: Output of “Select Directory” button

- Once the folders have been selected a treeview of your folder will show in to empty box below the sets of buttons as shown in figure 5.30.

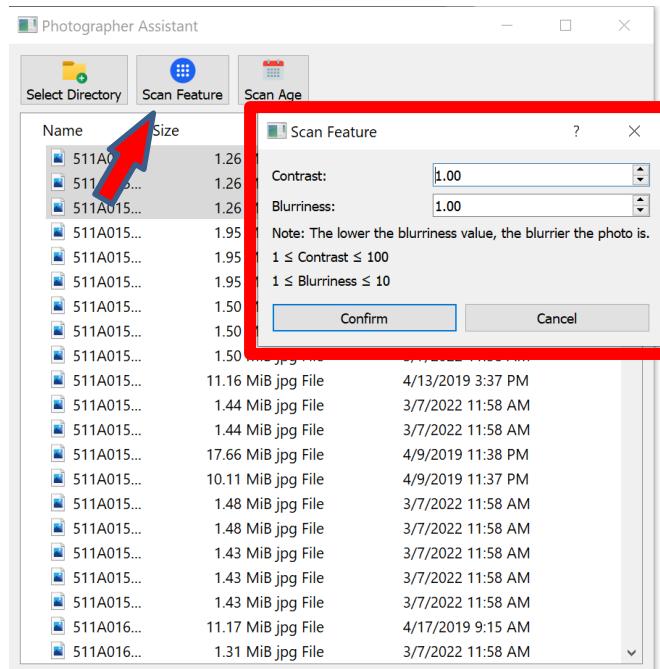


Figure 5.31: “Scan Feature” button

- You may select the selection of images you wish to scan in the treeview, once the images has been selected, press “Scan Feature” button and a dialog will prompt up as shown in figure 5.31.

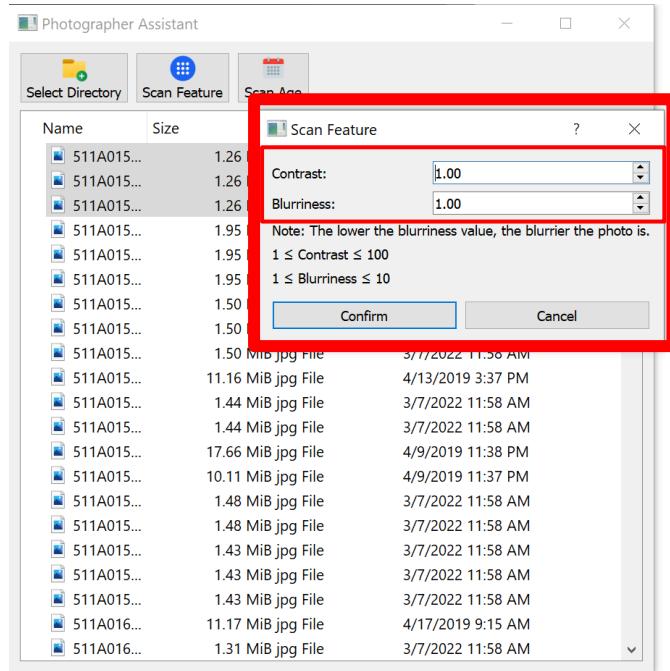


Figure 5.32: “Scan feature” dialog box

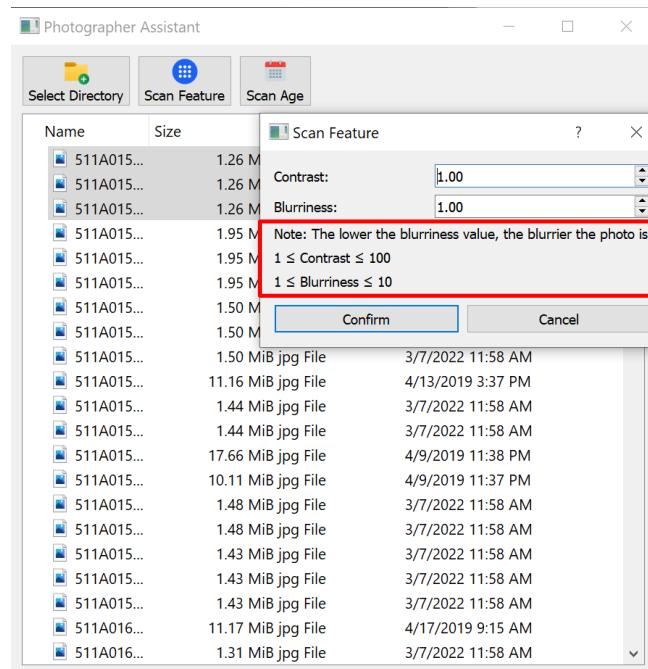


Figure 5.33: Notes for “Scan Feature”

5. In the dialog, you will be required to input 2 different values for contrast and blurriness as shown in figure 5.32. The value should set according to instructions displayed in the dialog menu following the restrictions as shown in figure 5.33.

