



COMSATS University Islamabad (CUI)

**Software Requirement Specification
(SRS DOCUMENT)**

for

AI Powered Image Transformation Tool

Version 1.0

By

Muhammad Usman Saleem

CIIT/FA21-BSE-039/VHR

Muhammad Fareed

CIIT/FA21-BSE-032/VHR

Supervisor

Dr Rab Nawaz Bashir

**Bachelor of Science in Software Engineering
(2021-2025)**

Revision History

Name	Date	Reason for Changes	Version

Application Evaluation History

Comments (by committee)	Action Taken

Supervised By
Dr Rab Nawaz Bashir

Signature

TABLE OF CONTENTS

List of Figures	v
List of Tables	vi
1 Introduction	2
1.1 Purpose	2
1.2 Scope	2
1.3 Intended Audience	3
2 Overall Description	4
2.1 Product Perspective	4
2.1.1 System Interfaces	4
2.1.2 User Interfaces	5
2.1.3 Hardware Interfaces	5
2.2 Product Functions	5
2.3 User Characteristics	6
2.4 Constraints	6
2.5 Assumptions and Dependencies	7
3 Requirement Collection Techniques	8
3.1 Introduction	8
3.2 Use Cases for Requirement Collection Techniques	9
4 Functional Requirements	14
4.1 Overview	14
4.2 Functional Requirements	14
4.2.1 User Interface	14
4.2.2 Style Transfer	15
4.2.3 User Management	15
4.2.4 Image Processing and Format Support	16
4.2.5 Performance Requirements	16
5 Non-Functional Requirements	17
5.1 Performance Requirements	17
5.2 Usability Requirements	18
5.3 Reliability Requirements	18

5.4	Security Requirements	19
5.5	Scalability Requirements	19
5.6	Maintainability Requirements	19
6	External Interface Requirements	21
6.1	User Interfaces	21
6.2	Front End	22
6.3	Sign up screen	22
6.4	Login screen	23
6.5	Dashboard	24
6.6	Account Setting	25
6.7	Result	26
6.8	Hardware Interfaces	26
6.9	Software Interfaces	26
6.10	Communication Interfaces	27
7	Design Description	28
7.1	Entity Relationship Diagram	28
7.2	ERD Diagram	29
7.3	Activity Diagram	29
7.4	Activity Diagram	30
7.5	Class Diagram	32
7.6	Sequence Diagram	32
7.7	Sequence Diagram	33
7.8	Data Flow Diagram	34
7.9	State Transition Diagram	34
7.10	Algorithm Viewpoint for Image Style Transfer System	34
7.11	Pseudocode	36

LIST OF FIGURES

3.1 Use Case Diagram for Requirement Collection Techniques	9
6.1 GUI _{Diagram}	22
6.2 GUI _{Diagram}	23
6.3 GUI _{Diagram}	24
6.4 GUI _{Diagram}	25
6.5 GUI _{Diagram}	26
7.1 ERD Diagram for Requirement Collection Techniques .	29
7.2 Activity Diagram for Requirement Collection Techniques	30
7.3 Class Diagram for Requirement Collection Techniques	32
7.4 Sequence Diagram for Requirement Collection Techniques	33
7.5 Data Flow Diagram for Requirement Collection Techniques level 1	34
7.6 Data Flow Diagram for Requirement Collection Techniques level 2	34
7.7 State Transition Diagram for Requirement Collection Techniques	34

LIST OF TABLES

Abstract

AI Powered Image Transformation Tool is a web application built to use artificial intelligence to draw artistic styles from uploaded images. The application has a simple **React.js**-fronted where users can upload images, apply pre-trained styles, and download images. The backend is created with help of **Laravel** , and it contains a function for processing the images by using **TensorFlow** models and handling user accounts. The process involves extracting content and style representations from images, computing Gram matrices for style transformation, and generating a stylized output through convolutional neural networks (CNNs). The algorithm supports both single and blended styles, allowing users to adjust the intensity of multiple artistic effects. The system is deployed as an API using FastAPI, ensuring efficient and scalable image processing. This document provides an in-depth analysis of the algorithmic structure, implementation, and evaluation of the style transfer system. Served on services as **AWS** or **Heroku** for scalability, the application includes real time, multiuser compute and an ergonomic, informal design for casual or creative users interesting in AI-infused graphics.

Chapter 1

Introduction

1.1 Purpose

The objective of this document called the Software Requirements Specification (SRS) is to outlines the necessary requirements of creating the **AI Powered Image Transformation Tool**. The present document outlines the functional and non-functional requirements of the system as far as it provides a clear picture of the overall goals of the project and detailed specifications to the developers and stakeholders.

1.2 Scope

AI Powered Image Transformation Tool is an online tool aimed to produce artworks from photos using the style transfer with the help of AI. Users will be able to post images and then choose among a set of artistic filters for further manipulation with the pictures. Key functionalities include

- Loosely Realtime style transfer utilizing pre-trained deep learning models.
- The four interface elements are the option to upload images, select styles, and download images transformed based on the selected style.
- They include support for several image formats and several users at a time.

