



# Bilkent University

DEPARTMENT OF COMPUTER ENGINEERING

## SENIOR DESIGN PROJECT

### PHOTONOM

---

## Final Report

---

Group Members	Omer Faruk BABADEMEZ Idil HANHAN Beyza KALKANLI Berfin KUCUK Kerem YILMAZ	21601216 21601289 21600944 21502396 21601223
Supervisor	Asst. Prof. Hamdi DIBEKLIOĞLU	
Jury Members	Prof. Ozcan OZTÜRK Assoc. Prof. Selim AKSOY	
Innovation Expert	Mehmet Surav	

Final Report  
May 27, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Senior Design Project course CS491/2.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Requirement Details</b>	<b>4</b>
2.1	Functional Requirements . . . . .	4
2.1.1	Uploading & Taking Photos . . . . .	4
2.1.2	Remove Objects . . . . .	4
2.1.3	Rain Removal . . . . .	4
2.1.4	Haze Removal . . . . .	5
2.1.5	Theme Transfer . . . . .	5
2.1.6	Image Stitching . . . . .	5
2.1.7	Image Quality Evaluation . . . . .	5
2.1.8	Panorama . . . . .	5
2.1.9	Facial Expression Modifier . . . . .	6
2.1.10	Saving the Edited Photo . . . . .	6
2.2	Nonfunctional Requirements . . . . .	6
2.2.1	Accessibility . . . . .	6
2.2.2	Availability . . . . .	6
2.2.3	Sufficient Network Bandwidth . . . . .	7
2.2.4	Licence . . . . .	7
2.2.5	Legal . . . . .	7
2.2.6	Scalability . . . . .	7
2.2.7	Security . . . . .	7
2.2.8	Testing . . . . .	8
2.2.9	Usability . . . . .	8
2.2.10	Performance . . . . .	8
2.2.11	Maintainability . . . . .	8
<b>3</b>	<b>Final Architecture &amp; Design Details</b>	<b>8</b>
3.1	Client Architecture . . . . .	9
3.2	Server Architecture . . . . .	10
<b>4</b>	<b>Development/Implementation Details</b>	<b>10</b>
4.1	Client . . . . .	10
4.2	Server . . . . .	11

4.2.1	Programming Language and Framework . . . . .	12
4.2.2	Request & Response Structure . . . . .	12
4.2.3	Git Usage . . . . .	13
4.2.4	Cloud Details . . . . .	13
4.2.5	Dependency Handling and Python Environments . . . . .	14
<b>5</b>	<b>Testing Details</b>	<b>15</b>
<b>6</b>	<b>Maintenance Plan &amp; Details</b>	<b>15</b>
<b>7</b>	<b>Other Project Elements</b>	<b>16</b>
7.1	Consideration of Various Factors . . . . .	16
7.2	Ethics & Professional Responsibilities . . . . .	17
7.3	Judgements and Impacts to Various Contexts . . . . .	17
7.4	Teamwork and Peer Contribution . . . . .	19
7.5	Project Plan Observed and Objectives Met . . . . .	19
7.6	New Knowledge Acquired and Learning Strategies Used . . . . .	21
<b>8</b>	<b>Conclusion and Future Work</b>	<b>21</b>
<b>9</b>	<b>Glossary</b>	<b>22</b>
<b>A</b>	<b>Installation</b>	<b>25</b>
<b>B</b>	<b>User Manual</b>	<b>25</b>
B.1	Main Menu . . . . .	25
B.2	Take Photo . . . . .	26
B.3	Choose Photo from Gallery . . . . .	27
B.4	Tool Selection Page . . . . .	28
B.5	Object Removal . . . . .	29
B.6	Rain Removal . . . . .	30
B.7	Haze Removal . . . . .	31
B.8	Theme Transfer . . . . .	32
B.9	Image Stitching . . . . .	34
B.10	Image Quality Evaluation . . . . .	36
B.11	Panorama . . . . .	37
B.12	Face Expression Modifier . . . . .	38

# 1 Introduction

Taking photos have turned into a daily practice, whether we are visiting a new country, celebrating a birthday or just having a normal day. In fact it has been estimated that more than 100 million photos and videos are uploaded to Instagram everyday [1]. Even though this is an impressive number, it does not even include the number of photos that people take for each upload. For some this number can be 20 and for other it might go up to 200 [2].

There are many reasons why one would have to take a lot of photos to get one shot they like. The light does not look right in one of them, someone walked behind them in another one, the building they want to be on the background did not fit the frame, they look too serious in that one and so on. Taking more photos can be a possible solution, but in most cases you might realise these problems when its too late. Going through your vacation photos to realise that picture you took in front of a historical church is missing the tip of the church is frustrating. Even if you do realise these problems when you have the chance to take more photos, some things cannot be changed. If it is raining and water droplets are in your lens, it is impossible to change the weather. And even if the problem is something you can change, why waste time taking tons of photos when it is possible for you to enhance or modify the ones you already have.

There are multiple tools that one can use to alter their photos. Advanced photoshop tools are often too complicated for occasional use. It is possible to find a tool that will remove a background item and then another one to alter your facial expression and then find a filter in another app that reflects your style. In this case, it is not common to find all these features in one place and the users have to jump from one app to another to get what they want.

With Photonom, we are proposing a mobile application that will allow users to modify their photos interactively and in one place whilst conserving the true nature of the image itself. In the next section the requirement details of Photonom will be explained. In the following sections, the final architecture of the application as well as design and other implementation details will be given. Then the elements related to the project such as project plan and teamwork will be given. The future work on Photonom will be explained in the final section. The references can be found at the end of this report. The user manual of Photonom can be found in the appendix.

## 2 Requirement Details

Photonom is a mobile application that provides various photo editing tools to its users. The tools of Photonom aim to offer an interactive space for the users to edit and enhance their photos. The users can upload their photo or take photos using Photonom. The tools provide the following functionalities: object removal, rain and haze removal, theme transfer, image stitching, image quality evaluation and changing facial expressions.

In this section the functional and nonfunctional requirements of Photonom will be explained.

### 2.1 Functional Requirements

In this section the functional requirements of Photonom will be explained.

#### 2.1.1 Uploading & Taking Photos

Once the application is open, the first thing the users will do is to choose the photo(s) they will be editing. The users will be able to access the other features only after they have selected a photo. If they want to upload their photo(s), they will select them from the gallery. Or they will also be able to take photo(s) using the applications camera function. If only one photo is selected then the user will be directed to a page which includes the photo itself and all available tools. If multiple photos are selected, they will be directed to a page which includes the photos in a slideshow, with the quality score written under the photo, and the panorama function as an available tool.

#### 2.1.2 Remove Objects

The users will be able remove items from their photo by using this tool. When the tool is selected, the user will select the object they want to remove by painting on it. The brush size for the task of painting can be changed according to the user's need. Once the object is specified it will be removed and the resulting image will be shown to the user. At this stage the user can finish using this tool, paint on another object for it to be removed, or clear the photo from all of the changes they have made.

#### 2.1.3 Rain Removal

With this functionality, the users will be able to remove the effects of rain droplets from their photo. The tool mostly removes the droplets that are in front of objects. The resulting

photo is shown to the user.

#### **2.1.4 Haze Removal**

With this functionality, the users will be able to remove haze from their photo. Once the tool is selected haze removal will be directly applied and the resulting image will be shown to the user.

#### **2.1.5 Theme Transfer**

Theme transfer tool will allow users to transfer the theme of one photo to another. Once this tool is selected the user will have to select another photo whose theme will be transferred to the photo they have originally chosen. After both photos are selected theme transfer will be applied and the results will be shown to the user.

#### **2.1.6 Image Stitching**

The image stitching tool allows users to stitch images together. When this tool is selected, the user will have to select another photo which will be stitched to the original photo. The system will locate the appropriate location for stitching and will show the resulting image to the user. At the moment, this tool can be used to stitch two images together.

#### **2.1.7 Image Quality Evaluation**

With this tool the users will be able to evaluate the quality of their photo(s). This tool is accessible for both single and multiple photo selection. With a single photo, the tool will be available in the tool selection page and once selected will show the photo with the score written underneath it. With multiple photos, the system will automatically apply this tool and in the resulting tool selection page will show the photos in a slideshow with the score written underneath each photo.

#### **2.1.8 Panorama**

Panorama tool that is only available when multiple photos are chosen. As mentioned before, when the user selects multiple photos, they will be directed to a page that shows the photos and panorama as an available tool. Once the tool is selected the system will attach the photos to each other to produce a panorama.

### **2.1.9 Facial Expression Modifier**

With this tool the users will be able to change their facial expression. When this tool is selected, the faces in the photo will be highlighted by the application and the user will select the face whose expression they want to change. The change itself will be done gradually with the help of a slider. Moving the slider to the right will result in a happier expression and moving it to the left will result in a sadder expression.

### **2.1.10 Saving the Edited Photo**

After a tool is applied and the resulting image is shown to the user, they will be able to save the photo to their device's gallery. At this point, they will also be able to continue editing by using the result as the original photo for the other tools.

## **2.2 Nonfunctional Requirements**

In this section the nonfunctional requirements of Photonom will be explained.

### **2.2.1 Accessibility**

Photonom should be accessible to everyone around the world. In order to ensure this, English will be the default language of the application.

### **2.2.2 Availability**

Photonom aims to provide many functionalities to its users and is created to have a modular easily-pluggable design. We also aim to ensure that any update to Photonom do not affect the availability of the application. In order to provide high availability but not spend too much monetary resources we have two servers and a simple orchestration tool supported cluster with 3 containers. Here is a simple flow for our release mechanism:

1. When there will be a new release for any of the modules, the load will be balanced between two containers instead of three.
2. The separated cluster will be deployed with the new changes and will be made sure its alive and well.
3. The load will be shifted to the newly deployed cluster.

4. After all previous jobs are completed in the previous two machines they will be deployed with the updated changes and they will be health checked.
5. The load will be separated between all clusters again.

### **2.2.3 Sufficient Network Bandwidth**

Photonom will require around 10 Mbps (Megabits per second) download and upload speed. This is due to the fact that Photonom performs computationally heavy operations and the transfer of image will require such speed to the increase in the quality of images taken by mobile phones. As we progress into a new era of 5G, these bandwidth requirements are simply minimal.

### **2.2.4 Licence**

Any dataset, product, library used in the product should be referenced properly.

