Software Requirement Specification Document for Project LoGo: Logo Generation Through Generative Adversarial Networks

Alyeldin Khairy, Fady Tarek, Hady Hany, Mohamed Yasser, Youssef Abdelwahab
Supervised by: Dr. Hossam Abdelrahaman, Eng. Noha Elmasry

December 17, 2022

Table 1: Document version history

Version	Date	Reason for Change
1.0	25-Nov-2022	SRS First version's specifications are defined.
1.1	15-Dec-2022	Add system overview and context diagrams

GitHub: `https://github.com/Alyeldin/Lo-Go_Logo-on-the-go`.

Contents

Abstract

Many startups fail to advertise their services due to the lack of knowledge of the importance of a logo. A logo is a business' face as it is what grabs the attention and stays in the memory of the consumers right away. The main aspects of a logo are that it must be memorable, unique, and most importantly carry meaningful insights regarding the business services. Artificial intelligence can elevate the productivity and overall effectiveness of the logo designing process by automating a substantial amount of the tasks needed to design it. In this project, we will be focusing on designing logos using neural networks by not only focusing on the business services but also the characteristics of the target audience of the user, such as age, job, social standard, and gender. Unlike other projects, this will generate more personalized logos due to the multiple classifications used to render the logos.

1 Introduction

1.1 Purpose of this document

This document's goal is to evaluate and deconstruct the software specifications for developing a web-based project management tool for software engineering students. It will outline the functional specifications as well as the anticipated conduct in a certain setting. All the tools and techniques utilised to complete this project will be examined in this document. All parties involved in this project, including students, teachers, and any developers who might work on it, will be the intended audience for this document.

1.2 Scope of this document

This document's purpose is to explain the framework of our project so that stakeholders and developers can both comprehend the solution we are offering. Additionally, the database and inheritance connections in this paper are still subject to change.

1.3 Business Context

The Beneficiaries of this solution will be for everyone needs logo for his own use. The users running this are startups who needs a professional designed and unique logos with also low price. This web application will increase the logo creation for small startups and any other people who wants to create logos for their own needs. This application will not be customized for certain people only or professionals but it will be available for all ages and nonprofessional people.

2 Similar System

2.1 Academic

- This paper [1] focuses on creating a logo renderer using deep learning to combat the ever increasing demand for professionally designed logos in various fields and industries. The author proposes using a convolutional neural network to renderer the logo. After the logo is

rendered, it is then separated into different elements and each element is rated and changed to fit certain parameters set by the user such as color scheme, shape, size, and symmetry. Using this method the author is able to render the logos with the input "garden" and name "botanysad.ru", they also able to render any logo regardless of size. The author fails to mention several key factors in his paper such as the specific architecture of the neural network used and the dataset used in training said network. In addition, the logos rendered lack any complexity this is likely due to the data-set used while training the network.

- This paper [2] aims to tackle the problem with the long and arduous process of designing a logo. Due to logos' multimodal nature, it is extremely difficult to develop a working neural network that can generate logos. The author proposes using Generative Adversarial Networks (GANs) combined with clustering the dataset into separate clusters to stabilize the GAN's training. The author used a dataset that consists of over 600,000 logo images called Large Logo Dataset (LLD) divided into 486,377 32x32 pixel images and 122,920 400x400 pixel images called LLD-icon: Favicons and LLD-logo: Twitter respectively for training, and the CIFAR-10 dataset that consists of 60,000 32x32 pixel images for testing. The author achieved the best result of 8.67 inception score using an improved Wasserstein Generative Adversarial Network developed by Gulrajani et al [3] with an Auxiliary Classifier and 32 ResNet-50 Classifier. The logos created by this model lack any text element and are to create a specific design due to the lack of input.

- This paper [4] aims to reduce the workload on designers and help increase their productivity through a logo generator that uses a wavelets-enhanced adversarial learning model. The author proposes using wavelet transform to decompose the input logo images into four frequency bands, Low Low(LL), Low High(LH), High Low(HL), High High(HH) to expand on the network developed by Sage et al. The LL band will preserve the primary content of the image which will then be labeled and clustered using a ResNet-50 classifier, and the LH, HL, HH bands are used to extract the image's vertical, horizontal, and diagonals edges respectively which are then fed to the network. The author uses the LLD-icon has 548, 210 standardized 32×32 favicons and a LLD-logo contains 122, 920 high-resolution logos (most in 400×400-pixels) crawled from twitter constructed by Sage et al. The author the highest inception score achieved was 5.40 on the LLD-icon data-set.
  The model developed by this author, while is more refined, still suffer the same problem as the previous author in lacking any real indication as to what theme or business they represent.

- This paper [5] aims expand on Sage's et al work in the field of logo generation and aid in reducing burden designer in the process of creating a logo. Contrary to the previous study, researchers here focused on dividing the logos by colors rather than labels. The network called LoGAN consists of an improved auxiliary classifier Wasserstein generative adversarial neural network that is able to generate logos conditioned on twelve different colors. The data-set used to train LoGAN is the LLD-icons data-set, which consists of 486,377 32×32 icons. The author evaluates his model based on how the logo was labeled in relation to its real color. Black achieved the highest F1-Score with 0.90. This model might benefit from adding more conditions to labels other than colors such as shapes, as the author mentioned, or names. This model suffers when dealing with a multi-colored logos or colors that are

similar in hue and often miss labels them as observed by pink achieving the lowest recall at 0.3.

- This paper [6] targets the multimodal nature of logos and aim to propose a solution for the challenge this cause to GANs. The author proposes using the StyleGAN architecture developed by NVIDIA as a base for his network enhanced with a conditional layer to gain more control over the output of the network. The data-set used in training is a boosted version of the Large Logo Data-set (LLD). The boosted data-set consists of over 120,000 logo, however after removing all text centric logos the author was left with circa 40,000 logos. To increase the size of the data-set the author scraped Google for new logo-like images. Keywords pertaining to topics such as nature, technology and illustrated characters were used in what resulted in circa 15 000 additional images. The author achieved the best result in labeling the images using a ResNet embedding layer to cluster images with similar visual characteristics together. Using this method, the author was able to generate these logos and achieved a FID score of 101.9211. This model might produce superior results had the data-set been of already existing logos in various fields as it might lead to more homogeneity in each class and create more memorable logos.

