Software Requirement Specification Document for StyleBusters

Walid Mohamed, Sherif Wael, Abdelrahman Hesham, Adham Moataz
Supervised by: Dr.Fatma Helmy, Eng.Mahmoud ElSahhar

May 15, 2023

Table 1: Document version history

Version	Date	Reason for Change
1.0	25-Oct-2022	SRS First version's specifications are defined.
2.0	2-May-2023	SRS latest updates are defined.

GitHub: https://github.com/walidwalid23/StyleBustersApplication

Contents

1 Introduction **3**

1.1 Purpose of this document . . . 3
1.2 Scope of this document . . . 3
1.3 Business Context . . . 3

2 Similar Systems **4**

2.1 Academic . . . 4
2.2 Business Applications . . . 6

3 System Description **7**

3.1 Problem Statement . . . 7
3.2 System Overview . . . 7
3.3 System Scope . . . 10
3.4 System Context . . . 11
3.5 Objectives . . . 11
3.6 User Characteristics . . . 11

4 Functional Requirements **12**

4.1 System Functions . . . 12
4.2 Detailed Functional Specification . . . 14

5 Design Constraints **15**

5.1 Standards Compliance . . . 15
5.2 Hardware Limitations . . . 15

6 Non-functional Requirements **15**

6.1 Maintainability . . . 15
6.2 Availability . . . 15
6.3 Usability . . . 15
6.4 Portability . . . 16
6.5 Security . . . 16
6.6 Reliability . . . 16

7 Data Design **16**

8 Preliminary Object-Oriented Domain Analysis **17**

9 Operational Scenarios **18**

10 Project Plan **19**

11 Appendices **20**

11.1 Definitions, Acronyms, Abbreviations . . . 20
11.2 Supportive Documents . . . 20

Abstract

The main idea of this project is to implement deep learning techniques to extract style features from images of different clothes, artworks, and logos and then use these features to retrieve similar clothes, logos, and artworks from various websites, which is going to help the online shoppers to shop faster, the fashion designers to find out if their designs were stolen, in addition to that it will also help in the prevention of stealing artworks and replicating logos.

1 Introduction

1.1 Purpose of this document

This document's purpose is to provide a detailed description of the Similar Designs Detection System goals and parameters to be used as a guide for developers. It also describes the system's target users, data design as well as the software and hardware requirements of the system.

1.2 Scope of this document

This document covers the functional, non-functional, and system requirements as well as the design constraints, operational scenarios, data design, and class diagram, which are required to develop the Similar Designs Detection System.

1.3 Business Context

According to E-commerce benchmarks, 2022 [1], 27% of the global population shop online, and during 2022, global online sales are estimated to be around 5 trillion dollars. Also, The global apparel market has grown from $551.36 billion in 2021 to $606.19 billion in 2022 [2], and according to Statistics [3], the global art market has been valued at 65.1 billion dollars in 2021 and according to financesonline.com [4], logos are the most recognizable brand identifiers. Therefore any solution that will help online shoppers to have a better shopping experience as well as help fashion designers to search for copies of their designs, help in detecting stolen artworks, and prevent the replication of logos, is going to have a great impact on the business industry.

2 Similar Systems

2.1 Academic

1. Tuinhof et. al. [5]:

   This paper discusses the creation of a project based on recommending products that are suitable for the user to increase the probability of a user's purchase. To enhance the performance of the model, it's trained on a convolutional neural network (CNN) to solve classification tasks and extract hidden features by composing many convolutional layers. In this paper, researchers were using two dataset Fashion category and Fashion texture. The Fashion category dataset consists of 11,851. Also, The Fashion texture dataset consists of 7,342 images, further information is shown in figure 1. The highest accuracy model for the category dataset was 87.0% with BN-Inception. Furthermore, The highest accuracy model for the texture dataset was 80.0% with BN-Inception. The researchers should have taken into consideration Siamese networks, so they would get higher accuracy than what they got.

Method	Params. ($\times 10^6$)	Accuracy	
		top-3	top-5
WTBI [2]	-	43.73	66.26
DARN [11]	-	59.48	79.58
FashionNet+500 [20]	-	57.44	77.39
FashionNet+Joints [30]	-	72.30	81.52
FashionNet+Poselets [30]	-	75.34	84.87
Deepfashion [20]	$\sim$134	82.58	90.17
Lu et al.VGG-16 [21]	134.4	86.72	92.51
Weakly [3]	-	86.30	92.80
ResNet50+OC	28.1	87.34	93.42
DenseNet121+OC	7.9	87.58	93.39
SEResNet50+OC	28.1	87.58	93.58
SEResNeXt50+OC	27.6	**88.42**	**93.93**

Figure 1: Quantitative comparison of category classification on category prediction of DeepFashion dataset

2. Sun et. al. [6]:

   Mainly in this paper, the problem was recommending products that are suitable for the user to increase the probability of a user's purchase. Throughout the paper, the researchers have proposed a version of two deep CNNs like VGG-19 and AlexNet, which are pre-trained on the ImageNet classification dataset and are fine-tuned for this task. The researchers were using three benchmark datasets such as Flickr Style, AVA Style, and WikiPaintings. The AVA Style dataset has about 14,000 images with different styles. Also, The Flickr Style dataset and WikiPainting dataset consist of 85,000 images and 80,000 images. The difference between datasets is shown in Figure 2. The best results for the WikiPainting dataset by using VGG-19-based MNet was 60.4% mAP. Conversely, the best result for the Flickr dataset using VGG-19-based MNet was 41.0% mAP. Last but not least, The best results for the AVA Style dataset by using VGG-19-based MNet was 65.5% mAP. This research paper can improve its results by choosing an optimal list of object features.