### **2.2.5 Legal**

The system will perform on the users' private photographs. Therefore, we need to obtain the necessary permissions from the user as they start using the app. For general data protection regulation (GDPR) compliance, the system will show all the operations performed on the photographs and if the users want their data to be deleted from the system, it will also be available as an option [3]. For example their data could be their action logs in our database which we may use the track in order to improve our user experience.

### **2.2.6 Scalability**

The system is aimed to provide its service globally. As the usage increase, the system will need to scale up to stay alive during high loads. To serve this purpose, we aim to design a dockerized server which will be easily deployed [4]. This way the system can scale itself up in case of high loads automatically by the use of cloud services such as AWS, GCP [5] [6].

### **2.2.7 Security**

The personal data of the users will persist on the server during the operations. We will ensure our server is secure for the sake of our user's privacy. In case there is an intrusion, all operations will be cancelled and the corresponding container will be quarantined.

### **2.2.8 Testing**

The system is promised to be available at all times. To achieve this, there will be unit and integration tests that check the main functionality and the health of the system.

### **2.2.9 Usability**

Photonom is a mobile app and naturally will have a learning curve. One of our design goals is minimising the learning time as much as possible. Photonom includes many functionalities which potentially may detriment the usability of the application. In order to avoid this, the interface of Photonom should be as simple as possible and include only the necessary structures.

### **2.2.10 Performance**

Each action user wants to perform should not take longer than 3 seconds. Transition between screens and response to user functions should not take longer than a second.

### **2.2.11 Maintainability**

The system is going to have a modular design, hence it should not be highly coupled and a change in one of the modules should not affect the others. Moreover, there will be a defined code standard along with additional automated checks regarding the new code changes. There will be a reviewer to each pull request to increase the maintainability of the code in the future changes.

## **3 Final Architecture & Design Details**

The details will be discussed in two separate sections: Client Architecture and Server Architecture.

### 3.1 Client Architecture

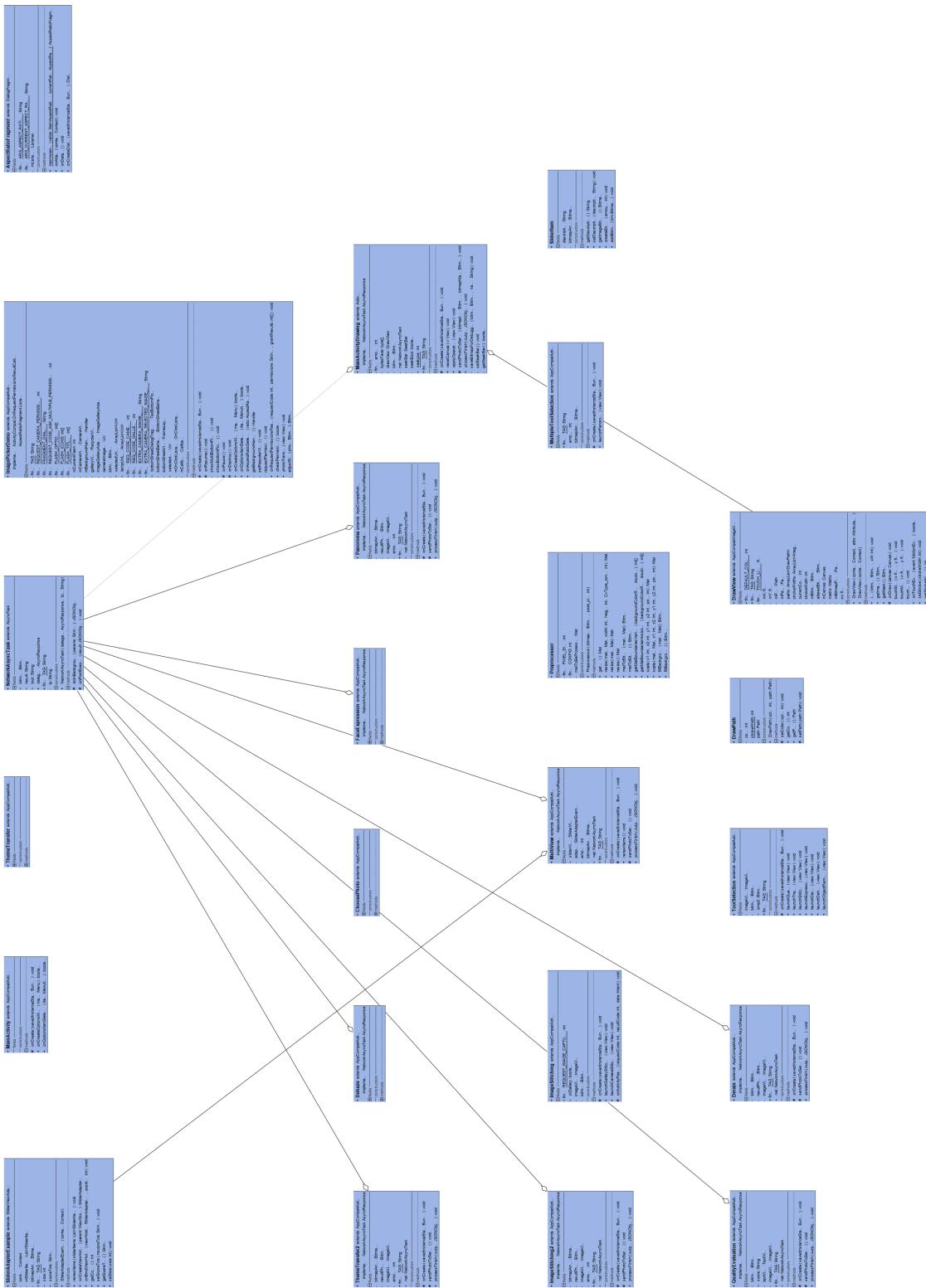


Figure 1: Class Diagram for Android Application

## 3.2 Server Architecture

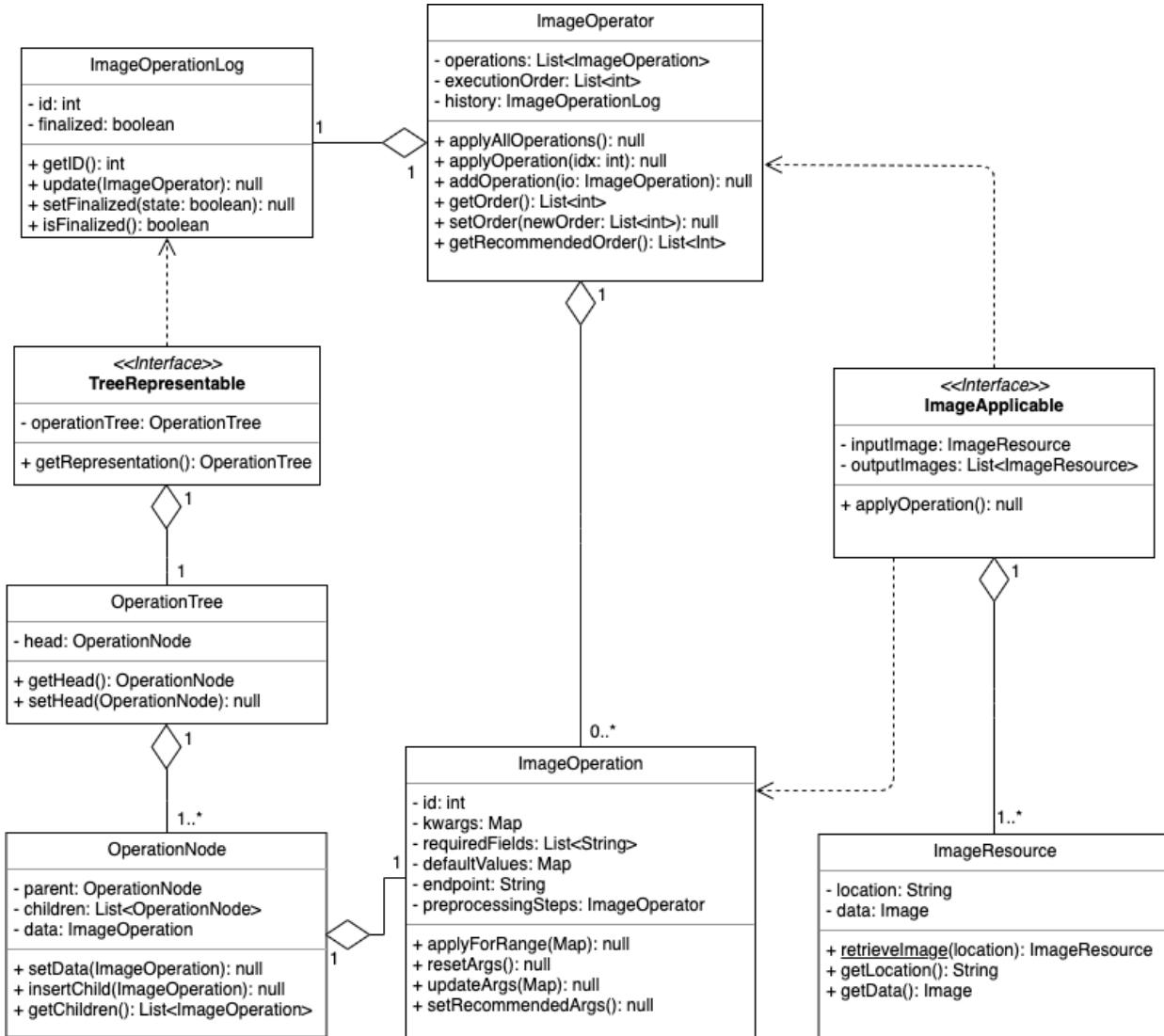


Figure 2: ImageOperator Package UML Diagram

## 4 Development/Implementation Details

The details will be discussed in two separate sections for the client and the server.

### 4.1 Client

The client application is available for Android and it is written in Java. This part of the system is mostly responsible for viewing elements of application to the user.

The first thing that client provides is choosing a photo or photos. Android Studio has default camera and gallery options. However, there is no option that the camera and the gallery can be used on the same screen. Also, it does not provide a proper multiple photo capturing activity. Therefore, we integrated a package, TedBottomPicker[7] which provides a combined camera/gallery activity. Users are able to choose and capture single and multiple photo(s) without changing any screen with that package.

Another activity that requires an additional package was image slider. It is on the screen that shows multiple images. The AutoImageSlider[8] package has been integrated for that part.

