



# Bilkent University

DEPARTMENT OF COMPUTER ENGINEERING

SENIOR DESIGN PROJECT

PHOTONOM

---

## Project Specifications Report

---

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

Project Specifications Report

Oct 14, 2019

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>
1.1	Description . . . . .	3
1.2	Constraints . . . . .	4
1.2.1	Implementation Constraints . . . . .	4
1.2.2	Dataset Constraints . . . . .	5
1.2.3	Economic Constraints . . . . .	5
1.2.4	Ethical Constraints . . . . .	6
1.2.5	Reliability Constraints . . . . .	6
1.3	Professional and Ethical Issues . . . . .	6
<b>2</b>	<b>Requirements</b>	<b>7</b>
2.1	Functional Requirements . . . . .	7
2.1.1	Taking Pictures and Uploading Photos . . . . .	7
2.1.2	Removing Objects from Background . . . . .	7
2.1.3	Rain Removal . . . . .	8
2.1.4	Theme Transfer . . . . .	8
2.1.5	Image Stitching . . . . .	8
2.1.6	Face Expression Modifier . . . . .	8
2.1.7	Image Quality Evaluation . . . . .	8
2.1.8	Saving and Sharing the Edited Photo . . . . .	9
2.1.9	Multi-language Support . . . . .	9
2.1.10	Head Pose Modifier . . . . .	9
2.2	Nonfunctional Requirements . . . . .	9
2.2.1	Accessibility . . . . .	9
2.2.2	Availability . . . . .	9
2.2.3	Sufficient Network Bandwidth . . . . .	10
2.2.4	Platforms and Portability . . . . .	10
2.2.5	Backup . . . . .	10
2.2.6	License . . . . .	11
2.2.7	Legal . . . . .	11
2.2.8	Scalability . . . . .	11
2.2.9	Security . . . . .	11
2.2.10	Testing . . . . .	11

2.2.11 Usability . . . . .	11
2.2.12 Performance . . . . .	12
2.2.13 Maintainability . . . . .	12

# 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 head position and/or 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, Photonom will be described in more detail and constraints related to it will be explained. Then both the functional and nonfunctional requirements of Photonom will be explained. The references can be seen at the end of this report.

## 1.1 Description

Photonom is a mobile application that will provide tools to modify photos interactively. The users can either take photos from the application or upload photos from their gallery. Once

the selection is done, one of the modifications that they can do is removing objects or people from their photo. A car that is in the background of a photo, for example, can be removed using this feature. The users will also be able to remove the effects of rain from their photos. Another tool that the user can use is referred to as image stitching. With this feature, the users can stitch different images, given that they have some common component, to each other. For example, if the photo is missing a certain part of the background, another photo of this background can be used to stitch these two images together. The user will also be able to modify their photos to fit a certain theme, such as a specific colour palette. In this feature, the user will first select the photo they want to modify, then they will upload another photo from their gallery that represent the theme they want to implement. Photonom will modify the former photo so that it has the theme of the latter. Photonom will also include a feature that will evaluate the quality of the selected image, and when given multiple photos will evaluate each of them and show the user the results. The user can then choose from this group of photos with the help of quality evaluation. Another feature of Photonom is related to facial expressions. By selecting their face, the user will be directed to an interface that will allow them to change their facial expression. Using Photonom, the users will also be able to save their progress and share the final product. If time permits, Photonom will also include a feature for head pose modification.

## **1.2 Constraints**

In this section, constraints of the project will be explained.

### **1.2.1 Implementation Constraints**

- For the maintenance of the code, GitHub will be used since it makes the collaboration easier.
- To track our sprints and balance the focus spent on the project stages, we aim to use Jira by Atlassian.
- In order to keep the topics separate to the channels and use the thread functionality to have more organized conversations Slack will be used.
- Open source libraries will be used during the implementation process.
- Images that are going to be used for the modification will be obtained from the phone camera or the gallery of the phone.

- Since it might be insufficient to use the phone for the necessary computations, instead of this a server will be used.
- This project will be created as a mobile application. In order to make this project cross-platform, Flutter or some other cross-platform mobile application framework will be used for the implementation of the project.
- Python will be used for the implementation of the neural networks.