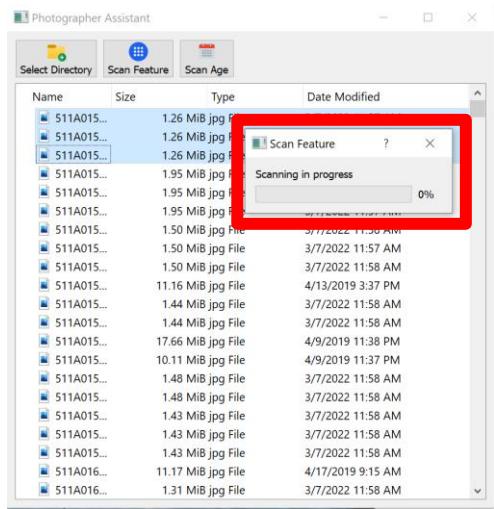


Figure 5.34: Progress bar dialog box

6. Once you have input your desired value, press “confirm” and a progress bar dialog menu will prompt showing the progress of your program as shown in figure 5.34.

sharpness_LT_4.0_contrast_LT_45.0	3/7/2022 12:45 PM	File folder
sharpness_LT_4.0_contrast_MT_45.0	3/7/2022 12:46 PM	File folder
sharpness_MT_4.0_contrast_LT_45.0	3/7/2022 12:46 PM	File folder
sharpness_MT_4.0_contrast_MT_45.0	3/7/2022 12:46 PM	File folder

Figure 5.35: output of “Scan Feature” function

7. Once the program has finished scanning your images, it will divide the files into 4 different files as shown in Figure 5.35.

NOTE: the files are namely:

1. Contrast more than X, Blurriness more than X
2. Contrast more than X, Blurriness less than X
3. Contrast less than X, Blurriness more than X
4. Contrast less than X, Blurriness less than X

5.4.3 Sorting images based on last modified date

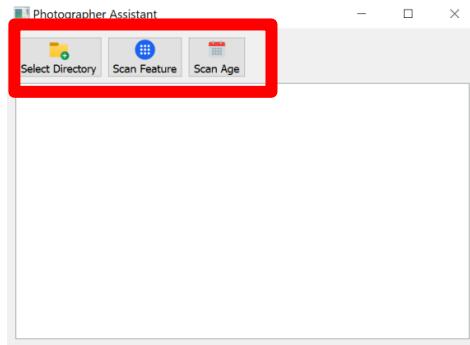


Figure 5.36: Main Window

1. When you first launch the program you will encounter a screen with 3 buttons on the top left corner and an empty menu below as shown in **Figure 5.36**.

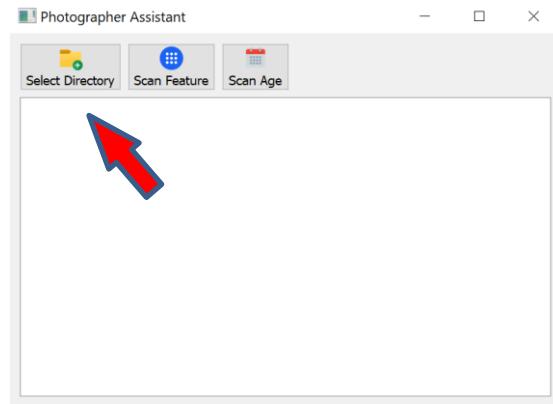
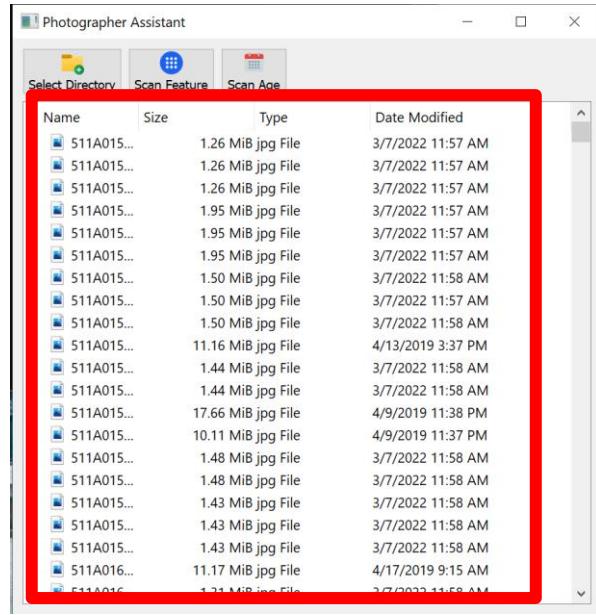