The challenge about these packages were the different dependencies caused by using them and changing their implementations to align them with Photonom.

Sending photos to server was one of the important duties of client part. Photos are first compressed and then encoded into a byte array. Sending process has been handled by a class which extends AsyncTask class[9]. When server completed its work, photos come to client as byte arrays and they are being decoded before showing them on the screen.

Activities for all tools have been created with some necessary tools for some activities. For instance, a seek bar feature added to facial expression screen. Among all seven activities, two of the activities required different handling. These are facial expression and object removal.

Since there can be multiple faces in a photo, face detection was necessary before the operation and the model used for facial expression did not provide it. To handle it in client part of the system, we used Face Detection with the Mobile Vision API[10] provided by Code Labs. After the user chose it, we cropped the chosen face and just send it to the server. When it returns, we put the modified parts to their old place.

Lastly, for the object removal feature, a paint system is created on the image. We have done it from scratch and no library or package have been used for it. Basically, the are two different images on the same place of the screen and one of them is transparent. User sees the real image while s/he actually paints the transparent one.

## 4.2 Server

The server's task is to handle requests coming from the client and return a meaningful response to the client.

#### 4.2.1 Programming Language and Framework

The server is written in Python and it is based upon Django framework. Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design [11]. The reasons we decided to use Django framework is as follows.

- **Scalability:** Django framework is proven to be scalable both horizontally and vertically. There are many huge companies using or have used Django at some point during their life cycles which hosted millions of unique users on daily basis including Instagram, Quora, Spotify, Bitbucket, etc [12].
- **Versatility:** For our project, we use many other projects from many different sources as submodules to our system. Django is used for many areas and has a modular reusable system which lets the developers work independently and makes it way easier to add new functionalities, in our case submodules, for the developers.
- **Security:** Usually there are people in teams who are focused on satisfying the security of the projects. We do not have such a resource to spare nor a knowledgeable person to allocate for this task. Hence, we looked for out of the box security features. Django provides many out-of-the-box security improvements that relieves us, the developers. These include but are not limited to SQL injection prevention, cross-site scripting, cross-site request forgery [11].
- **Long term support:** The life-time of our project is uncertain, and we are only 5 people and we can't afford to change frameworks in the middle of the road. Django provides LTS for bug fixes and security patches which makes it a good candidate among other Python web development frameworks.

#### 4.2.2 Request & Response Structure

For each submodule we have in our system we created an endpoint which the server listens and the client sends requests. Django allows us to easily create endpoints and connect them to basic methods. We chose to connect the endpoints to methods since we do not utilise Django models and we use submodules that are self-contained. For example when a request comes to an endpoint, the Django middle-wares redirect the request to the corresponding handling method and in that method we apply the operation we imported from the submodule. Then, return the appropriate response to the client. This does not only helps us treat to submodules as black boxes but it also allows us to keep the code clean and separated.

For each endpoint, there is a decided contract and the requests are all POST requests between the client and the server since the personal images of the users are being sent back and forward. POST requests provide the most secure connection along with HTTPS for our current use with the least effort. To transfer the images we use base 64 encoding. The client sends the request which contains a JSON that is in line with the decided contract between server and client.

The server receives the request and then parses the JSON in order to extract the image. After the image is extracted, the image is decoded using base 64 package in Python standard library. After this decoding operation we pass this image to the corresponding operation submodule, receive the output and send it back to the client in a POST request, after encoding it with base 64 package as the request.

#### 4.2.3 Git Usage

We have many submodules that we have to include in the git repository of the server. These submodules can be thought of as other projects that we need to use in our main projects. For similar use cases, git provides the developers with a functionality called submodules. Submodules allow you to keep a Git repository as a subdirectory of another Git repository. This lets you clone another repository into your project and keep your commits separate [13].

#### 4.2.4 Cloud Details

From the first day forward we used Google Cloud Platform (GCP) to deploy our project. While this decision had its perks, it also introduced an overhead and costs to our team. First of all none of us had powerful computers powerful enough to handle many GPU oriented operations concurrently. Moreover, considering the COVID-19 situation and the need to work remotely, it would be way harder to use a local PC to deploy our projects. GCP provides 300\$ of free credits during the trial period and we utilised these credits to work on our project.

In order to create virtual machines that has allocated GPUs, we contacted GCP support since there is a GPU limit for each account and its 0 by default. It was increased to 1. We used N1 type virtual machines which are Compute Engine's first generation general-purpose machine types [6].

The specific model we used is "n1-standard-4". It provides 4 vCPUs, 15GB memory, a local ssd with 100GB capacity, and network egress bandwith of 10 Gbps. Upon customizing

this model, we added a Tesla T4 GPU. The GPU that GCP allocates provides a clock frequency of 1590 MHz with boost, memory clock of 5000 MHz, VRAM size of 16GB, memory bus width of 256-bits, and the Turing architecture [14].

To be prepared for any case, we also utilize machine images functionality of GCP and generate machine image of our main virtual machine weekly. We keep the last two weeks' images and delete the older ones. Since the GPU is costly, we do not leave the server open at all times and we start it whenever we need. This introduces another problem we faced many times. GCP may not be able to provide a Tesla T4 GPU at all times. If all T4 GPUs in our region are allocated, we can't start our server. To overcome this issue, we always have 3 synced virtual machines in 3 different regions: eu, us, and asia.

#### 4.2.5 Dependency Handling and Python Environments

We use the following projects as our submodules.

- Quality Evaluation [15]
- Image Styling [16]
- Image Stitching [17]
- Object Removal [18]
- Facial Expression Modification [19]
- Rain and Haze Removal [20]

Even though all of these use similar libraries as NumPy, Pillow, OpenCv, CuDNN, Cuda, PyTorch, TensorFlow, Sci-kit, and so on, they are all developed with different versions of these libraries [21, 22, 23, 24, 25, 26, 27, 28]. We utilized Anaconda to handle environments. Conda makes it easy to manage multiple data environments that can be maintained and run separately without interference from each other [29].

Another problem is that these are supposed to run in a single Django project. Initially we tested out many versions to find the most suitable version for the main environment. At first, we managed to be able to continue with a single environment and a single server.

Since we also use the pretrained models of these projects, at some point we were not able to continue with the approach we described above. Different versions interpreted same models differently and the model weights did not match the model descriptions of these

projects. At this point we separated outlying projects and ran them inside different Django servers and we deployed their endpoints to different servers.

The next major problem we faced was not related to Python package management. It was CUDA versioning. The CUDA versions these projects use ranged from 8.0 to 10.1. There was no way to manage different versions in a single server but we got lucky and CUDA distribution with version 9.1 seemed to operate successfully for all of our projects that require CUDA. Otherwise it would be very problematic since our *GLOBAL\_GPU\_LIMIT* for GCP is limited to a single GPU. We can't run multiple virtual machines with allocated GPUs simultaneously.

## 5 Testing Details

We wrote tests simulating the request response cycle during the integration of submodules to the server. Each operation had its own specific test demonstrating the contract between the client and the server. We used these tests later to debug our implementation and make sure the components work. The tests were most useful during the environment setups since one change had the possibility to fail a feature.

## 6 Maintenance Plan & Details

During the design process, our concern was coming up with a system that is not highly coupled. This logic allowed us to obtain a final product which is easy to modify. Due to our system design, the maintenance of the product won't be cumbersome. In order to ensure that the application works flawlessly, it is required to make some controls regularly within a planned schedule and also when a problem arises. Considering the nature of the application, it is more likely to have some problems with the server-side of the application. So, the maintenance should be focused on the server-side. With our experiences, we decided that the maintenance for the server-side should be conducted once a month. Also, we should be prepared for the unexpected errors that arise from the server-side. These problems are considered as unexpected since they are due to the intensity of the usage of the credits in our accounts. In addition to this maintenance plan, the user feedback regarding the user experience and the problems they encounter while using our application will be important for us. By that way, we will be able to determine the parts we need to focus on the client-side.

## 7 Other Project Elements

### 7.1 Consideration of Various Factors

While deciding the functionalities of the end product, various factors are considered related to the demands and needs of the users. In addition to affecting which functionality is added to the application, these factors also affected the design of the project. The said factors can be seen in Table 3.

	Effect Level	Effect
Technological factors	5/5	As the effect of the social media applications on the society increases, the people value the photographs they post more. Our application can help improve these photographs by providing editing tools.
Social factors	3/5	While adding the object removal functionality, the privacy of the people on the background is also considered. It might be disturbing for them to be shown in a photograph of a stranger. In order to avoid this situation, the object removal is added to the design of the project.
Public welfare	4/5	The increase of a public welfare enables people to spend more time for social activities such as managing social media accounts. Since more people are involved in this, the application should reach more people. That's why, during the design of the project usability was one of our priorities.

Cultural factors	4/5	In the cases where the photo someone took in front of a historical building but some parts of it is missing, we wanted to ensure people would still be able to look back on those photos and remember that historical structure in its true form. As a result we added image stitching during analysis. Cultural factors were effective in this decision since we wanted to conserve the nature of the culture being reflected on the photo.
------------------	-----	--

Table 1: Factors that affected analysis and design

## 7.2 Ethics & Professional Responsibilities

In this section the ethical and professional responsibilities of this project is specified.

- The user should consent to Photonom accessing and modifying their photo. However, we cannot assure that the user of Photonom is the actual owner of the photo or whether the consent of the owner is taken before photo is uploaded. Photonom does not take further responsibility of taking consent of the owner and it is assumed that the user has taken the consent of the owner.
- Pictures are not stored in our servers after the user has completed editing. However, editing is done in our servers, in the meantime, security for the user's data will be provided. User's data will not be shared with other applications or people without user's consent.
- The user is able to save the processed picture so it is the user's responsibility to abide by the copyright of pictures.
- Licenses for third-party APIs and libraries is checked before usage.

## 7.3 Judgements and Impacts to Various Contexts

Judgement Description:		Pictures are not stored in our servers after the user has completed editing.
	Impact Level	Impact Description
Impact in Global Context	1/5	This decision does not have any impact in global context.
Impact in Economic Context	5/5	Not storing pictures in server help us not to pay extra money to the server for keeping such data.
Impact in Environmental Context	1/5	This judgement does not have any negative or positive impact on environmental context.
Impact in Societal Context	4/5	User will feel secure that their pictures will not be saved and used for any other purposes. Thus, they will be more engaged in terms of social comfort.