### 1.2.2 Dataset Constraints

- To implement the image inpainting feature we will use four different datasets during training of the neural networks to be able to recover the erased parts after the removal of the unwanted objects:
  - **CelebA [3]:** Faces
  - **Paris [4]:** Buildings
  - **ImageNet [5]:** Objects
  - **Places2 [6]:** Natural scenes
- For the modification of facial expressions feature, we will use EmotioNet dataset [7] during the training.
- In order to be able to assess the quality of the photos, neural networks will be trained with following two datasets:
  - **AVA [8]:** To be able to consider aesthetic properties of the photograph
  - **TID2013 [9]:** To be able to consider technical details such as contrast, brightness etc.

### 1.2.3 Economic Constraints

- The usage of the open source libraries for the project is going to be free of charge.
- Currently, it is not planned to earn money from this project. So, users will be able to download and use the app without any fees or in app purchases.
- We are not also planning to use advertisements in the app as a source of income.

- In case of a need of an additional computational power, Google Cloud Platform or Amazon Web Services' free credits may be used. [10] [11]
- Publishing the app in Google Play Store may require additional costs.

#### **1.2.4 Ethical Constraints**

- Since the application requires users to take pictures using this application or an access to the gallery of the phone, it is important to ensure that these photos are safe. In order to guarantee this, we won't be sharing these photos with other applications or people without the necessary permissions of the users.
- In order to guarantee the safety of the users' data which is being kept at the server, in case of a security breach these operations will be aborted and reported to the user.
- During the use of the datasets and libraries their licenses will be taken into account and necessary procedures will be followed to get the permissions.
- In order to recommend better image stitching options it may also be necessary to use the GPS information of the user but it won't be used without the permission of the user.

#### **1.2.5 Reliability Constraints**

- In order to obtain more personal results during the inpainting of the faces after the removal of the objects, we will also consider the previous photos of the user as an input.
- After a shot, the application will demonstrate the multiple photograph options to the user but the photo that has the highest quality score according to the application will be showed as a first option. According to the choices of the user these recommendations will be updated accordingly.

### **1.3 Professional and Ethical Issues**

- Consent of the owner of the photo is required. However, since 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, we do not take any responsibility beyond this. Taking consent for any kind of photos to be uploaded are the user's responsibility.

- Since Photonom edits and changes original pictures, it is assumed that the user of Photonom accepted the reformation of pictures in advance.
- Pictures will not be stored in our servers after the user is done with editing. However, editing will be done in our servers, in the meantime, security for the user's data will be provided and will not be shared with third parties. However, the user will be able to share and 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 will be checked before usage.

## 2 Requirements

### 2.1 Functional Requirements

In this section, the functional requirements of our system will be explained.

#### 2.1.1 Taking Pictures and Uploading Photos

In order to use the other functionalities, the users first has to take a picture on the application or upload a photo from their phone's gallery. During the upload, the user will be able to crop the photo to adjust size. Once this step is done, the user will see the uploaded photo with the tools of the application on the screen.

#### 2.1.2 Removing Objects from Background

The user will be able to remove the items that s/he does not want from the photo. These items can be humans, animals, cars etc. Additionally, the said item should be in front of objects, people, buildings and natural scenes. The user will select the items that s/he wants to remove, and the application will remove these items without ruining the reality of the photo. In more detail, after the item to be removed is selected, the application will propose possible end results and the user will be able to choose from them. For example, if the user wants to remove a bag that is in front of the building, they will first select this bag. Then the application will show the edited photos, where the bag is removed and it's place is filled accordingly, the user will choose from these alternatives.



### **2.1.3 Rain Removal**

With this functionality, the user will be able to remove the water droplets which are on the camera lens because of the humidity or rain. Also, with this feature user can eliminate the negative effects of the rainy weather on the clarity of the image.

### **2.1.4 Theme Transfer**

Using Photonom, the user will be able to change the theme of their photo. The users will select a photo that represents the theme they want to apply. Then their own photo will be changed to fit this theme. For example there are people who really try to keep a consistent colour palette in their photographs. Instead of trying so hard to do so, they can use this feature by giving a sample photograph and the photograph they want to change. The resulting photograph will have a similar palette with the given samples.

### **2.1.5 Image Stitching**

In case the user's photograph is missing some parts, they will be able to use Photonom to stitch other photos that contain the missing parts of the original photo. The missing part of a photo could be the top of a church or tower or a section of a sculpture. The photos containing the missing parts will either be provided by the user from their gallery or will be found from the web. If these photos will be found from the web, the application will find them either by looking at the location of the photo if the user allowed the application to access the location data, or using object recognition to recognise famous places. Once all images are stitched, the user can crop the resulting photo to get the desired frame.