Figure 5.37: “Select Directory” button location in the Main Window

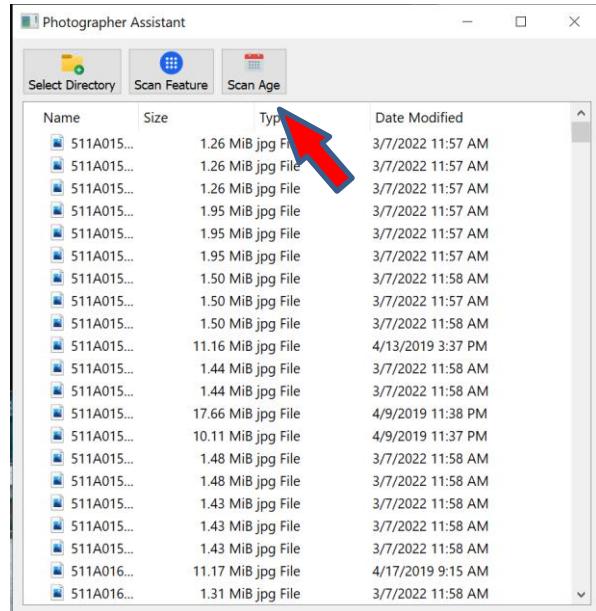
2. Press “select directory” button to choose the specific folders containing your Images.



Name	Size	Type	Date Modified
511A015...	1.26 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.26 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.26 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.95 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.95 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.95 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.50 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.50 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.50 MiB	jpg File	3/7/2022 11:58 AM
511A015...	11.16 MiB	jpg File	4/13/2019 3:37 PM
511A015...	1.44 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.44 MiB	jpg File	3/7/2022 11:58 AM
511A015...	17.66 MiB	jpg File	4/9/2019 11:38 PM
511A015...	10.11 MiB	jpg File	4/9/2019 11:37 PM
511A015...	1.48 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.48 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.43 MiB	jpg File	3/7/2022 11:58 AM
511A016...	11.17 MiB	jpg File	4/17/2019 9:15 AM
511A016...	1.31 MiB	jpg File	3/7/2022 11:58 AM

Figure 5.38: Output of “Select Directory” function

- Once the folders have been selected a treeview of your folder will show in to empty box below as shown in Figure 5.38.



Name	Size	Type	Date Modified
511A015...	1.26 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.26 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.26 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.95 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.95 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.95 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.50 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.50 MiB	jpg File	3/7/2022 11:57 AM
511A015...	1.50 MiB	jpg File	3/7/2022 11:58 AM
511A015...	11.16 MiB	jpg File	4/13/2019 3:37 PM
511A015...	1.44 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.44 MiB	jpg File	3/7/2022 11:58 AM
511A015...	17.66 MiB	jpg File	4/9/2019 11:38 PM
511A015...	10.11 MiB	jpg File	4/9/2019 11:37 PM
511A015...	1.48 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.48 MiB	jpg File	3/7/2022 11:58 AM
511A015...	1.43 MiB	jpg File	3/7/2022 11:58 AM
511A016...	11.17 MiB	jpg File	4/17/2019 9:15 AM
511A016...	1.31 MiB	jpg File	3/7/2022 11:58 AM

Figure 5.39: “Scan Age” button location

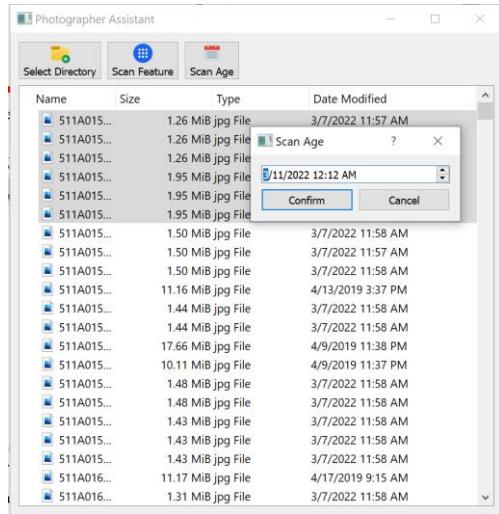


Figure 5.40: “Scan Age” dialog prompt

4. You may select the selection of images you wish to scan in the treeview, once the images has been selected, press “Scan age” button as shown in figure 5.39 and a dialog will prompt up as showed in figure 5.40