Dataset	Attribute Precision			Attribute Recall		
	baseline	fc-fc	conv-fc	baseline	fc-fc	conv-fc
DeepFashion	0.0448	0.0899	-	0.2932	0.2127	-

Figure 2: Test Results

3. Shen et. al.[7]:
   The main problem in this research paper was discovering close duplicate patterns in large numbers of artworks. Researchers aim to solve this problem by fine-tuning a standard deep feature using self-supervised learning on a specific art collection. The researchers have used a dataset of artworks for Jan Brueghel and his workshop1 which consist of 1587 artworks. The researchers have annotated 273 near duplicate details. In this paper, the researcher's method achieved 88.5% as the highest accuracy for LTLL and 85.7% as the highest accuracy for Oxford, as shown in Figure 3. The model created can lead to better results by training it on a larger dataset.

Method	LTLL (%)	Oxford (%)
B. Fernando et al.[16]	56.1	-
F. Radenović et al.[35]	-	87.8
ResNet18 max-pool, image level	59.8	14.0
ResNet18 + discovery	80.9	85.0
Ours (trained LTLL + discovery)	**88.5**	83.6
Ours (trained Oxford + discovery)	85.6	**85.7**

Figure 3: Classification accuracy on LTLL and retrieval mAP on Oxford5K

2.2 Business Applications

1. StyleSnap [8] : StyleSnap is an AI-powered feature created by Amazon that helps online shoppers in shopping through Amazon by allowing them to upload screenshots of the designs that they want to search for and getting Amazon products with similar designs as a result.

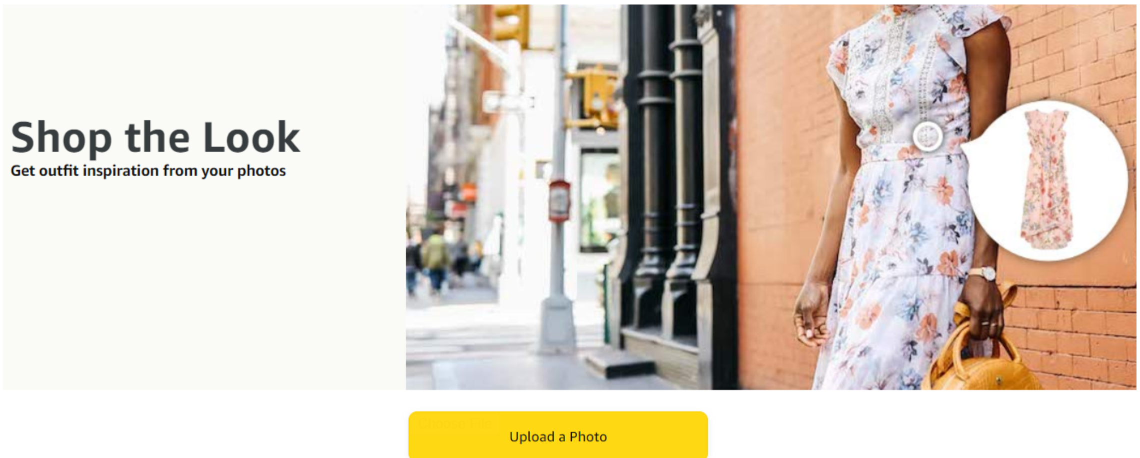

Figure 4: StyleSnap

2. Oxford Painting Search [9] : Oxford Painting Search is an AI-powered feature created by Oxford to help people in searching for paintings through Oxford by allowing them to upload screenshots of the paintings that they want to search for and getting similar paintings found in BBC and ART uk.

Figure 5: Oxford Painting Search

3 System Description

3.1 Problem Statement

The main problems that our system is trying to solve are helping online shoppers to shop faster across e-commerce websites as well as helping fashion designers in finding out if their styles are being copied or not by allowing them to search for clothes by designs. Our system is also going to help in preventing artworks from being stolen by allowing users to search for similar style artworks as well as preventing the replication of logos using deep learning techniques.

3.2 System Overview

Clothes System Overview Diagram:

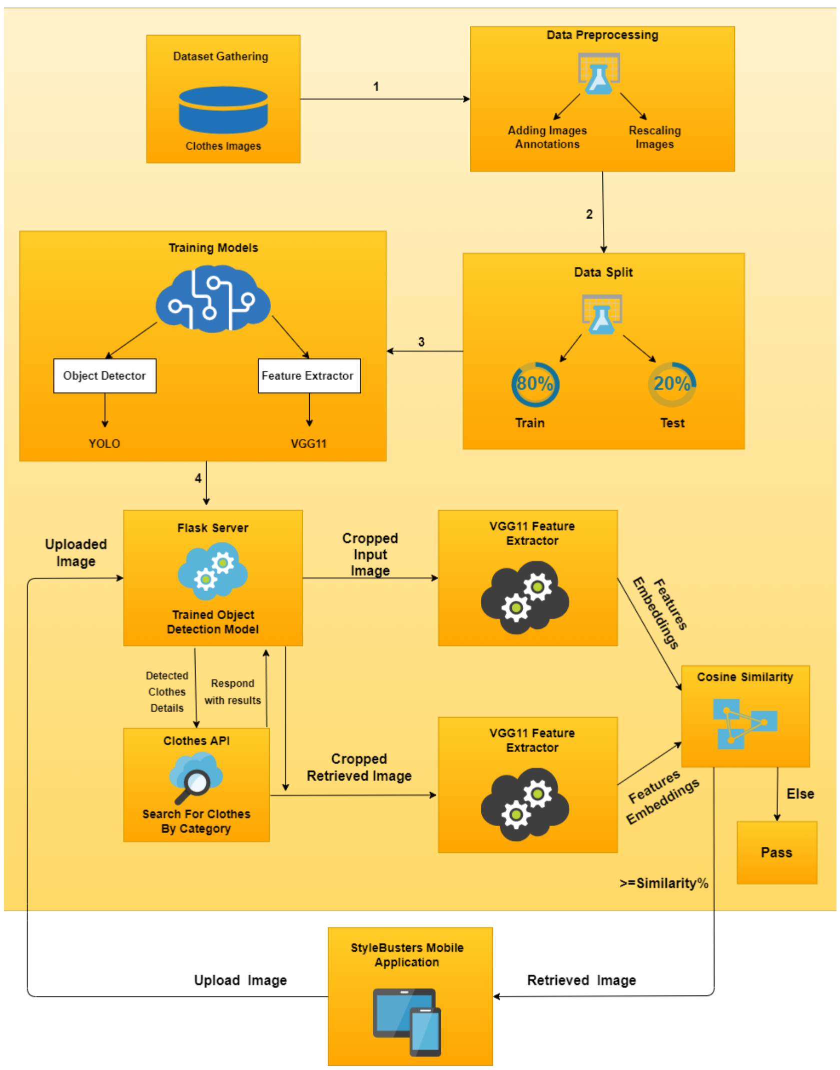

Figure 6: Clothes System Overview

Logos System Overview Diagram:

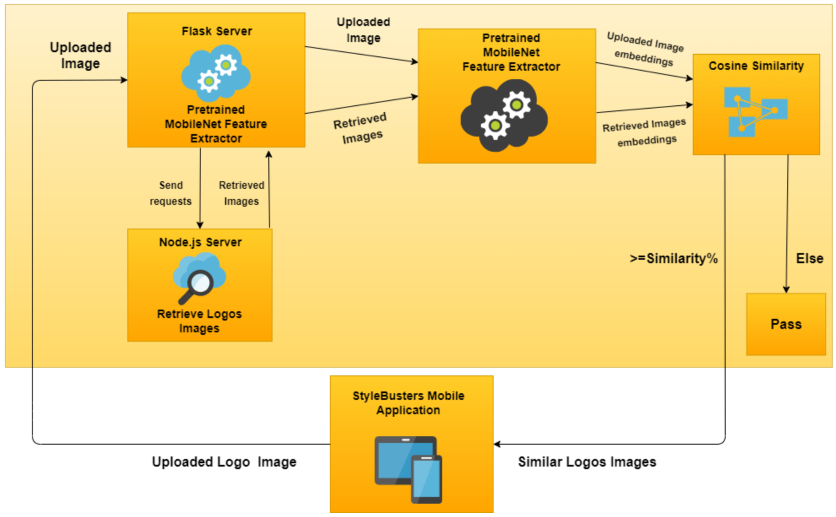

Figure 7: Logos System Overview

Artworks System Overview Diagram:

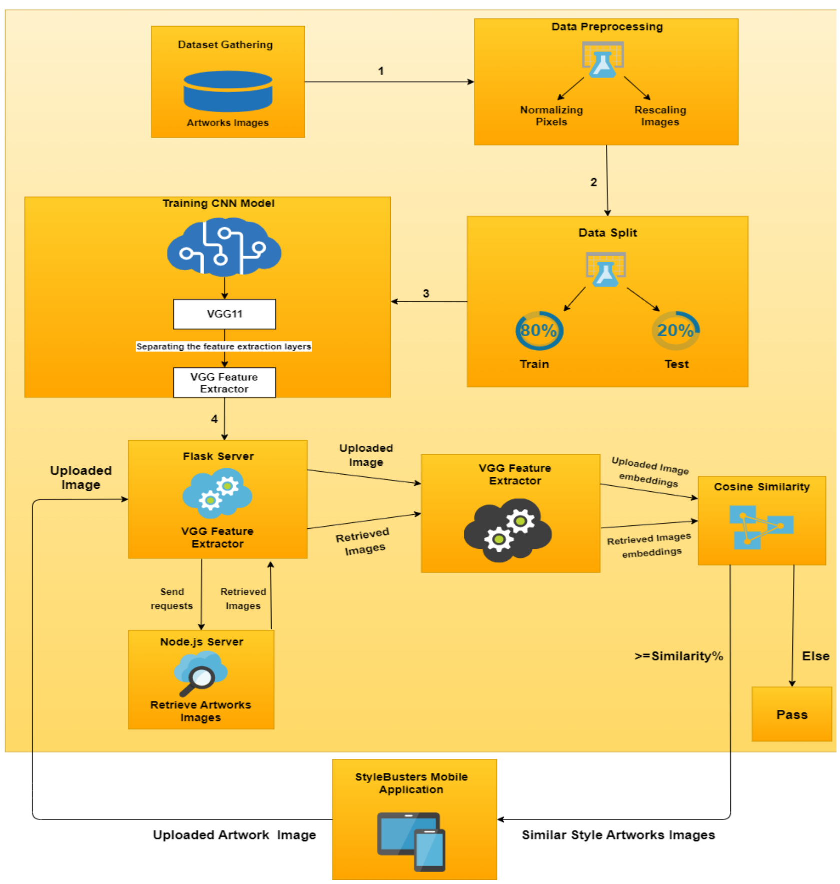

Figure 8: Artworks System Overview

Systems Description:

- Dataset Gathering:
  - For the Clothes System: We are going to use the Category and Attribute Prediction subset of the DeepFashion dataset [10] which contains 289,222 diverse clothes images which are divided into 46 different clothing categories from both consumers and commercial shopping stores, in order to train the classifier model. Additionally, we are going to use the DeepFashion2 dataset [11] which contains 491K diverse images of 13 popular clothing categories from both the commercial shopping stores and consumers in order to train the object detection model.
  - For the Artworks System: We are going to use the refined WikiArt Dataset [12] which contains more than 80000 unique images from 1119 different artists in 27 styles.
- Data Preprocessing: the data that has been gathered will be preprocessed by adding the suitable annotations to each image and rescaling the images as well as normalizing the pixels values; then, the data will be split into a training set of 80% and a testing set of 20%.
- Deep Learning Models:
  - For the Clothes System: YOLO V8 [13] is used as an object detector and classifier to be able to detect, classify and crop the clothes from the images that will be uploaded by the users and the retrieved images. The VGG (Visual Geometry Group) model will be trained for classification then it will be used as a feature extractor to extract features from the cropped input clothes images and the cropped clothes images retrieved from the web scraping server.
  - For the Logos System: MobileNet model [14] which is pretrained on ImageNet Dataset [15] will be used as a feature extractor to extract features from the uploaded logo and the logos retrieved from the web scraper query server.
  - For the Artworks System: VGG (Visual Geometry Group) deep learning model will be trained for classification and after the training process the feature extraction part of the model will be separated and used as a feature extractor in our system to extract style features from the artwork uploaded to the system and the artworks retrieved from the web scraper query server
- Similarity Comparison: The Cosine Similarity [16] will be used to get the similarity percentage between the feature embeddings that are resulted from the feature extractors then the retrieved image will be displayed to the users if its similarity percentage is above a certain limit.
- Web Scraping: The Node.JS Server will be responsible for web scraping and retrieving artworks, clothes and logos from various reliable websites.
- Models Deployment: the trained feature extraction models will be deployed in a flask server to serve the requests sent by the users of the mobile application and send requests to the Node.js web scrapers then respond with the results to the users of the mobile application.

- Mobile Application: A Flutter mobile application will be developed which will allow the users of the application to upload images of clothes,logos and artworks and retrieve clothes,logos and artwork of similar styles.

3.3 System Scope

The Main Features Of The System:

- The system will allow the users to search for clothes by styles across e-commerce websites by uploading an image of the desired piece of clothes.
- The system will allow the users to search for artworks that have a similar drawing style by uploading an image of an artwork.
- The system will allow the users to search for identical logos by uploading an image of a logo.

The Expected Outcomes Of The System:

- A mobile application that will allow the users to search for similar style clothes or artworks or logos.

The Boundaries Of The System:

- The system will not be able to query through all the websites on the internet as this is not technically possible.