- In this paper [7],The author addresses the difficulty of generating Text To Image problem ,by proposing a new-framework called Mirror-GAN. Mirror-GAN successfully applies the concept of learning text-to-image generation through redescription. STEM generates embeddings at the word and sentence levels. GLAM uses a cascaded architecture to generate target images at various scales, taking advantages of both local word attention and global sentence attention to improve the diversity and semantic consistency of the generated images. STREAM also monitors the generators by regenerating the text description from the generated image, which is semantically aligned with the given text description. On two benchmark data sets, they demonstrate that Mirror-GAN achieves new state-of-the-art performance.

- [8] The layout of generative models has a place for generative adversarial networks. It suggests that they are able to produce new models. Pixels are used to transform text into images. "Flower with pink petals," for instance. GAN is made up of a combination of two competing neural network models that track, capture, and replicate the variations present in a dataset. Transforming written descriptions into the right images is the main goal of text to image synthesis. GAN models are frequently utilised today for better outcomes. Another issue with deep learning is that there are numerous potential arrangements for a single text description, however this issue can be solved by training the model.

- In this paper [9] Medical imaging methods like MRI and CT can include missing or distorted areas, which is of particular importance to automated image analysis frameworks to improve post-processing tasks like segmentation or classification. In this paper, we introduce IPA-MedGAN, a brand-new framework for the integration of medical modalities. In order to build the proposed IPA-MedGAN, two discriminator networks that were jointly trained as a conditional GAN (cGAN) and a new cascaded generator network that is based on MultiRes-UNets are used. To calculate an additional non-adversarial loss, a feature extractor network that has already been trained is used. The proposed IPA-MedGan framework is assessed

using MR data from a brain region. The results of the IPA-MedGan was that it completed the missing parts of the trained image. One of the problems researchers must make sure of it that completing the missing part of MRI image must not have any error as this may cause wrong diagnosis.

- In this paper [10] Yuchuan Zheng the author of research article titled" Visual Memory Neural Network for Artistic Graphic Design" proposed to create a new model that makes computers have the ability to recognize and differentiate between graphic designs and graphic artistic design with help of labels and free combination of graphical solutions, he mentioned that their main problem was artistic graphic region segmentation and he thought it could be solved by using self-attentive mechanism, he generated a data-set with the help of professional art aestheticians to evaluate different art graphics and label them, as a results the experiment achieved a higher accuracy than compared algorithms.

- In this paper [11] Stochastic variation, Think about the implementation of a conventional generator. Stochastic change. Given that the input layer is the sole way for the network to receive input, the network must develop a technique for producing spatially variable pseudo random numbers .Separation of global effects from stochasticity, because whole feature maps are scaled and biased with the same values, the style in our style-based generator influences the entire image. Perceptual path length this indicates that the components of variation are not effectively segregated and that the latent space is intertwined. The FFHQ data-set, they have created a brand-new set of human face data called Flicker Faces-HQ (FFHQ), which consists of 70,000 high-resolution pictures the data-set provides significantly greater coverage of accessories.

2.2 Business Applications

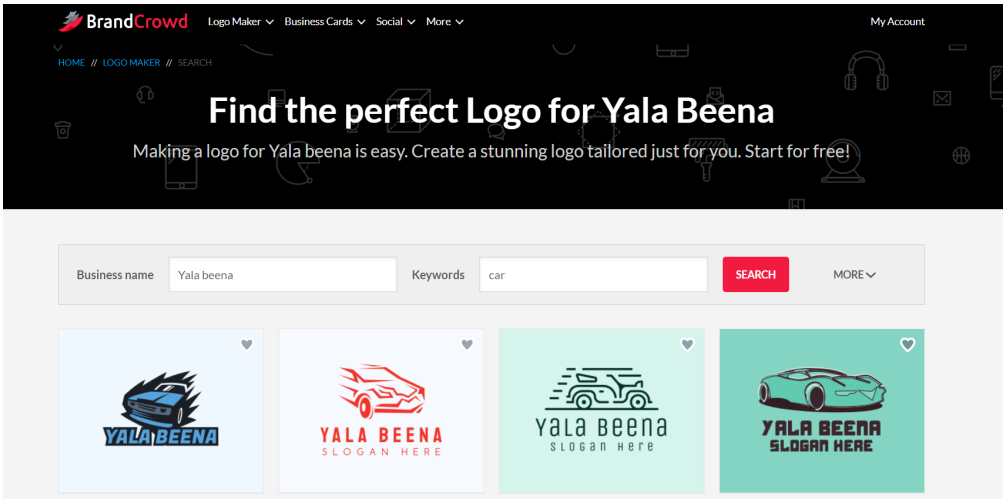

Figure 1: Brand Crowd

- Brand Crowd is a website that offer logo generation services for the user. It take the business's name and the user can input keywords to affect the overall design of the logo. As seen

in the figure below we created a logo using the name "Yala Beena" and the keyword "car".

3 System Description

3.1 Problem Statement

Logo nowadays have been one of the most important thing in any product, brand or companies. Logo designers spent allot of time in creating a logo and may not reach the target audience of the client and this also may cost the client allot of money till he reach the best logo design. This what our project is going to give a solution in the logo generating field as it will ask the user to give some information about his target audience and some of the logo style and this inputs will generate 2 logos to choose the best of them in a few minutes.

3.2 System Overview

This web application will be a professional logo generator for every nonprofessional designer. The app will offer the users professional and unique logo designs to help them reach their all needs of a logo with just some inputs about their logo specs.