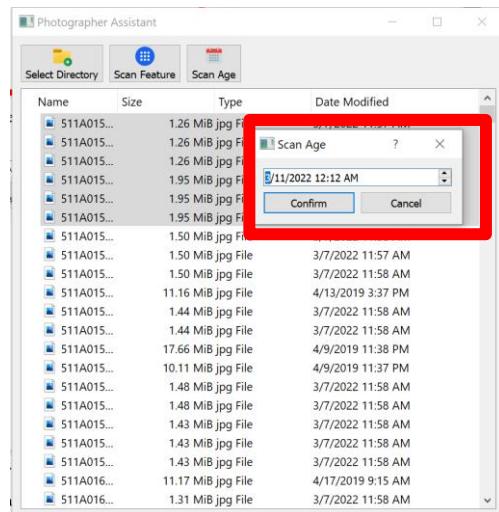


Figure 5.41: “Scan Age” dialog box

5. In this file dialog, you may input the date you wish for the program to sort the images according to last modified date

📁 after_01-01-2020_16-59-54	3/7/2022 5:06 PM	File folder
📁 before_01-01-2020_16-59-54	3/7/2022 5:05 PM	File folder

Figure 5.42: Output of “scan age” button function

6. Once the program has finished scanning, it will divide it into 2 files: before the input date and another after the input date

5.4.3.1 Notes

1. Note all the images in the output folders are copied and the originals are uninterrupted for safety and backup purposes
2. There are limit to the range value you may input in the Contrast and Blurriness scanning function, the limits are 1.00 – 100.00 for contrast and 1.00 – 10.00 for Blurriness
3. For Blurriness function, the higher the amount of blur in the image the lower the value will be.

CHAPTER 6

TESTING AND EVALUATION OF FINDINGS

6.1 Overview

This chapter focuses on the testing of the developed software application. The testing includes in terms of verification of quality, reliability and performance. This is to ensure the chances of an error and minimizes furthermore it also allows for checking whether the application meets the requirements and expected outcomes. A table of tests outline was carried out to test the functions of the application

6.2 Testing Outline

Table 6-1: Testing outline table

#	Features	Action	Expected Outcome	Result
1.	Main Menu	Press “select directory button”	Prompt windows selection menu	✓
2.	Main Menu	Press “Scan Feature” button	Prompt Scan Feature dialog box	✓
3.	Main Menu	Press “Scan Age” button	Prompt Scan Age dialog box	✓
4.	Main Menu	Selecting folder for treeview	Display folder content in treeview	✓
5.	Select Directory Prompt	Empty field on folder selection	Show default directory in treeview	✓
6.	Select Directory Prompt	Select a folder	Show folder contents in treeview	✓
7.	Select Directory prompt	Select multiple folder	Deselect first folder and select	✓

			newest selected folder	
8.	Main menu	Selecting images for scanning	Output to folders depending on function	✓
9.	Main menu	Selecting 0 images for scanning	Prompt error message	✓
10.	Scan Feature Dialog box	Select multiple images for feature scanning	Create 4 folders and copy selected images into suitable folder	✓
11.	Scan Feature Dialog box	Select single image for feature scanning	Create 4 folders and copy selected image into suitable folder	✓
12.	Scan Feature Dialog box	Input value over the specified range	Unable to do so due to locked range	✓
13.	Scan Age Dialog box	Input date for scanning	Create 2 folders and copy scanned images into the folders	✓

CHAPTER 7

CONCLUSION

7.1 Overview

This chapter summarizes the work that has been completed in the second phase of the final year project and well as a conclusion.

7.2 Summary of work done

In the beginning of this project, the chapters of phase 1 was rechecked and reconfirmed. Once that has been done, the author proceeded to implementing the system as well as the algorithms. The author started by setting up the user interfaces followed by setting up classes and method functions for buttons and fields within the user interface. This is followed by implementation of the algorithms within the buttons. Once this has been done it is followed by testing the software and debugging and changing functions output for practical reason. Once testing is complete and debugged, the author proceeded with completion of the report by altering parts which required alteration and finalization as well as final checking.