Table 2: Not Storing Pictures as a Judgement in Different Contexts

Judgement Description:		We use globally understandable icons in user interface
	Impact Level	Impact Description
Impact in Global Context	5/5	Even though English is the only language we offer, through using icons in user interface rather than texted buttons make our app usable by everyone.
Impact in Economic Context	5/5	Using icons that are available for everyone, allows us to not spend money on specialised icons.
Impact in Environmental Context	1/5	This judgement has no impact on environmental context.

Impact in Societal Context	4/5	Since our app can be used to edit the pictures that are shared on social media, this judgement helps more people to be socially engaged.
----------------------------	-----	--

Table 3: Global Icons as a Judgement in Different Contexts

## 7.4 Teamwork and Peer Contribution

All team members did their best to complete all functionality of the app. Work division has been done as in Table 4 and all team members helped each other when needed. This project also improved us to work effectively with others and allowed us to shape and better our skills in computer science and teamwork.

WP#	Work Package Title	Leader	Members Involved
<b>WP1</b>	<b>Reports</b>	Idil Hanhan	All Team
WP2	High Level Design Report	Idil Hanhan	Idil, Kerem, Berfin, Beyza
WP3	Low Level Design Report	Kerem Yilmaz	Omer, Beyza, Berfin
WP4	Final Report	Omer F. Babademez	All Team
WP5	Demo	Berfin Kucuk	Omer
WP6	Presentation	Beyza Kalkanli	Kerem
<b>WP7</b>	<b>Development</b>	Kerem Yilmaz	All Team
WP8	Research	Beyza Kalkanli	All Team
WP9	Server	Omer F. Babademez	Idil, Kerem, Omer
WP10	Mobile Application	Berfin Kucuk	All Team

Table 4: Distribution of list of work packages

## 7.5 Project Plan Observed and Objectives Met

In the beginning of the project, we made a plan to decide how to complete the project and in which order we should finish the tasks with specified deadlines. We also divide tasks to group members such that everyone can focus on the specified tasks. When we divide the work, we also put estimated completion time for each tasks. However, we realize that tasks usually take more time than we have estimated. Although this situation affect the workflow of the

project, at the end, we were able to complete tasks before the specified deadlines. Also, at the beginning of the project, we decided to use Flutter to implement our user interface. However, we realize that Flutter does not have some functionalities or not scalable to our project such as multiple photo selection from gallery due to conflict in image format. Then, we decided to migrate our project to Android. This situation costs us extra implementation efforts.

In the research components of the project, we managed to implement all of the core functionalities we decided before. Before the implementation of the project, we determined the repositories and datasets we need for this project. However, during the implementation of the project, we didn't use some of these repositories and datasets since they didn't meet our expectations. The main problem with these was the performance difference between what we obtain when we ran the code and what we obtain with the demo. So, we tried multiple models for each functionality in order to obtain the best results.

In our previous reports, we emphasized that there is an additional tool for the head pose modification which is expected to be integrated into the application if time permits. The problem we had with this tool was not due to the time. For this tool, we tried multiple methods. The first one was using a model that directly generates different photos with different rotation angles. However, the resulting images were not successful enough to use in the implementation. So, we decided to follow a different method which is generating the 3D model of a face from a single image. We managed to obtain a per-vertex color .obj file that keeps the necessary information to visualize the 3D model. The photo that we used as an example is given in Figure 3a. The visualization of the resulting 3D model with Meshlab we obtained with this photo is given in Figure 3b. Although we managed to visualize such models with Meshlab without any problem, the framework that we used in Android for the visualization purposes, min3D, was expecting to work with Wavefront .obj type. According to our research, we found that additional steps with different software such as Meshlab or Blender were needed for this conversion from per-vertex color .obj file to Wavefront .obj file. After these steps, we visualized the face model with Android. However, there were problems regarding the texture of the model which may be originated from the conversion we had to do before. Since there are additional operations needed to be conducted via other software, we didn't include this tool in our final product.

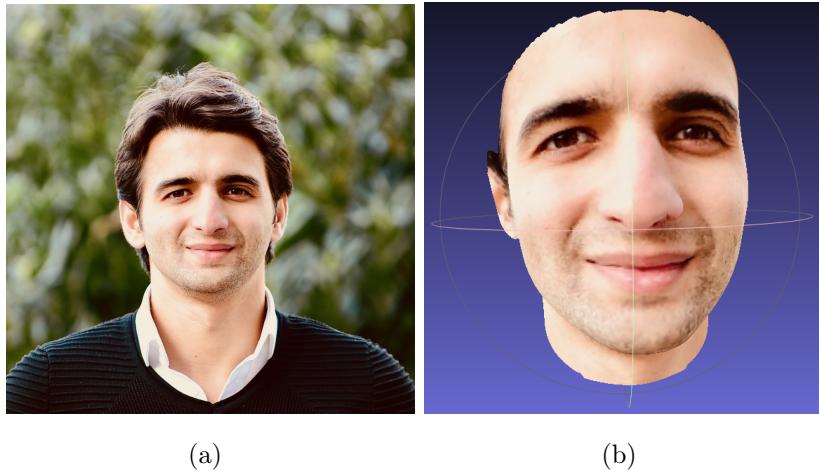


Figure 3: (a) Single image given to model (b) Resulting 3D model

## 7.6 New Knowledge Acquired and Learning Strategies Used

During our senior design project, our team was required to learn many things regarding both the research and the development work packages. In order to learn the necessary subjects we've come up with two different strategies. The learning items were as follows.

For the research related subjects, we were required to read a few general articles or research papers regarding the subject. After we do so, we needed to find an open source project and examined its code to have a somewhat engineering perspective to it. After these steps were completed, we were on-boarded enough to focus the main task.

For the development related subjects, for us, the best way to learn a new framework was to start using it. The strategy was as follows. We first used the original quick start tutorial if it exists, otherwise we followed a comprehensive tutorial to set up the framework and learn about its philosophies. Then, the team members focused on the main focus of the assigned task. Along the way, when necessary, purpose focused tutorials helped us learn the tools we will be using while at the same time making some progress regarding our senior design project, Photonom.

## 8 Conclusion and Future Work

Our aim for this project was providing users an application where they can

- change the facial expressions of the people in their photo,
- remove the objects they don't want to exist in their photos,

- stitch two photos that have common parts,
- create panorama using multiple photos,
- remove the unwanted effects of rain and haze,
- transfer the color scheme of an image to the target image,
- view the quality scores of their photos.

We used multiple models to accomplish these tasks and utilized different frameworks such as PyTorch and Tensorflow. For the implementation of the application, we decided to work with Android Studio. At the end of the implementation process, we managed to provide the core functionalities we decided before. So, in our final product the functionalities listed above are included.

In our previous reports, we underlined that an additional tool which is head pose modification might be provided if time permits. We were hoping to include this tool which allows users to rotate the faces in their photo with a certain degree. Due to the not time limitations but technical limitations we have encountered during the implementation of this tool, we couldn't finalize the use of this property. Although we managed to obtain the 3D model of a face from a single photo, we couldn't integrate this functionality to the final product since the resulting 3D models required further operations with additional tools such as Meshlab or Blender. As future work, if the additional operations can be completed through coding, the head pose modification tool might also be included in the final product.

In one of our functionalities, we provide the image stitching feature which stitch selected 2 images. However, if user wants to stitch more than 2 images, they have to do it iteratively because user interface allows to select only one image to be stitched. Thus, as future work, if the operations can be completed without preventing the usability of user interface, more than 2 images can be stitched at one operation.

As a result, our application, Photonom, is ready to be used and includes the functionalities listed above.

## 9 Glossary

**Photonom:** A word derived from the merging of two words: photoshop and autonom.

## References

- [1] “Instagram by the numbers (2019): Stats, demographics fun facts,” 2019. [Online]. Available: <https://www.omnicoreagency.com/instagram-statistics/>
- [2] J. Brucculieri, “4 instagrammers show us how many photos they took before nailing ‘the shot’,” 2018. [Online]. Available: [https://www.huffpost.com/entry/instagram-photocamerarolls\\_n\\_5ac4ed48e4b063ce2e58131f](https://www.huffpost.com/entry/instagram-photocamerarolls_n_5ac4ed48e4b063ce2e58131f)
- [3] 2019. [Online]. Available: <https://gdpr.eu/>
- [4] 2019. [Online]. Available: <https://www.docker.com/>
- [5] 2019. [Online]. Available: <https://aws.amazon.com/>
- [6] 2019. [Online]. Available: <https://cloud.google.com/>
- [7] 2016. [Online]. Available: <https://github.com/ParkSangGwon/TedBottomPicker>
- [8] 2018. [Online]. Available: <https://github.com/smarteist/Android-Image-Slider>
- [9] [Online]. Available: <https://developer.android.com/reference/android/os/AsyncTask>
- [10] [Online]. Available: <https://codelabs.developers.google.com/codelabs/face-detection/index.html?index=..%2F..index0>
- [11] 2020. [Online]. Available: <https://www.djangoproject.com>
- [12] 2020. [Online]. Available:  
<https://coderbook.com/@marcus/how-scalable-are-websites-built-in-django-framework>
- [13] 2020. [Online]. Available: <https://git-scm.com/book/en/v2/Git-Tools-Submodules>
- [14] 2020. [Online]. Available: <https://www.nvidia.com/content/dam/en-za/Solutions/Data-Center/tesla-t4/t4-tensor-core-product-brief.pdf>
- [15] 2020. [Online]. Available: <https://github.com/idealo/image-quality-assessment>
- [16] 2020. [Online]. Available: <https://github.com/NVIDIA/FastPhotoStyle>
- [17] 2020. [Online]. Available: <https://github.com/ppwwyyxx/OpenPano>
- [18] 2020. [Online]. Available: <https://github.com/MathiasGruber/PConv-Keras>

- [19] 2020. [Online]. Available: <https://github.com/Puzer/stylegan-encoder>
- [20] 2020. [Online]. Available: <https://github.com/cddlyf/GCANet>
- [21] 2020. [Online]. Available: <https://numpy.org/>
- [22] 2020. [Online]. Available: <https://pillow.readthedocs.io/en/stable/>
- [23] H. Partnership, “Opencv,” 2020. [Online]. Available: <https://opencv.org/>
- [24] 2020. [Online]. Available: <https://developer.nvidia.com/cudnn>
- [25] 2020. [Online]. Available: <https://developer.nvidia.com/cuda-toolkit>
- [26] 2020. [Online]. Available: <https://pytorch.org/>
- [27] 2020. [Online]. Available: <https://www.tensorflow.org/>
- [28] [Online]. Available: <https://scikit-learn.org/stable/>
- [29] 2020. [Online]. Available: <https://www.anaconda.com>

## A Installation

The users can install this manually by the APK provided by us until we publish this application in Google Play Store.