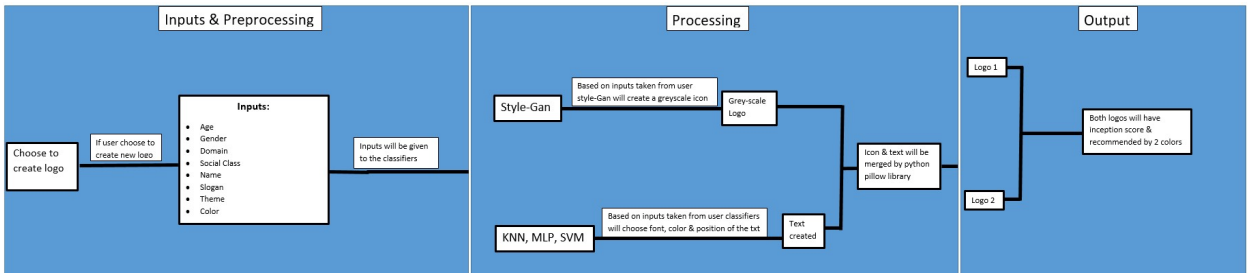

Figure 2: System Overview

3.3 System Scope

- Logo designers spend a lot of time to create a logo
- Logo designers asks for a large amount of money
- Logo generator is seen as a solution to tackle one of startup's most common problems

3.4 System Context

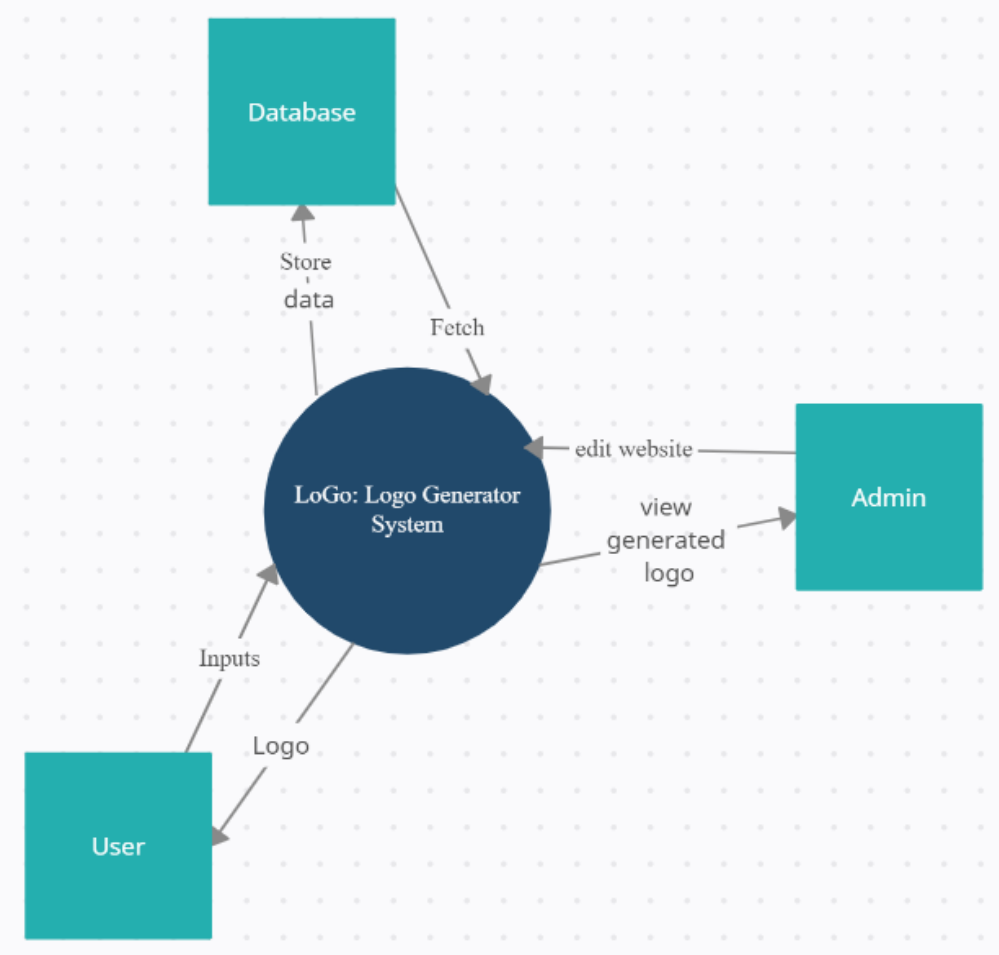

Figure 3: Context Diagram

3.5 Objectives

- The system shall be easy to use by non professional designers
- The system shall generate the logos in a few minutes
- User can be able to generate more than one logo

3.6 User Characteristics

Expected user are the startups who want to not pay a lot of money and have their logo in short time. And any other user who already have his own logo but just needs some upgrades or change in logo style. The user should not have any basics about logo design he just need to fill the questions about his target audience and logo style and system will complete for him everything till the logo be generated for him.

4 Functional Requirements

4.1 Detailed Functional Specification

4.1.1 Admin

- The admin shall be able to view the feedback sent by the users.
- The admin shall be able to view client's registration and accounts.

4.1.2 User

- The user shall be able to view the website.
- the user shall be able to register to the system
- the user shall be able to login to the system
- The user shall be able to generate a logo based on several inputs: age, gender, domain, color, theme, social class.
- The user shall be able to buy the generated logo if they want to ensure that it will not be generated again
- The user shall be able to view the website.
- The user shall be able to choose between generating mono-color logos or multi-color logos.
- The user shall be able to pay a fee to get feedback from a professional designer on their logo and how to upgrade it.
- The user shall be able to rate their experience using the website.
- The user shall recommend a primary color and a secondary color for each logo.

4.1.3 System

- the system shall generate a minimum of 2 logos per user input.
- the system shall generate unique logos even if the input is the same.
- The system shall rate each generated logo using inception score.