3.4 System Context

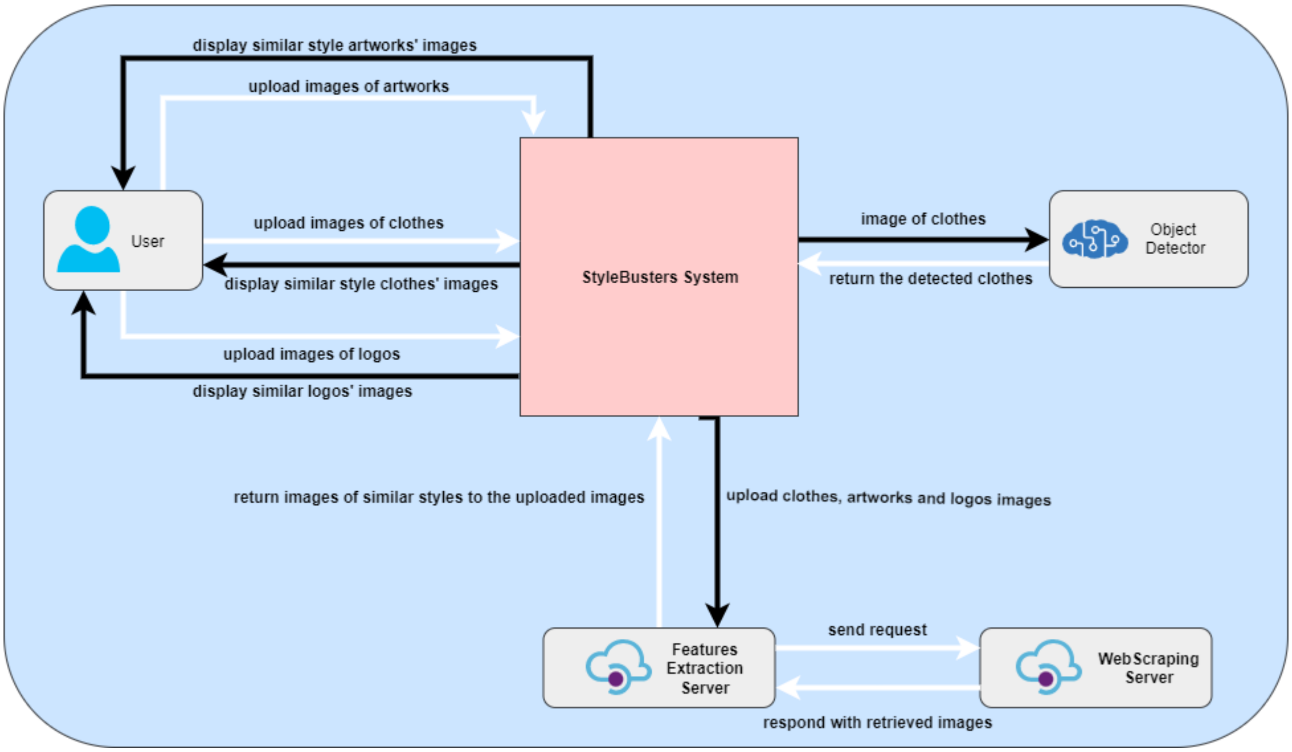

Figure 9: System Context

3.5 Objectives

- To implement a system that will allow the users to find clothes from various e-commerce websites that have similar styles to the clothes' images that they upload.
- To implement a system that will help in the prevention of stealing artworks by searching for artworks that have a similar drawing style.
- To implement a system that will help in the prevention of the replication of logos by searching for identical existing logos.

3.6 User Characteristics

- Users of all ages and genders can use the system's mobile application.
- The users should only have a basic knowledge of how to download and use a mobile application.

4 Functional Requirements

4.1 System Functions

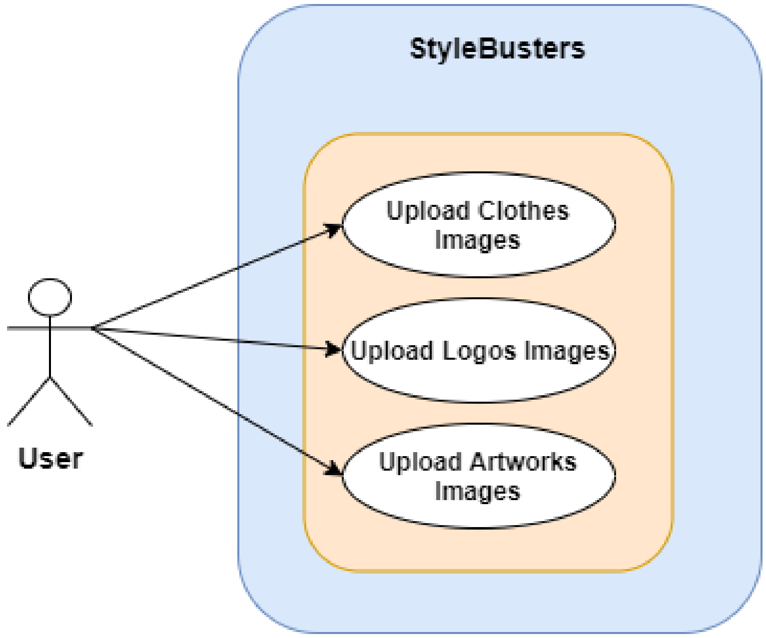

Figure 10: Use Cases Diagram

- FR01: The users shall be able to upload an image of clothes to retrieve clothes of similar style.
- FR02: The users shall be able to upload image of artwork to retrieve artworks of similar drawing style to the uploaded image on email.
- FR03: the user shall be able to upload a logo image to retrieve identical logos to the uploaded image on email.

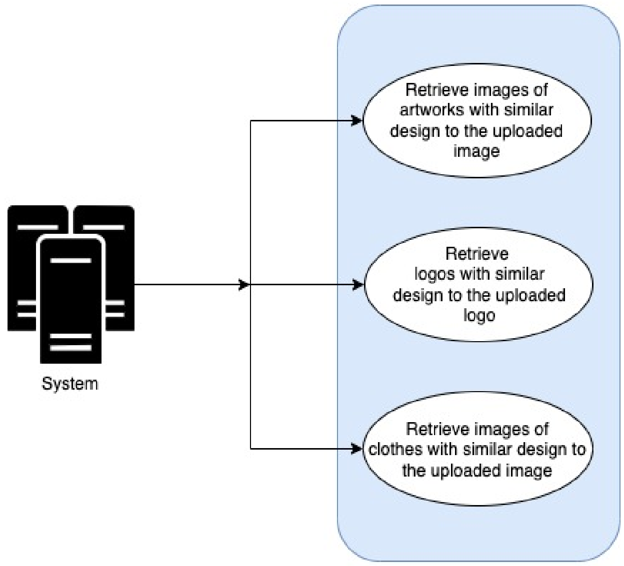

Figure 11: Use Cases Diagram

1. The system shall be able to retrieve images of artworks with similar style to the uploaded image.
2. The system shall be able to retrieve images of clothes with similar style to the uploaded image.
3. The system shall be able to retrieve logos that are identical to the uploaded logo.

4.2 Detailed Functional Specification

Table 2: Process Function

Name	process
Code	FR01
Priority	Extreme
Critical	This function is essential for the system to work
Description	This function process an image then extract its features using a feature extraction model then return the extracted features vector as output
Input	imagePath
Output	List of features
Pre-condition	image path must be valid and feature extraction model must be loaded
Post-condition	retrieving image features
Dependency	it depends on the loaded feature extraction model
Risk	invalid image path or model path

Table 3: Cosine Similarity Function

Name	cosineSimilarity
Code	FR02
Priority	Extreme
Critical	This function is essential for the system to work
Description	This function takes two feature vectors as an input and compute the similarity between them then returns the similarity percentage
Input	featureVector1, featureVector2
Output	Double similarity percentage
Pre-condition	feature vectors must be in valid format
Post-condition	similarity percentage will be computed
Dependency	it depends on the process function
Risk	invalid feature vectors format

Table 4: Search Function

Name	search
Code	FR03
Priority	Extreme
Critical	This function is essential for the system to work
Description	This function takes an image as input then search for images of similar style features
Input	image
Output	List of similar Images
Pre-condition	image path must be valid
Post-condition	images with similar designs should be retrieved
Dependency	it depends on the process and cosineSimilarity functions
Risk	invalid image path

5 Design Constraints

5.1 Standards Compliance

The Smartphone that will run the system's mobile application must have an internet connection to connect to the system's API.