REFERENCES

- DeMuro, J. P., & James, S. (2021, July 12). *Amazon Photos Cloud Storage Review*. TechRadar. Retrieved October 27, 2021, from <https://www.techradar.com/sg/reviews/amazon-photos>.
- Ben-Yair, S. (2020, November 11). *Updating google photos' storage policy to build for the future*. Google. Retrieved October 27, 2021, from <https://blog.google/products/photos/storage-changes/>.
- Muchmore, M. (2021, June 1). *Google photos*. PCMag Asia. Retrieved October 27, 2021, from <https://sea.pcmag.com/photo-management-sharing-products/3795/google-photos>.
- Mitroff, S., & La, L. (2015, May 28). *Google Photos Review: Hands-off photo organization meets eye-catching editing tools*. CNET. Retrieved October 27, 2021, from <https://www.cnet.com/reviews/google-photos-review/>.
- Muchmore, M. (2021, February 22). *Microsoft Photos Review*. PCMag. Retrieved October 27, 2021, from <https://www.pcmag.com/reviews/microsoft-photos-for-windows-10>.
- Muchmore, M. (2021, June 14). *Adobe Lightroom Classic*. PCMag Asia. Retrieved October 27, 2021, from <https://sea.pcmag.com/photo-editing/3037/adobe-lightroom-classic>.
- Faststone Image Viewer Review*. Clifton Web Design. (2020, January 12). Retrieved October 28, 2021, from <https://www.cliftonwebdesign.co.uk/archive/faststone-image-viewer-review-free-image-browser-editor/>.
- Faststone Image Viewer Review - faststone image viewer is great when photoshop is too much*. Bright Hub. (2008, July 26). Retrieved October 28, 2021, from <https://www.bighthub.com/multimedia/photography/articles/3828/>.
- Idesis, S. (2020, October 9). *Rapid application development: Why Rad and why now*. Build Applications Fast, Right and For the Future. Retrieved October 27, 2021, from <https://www.outsystems.com/blog/posts/rapid-application-development/>.
- Thompson, P. (2021, October 29). *PYQT5*. PyPI. Retrieved March 24, 2022, from <https://pypi.org/project/PyQt5/>
- What is PyQt?* Riverbank Computing | Introduction. (n.d.). Retrieved March 24, 2022, from <https://riverbankcomputing.com/software/pyqt/intro>
- OS - miscellaneous operating system interfaces¶*. os - Miscellaneous operating system interfaces - Python 3.10.3 documentation. (2022, March 24). Retrieved March 24, 2022, from <https://docs.python.org/3/library/os.html>
- Python datetime*. Python Dates. (n.d.). Retrieved March 24, 2022, from https://www.w3schools.com/python/python_datetime.asp

Sys - system-specific parameters and functions¶. sys - System-specific parameters and functions - Python 3.10.3 documentation. (2022, March 24). Retrieved March 24, 2022, from <https://docs.python.org/3/library/sys.html>

Dizon, K. K. (2018, October 10). *KDIZON/image-contrast: Using standard deviation to quantify image contrast*. GitHub. Retrieved March 24, 2022, from <https://github.com/kdizon/image-contrast>

Tolga Birdal (2022). Sharpness Estimation From Image Gradients (<https://www.mathworks.com/matlabcentral/fileexchange/32397-sharpness-estimation-from-image-gradients>), MATLAB Central File Exchange. Retrieved March 24, 2022.

APPENDICES

Appendix A: Meeting Logs



Faculty of Information Science and Technology (FIST) Final Year Project Meeting Log

MEETING DATE: 3-1-2022	MEETING NO.: 1
PROJECT ID: T771871	
PROJECT TITLE : PHOTOGRAPHY ASSISTANT	
SESSION : Phase 2: Trimester 2 2021/2022	SUPERVISOR : Ts. Liew Tze Hui
STUDENT ID & Name: 1171103901 Chai Jay Jay	CO- SUPERVISOR :

All to be filled in by student

1. WORK DONE [Please write the details of the work done after the last meeting.]
Discussion in regards of overall formatting of the reports and system development

2. WORK TO BE DONE

- Chapter 1 : introductions
- Chapter 2 : literature review
- Chapter 5 : Implementation Process and Result

3. PROBLEMS ENCOUNTERED

Rechecking and coding for Chapter 5

4. COMMENTS

For FYP Meeting Log Tri 2 -2022
LIEW TZE HUI
LECTURER
Faculty of Information Science & Technology
Multimedia University
Jalan Ayer Keroh Lama
71450 Melaka, Malaysia.

Supervisor's Signature &
Stamp

Co-Supervisor's Signature
& Stamp (if any)

Student's Signature

NOTES:

1. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
2. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
3. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
4. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 &

14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission,

weeks 13 & 14: presentation)



**Faculty of Information Science and Technology (FIST)
Final Year Project Meeting Log**

MEETING DATE: 17-1-2022	MEETING NO.:2
PROJECT ID: T771871	
PROJECT TITLE : PHOTOGRAPHY ASSISTANT	
SESSION : Phase 2: Trimester 2 2021/2022	SUPERVISOR: TS. Liew Tze Hui
STUDENT ID & Name: 1171103901 Chai Jay Jay	CO- SUPERVISOR :

All to be filled in by student

1. WORK DONE [Please write the details of the work done after the last meeting.]
Chapter 1: introduction & Chapter 2 : literature review

2. WORK TO BE DONE

Chapter 3 : Methodology

Chapter 5 : Implementation Process and Result

3. PROBLEMS ENCOUNTERED

Rechecking and coding

4. COMMENTS

For FYP Meeting Log Tri 2 -2022
LIEW TZE HUI
LECTURER
Faculty of Information Science & Technology
Multimedia University
Jalan Ayer Keroh Lama
71400 Melaka, Malaysia.