4.1.4 System Use-case Diagram

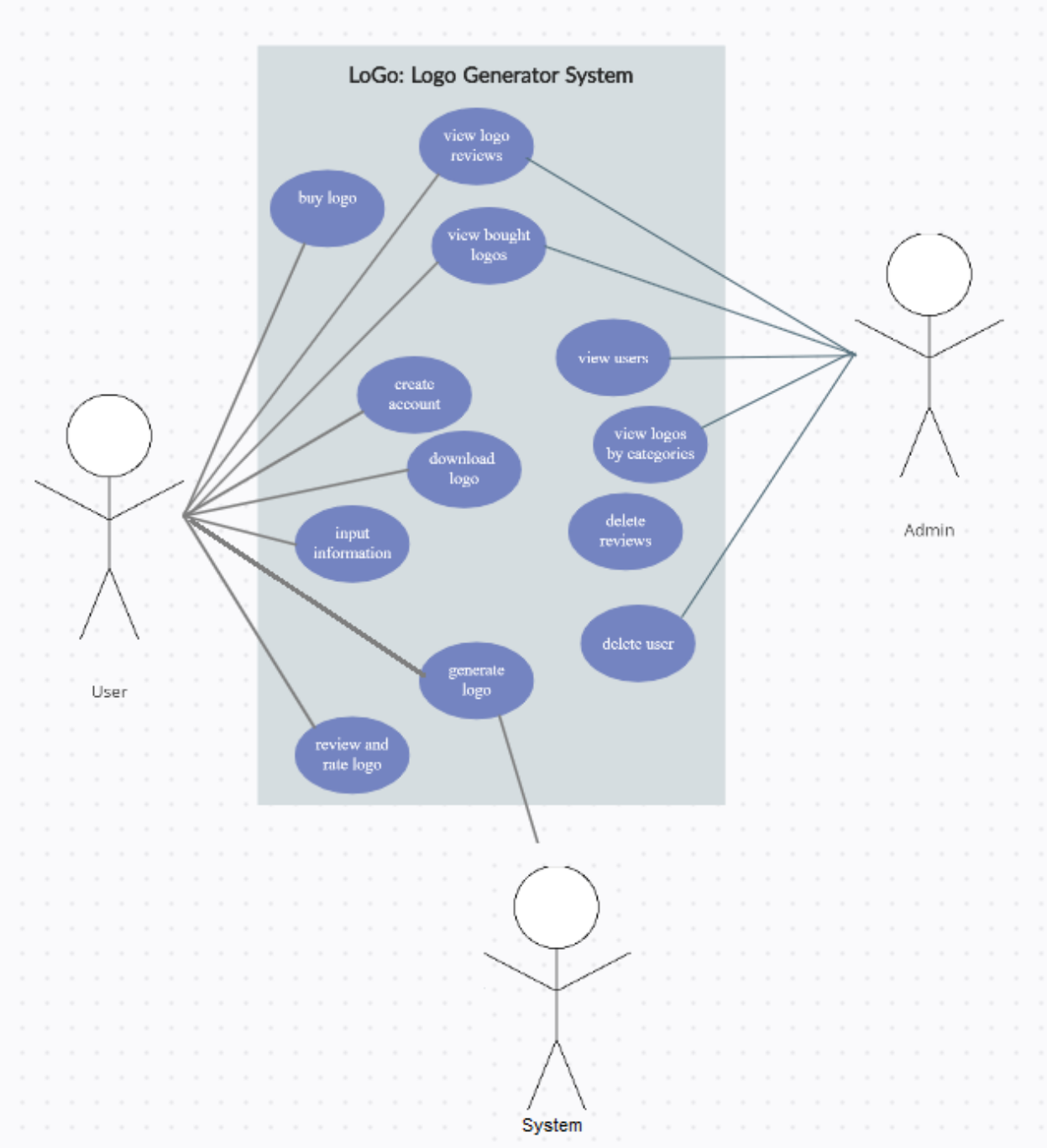

Figure 4: Use-Case Diagram

4.2 System Functions

Name	logIn
Code	US01
Priority	Extreme
Critical	highly essential for user authentication
Description	takes the email and password and searches for them in the db until it finally finds a match
Input	email,password
Output	Boolean(true for success)
Pre-condition	User needs to have an existing account
Post-condition	if successful start a user session if its not output email or password doesn't match
Dependency	- - - - - - - - - - -
Risk	None

Name	User Registeration
Code	US02
Priority	High
Critical	- - - - - - - - - - -
Description	User can sign up in our system with his information
Input	Sign up form including username, password, email, name, address, phone number.
Output	Message to the user 'Successfully registered to the system'
Pre-condition	User's email is not registered in our database.
Post-condition	Back to homepage and can generate logos now.
Dependency	- - - - - - - - - - -
Risk	Risk Email could be registered before.

Name	Create or update logo
Code	US03
Priority	High
Critical	Critical because inputs may change requirements if user choose to update logo as he have to give a photo of old logo while entering inputs
Description	User must choose to between creating a new logo or not as the inputs will change if user choose to update old logo and some system processing as creating an icon will change
Input	choose between create logo or update logo
Output	- - - - - - - - -
Pre-condition	User is log-ed in
Post-condition	User will move to the inputs form to fill his logo details
Dependency	- - - - - - - - - - -
Risk	If user doesn't choose will not pass to the inputs form

Name	User's inputs
Code	US04
Priority	Extreme
Critical	Critical because inputs will change everything in logo processing
Description	User will enter Age/Gender/Domain/Social Class/Name/Slogan/Theme/Color/Logo Style/Logo shape/Choose mono or multi -color
Input	Fill the inputs form
Output	Inputs will starts processing
Pre-condition	User choose whether create new logo or update old logo
Post-condition	Inputs will pass to the system classifiers
Dependency	US03 Create or update logo
Risk	If user doesn't fill all input's text boxes exception handling will appear to fill all the form

Name	Inputs divide between classifiers
Code	SY01
Priority	High
Critical	Important for system to give all classifiers their inputs
Description	Style-gan will take user inputs to create the logo's icon and KNN,SVM,MLP will choose the font,color,position of logo's text
Input	User inputs
Output	
Pre-condition	System should check that all inputs are taken from the user
Post-condition	inputs will be processed to start processing the logo parts
Dependency	US04 User's inputs
Risk	System must check that inputs are divided correctly to prevent errors in processing

Name	Style-gan generating icon
Code	SY02
Priority	Extreme
Critical	Style-gan will check all inputs and try to generate the most matching icon to user's inputs
Description	Style-gan will generate an icon based on the user's inputs
Input	User input from function US04
Output	An icon
Pre-condition	Inputs taken from the user and divided between classifiers
Post-condition	Icon will be colored by style-gan
Dependency	User input in function US04
Risk	Generated icon doesn't match user the inputs

Name	KNN,SVM,MLP creating text
Code	SY03
Priority	Extreme
Critical	KNN,SVM,MLP doesn't create best text based on the inputs
Description	KNN,SVM,MLP will take user inputs to choose the font,color,position of the logo's text
Input	User input from function US04
Output	Text with it's font,color,position
Pre-condition	Inputs taken from the user and divided between classifiers
Post-condition	Text will be added to the logo icon
Dependency	User input in function US04
Risk	Generated text doesn't match user the inputs

Name	Collecting logo parts from classifiers
Code	SY04
Priority	High
Critical	All inputs should not change or missed
Description	Classifiers outputs will be collected by python pillow library
Input	Style-gan icon and Classifiers text
Output	
Pre-condition	Style-gan and classifiers generate icon and text of the logo
Post-condition	python pillow will merge these logo parts
Dependency	SY02,SY03
Risk	