## B User Manual

The user manual of Photonom is given in this appendix.

### B.1 Main Menu

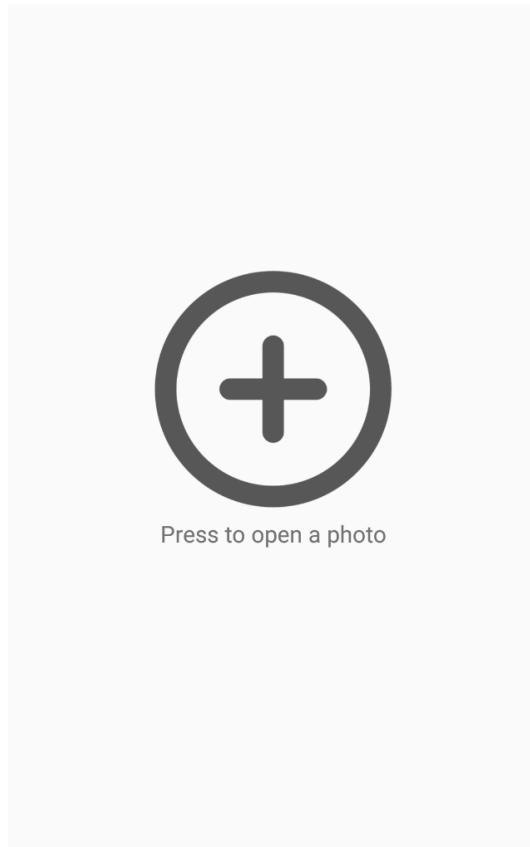


Figure 4: Main Page of Photonom

The main menu is the first page of Photonom. By clicking on the plus at the centre of the photo, the user's will be able to either take a photo or choose a photo from their gallery. These actions can be seen in section 2 and 3 of this appendix.

## B.2 Take Photo

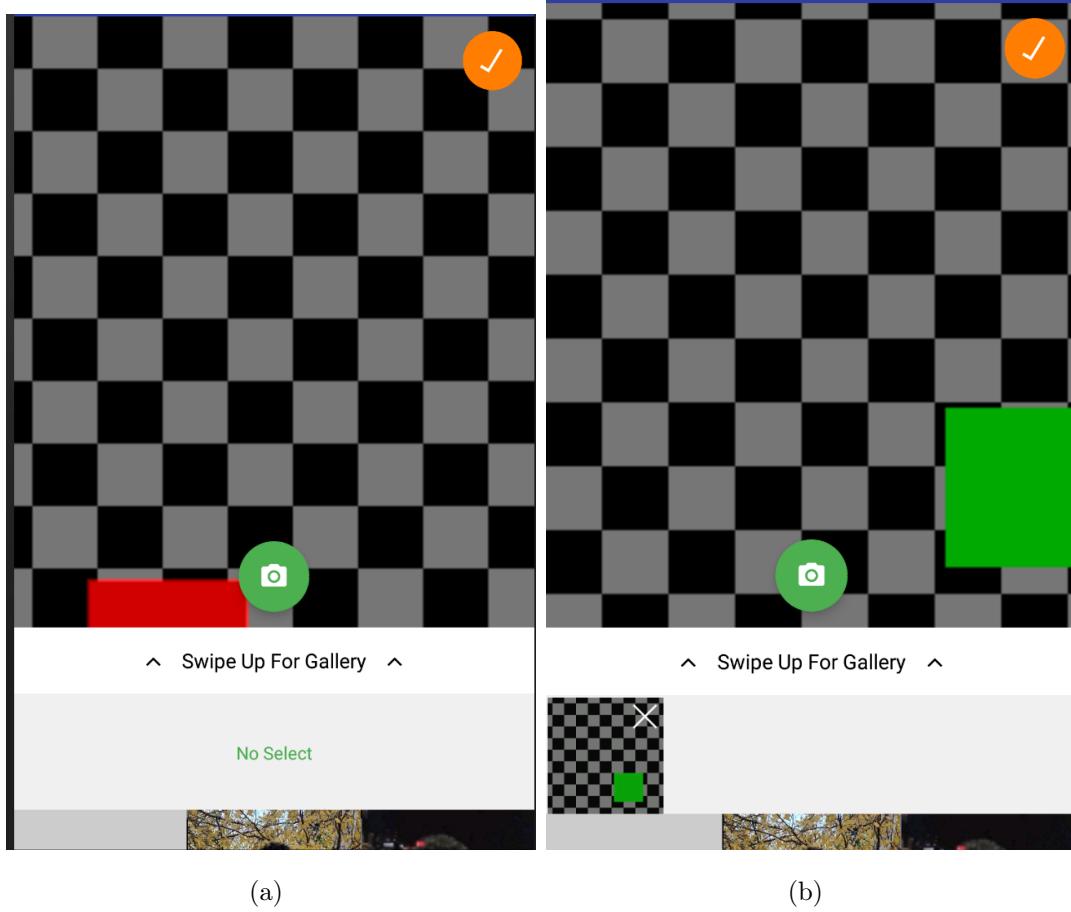


Figure 5: (a)Initial Camera Screen (b) Camera Screen after user takes a photo

This is the page the user is directed to after they pressed the plus on main page of Photonom, seen in Figure 4.

In this page the user can take photos by pressing the green camera button on the page. They can take multiple photos by pressing the button. Once they are done, they will press the orange check button seen in the top right corner of the screen.

If the user only took one photo, then they will be directed to the Tool Selection page, seen in Figure 7. If they took multiple photos then they will be directed to the page seen in Figure 14a and 14b.

If the user prefers to select photos from the gallery, they will be able to reach it by using the "Swipe Up" feature seen in the screen which will direct the user to the pages seen in section 3.

### B.3 Choose Photo from Gallery

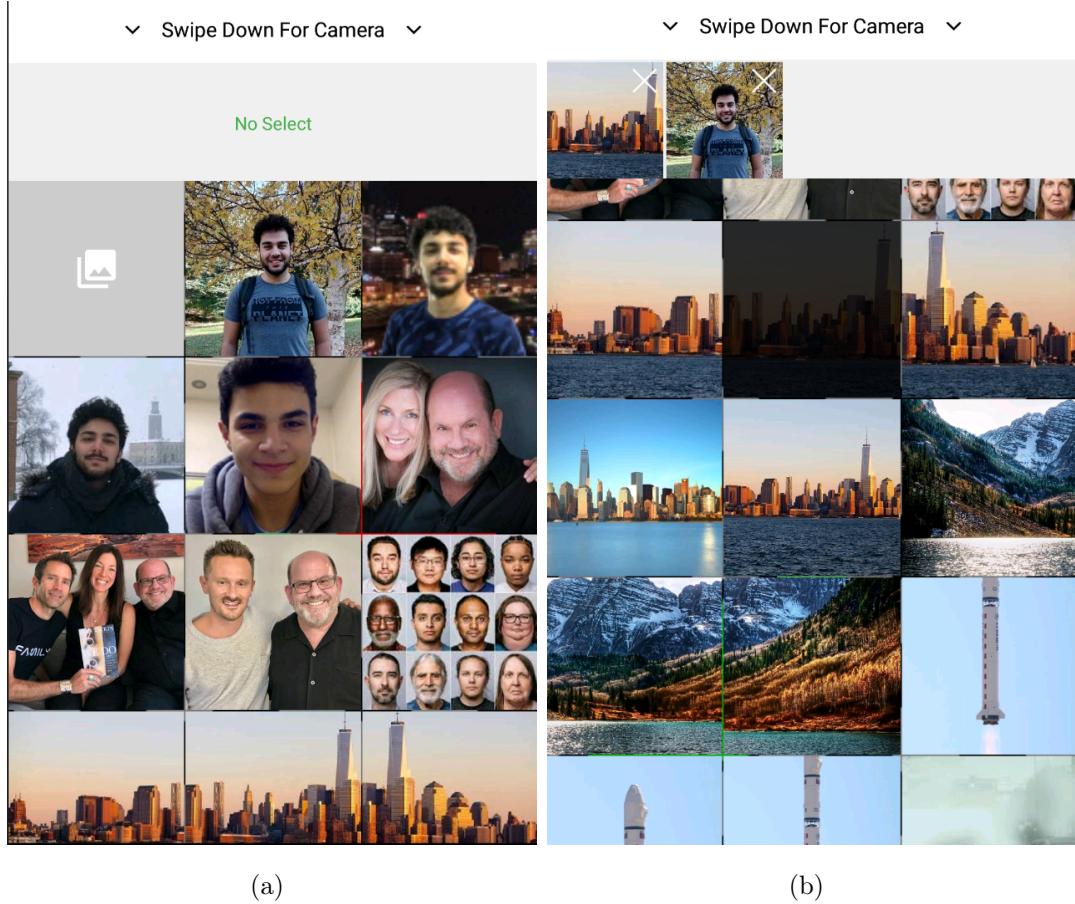


Figure 6: (a) Initial Gallery Screen (b) Gallery Screen after a photo is selected

This is the page the user is directed to after they swiped up in the camera screen of Photonom, seen in Figure 6a.

In this page, the user can select the photo(s) they want to choose by directly selecting them.

If the user only selected one photo, then they will be directed to the Tool Selection page, seen in Figure 7. If they selected multiple photos then they will be directed to the page seen in Figure 14a and 14b.

If the user wants to go back to the camera screen, they will be able to do so by using the "Swipe Down" feature seen in the screen which will direct the user to the pages seen in section 2.