This tool is suitable for non-professional users who want to play around with images, and then also for digital artists who want to improve their production creativity.

1.3 Intended Audience

This document is intended for:

- **Developers:** In order to offer the specific requirements needed to build the system that shall be used in the company.
- **Project Supervisors:** To look in depth at the objectives of the system, its structure and the way it works.
- **End Users:** In order to know which functionalities they can get from the final product.
- **Future Developers/Maintainers:** In order to identify the characteristics that the construction could possess and technical requirements for possible improvements.

Chapter 2

Overall Description

2.1 Product Perspective

The AI Powered Image Transformation Tool is a web application dedicated to transform the user's image into different artistic technique. Next would be style transfer where the system would utilize pre-trained model for the purpose in order to allow users apply a creative twist to their pictures. What makes this tool great is that it is a powerful AI driven platform designed at the same time for novices and pro graphical artists.

2.1.1 System Interfaces

The system will interact with various external services and components:

- **Frontend Interface:** Developed using the website development framework, React.js that enhances its ability to load in a responsive manner and be easily navigable.

- **Backend Interface:** Ensures that the server's requests are managed using Systems such as Flask or Django while AI models are managed together with the database.
- **Database Interface:** Both user data or image history will be archived on PostgreSQL or MongoDB.
- **AI Model Interface:** The model for the implementation of style transfer is based on TensorFlow.

2.1.2 User Interfaces

The user interface will feature:

- Clean and easy to manage layout for adding the pictures, choosing the attributes and viewing the changes.
- Save buttons to encode the picture in several formats for downloading.
- All pages with account management information and access to extensive history of viewed images by a particular user.

2.1.3 Hardware Interfaces

The application will be likely to be running on a cloud platform like AWS or Heroku and will involve little hardware interaction on the client side. This information will be accessible to the users and they will only require a device with internet connection and a web browser of right standards.

2.2 Product Functions

The main functions of the **AI Powered Image Transformation Tool** include:

- **Image Upload:** Allows users to upload an image to be transformed.
- **Style Selection and Transfer:** Applicants are provided with an opportunity to select a particular artistic approach and preview how it looks when being added to the uploaded picture instantly.
- **Image Download:** Offers the stylized image to download by the formats of JPEG or PNG.
- **User Accounts (Optional):** Which provides an interface for users to set up an account and track the changes, known as transformation history.

2.3 User Characteristics

The application is designed for:

- **Casual Users:** New users who have minimum computer proficiency and who are open to play around with artistic content transformations.
- **Digital Artists:** Users who want to draw additional effects to their artistic pieces using AI.
- **Technical Users:** It will be those occasional users who may want to learn more about the features or styles the application offers.

2.4 Constraints

- **Performance Constraints:** Image processing should be completed within 5 seconds for a smooth user experience.

- **Concurrency Constraints:** The system should support up to 100 concurrent users.
- **Security Constraints:** HTTPS will be enforced to ensure secure data transmission.

2.5 Assumptions and Dependencies

- Reliable internet connectivity is assumed for users to interact with the platform.
- The system is dependent on the availability of pre-trained AI models and stable cloud hosting services.
- User devices are assumed to support modern web browsers for compatibility.

Chapter 3

Requirement Collection Techniques

3.1 Introduction

Requirement collection is a critical phase in the software development lifecycle, where stakeholders' needs and expectations are gathered to form the foundation of the project. Various techniques can be employed to collect requirements effectively, ensuring that the final product meets the users' needs.