Name	Merge logo parts from classifiers
Code	SY05
Priority	High
Critical	All inputs should not change or missed
Description	Logo icon and text generated will be merged together and text will will have the position chosen by classifiers
Input	Style-gan icon and Classifiers text
Output	Logo icon and text as one part
Pre-condition	Style-gan and classifiers generate icon and text of the logo
Post-condition	Logo samples will be generated
Dependency	SY02,SY03
Risk	

Name	2 logos generated as samples
Code	O(01)
Priority	low
Critical	2 logos should have same outline but a little difference in appearance
Description	After system finishes processing and generating logos 2 samples will be given to user to choose between them
Input	
Output	2 logos
Pre-condition	processing and generating logos
Post-condition	user choose best logo
Dependency	SY05
Risk	Logos doesn't reach user needs

5 Design Constraints

5.1 Hardware Limitations

The model must be trained on a PC with an NVIDIA GPU and high processing power in order to be able to handle the heavy computations needed to train it.

5.2 Other Constraints as appropriate

The model must be trained using Python version 3.7, Tensorflow version 1.14, and CUDA 10.0. The application shall require a server to be hosted on.

6 Non-functional Requirements

6.1 Maintainability

The program is built in such a manner that it collects information of data to produce a logo. Additionally, the system's documentation is comprehensive and concise. The system is easy maintenance and adaptable for future modifications or enhancements.

6.2 Usability

The logo generator will provide consumers with a fantastic user experience while saving them time. The majority of users won't require any training and won't need to spend much time interacting with and using the logo generator's features and services.

6.3 security

The system must be able to protect information such as user entered data and the owner's login details. Strong passwords can be made for logins, security questions may be made to confirm

users, and firewalls can be used to block unauthorized access to user data.

6.4 Reliability

Given the importance of this system to the users, it should offer the most distinctive logo design possible to produce a logo that is appropriate and well-designed for the information that the user entered.