Supervisor's Signature &
Stamp

Co-Supervisor's Signature
& Stamp (if any)

Student's Signature

NOTES:

5. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
6. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
7. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
8. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



**Faculty of Information Science and Technology (FIST)
Final Year Project Meeting Log**

MEETING DATE: 31-1-2022	MEETING NO.:3
PROJECT ID: T771871	
PROJECT TITLE : PHOTOGRAPHY ASSISTANT	
SESSION : Phase 2: Trimester 2 2021/2022	SUPERVISOR : TS. Liew Tze Hui
STUDENT ID & Name: 1171103901 Chai Jay Jay	CO- SUPERVISOR :
All to be filled in by student	
1. WORK DONE [Please write the details of the work done after the last meeting.] Chapter 3 : methodology Chapter 5 : Implementation Process and Result	
2. WORK TO BE DONE Chapter 4 : System designs Chapter 5 : Implementation Process and Result	
3. PROBLEMS ENCOUNTERED Rechecking and amendment	
4. COMMENTS	

For FYP Meeting Log Tri 2 -2022
 LIEW TZE HUI
 LECTURER
 Faculty of Information Science & Technology
 Multimedia University
 Jalan Ayer Keroh Lama
 71450 Melaka, Malaysia.

Supervisor's Signature &
Stamp

Co-Supervisor's Signature
& Stamp (if any)

Student's Signature

NOTES:

9. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
10. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
11. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
12. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



**Faculty of Information Science and Technology (FIST)
Final Year Project Meeting Log**

MEETING DATE: 14-2-2022	MEETING NO.:4
PROJECT ID: T771871	
PROJECT TITLE : PHOTOGRAPHY ASSISTANT	
SESSION : Phase 2: Trimester 2 2021/2022	SUPERVISOR : TS. Liew Tze Hui
STUDENT ID & Name: 1171103901 Chai Jay Jay	CO- SUPERVISOR :
All to be filled in by student	
1. WORK DONE [Please write the details of the work done after the last meeting.] Chapter 4: System Design	
2. WORK TO BE DONE Chapter 4 : User interface design Chapter 5 : Implementation Process and Result	
3. PROBLEMS ENCOUNTERED Rechecking and amendment and coding	
4. COMMENTS	

For FYP Meeting Log Tri 2 -2022
 LIEW TZE HUI
 LECTURER
 Faculty of Information Science & Technology
 Multimedia University
 Jalan Ayer Keroh Lama
 71400 Melaka, Malaysia.

Supervisor's Signature &
Stamp

Co-Supervisor's Signature
& Stamp (if any)

Student's Signature

NOTES:

- 13. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
- 14. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
- 15. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
- 16. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



**Faculty of Information Science and Technology (FIST)
Final Year Project Meeting Log**

MEETING DATE: 14-3-2022	MEETING NO.: 5
PROJECT ID: T771871	
PROJECT TITLE : PHOTOGRAPHY ASSISTANT	
SESSION : Phase 2: Trimester 2 2021/2022	SUPERVISOR : TS. Liew Tze Hui
STUDENT ID & Name: 1171103901 Chai Jay Jay	CO- SUPERVISOR :

All to be filled in by student

1. WORK DONE [Please write the details of the work done after the last meeting.] User interface design & Conclusion
2. WORK TO BE DONE Final checking and report writing
3. PROBLEMS ENCOUNTERED Changes to several parts of the project
4. COMMENTS

For FYP Meeting Log Tri 2 -2022
 LIEW TZE HUI
 LECTURER
 Faculty of Information Science & Technology
 Multimedia University
 Jalan Ayer Keroh Lama
 71400 Melaka, Malaysia.

Supervisor's Signature &
Stamp

Co-Supervisor's Signature
& Stamp (if any)

Student's Signature

NOTES:

17. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
18. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
19. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
20. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)



**Faculty of Information Science and Technology (FIST)
Final Year Project Meeting Log**

MEETING DATE: 21-3-2022	MEETING NO.: 6
PROJECT ID: T771871	
PROJECT TITLE : PHOTOGRAPHY ASSISTANT	
SESSION : Phase 2: Trimester 2 2021/2022	SUPERVISOR : TS. Liew Tze Hui
STUDENT ID & Name: 1171103901 Chai Jay Jay	CO- SUPERVISOR :

All to be filled in by student

1. WORK DONE [Please write the details of the work done after the last meeting.] User interface design & Conclusion
2. WORK TO BE DONE Final checking and amendment
3. PROBLEMS ENCOUNTERED Changes to several parts of the project
4. COMMENTS

For FYP Meeting Log Tri 2 -2022
 LIEW TZE HUI
 Faculty of Information Science & Technology
 Multimedia University
 Jalan Ayer Keroh Lama
 71400 Melaka, Malaysia.