3.2 Use Cases for Requirement Collection Techniques

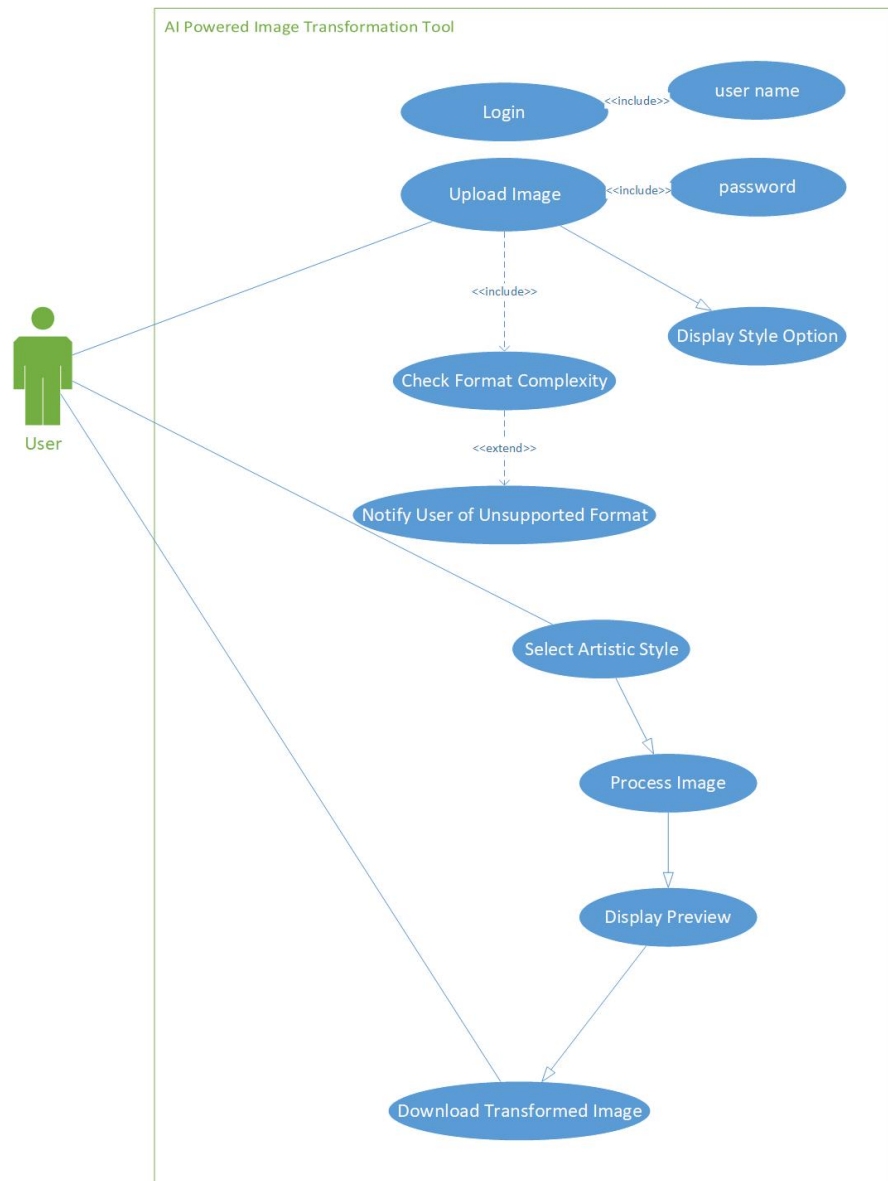


FIGURE 3.1: Use Case Diagram for Requirement Collection Techniques

1. Use Case IDs and Descriptions

Use Case ID	Use Case	Description
UC-01	Login	Allows the user to log in to access the tool.
UC-02	Upload Image	User uploads an image for transformation.
UC-03	Check Format Complexity	System assesses image format and complexity.
UC-04	Notify Unsupported Format	Alerts user if image format is unsupported.
UC-05	Select Artistic Style	User selects an artistic style to apply to the image.
UC-06	Process Image	System processes the image with selected style.
UC-07	Display Preview	Shows preview of the transformed image.
UC-08	Download Transformed Image	User downloads the final transformed image.

2. Use Case Steps

Use Case ID	Steps
UC-01	1. Enter username 2. Enter password 3. Click Login
UC-02	1. Click Upload 2. Select image file 3. Confirm upload
UC-03	1. System checks file format 2. Analyzes image complexity
UC-04	1. Display unsupported format message to user
UC-05	1. User selects an available artistic style
UC-06	1. System processes image 2. Applies selected style
UC-07	1. System displays preview 2. User reviews and approves or re-selects style
UC-08	1. User clicks download 2. System initiates download of transformed image

3. Alternative Flows

Use Case ID	Alternative Flow
UC-01	Incorrect credentials: Show error message and prompt re-entry.
UC-02	If no file is selected, prompt user to select a file before proceeding.
UC-05	If user decides to change style after preview, system allows re-selection of style.

UC-07	If user disapproves preview, system returns to style selection for adjustments.
UC-08	If download fails, system provides retry option.

4. Exception Scenarios

Use Case ID	Exception Scenario
UC-01	System down: Login temporarily unavailable, displays error message.
UC-02	File exceeds upload limit: Displays warning to choose smaller file.
UC-03	Unsupported format: System prevents further processing and displays error.
UC-06	Processing failure: System displays error and allows user to try again or select another image.
UC-08	Download interrupted: System prompts to retry download.

5. Functional Requirements

Use Case ID	Functional Requirement
UC-01	System must validate credentials and provide access upon successful login.
UC-02	System must allow image uploads within specified format and size limitations.
UC-03	System must analyze uploaded image format and complexity to determine compatibility.
UC-04	System must notify the user if uploaded image format is unsupported.

UC-05	System must display available artistic styles for user selection.
UC-06	System must apply the chosen style to the uploaded image and begin processing.
UC-07	System must display a preview of the transformed image for user review.
UC-08	System must enable download of the final transformed image after user approval.

Chapter 4

Functional Requirements

4.1 Overview

The functional requirements describe the key features and capabilities that the **AI Powered Image Transformation Tool** must provide. This includes functionalities for image upload, style selection, image transformation, and user account management.

4.2 Functional Requirements

4.2.1 User Interface

The application will provide a user-friendly interface that includes:

- **Image Upload:** JPEG and PNG formats of images are allowed to be uploaded by the users.
- **Style Selection:** One was able to choose between the available art periods (Van Gogh, Picasso etc.) that they wish to use on their pictures.

- **Preview Display:** This will be in real time where the user will get a preview of what the transformed image will look like before downloading.
- **Download Image:** In the case of the stylized image, the users are able to download it in any format accessible.