7 Data Design

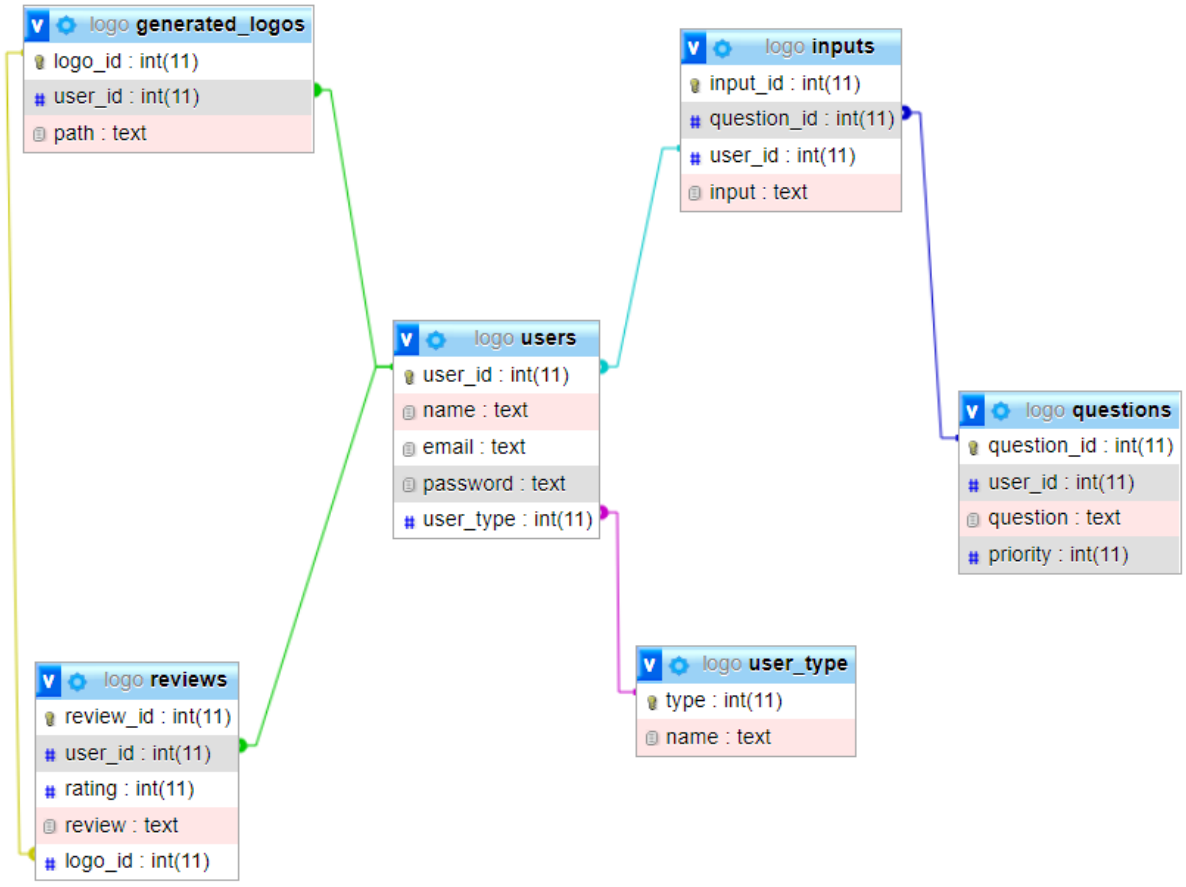

Figure 5: Database Design

8 Preliminary Object-Oriented Domain Analysis

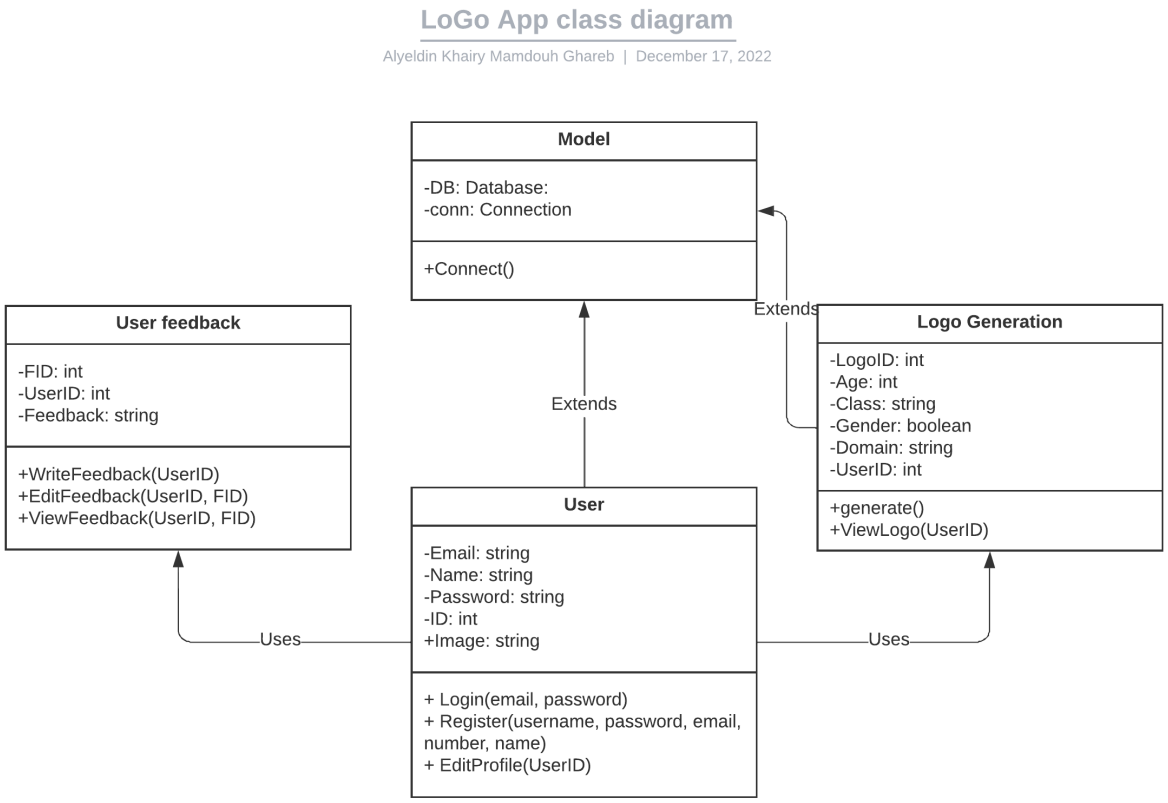

Figure 6: System Class Diagram

9 Operational Scenarios

- The user will open the website or the webapp The app will ask him to enter some inputs to give a unique generated logo the more details given by the user the better the logo and it will deliver the user needs for the logo
- The app will recommend some logos for the user to choose from based on his inputs.
- After he downloads one of the recommended it will be saved in our database to him and won't generate again for any other users

10 Project Plan

Task No.	Task description	Task date	Task due date	Duty of	Done by members
1	Idea Approval by Dr. Hossam	21 Aug 2022	25 Aug 2022	All team members	☑
2	Project Title	24 Sep 2022	25 Sep 2022	All team members	☑
3	Project Abstract	24 Sep 2022	26 Sep 2022	All team members	☑
4	Related papers about the project	27 Sep 2022	1 Oct 2022	All team members	☑
5	Summary about related papers found	27 Sep 2022	1 Oct 2022	All team members	☑
6	Related Dataset for input	27 Sep 2022	1 Oct 2022	Fady, Mohamed	☑
7	Overleaf creation and proposal project	27 Sep 2022	1 Oct 2022	Fady, Aly	☑
8	Github repository creation	27 Sep 2022	1 Oct 2022	Hady	☑
9	Search & study IPMedGAN algorithm	5 Oct 2022	8 Oct 2022	Fady, Mohamed	☑
10	Study GAN Algorithm and know their inputs & outputs	5 Oct 2022	8 Oct 2022	Youssef, Hady	☑
11	Compare any results found between StyleGAN & IPMedGAN	5 Oct 2022	8 Oct 2022	Fady, Aly	☑
12	Comparison between StyleGAN & IPMedGAN	5 Oct 2022	8 Oct 2022	Aly, Mohamed	☑
13	System overview for the project	8 Oct 2022	11 Oct 2022	Fady	☑
14	System overview for the project	8 Oct 2022	11 Oct 2022	Mohamed	☑
15	Study project inputs and outputs and how to run algorithm	8 Oct 2022	11 Oct 2022	Aly, Hady	☑
16	Start filling proposal document with the gathered information	8 Oct 2022	11 Oct 2022	Hady, Youssef	☑
17	Comparison between styleGAN and MirorrGAN	13 Oct 2022	15 Oct 2022	Mohamed	☑
18	System overview diagram need more details and information of processes	13 Oct 2022	15 Oct 2022	Fady	☑
19	Comparsion between all recommended algorithms (KNN, SVM, MLP)	13 Oct 2022	15 Oct 2022	Hady	☑
20	Overleaf proposal document writing	13 Oct 2022	15 Oct 2022	Fady	☑
21	Run code of styleGAN	13 Oct 2022	18 Oct 2022	Fady, Mohamed	☑
22	find datasets for all inputs	13 Oct 2022	16 Oct 2022	Youssef	☑
23	related work for proposal	13 Oct 2022	16 Oct 2022	Aly	☑
24	Draft for system overview in the proposal document	13 Oct 2022	16 Oct 2022	Youssef, Hady	☑
25	Proposel section 1	18 Oct 2022	19 Oct 2022	Hady , Fady	☑
26	Proposel section 2	18 Oct 2022	19 Oct 2022	Fady, Aly	☑
27	Proposel section 3	18 Oct 2022	19 Oct 2022	Youssef	☑
28	Proposel section 4	18 Oct 2022	19 Oct 2022	Hady	☑
29	Proposel section 5	18 Oct 2022	20 Oct 2022	Fady , Mohamed	☑
30	Proposel section 6	18 Oct 2022	19 Oct 2022	Aly, Hady	☑
31	Proposel section 7	18 Oct 2022	20 Oct 2022	Fady	☑
32	Create a survey to prove a concept	18 Oct 2022	19 Oct 2022	Aly	☑
33	Proposal document (Add more related work)	20 Oct 2022	21 Oct 2022	Mohamed, Hady	☑
34	Datasets for proposal	21 Oct 2022	21 Oct 2022	Fady, Mohamed	☑
35	Update for the system overview with full details regarding, Input/Output and processing	16 Nov 2022	25 Nov 2022	Hady Hany	☑