Supervisor's Signature &
Stamp

Co-Supervisor's Signature
& Stamp (if any)

Student's Signature

NOTES:

21. Items 1 – 3 are to be completed by the students before coming for the meeting. Item 4 is to be completed by the supervisor.
22. For FYP Phase 1, total six log sheets are to be submitted (every other week*).
23. For FYP Phase 2, total six log sheets are to be submitted (every other week**).
24. Log sheets are compulsory assessment criteria for FYP. Student who fails to meet the requirements of log sheets will not be allowed to submit FYP report.

*: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the first trimester (week 11: report submission, weeks 13 & 14: presentation)

**: week 1, 3, 5, 7, 9, 11 or 2, 4, 6, 8, 10 of the second trimester (week 11: report submission, weeks 13 & 14: presentation)

I hereby confirm the following students has attended the FYP for FIST session.

Trainer:
Nurul Irtika Mohamad Nori
Librarian, Siti Hasmah Digital library

No	Full Name	MMU ID No.	FYP title
1	Soh Chun Wei	1181100955	Blockchain On Tam
2	Nurul Sa'adah Binti Zabidi	1151205539	Basic Subtraction Application
3	Liuwe Yee Jia	1181101480	Human Activity Recognition
4	Hew Zhong Ken	1181101119	Covid-19 Recognition Based On Chest X-Ray
5	Suvithra Vani A/P Chandrasekaran	1161204216	Online Food Ordering System
6	Nurul Radhiatal Aleen Binti Mohd	1161200319	Enterprise Resource Planning For Online Business
7	Nur Ayuni Binti Zukarnain	1181120457	iShare Student Networking Apps
8	Ong Han Bin	1181100723	Health And Diet Guide System
9	Nurul Nafiah Binti Nor	1181101455	Smart Home System
10	Nurul Azura Binti Mohamad Rizal	1191300699	Digitalisation Of Individualised Education Programme
11	Chong Jun Xiong	1171100229	Deepfakes Detection Using Computer Vision And Deep Learning Approaches
12	Lim Yu Jie	1181101201	Medical Image Analysis With Deep Learning
13	Kannan Suppiah	1181302865	Surveillance Camera Using IoT
14	Alyssa Mary Carvalho	1171201821	E-Voting System For Student Council Elections
15	Chai Jay Jay	1171103901	T771871: Photography Assistant
16	Wong Cong Zhi	1181100850	Maize Leaf Disease Identification By Image-Based Deep Learning
17	Victor Tan Peng Sheen	1171202971	Image Noise Variance Classification With Deep Learning
18	Chu Lei Wern	1181101031	Inventory Management System For E-Commerce
19	Muhammad Amirul Zafwan Bin Ramli	1171103364	Smart Sorting/Defect Detection On Conveyor Belt
20	Woon Hual Chang	1181100735	Library Bot
21	Bin Nouh Abdulaziz Fahad A	1171302967	Doctor Hub
22	Adham Haziq Bin Adam	1181302917	Creative Marketweb
23	Tong Li Fong	1191301862	Cosmetics Consultant
24	Yew Ley Theng	1191301929	Qr-Enabled Smart Receipt
25	Lee Wei Jye	1181100428	Leave Application System
26	Zhou Yin Yen	1191300910	AI-Enabled Movie Assistant
27	Eugene Tan Kok Yew	1181101030	Application For Tracking The Status Of Covid-19 In Malaysia
28	Bok Hui Ling	1181100160	Occupational Safety And Health Management System
29	Tan Zhan Jun	1181100805	Residential Visitor Management Mobile Application
30	Jeffry Joaquin Iman Bin Alamsyah	1181101168	T771003: Basic Addition Application
31	Muhammad Hakimi Bin Hamidzul	1181101188	E-Pharmacy System
32	Ge Yong Chang	1181101641	Interactive Teaching For Chinese Characters (Primary School)
33	Aidil Mohamed Afif Bin Ghazali	1181303011	Pc Part Picker Builder Website
34	Anas Bin Abdul Mutaleb	1161204517	Finance Tracking System
35	Harith Danial Poh Bin Jeffry Poh	1142700661	Pc Builder: Picks Parts, Build Your Pc
36	Mok Li Hao	1191301860	Colloquial Dictionary Platform : Categories Bridge
37	Mohamed Hizzudin Mohamad	1181101153	Fatigue Detection Using Embedded Devices
38	Tay Kim Soon	1171202921	Homeworkout Application
39	Rivetha Raman	1171200830	Gamification Of Online Assessment Platform
40	Ng William	1171201121	Gamification In Online Learning
41	Ng En X	1181100811	Student Registration With Voice Recognition Chatbot
42	Tan Soo Chin	1171103570	Secure Cryptographic E-Auction System
43	Ng Wei Khang	1171202181	Online Street Jeker
44	Mohd Norman Bin Bakari@Bakri	1181101175	Blockchain And Tourism
45	Loura Christie Lawrence	1181302964	Sentiment Analysis Using Machine Learning
46	Zachary Isaac Lazaroo	1171200530	Augmented Reality Home Interior Designer Application
47	Lau Qi Wei	1181101468	Security Guard Patrol Application
48	Soo Hui Zheng	1161203330	Employee Attendance And Payroll Management System
49	Chow Tang Soon	1171200142	Interactive Augmented Reality Book Application
50	Yeow Jian Cheng	1171202978	Food Recipes (Fr)
51	Fong Li Tat	1181102012	IoT Based Indoor Appliance Management
52	Nurmiza Aqilah Binti Rosli	1191301924	Call Admission Control Framework For 4g/5g Wireless Broadband Network
53	Ahmad Syami Bin Che Roslan	1171302695	Personal Budget And Expenses Tracking Cloud Based Chatbot With Otp Authentication
54	Tan Siao Wah	1181100430	Food Detection And Recognition Using Machine Learning
55	Koh Jun Hong	1171203668	An Effective Transmission Of Environmental Sensor Data In Lora Network
56	Ahmad Hasree Hakimie Bin Abang	1191301848	Agromobileapps : A Cloud-Based Framework For Agriculturists On Mobile Platform
57	Muhammad Irfan Syahmi Bin Mohd	1181101191	IoT Based Smart Sensors
58	John Nathan	1171201833	Traffic Sign Recognition With Machine Learning
59	Sei Liang Hock	1171202904	My Healthcare – A Remote Telemedicine Application
60	Chedi Jia Hao	1171201061	Application Using Blockchain In E-Commerce
61	Ahmad Danish Mikhai Bin A. Wirah	1181101177	Smart Speech-To-Text Meeting Minutes Recorder
62	Mohammad Hadiff Bin Hamidey	1181101152	Secure Cryptographic E-Voting System
63	Khaled Androu Abdelsaheer M K	1181302338	Visitor Identification system
64	Muhammad Haigal Bin Bahrudin	1181100453	IoT Based Smart Room Lighting System
65	Lee Zhi Yang	1181100967	Acoustic Event Classification Using Deep Learning
66	Tan Kah Feng	1171203759	Patient Health Monitoring System
67	Chang Kang Hong	1171203135	Neurodegenerative Disorders Detection Using Gait Analysis - Pose Estimation
68	Tan Jie Hang	1191301635	Real Time Fast Ensemble Learning Movie Auto Subtitle Synchronization
69	Muhammad Afif Iskandar Bin Eddie	1171103647	AIot : Smart Room System
70	Kang Chuen Yik	1181101967	T771577 : Cryptocurrencies Price Prediction Using Deep Learning
71	Kendy Wong Jia Kang	1171103369	Implementation Of Deep Learning For Text Classification Based On Customer Review
72	Ong Kai Jaz	1171103511	Juggling Pattern Recognition
73	Law Kin Aik	1171103920	Ar-Based Education For Special Needs