4.2.2 Style Transfer

This core feature, using AI models, overlays an art style to the uploaded image. Main functions include:

- **Apply Style:** The style transfer will be done using AI models based on TensorFlow.
- **Multiple Style Options:** Multiple architectural types will be allowed by pre-trained models.
- **Real-Time Processing:** Make sure the style transfer has been done within 5 seconds.

4.2.3 User Management

The application will support optional user accounts to enhance user experience:

- **Create an Account:** They can also choose transformation history to register and log in to transformation management.
- **View Image History:** Previously transformed Images can be viewed by the users.
- **Account Management:** Users are also allowed to edit personal data like password or email, if needed.

4.2.4 Image Processing and Format Support

The system should support various image formats for upload and download:

- **Supported Upload Formats:** JPEG, PNG.
- **Downloadable Formats:** The processed images can be downloaded in JPEG and PNG format.

4.2.5 Performance Requirements

To ensure efficient performance, the system will support:

- **Simultaneous Users:** The intended user should be able to work within the context of the application simultaneously with up to 100 users.
- **Processing Time:** Every single style transfer should take ideally less than 5 seconds.

Chapter 5

Non-Functional Requirements

5.1 Performance Requirements

The **AI Powered Image Transformation Tool** performance needs to be optimal for the user engagements to run their course without hitches. Key performance requirements include:

- **Response Time:** The style transfer should not take more than 5 seconds since there will be multiple images in this application.
- **Concurrent Users:** The disk space and the other resources required should provide response time that is not affected by up to 100 simultaneous users.
- **Throughput:** It should be also possible to perform at least 500 transformations per hour.

5.2 Usability Requirements

The application should be easy to use, even for users with minimal technical knowledge:

- **Intuitive Interface:** The user interaction layer should be simple and the users should be able to input pictures, decide on the styles and download the results.
- **Accessibility:** All current and future layouts should be compatible with the Assistive-technologies so that those with disabilities would be supported.
- **User Documentation:** Gladly include a short manual filled with tips or a screen prompt describing the most basic functions of the application.

5.3 Reliability Requirements

Reliability is crucial to provide consistent user experience:

- **System Uptime:** The main requirement here is the application should be available at 99.5% of the time.
- **Error Handling:** The system should also gracefully fail and include user friendly messages that can be logged and reviewed for developer analysis.
- **Data Backup and Recovery:** Users' call data and transformations should be processed every night and synchronized in case of a breakdown.

5.4 Security Requirements

Security measures are essential to protect user data and ensure safe interactions:

- **Data Transmission Security:** Secure the communication process by using HTTPS protocol, to protect the uploaded images and users data.
- **Authentication:** Use of strong passwords for user accounts and any method of log in and authentication.
- **Data Privacy:** Ensure adherence to the provisions of data protection laws especially as regard to stored image and user data.

5.5 Scalability Requirements

The system should be scalable to accommodate future growth:

- **Horizontal Scalability:** The application should be able to handle an increase in the amount of traffic in the system horizontally.
- **Database Scalability:** The database should be able to support a large amount of user data as well as transformation histories without affecting performance decisively.
- **Modularity:** The system should be logically decentralized and designed in a way that new feature additions should not seriously affect pre-existing features.

5.6 Maintainability Requirements

To ensure the system can be efficiently updated and maintained:

- **Code Modularity:** This should be done by following modular programming standards of work in order to be able to update and debug easily.
- **Documentation:** It is good practice to update the code and system documentation in helping developers with future maintenance work.
- **Error Logging:** Make use of use error logs in order to keep track of some problems and thereby easily solve system troubles.

Chapter 6

External Interface Requirements

6.1 User Interfaces

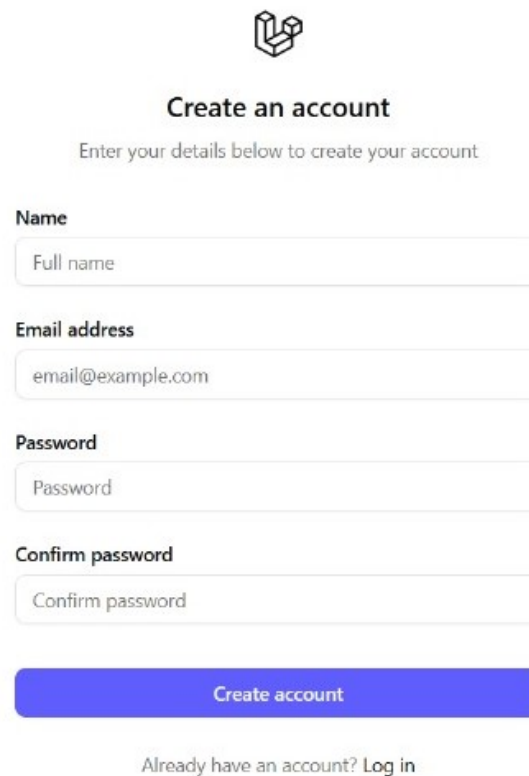
The **AI Powered Image Transformation Tool** will feature a clean simple user interface, along with UI that is quick to respond to so that it can be easy to use for every person. Key requirements include:


- **Image Upload Interface:** Allows users to upload images in common formats such as JPEG and PNG. The interface is basically drag and drop and there will be a button beside it for navigating through the files.
- **Style Selection Interface:** A dropdown or gallery view will display available artistic styles for users to choose from.
- **Image Preview Interface:** After style selection, a preview section will display the transformed image in real-time.