## B.4 Tool Selection Page

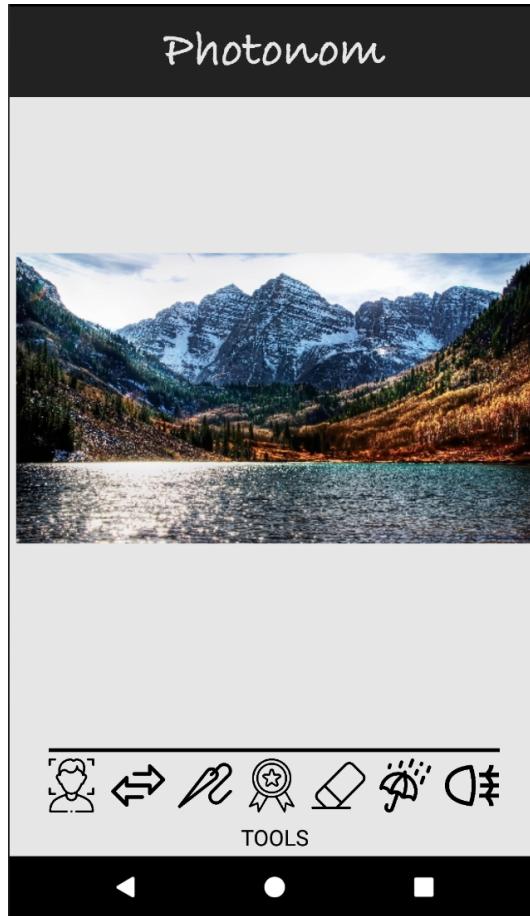


Figure 7: Tool Selection Page of Photonom

After the user has selected a single photo they want to edit, this is the page they will be directed to. The page includes the photo chosen by the user and the editing tools.

Each editing tool can be accessed by pressing the figure that represent that tool. The tools seen on this page, from left to right, are facial expression modifier, theme transfer, image stitching, image quality evaluation, object removal, rain removal and haze removal. The pages of this tools can be seen under the sections with the same name as the tool.

## B.5 Object Removal

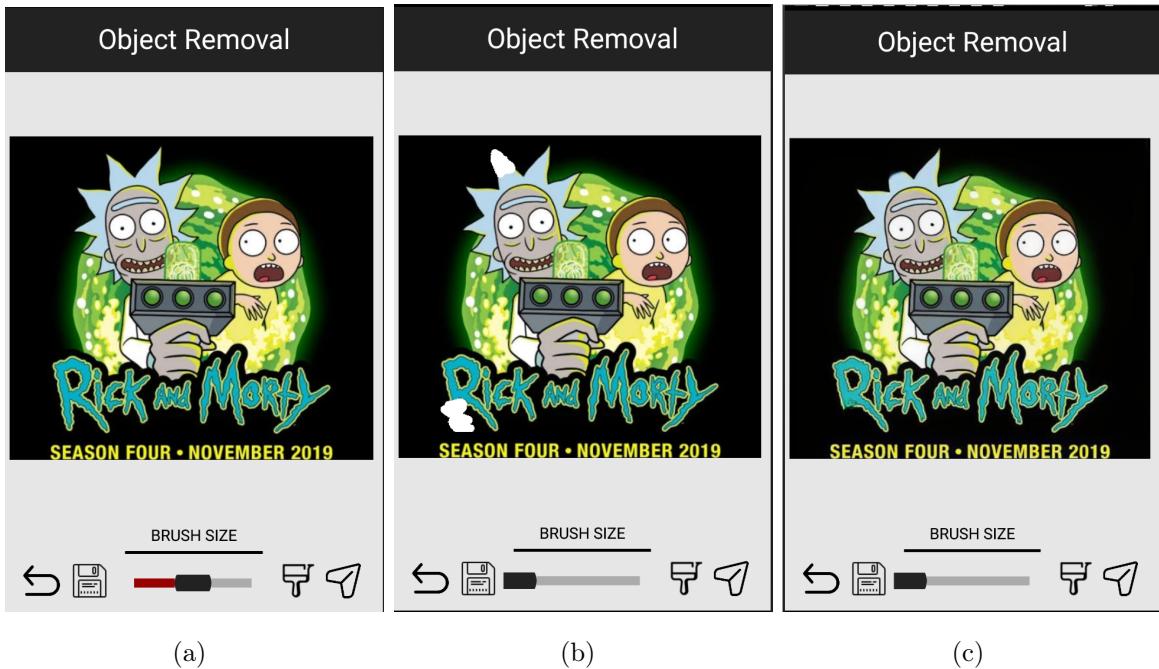


Figure 8: (a) Initial Object Removal Screen (b) Painting on the object to be removed (c) Object Removal Page after the tool is applied

The pages related to the Object Removal tool can be seen in this section. A sample usage of this tool can be seen in Figure 8, starting from a to b and to c.

The page seen in Figure 8a, is the page the user is directed to after object removal is selected. The slider at the bottom centre of the page can be used to adjust the brush size. Once the brush size is selected, the user can paint on the section they want to remove as seen in Figure 8b. By pressing the brush button they will be able to apply the removal and the resulting page can be seen in Figure 8c.

If they want to scratch all of the changes they made they can press the undo icon at the bottom left corner of the page. They can save the image to their device's gallery by pressing the "Save" icon. If the user wants to continue editing with another tool, they will be able to reach the tool selection page by pressing the paper plane icon at the bottom right corner of the page.

## B.6 Rain Removal

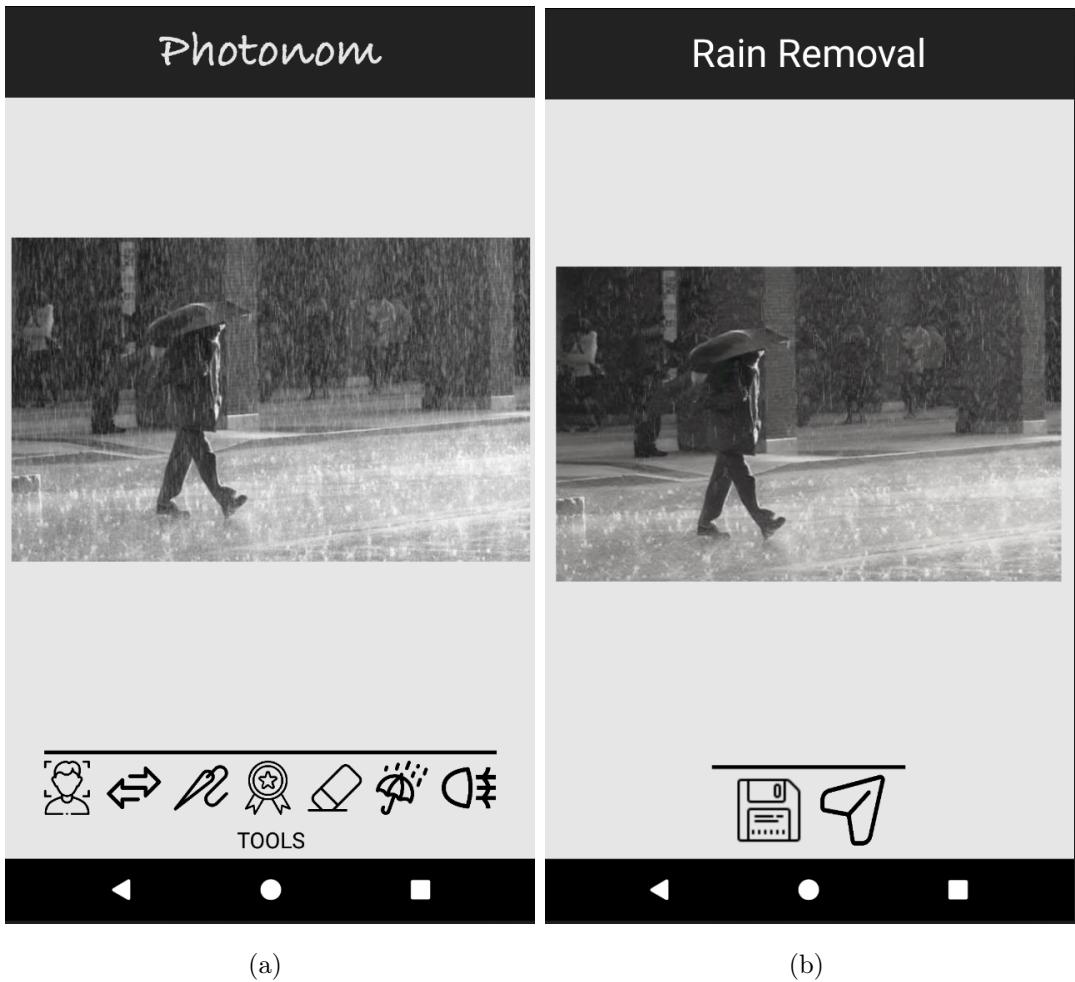


Figure 9: (a) Tool Selection Page (b) Rain Removal Page

In Figure 9a, another screen for the tool selection page can be seen. If the user selects the rain removal tool from this screen, then they will be directed to the page seen in Figure 9b. The photo after the tool is applied can be seen in this page.

They can save the image to their device's gallery by pressing the "Save" icon. If the user wants to continue editing with another tool, they will be able to reach the tool selection page by pressing the paper plane icon.

## B.7 Haze Removal



Figure 10: (a) Tool Selection Page (b) Haze Removal Page

In Figure 10a, another screen for the tool selection page can be seen. If the user selects the haze removal tool from this screen, then they will be directed to the page seen in Figure 10b. The photo after the tool is applied can be seen in this page. They can save the image to their device's gallery by pressing the "Save" icon. If the user wants to continue editing with another tool, they will be able to reach the tool selection page by pressing the paper plane icon.