Figure 7: Task Plan 1

35	Update for the system overview with full details regarding, Input/Output and processing	16 Nov 2022	25 Nov 2022	Hady Hany	☑
36	Describe an evaluation system for the ouput of the system	16 Nov 2022	25 Nov 2022	Mohamed	☑
37	Describe the differentiation process of the logos generated by the system	16 Nov 2022	25 Nov 2022	Aly	☑
38	Full details about the dataset needed	16 Nov 2022	25 Nov 2022	Fady	☑
39	Full details about the nature of the input	16 Nov 2022	25 Nov 2022	Hady Hany	☑
40	Decide on the different classifiers algorithms that will be used in the system	16 Nov 2022	25 Nov 2022	Aly, Youssef	☑
41	List the functional requirments for the SRS document	16 Nov 2022	25 Nov 2022	All team members	☑
42	Requesting a dataset from colorcom.com	25 Nov 2022	25 Nov 2022	Fady	☑
43	creating a survey to enchance the dataset	25 Nov 2022	25 Nov 2022	Aly	☑
44	creating a survey to get feedback on the system from graphic designers	25 Nov 2022	25 Nov 2022	Hady	☑
45	Conditional StyleGAN preprocessing	25 Nov 2022	30 Nov 2022	Yasser, Fady	☑
46	creating dummy dataset	30 Nov 2022	1 Dec 2022	Aly, Youssef	☑
47	dummy data preprocessing	30 Nov 2022	1 Dec 2022	Youssef, Yasser	☑
48	trainning a KNN classifier on the dummy dataset	1 Dec 2022	3 Dec 2022	Hady,Yasser	☑
49	testing a pretrained model	3 Dec 2022	7 Dec 2022	Fady, Yasser	☑
50	training a prototype model	7 Dec 2022	14 Dec 2022	Fady, Yasser, Aly	☑
51	creating the text component using the classifers output	8 Dec 2022	9 Dec 2022	Fady	☑
52	merge the text compnent with the icon compotnent from the styleGAN to create a logo	8 Dec 2022	10 Dec 2022	Youssef, Hady	☑
53	recolor the logo with the recomended color	8 Dec 2022	12 Dec 2022	Aly	☑
54	SRS section 1	8 Dec 2022	15 Dec 2022	Mohamed Yasser, Youssef Mohamed	☑
55	SRS section 2	8 Dec 2022	15 Dec 2022	Aly	☑
56	SRS section 3	8 Dec 2022	15 Dec 2022	Hady Hany	☑
57	SRS section 4	8 Dec 2022	15 Dec 2022	Fady Tarek,Hady Hany	☑
58	SRS section 5	8 Dec 2022	15 Dec 2022	Fady Tarek	☑
59	SRS section 6	8 Dec 2022	15 Dec 2022	Youssef Mohamed	☑
60	SRS section 7	8 Dec 2022	15 Dec 2022	Fady Tarek	☑
61	SRS section 8	8 Dec 2022	15 Dec 2022	Aly	☑
62	SRS section 9	8 Dec 2022	15 Dec 2022	Mohamed Yasser	☑
63	SRS section 10	8 Dec 2022	15 Dec 2022	All team members	☑
64	SRS section 11	8 Dec 2022	15 Dec 2022	Fady Tarek,Aly	☑
65	Techanical evaluation system for the project				☐
66	Market evaluation system for the project				☐
67	Logo simelarity checker				☐
68	training a complete conditional StyleGAN model with multiple inputs				☐
69	SDD document				☐