- **Download Interface:** A download button will allow users to save the stylized image to their device in various formats.
- **User Account Interface (Optional):** Users can register, log in, and manage their profile information through a simple account interface.

6.2 Front End

6.3 Sign up screen





Create an account

Enter your details below to create your account

Name

Email address

Password

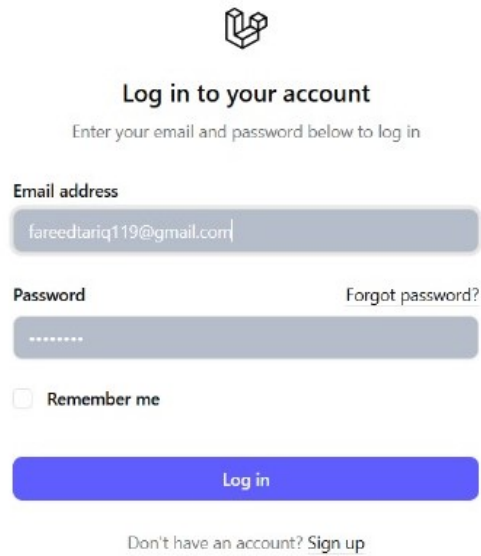
Confirm password

[Create account](#)

Already have an account? [Log in](#)

FIGURE 6.1: $GUI_{Diagram}$

6.4 Login screen



The diagram illustrates a login screen layout. At the top center is a logo consisting of three interlocking cubes. Below the logo is the heading "Log in to your account" in bold, followed by the instruction "Enter your email and password below to log in". The form contains two input fields: "Email address" with the text "fareedtariq119@gmail.com" and "Password" with masked characters "*****". To the right of the password field is a link "Forgot password?". Below the password field is a checkbox labeled "Remember me". At the bottom is a blue "Log in" button. Below the button is a link "Don't have an account? Sign up".