## B.8 Theme Transfer

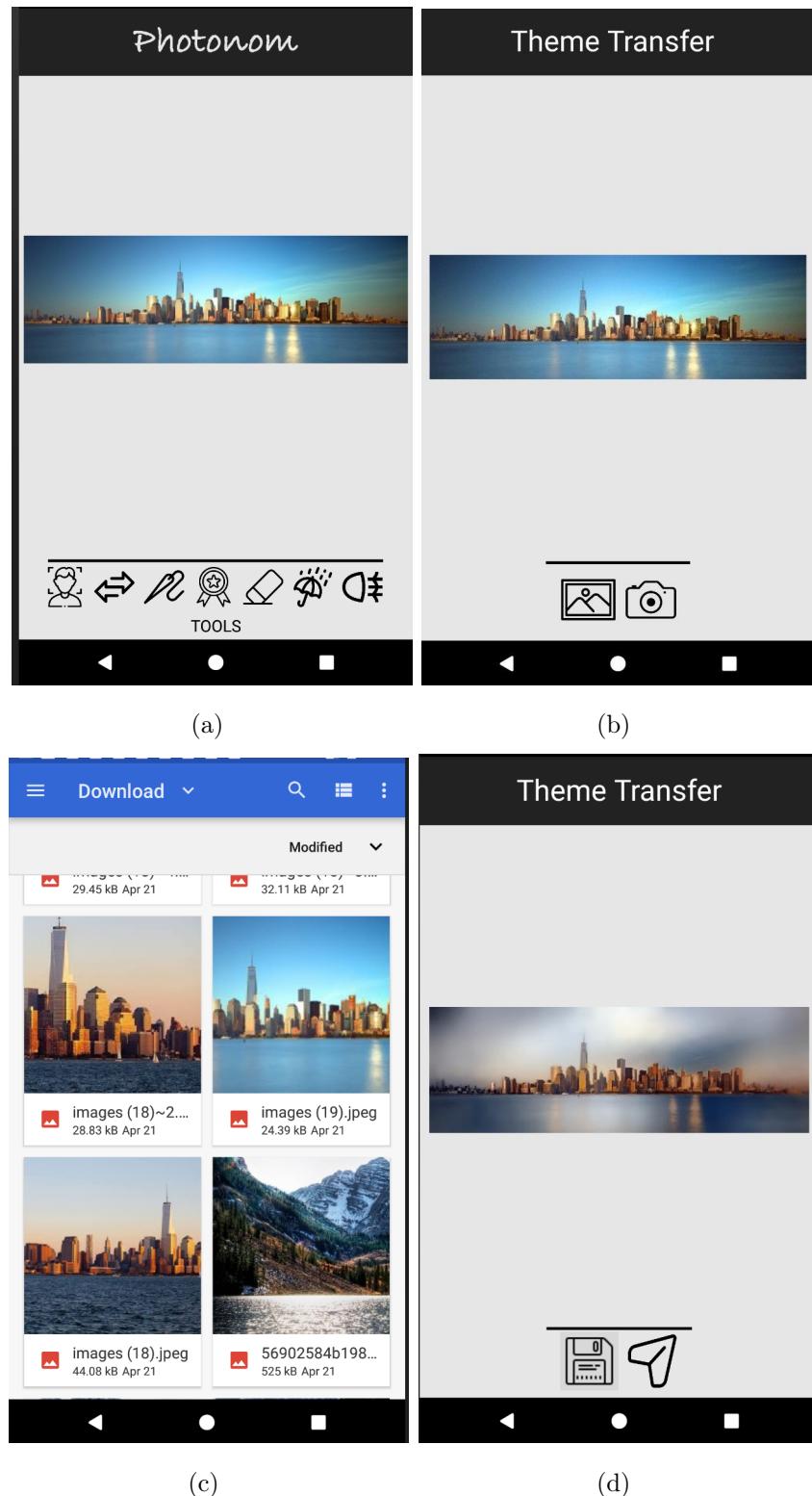


Figure 11: (a) Tool Selection Page (b) Original Photo (c) Theme Photo Selection (d) Theme Transfer page after the tool is applied

The pages related to the Theme Transfer tool can be seen in this section. A sample usage of this tool can be seen in Figure 11, starting from a and finishing at d.

The page seen in Figure 11a is the tool selection, if the user selects the theme transfer tool then they will be directed to page seen in Figure 10b. At this point, the user has to select a photo whose theme will be transferred to the original photo, seen at this page. This theme photo can be selected either from the gallery or it can be taken by using the camera. The selection of this photo can be seen in Figure 11c. Once the theme photo is selected, the tool is applied and the user is directed to the page seen in Figure 11d. The photo after the tool is applied can be seen in this page. The user can save the image to their device's gallery by pressing the "Save" icon. If the user wants to continue editing with another tool, they will be able to reach the tool selection page by pressing the paper plane icon.

## B.9 Image Stitching

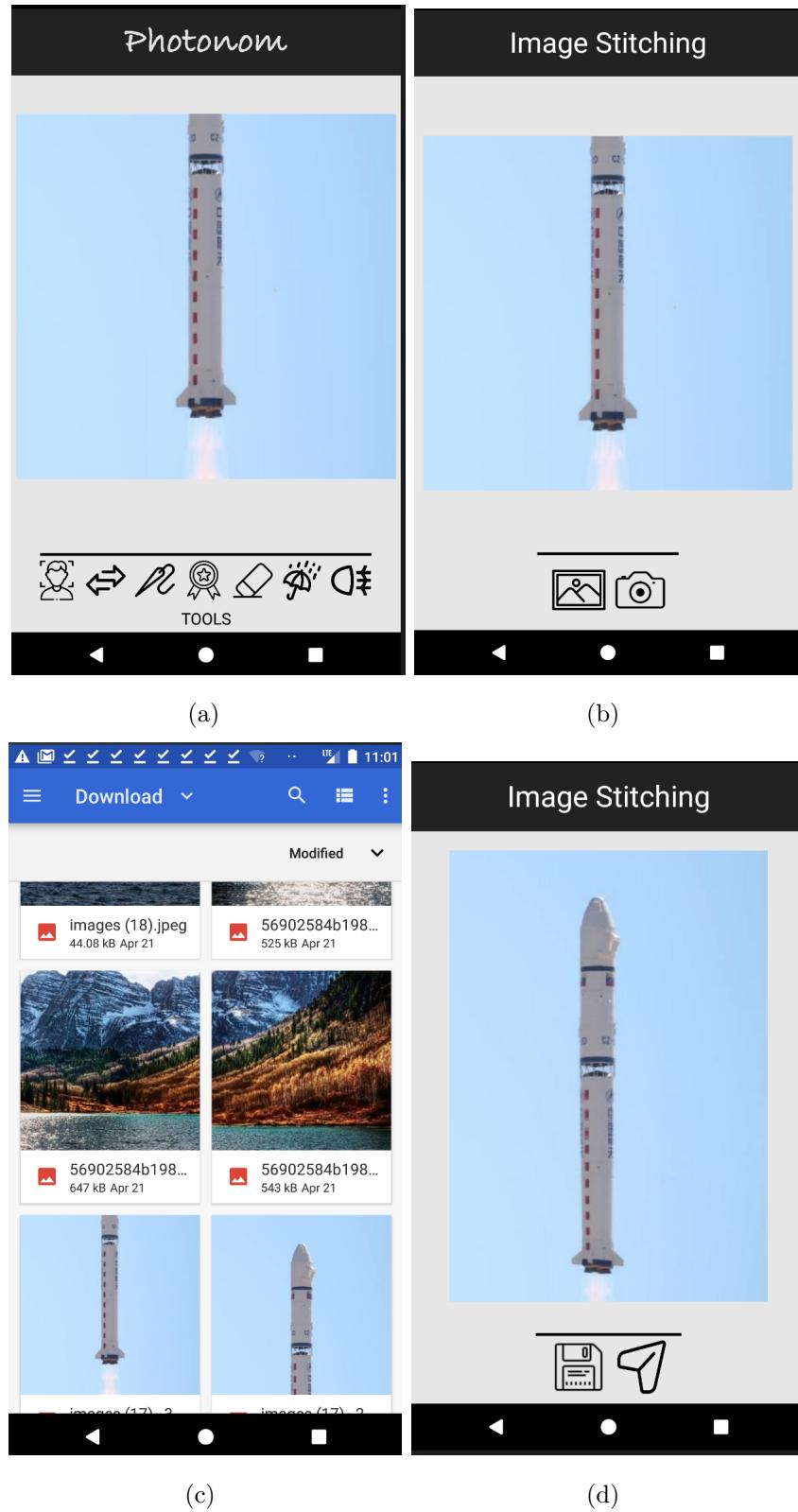


Figure 12: (a) Tool Selection Page (b) Original Photo (c) Image Stitching Photo Selection  
(d) Image Stitching page after the tool is applied

The pages related to the Image Stitching tool can be seen in this section. A sample usage of this tool can be seen in Figure 12, starting from a and finishing at d.

The page seen in Figure 12a is the tool selection, if the user selects the image stitching tool then they will be directed to page seen in Figure 12b. At this point, the user has to select the photo that will be stitched to the original photo. The selection of this photo can be seen in Figure 12c. Once the photo is selected, the tool is applied and the user is directed to the page seen in Figure 10d. The photo after the tool is applied can be seen in this page. The user can save the image to their device's gallery by pressing the "Save" icon. If the user wants to continue editing with another tool, they will be able to reach the tool selection page by pressing the paper plane icon at the bottom right corner of the page.