5.2 Hardware Limitations

The Smartphone that will run the system's mobile application must have an Android version of 4.1 and above or IOS 11 and above and it also must have a working camera to capture images.

6 Non-functional Requirements

6.1 Maintainability

The system should be maintainable and we will achieve this by documenting our system and using a clean architecture pattern when developing the system's mobile application.

6.2 Availability

The system shall be always up and running to receive requests from the users (internet connection is required to receive the results).

6.3 Usability

The system should be easy to use and this will be achieved by creating a user-friendly and organized user-interface.

6.4 Portability

The system shall work on any Android or IOS powered devices.

6.5 Security

The system must be secure and protected against any type of attacks and we will achieve this by taking various security measures such as encrypting the passwords of the users by hashing to keep their data secure.

6.6 Reliability

The system should be reliable and we will achieve this by reducing down times by choosing a reliable hosting service.

7 Data Design

- For the Clothes System: Our dataset that is used to train the object detector,is an extensive dataset of fashion designs, and it is called DeepFashion2 [11] . Deepfashion2 contains 292K diverse images from 13 different clothing categories that are famous for commercial shopping stores and consumers. In the dataset each item in the images is labeled according to scale, occlusion, zoom-in, viewpoint, category, style, bounding box, dense landmarks and per-pixel mask. The dataset is splitted into 191K images of the training set , 67K images of the test set ,and 34K images of the validation set.

- For the Artworks System: We are going to use the refined WikiArt Dataset [12] which contains more than Dataset contains 80000 unique images from 1119 different artists in 27 styles.

8 Preliminary Object-Oriented Domain Analysis

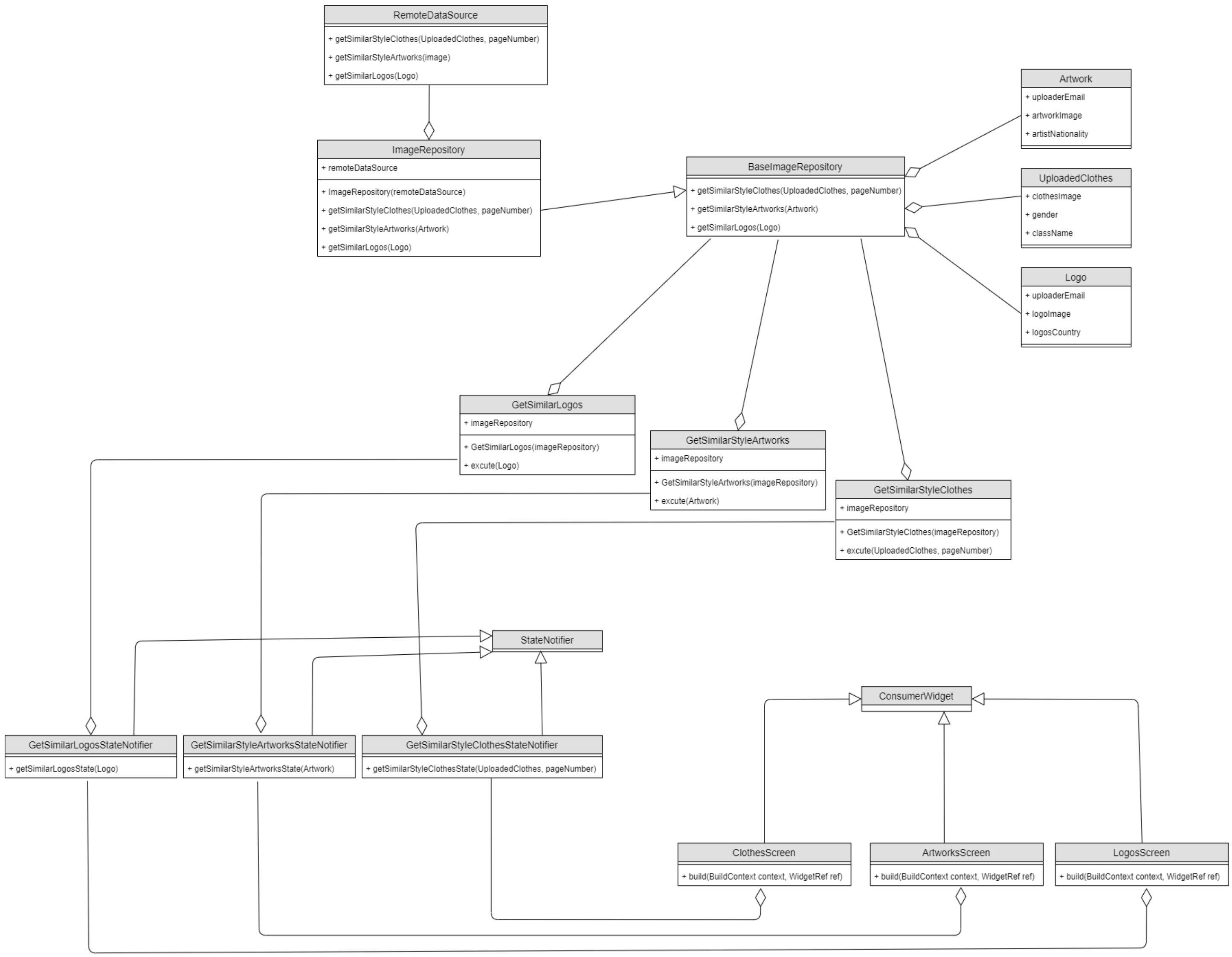

Figure 12: Class Diagram

9 Operational Scenarios

- Scenario 1
  In this scenario, the user will open the system's mobile application and press on the clothes navigation bar button then upload an image of the desired piece of clothes then the user will receive images of clothes that have a similar style to the uploaded image and each received image will include a link to the website where the clothes are being sold and their prices.