FIGURE 6.2: $GUI_{Diagram}$

6.5 Dashboard

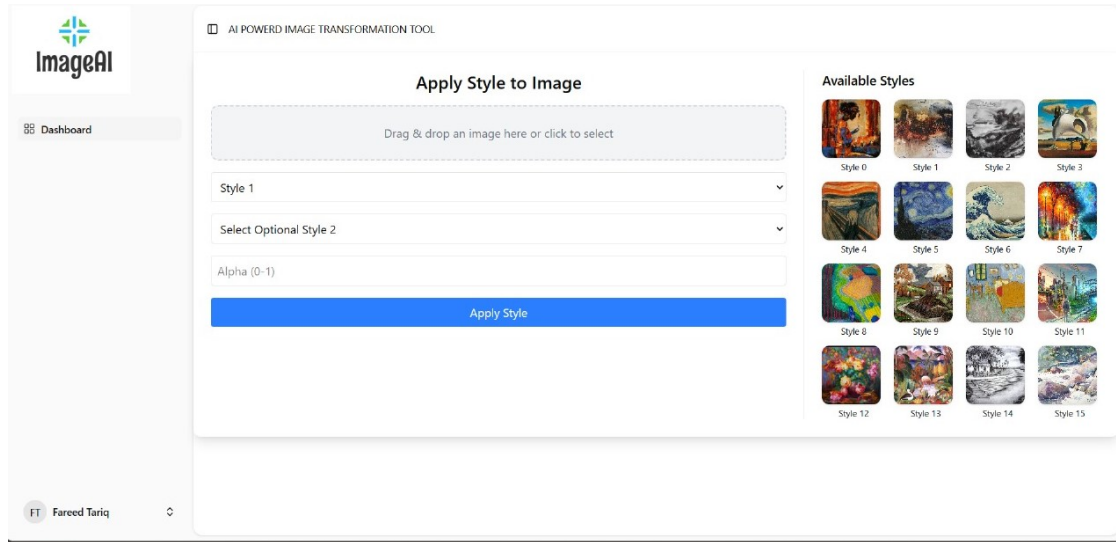


FIGURE 6.3: $GUI_{Diagram}$

6.6 Account Setting

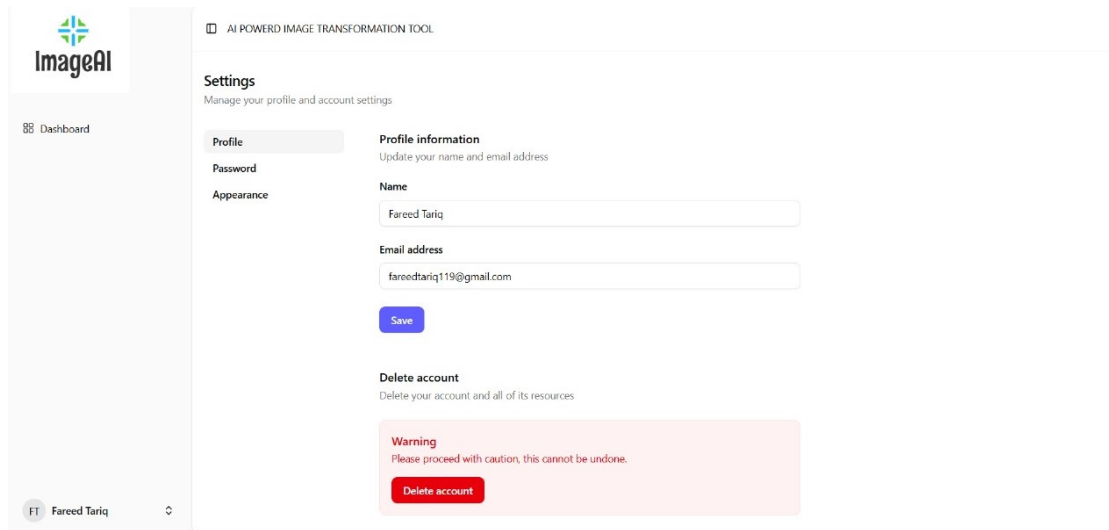


FIGURE 6.4: $GUI_{Diagram}$

6.7 Result

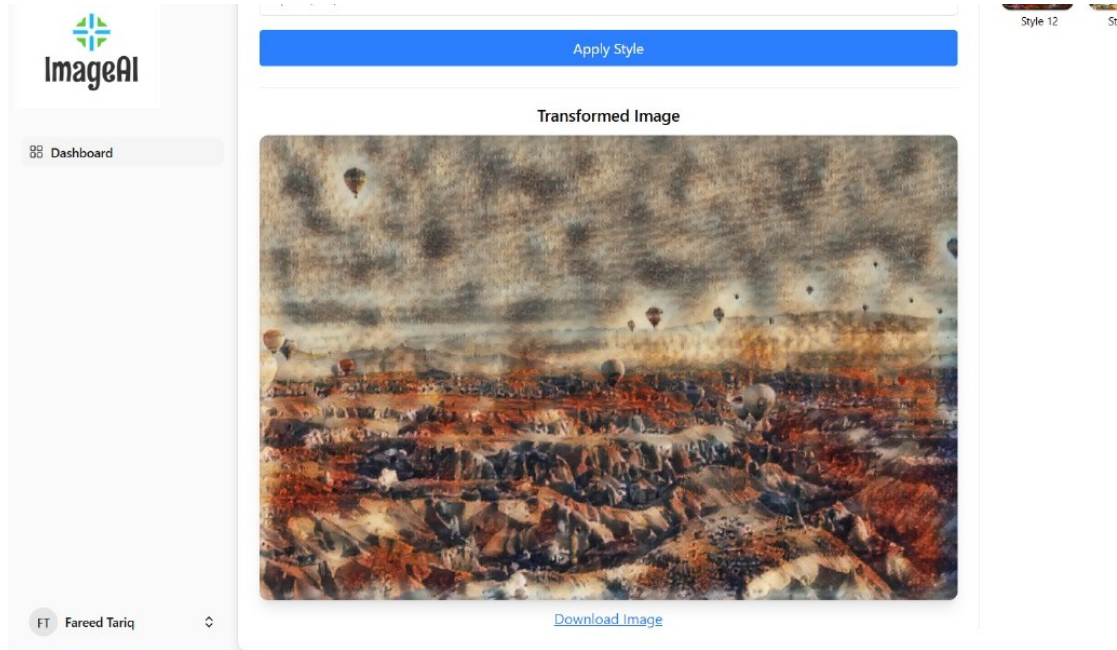


FIGURE 6.5: $GUI_{Diagram}$

6.8 Hardware Interfaces

The application will interact with client and server hardware as follows:

- **Client Device Requirements:** The application requires a device (e.g., desktop, tablet, smartphone) with an internet connection and a modern web browser.
- **Server Hosting Requirements:** The application will be hosted on cloud platforms such as AWS or Heroku, utilizing the platform's computing resources for AI model processing, storage, and scalability.

6.9 Software Interfaces

The application will interface with various software components and libraries:

- **Frontend Framework:** The user interface will be developed using React.js, and uses TypeScript Language which enables a responsive and dynamic user experience.
- **Backend Framework:** The backend will be Laravel, which will manage user requests, connect with the database, and handle AI model interactions.
- **AI Framework:** TensorFlow will be used for implementing the style transfer models. Pre-trained models will be loaded and executed to perform the artistic transformations.
- **Database Interface:** MySql will be used to store user data, transformation history, and account information.

6.10 Communication Interfaces

The system will use secure communication protocols to ensure data integrity and security:

Chapter 7

Design Description

This chapter outlines the design implications of the proposed **AI Powered Image Transformation Tool**. It encompasses various viewpoints, including

7.1 Entity Relationship Diagram

This **Entity-Relationship Diagram (ERD)** represents the database structure of the *image processing application*, where users upload images and apply AI-based styles. The **User** entity stores login credentials and personal details. Each **Image** uploaded by a user is stored with metadata such as format, file path, and size. The **Style** entity defines different artistic styles available for transformation. The **Transformation** entity records which style is applied to which image, along with a timestamp. This structured database ensures smooth tracking of user uploads, applied transformation, and processed results.