Appendix B: Checklist for FYP Interim Report Submission



Faculty of Information Science and Technology(FIST)
Checklist for Interim Report Submission
(To be filled in by Student)

STUDENT'S DETAILS

Project Code	T771871
Name	CHAI JAY JAY
ID No	1171103901
Title of Thesis	PHOTOGRAPHY ASSISTANT
Supervisor Name	Ts. Liew Tze Hui

REPORT ARRANGEMENT	✓	Comments by Supervisor (if any differences)
1. Cover of The Final Report	✓	
2. Title Page of the Final Report	✓	
3. Copyright page of I Final Report	✓	
4. Declaration Page of Final Report	✓	
5. Acknowledgement	✓	
6. Table of Contents	✓	
7. Abstract	✓	
8. List of Tables	✓	
9. List of Figures	✓	
10. List of Symbols	✓	
11. List of Appendices	✓	

12. Chapter 1: Introduction – objectives, scope	✓	
13. Chapter 2: Literature Review	✓	
14. Chapter 3: Title: Project Methodology	✓	
15. Chapter 4: Title: System Design	✓	
16. Chapter 5: Title: System Implementation	✓	
17. Chapter 6: System Testing and Results	✓	
18. Chapter 7: Conclusion	✓	
19. References - APA style	✓	
20. Appendices	✓	
21. CD/DVD and envelope as shown in Appendix K.		
22. Attachment: FYP Meeting Logs(all) - 1 set	✓	

FORMAT OF REPORT	✓	Comments by Supervisor
1. Page Numbering	✓	
2. Font and Type Face	✓	
3. Font Cover	✓	
4. Tables and Figures	✓	
5. Comb Bind	✓	
6. Colour of the Front Cover		
7. Number of words > 10000 (Main content only)	✓	

Checked by,



Chai Jay Jay

24/3/2022