- Scenario 2
  In this scenario, the user will open the system's mobile application and press on the artworks navigation bar button then upload an image of the desired artwork then the user will receive images of artworks that have a similar style to the uploaded image on email and each received image will include a link to the website where the artwork exists and details about the original artist of the artwork.

- Scenario 3
  In this scenario, the user will open the system's mobile application and press on the logos navigation bar button then upload an image of the desired logo then the user will receive images of logos that are identical to the uploaded logo on email, along with the name of the company that owns each logo.

10 Project Plan

ID	Task	Start Date	Number Of Days	Team Member
1	Clothes Dataset Gathering	5-Oct	2	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
2	Survey Creation	6-Oct	1	Sherif Wael, Walid Mohamed, Adham Moataz, Abdelrahman Hesham
3	Research Papers Gathering	7-Oct	2	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
4	Clothes Object Detection Model Training	8-Oct	1	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
5	Clothes Object Detection Model Testing	9-Oct	1	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
6	Proposal Document and 10% of Project	10-Oct	14	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
7	finding pretrained feature extraction model	20-Oct	2	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
8	Testing MobileNet feature extractor	23-Oct	2	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
9	Implementing local search	26-Oct	3	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
10	Creating logos API by Web Scraping	10-Nov	1	Walid Mohamed
11	Applying remote search for logos using API	11-Nov	1	Walid Mohamed
12	Applying feature extraction on logos	11-Nov	1	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
13	Finishing SRS Document	15-Nov	10	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
14	Artsy website Web Scraping API (Node.js)	25-Jan	1	Walid Mohamed
15	Logos Flask Backend Implementation	26-Jan	1	Adham Moataz, Walid Mohamed
16	Artworks Dataset Gathering	27-Jan	1	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
17	Artworks VGG Feature Extractor Training	28-Jan	3	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
18	Clothes Style Dataset Gathering	1-Feb	1	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
19	Clothes VGG Feature Extractor Training	1-Feb	3	Abdelrahman Hesham
20	Clothes Object Detector Training (YOLO V8)	2-Feb	3	Sherif Wael
21	SDD Document	2-Feb	20	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
22	Artworks Flask Backend Implementation	7-Mar	2	Walid Mohamed
23	Clothes Flask Backend Implementation	9-Mar	2	Sherif Wael, Abdelrahman Hesham, Adham Moataz
24	Deploying Servers	11-Mar	1	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
25	Developing The Mobile Application	13-Mar	30	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
26	System (Prototype)	21-Apr	1	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
27	Testing The System	22-Apr	7	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham
28	Technical Evaluation	13-May	7	Walid Mohamed, Sherif Wael, Adham Moataz, Abdelrahman Hesham