Figure 8: Task Plan 2

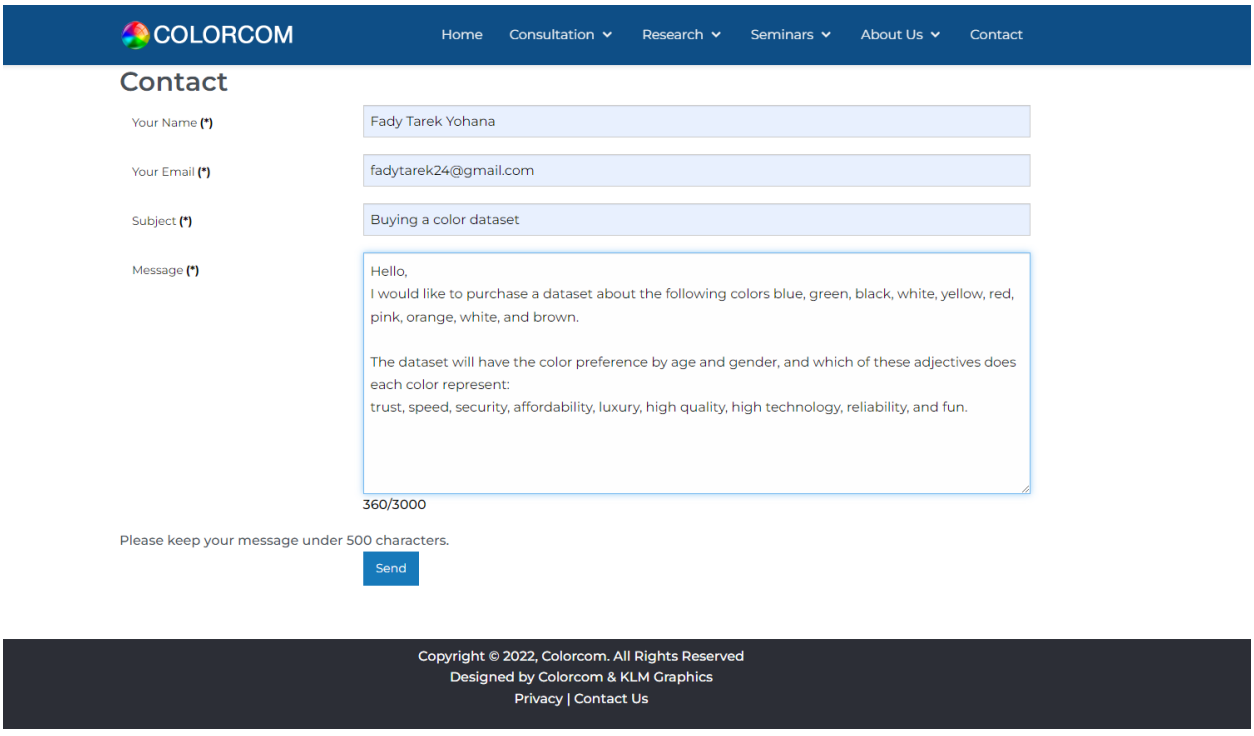

Figure 9: Color Dataset request from Colorcom.com

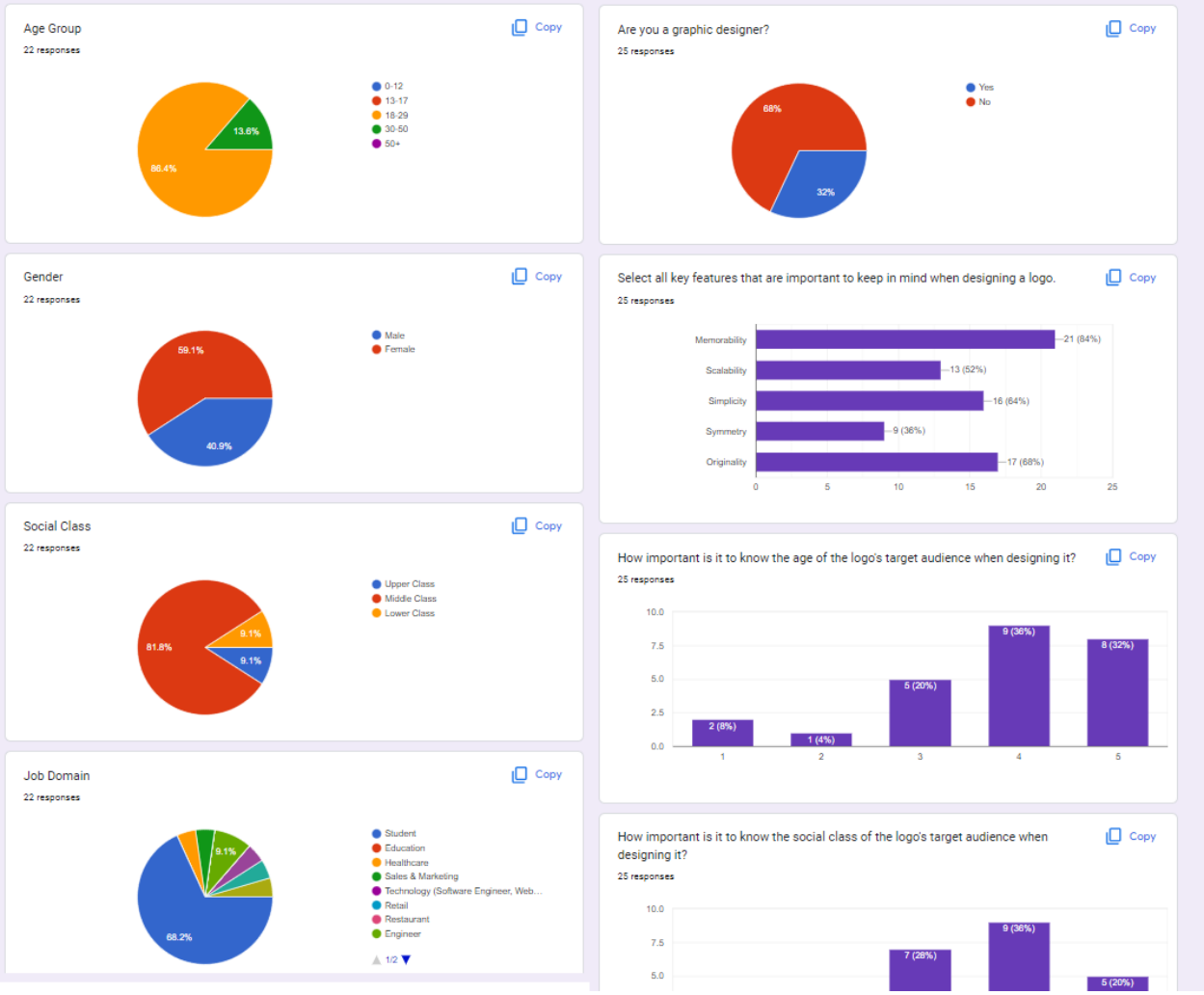

Figure 10: Survey Responses

11 Appendices

11.1 Supportive Documents

References

[1] Alexandr Kuzmenko, Sergey Kondratenko, Konstantin Dergachev, et al. “Ergonomic support for logo development based on deep learning”. In: *CEUR Workshop Proceedings*. Vol. 30. 2020.

[2] Alexander Sage, Eirikur Agustsson, Radu Timofte, et al. “Logo synthesis and manipulation with clustered generative adversarial networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5879–5888.

[3] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, et al. “Improved training of wasserstein gans”. In: *Advances in neural information processing systems* 30 (2017).

[4] Longchun Mao, Jinghua Wang, and Jianmin Jiang. “Computerized logo synthesis with wavelets-enhanced adversarial learning”. In: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2020, pp. 1–5.

[5] Ajkel Mino and Gerasimos Spanakis. “Logan: Generating logos with a generative adversarial neural network conditioned on color”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2018, pp. 965–970.

[6] Cedric Oeldorf and Gerasimos Spanakis. “LoGANv2: conditional style-based logo generation with generative adversarial networks”. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE. 2019, pp. 462–468.

[7] Tingting Qiao, Jing Zhang, Duanqing Xu, et al. “Mirrorgan: Learning text-to-image generation by redescription”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1505–1514.

[8] Akanksha Singh, Sonam Anekar, Ritika Shenoy, et al. “Text to Image using Deep Learning”. In: *International Journal of Engineering and Technical Research* (Mar. 2022).

[9] Karim Armanious, Vijeth Kumar, Sherif Abdulatif, et al. “ipa-medgan: Inpainting of arbitrarily regions in medical modalities”. In: *arXiv preprint arXiv:1910.09230* (2019).

[10] Yuchuan Zheng. “Visual Memory Neural Network for Artistic Graphic Design”. In: *Scientific Programming* 2022 (2022).

[11] Tero Karras, Samuli Laine, and Timo Aila. “A style-based generator architecture for generative adversarial networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.