7.2 ERD Diagram

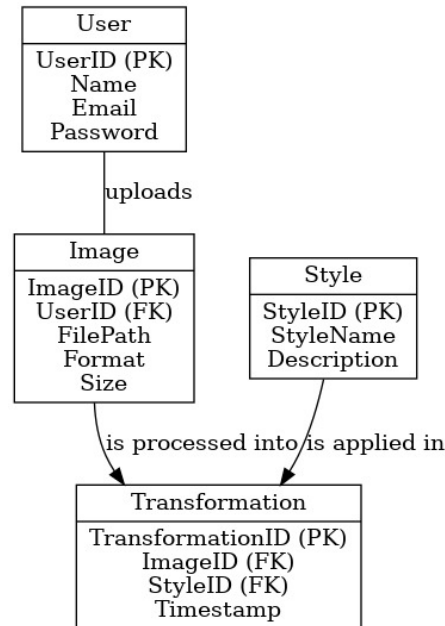


FIGURE 7.1: ERD Diagram for Requirement Collection Techniques

7.3 Activity Diagram

This **Activity Diagram** represents the step-by-step process of the *image processing application*. It starts with the user logging in or registering, followed by uploading an image. The user then selects an **artistic style** to apply to the image. The system processes the image using AI, and the transformed image is previewed. Finally, the user can download the **stylized image**, completing the workflow.

7.4 Activity Diagram

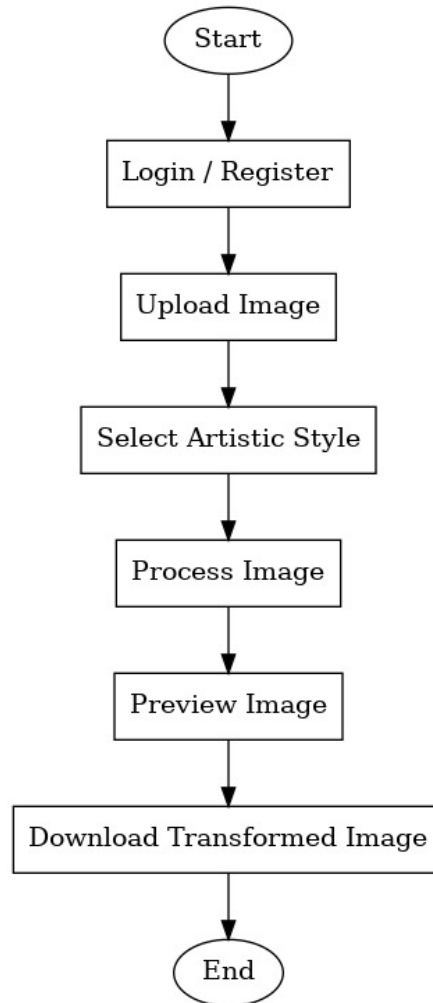


FIGURE 7.2: Activity Diagram for Requirement Collection Techniques

Class Diagram Explanation

This **Class Diagram** represents the structure of the *image processing application*, showing relationships between different classes.

- The **User** class contains attributes like `userID`, `username`, `password`, and `email`, with functions for login, registration, and account management. Users upload images.

- The **Image** class stores `imageID`, `file path`, and `format`, and has functions to upload, process, and download images.
- The **Style** class defines `styleID` and `style name`, with a function to apply styles to images.
- The **System** class manages concurrent users, validates images, processes them, and logs errors.
- The **User** interacts with the **System**, which manages processing and ensures smooth execution. Images are uploaded and styled using AI.

7.5 Class Diagram

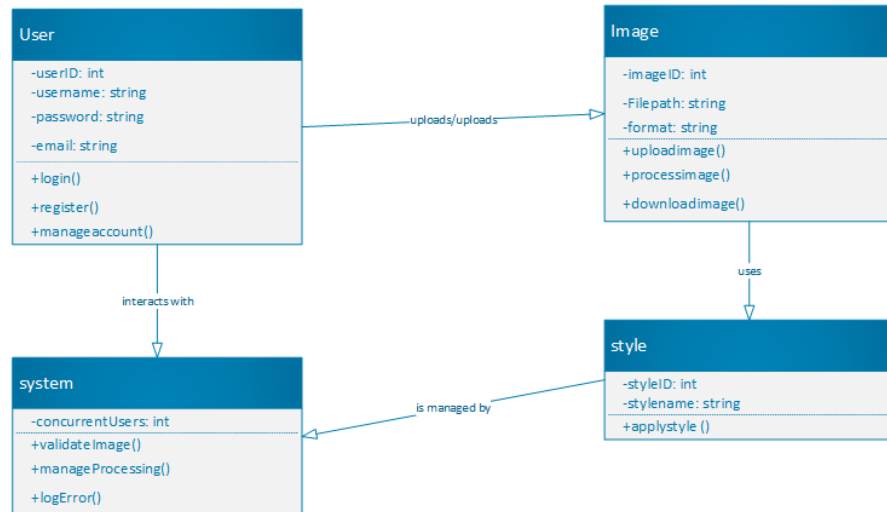


FIGURE 7.3: Class Diagram for Requirement Collection Techniques

7.6 Sequence Diagram

This sequence diagram represents the workflow of the **image processing application**, where users interact with the system to upload images, apply AI-based styles, and download the processed results. The process begins when the *user* submits login credentials via the *frontend UI*. The frontend forwards these details to the *backend system*, which authenticates the user by retrieving relevant data from the *database*. Once authentication is successful, the user can proceed to upload an image.