## B.10 Image Quality Evaluation

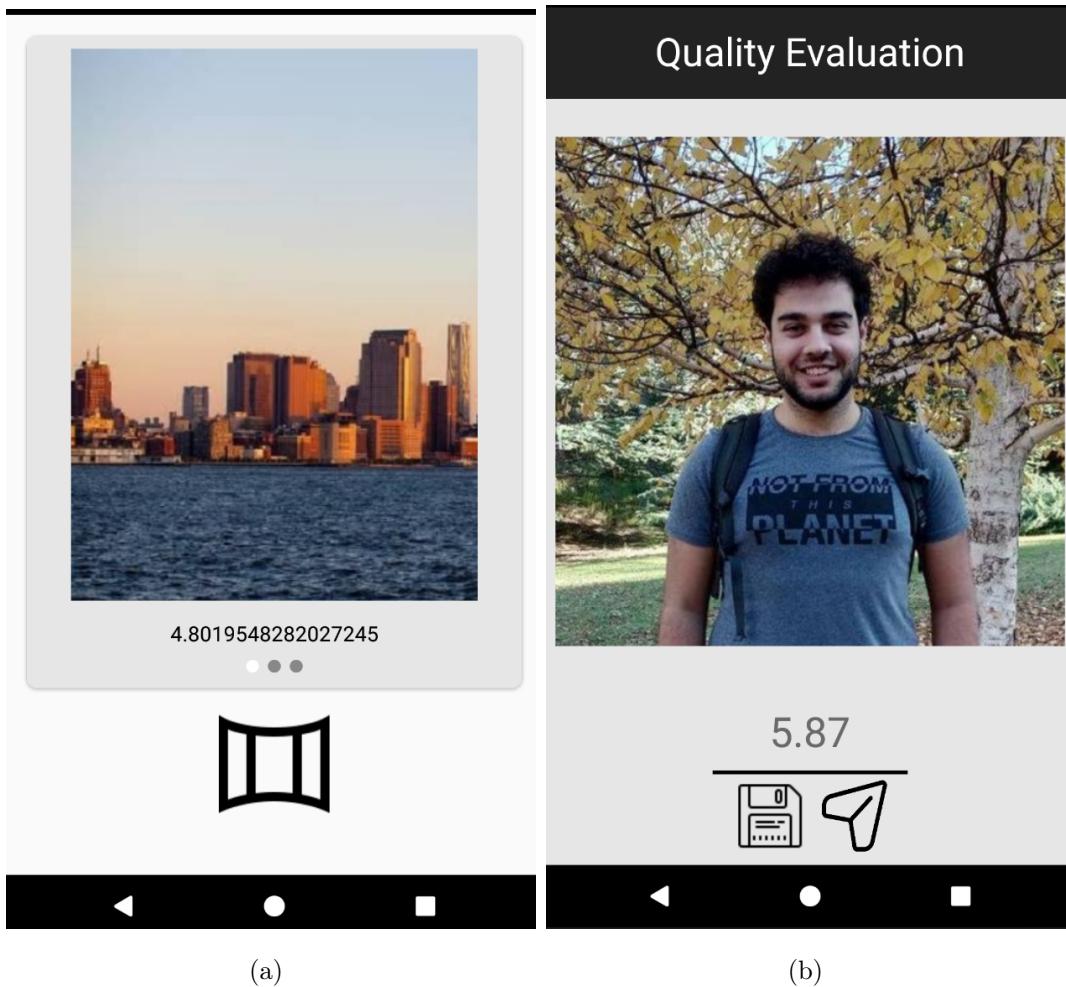


Figure 13: (a) Image Quality Evaluation with multiple photos (b) Image Quality Evaluation with a single photo

The pages related to the Image Quality Evaluation tool can be seen in this section. Figure 13a shows the tools usage with multiple photos and Figure 13b shows it's usage with a single photo.

If multiple photos are selected from pages explained in section 2 and 3 of this appendix, then the quality evaluation tool is automatically called and the photos are shown in a slideshow with their quality score written underneath each photo, as seen in Figure 13a.

If the image quality evaluation tool is selected from the tool selection page, explained in section 4 of this appendix, then the selected photo will be shown with the score written underneath it, as seen in Figure 13b. The user can save the image to their device's gallery by pressing the "Save" icon. If the user wants to continue editing with another tool, they

will be able to reach the tool selection page by pressing the paper plane icon at the bottom right corner of the page.

## B.11 Panorama

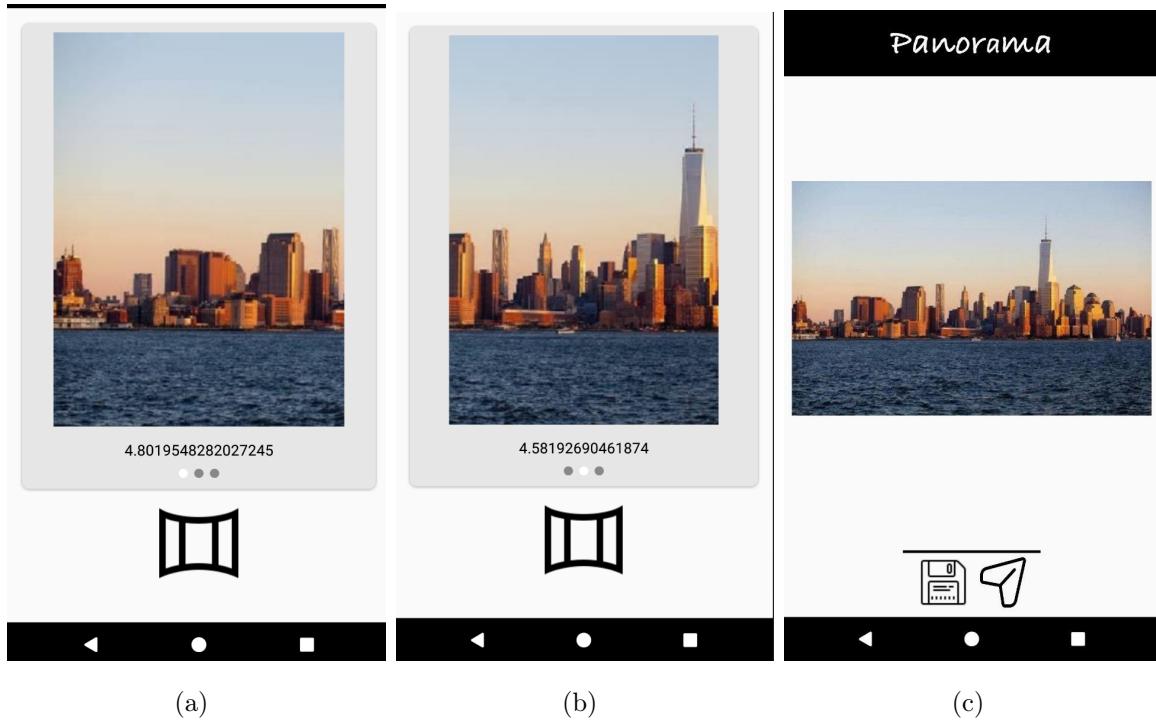


Figure 14: (a)-(b) Tool Selection Page for Multiple Photos(c) Panorama Page after the tool is applied

The pages related to the Panorama tool can be seen in this section. A sample usage of this tool can be seen in Figure 14, starting from a and finishing at c.

The page seen in Figure 14a and 14b is the tool selection page for multiple photos. At the moment, other than quality evaluation, Panorama is the only tool that can be used with multiple photos. The only icon present in this page is the icon for the Panorama tool.

Once the Panorama tool is selected, the tool is applied and the user is directed to the page seen in Figure 14c. This page shows the resulting panorama image. The user can save the image to their device's gallery by pressing the "Save" icon. If the user wants to continue editing with another tool, they will be able to reach the tool selection page by pressing the paper plane icon at the bottom right corner of the page.

## B.12 Face Expression Modifier

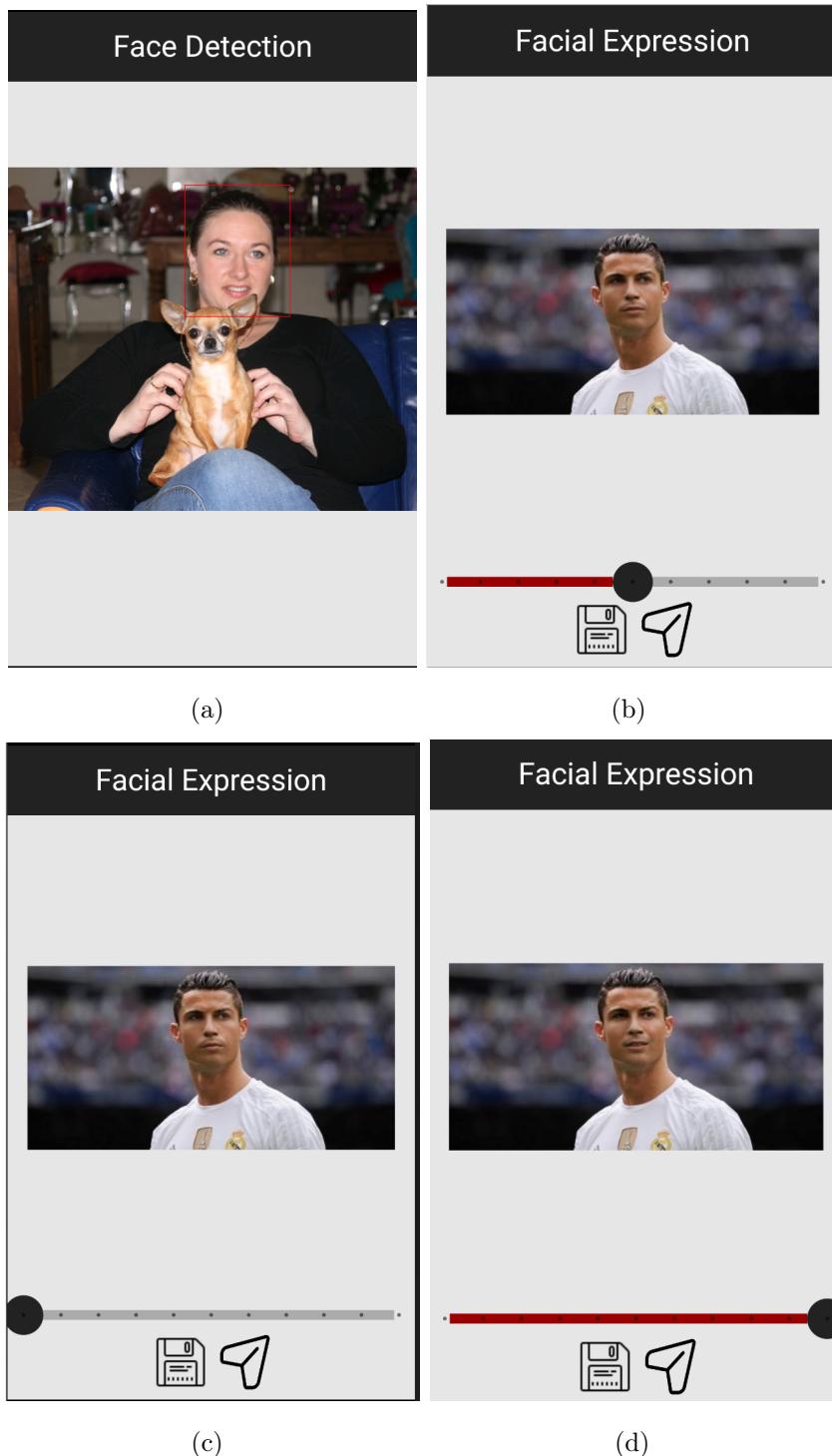


Figure 15: (a) Face Detection Page (b) Facial Expression Modifier Page (c) Facial Expression Modifier Page when slider is moved to left (d) Facial Expression Modifier Page when slider is moved to right

The pages related to the Face Expression Modifier tool can be seen in this section. After the user selects this tool from the tool selection page, explained in section 4, they will be directed to page seen in Figure 15a. In this page, the faces detected in the photo will be highlighted and the user will be able to select the face they want to modify by clicking on it. Once the face is selected, the tool is applied and the user is directed to the page seen in Figure 15b. In this page, the user can use the slider to change the facial expression of the face they have chosen. The resulting photo after the slider is moved to left and right can be seen in Figure 15c and 15d respectively.