### **2.1.6 Face Expression Modifier**

Using this feature, the users will be able to change their facial expression. There will be some expression styles such as happiness, anger etc. The user will be able to select among them and with the slider comes out, s/he will change the face expression gradually.

### **2.1.7 Image Quality Evaluation**

The quality of photos will be evaluated by the application according to both aesthetics and technical details. While evaluating the quality according to aesthetics, the conformity of the photograph to photographic rules and practices such as golden ratio, the harmony of the colors will be taken into account. [8] For the evaluation of technical quality of the photo,

multiple criteria such as resolution of the photo, contrast, and brightness will be considered. The user will select the photos that s/he wanted to compare and the photo with the best score will be showed. Also, it will be possible to see other photos with their own scores.

### **2.1.8 Saving and Sharing the Edited Photo**

If their editing is not completed, the user will be able to save it as a draft to a cloud storage service or directly to the application storage. If the photo is completed, the user will be able to share it. These photos will either be saved to the phone's gallery or shared via e-mail or WhatsApp, if these are available on the phone.

### **2.1.9 Multi-language Support**

This requirement provides the opportunity to choose a language among several languages. The user will be able to access language support inside the application.

### **2.1.10 Head Pose Modifier**

This tool will be added if time permits. If the user does not like the stance of their head, s/he will be able to change their head pose. Photonom will only be able to change the head pose slightly (up to 10%) in 4 directions (North, South, West, East).

## **2.2 Nonfunctional Requirements**

In this section the non-functional requirements of our system will be explained briefly.

### **2.2.1 Accessibility**

Photonom aims to be used by users all around the world. Hence we will provide many language options for our users worldwide and English as the default option.

### **2.2.2 Availability**

Our mobile application will provide many functionality and its aimed to have a modular easily-pluggable design. However any related updates should not affect the availability of the system. While making new changes to our models in the server, we will never let any modules to be down. In order to provide high availability for our users but not spend too much monetary resources we will 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**

Our system performs computationally heavy operations that most mobile devices will not be able to execute. Moreover the transfer of the image could be problematic as well due to new mobile phone cameras have increased their capability to take very high quality images. Hence we require around 10 Mbps (Megabits per second) download and upload speed. As we progress into a new era of 5G, these bandwidth requirements are simply minimal.

### **2.2.4 Platforms and Portability**

The system will persist on two different platforms: Server and the mobile applications. Since we are trying to produce a minimum viable product (MVP) at the end, we don't want to spend extra effort on our mobile application for different operating systems such as iOS, Android, or Windows Phone. We'll use a cross-platform framework such as Flutter by Google so that the apps are easily ported and our application is available on most of the mobile phones.

### **2.2.5 Backup**

The system will perform a set of operations on their photographs, if the user gives permission, we can also save the photo to their preferred cloud storage service such as Google Photos at different stages of the performed operations in case the user later wants to access one of the others.

### **2.2.6 License**

Any dataset, product, library used in the product should be referenced properly. Moreover they need to be paid if necessary in the case of commercial usage.

### **2.2.7 Legal**

The system will perform on the users' private photographs. Hence, 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 [12]. 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.

### **2.2.8 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 [13]. This way the system can scale itself up in case of high loads automatically by the use of cloud services such as AWS, GCP [11] [10].

### **2.2.9 Security**

The personal data of the users will persist on the server during the operations. Hence, our server should be secure and should not permit any intruders for the sake of the privacy of the users' data. In such a case, all operations will be cancelled directly and the corresponding container will be quarantined.

### **2.2.10 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. There will also be a circle CI system that includes a coverage test and will not let the new changes to be applied to master.

### **2.2.11 Usability**

The system will provide many functionalities and since the users will use it as a mobile app, the app should have a natural learning curve which may be supported with welcome tutorials

if seen necessary.

### **2.2.12 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.

### **2.2.13 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.

## 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] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [4] 2019. [Online]. Available: <https://github.com/pathak22/context-encoder6-paris-street-view-dataset>
- [5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.
- [6] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [7] [Online]. Available: <http://cbcs1.ece.ohio-state.edu/cvpr16.pdf>
- [8] [Online]. Available: <https://github.com/ylogx/aesthetics/tree/master/data/ava>
- [9] [Online]. Available: <http://www.ponomarenko.info/tid2013.htm>
- [10] 2019. [Online]. Available: <https://cloud.google.com/>
- [11] 2019. [Online]. Available: <https://aws.amazon.com/>
- [12] 2019. [Online]. Available: <https://gdpr.eu/>
- [13] 2019. [Online]. Available: <https://www.docker.com/>