Figure 13: Plan Table

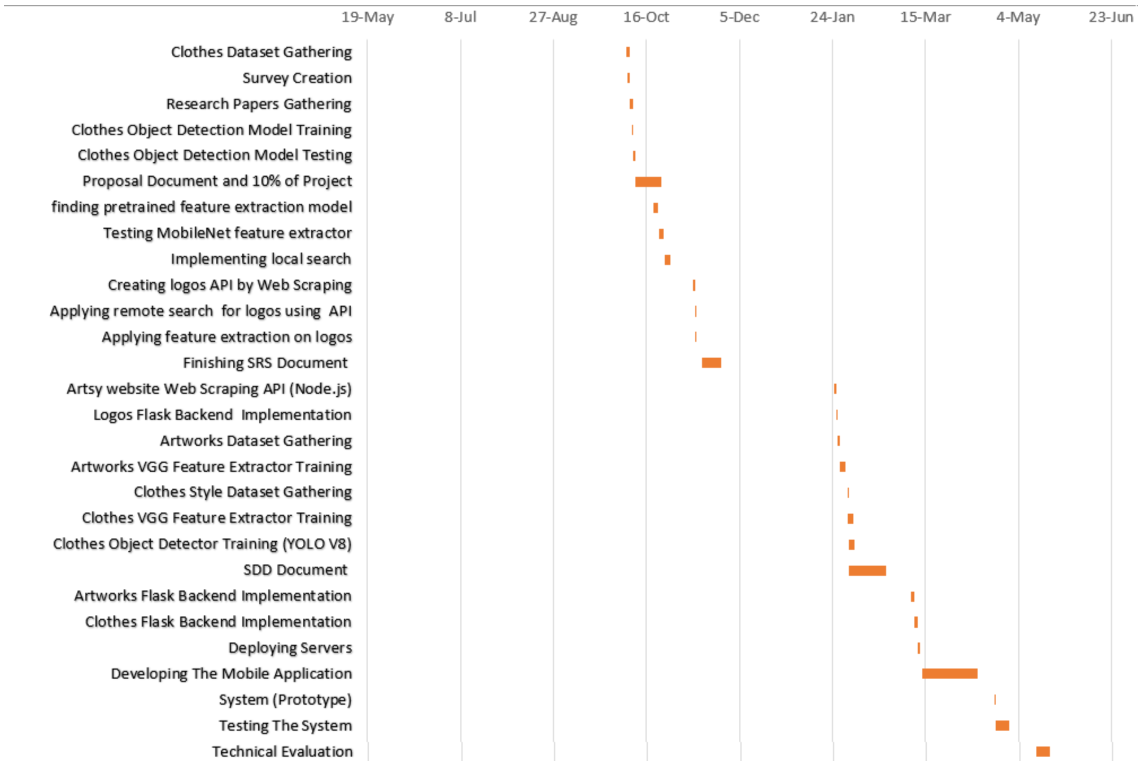

Figure 14: GANTT CHART

11 Appendices

11.1 Definitions, Acronyms, Abbreviations

Term	Stands For
API :	Applicaiton Programming Interface

11.2 Supportive Documents

Survey: We have conducted a survey asking teens the following two questions and the response ratios are displayed in the below figures:

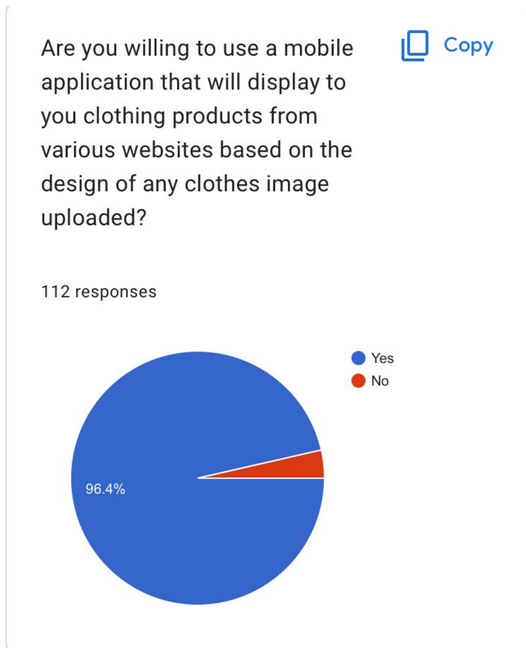

Figure 15: Question1

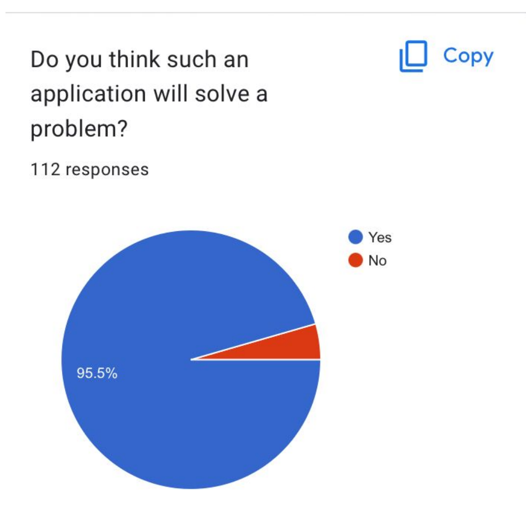

Figure 16: Question2

References

[1] https://www.tidio.com/blog/online-shopping-statistics/.

[2] https://www.thebusinessresearchcompany.com/report/apparel-global-market-report\#:~:text=The\%20global\%20apparel\%20market\%20grew,least\%20in\%20the\%20short\%20term..

[3] https://www.statista.com/topics/1119/art-market/.

[4] https://financesonline.com/logo-statistics/\#:~:text=Logos\%20are\%20the\%20most\%20recognizable,(Study\%20Finds\%2C\%202020)..

[5] Hessel Tuinhof, Clemens Pirker, and Markus Haltmeier. "Image-based fashion product recommendation with deep learning". In: *Machine Learning, Optimization, and Data Science: 4th International Conference, LOD 2018, Volterra, Italy, September 13-16, 2018, Revised Selected Papers 4*. Springer. 2019, pp. 472–481.

[6] Tiancheng Sun, Yulong Wang, Jian Yang, et al. "Convolution neural networks with two pathways for image style recognition". In: *IEEE Transactions on Image Processing* 26.9 (2017), pp. 4102–4113.

[7] Xi Shen, Alexei A Efros, and Mathieu Aubry. "Discovering visual patterns in art collections with spatially-consistent feature learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9278–9287.

[8] https://www.amazon.com/stylesnap.

[9] http://zeus.robots.ox.ac.uk/artsearch/.

[10] Ziwei Liu, Ping Luo, Shi Qiu, et al. "DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[11] Yuying Ge, Ruimao Zhang, Lingyun Wu, et al. "A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images". In: *CVPR* (2019).

[12] Wei Ren Tan, Chee Seng Chan, Hernan Aguirre, et al. "Improved ArtGAN for Conditional Synthesis of Natural Image and Artwork". In: *IEEE Transactions on Image Processing* 28.1 (2019), pp. 394–409. DOI: 10.1109/TIP.2018.2866698. URL: https://doi.org/10.1109/TIP.2018.2866698.

[13] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *CoRR* abs/1506.02640 (2015). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640.

[14] Andrew G. Howard, Menglong Zhu, Bo Chen, et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: *CoRR* abs/1704.04861 (2017). arXiv: 1704.04861. URL: http://arxiv.org/abs/1704.04861.

[15] Jia Deng, Wei Dong, Richard Socher, et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[16] https://en.wikipedia.org/wiki/Cosine_similarity.