After receiving the image, the frontend sends it to the backend, which processes it by communicating with the *AI model*. The AI model applies the selected style and returns the stylized image to the backend. The backend then forwards the processed image to the frontend, where it is displayed for preview. Finally, the user can download the transformed image. This system ensures seamless interaction between users, the frontend, backend, AI processing, and database storage.

7.7 Sequence Diagram

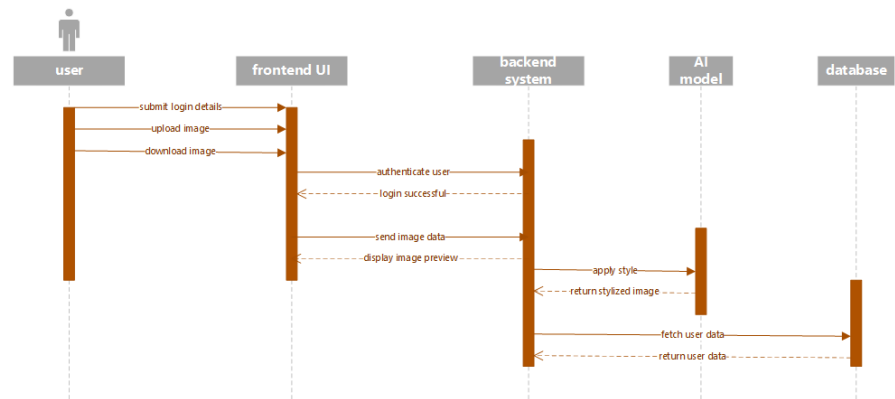


FIGURE 7.4: Sequence Diagram for Requirement Collection Techniques

7.8 Data Flow Diagram

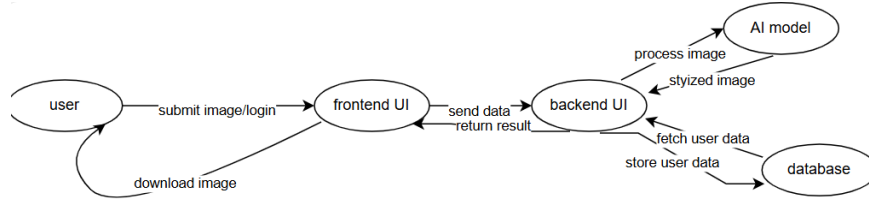


FIGURE 7.5: Data Fow Diagram for Requirement Collection Techniques level 1

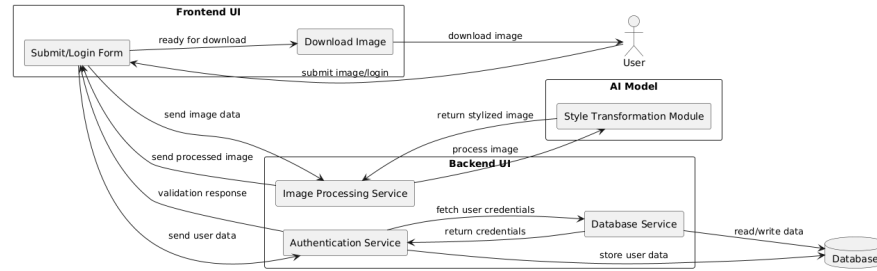


FIGURE 7.6: Data Fow Diagram for Requirement Collection Techniques level 2

7.9 State Transition Diagram



FIGURE 7.7: State Transition Diagram for Requirement Collection Techniques

7.10 Algorithm Viewpoint for Image Style Transfer System

– Library and Framework Setup

- * Import essential libraries such as **Torch**, **FastAPI**, **PIL** for deep learning and image processing.
- * Utilize pre-trained models for feature extraction and style transfer.

– Data Acquisition

- * Accept image uploads in formats like PNG, JPEG.

- * Validate image format and convert to a standardized tensor format.

– **Data Preprocessing**

- * Resize images to a fixed shape for uniform processing.
- * Normalize pixel values to improve model accuracy and contrast handling.

– **Style Selection & Feature Extraction**

- * Allow users to choose a single or blended style.
- * Extract Gram Matrices from selected styles for neural transformation.

– **Style Transfer Execution**

- * **Single Style**

- Apply a single transformation model to transfer the selected style.

- * **Blended Styles**

1. Generate transformed outputs for both styles.
2. Combine the transformations using a weighted approach.
3. Convert the output tensor back to an image format.

– **Model Evaluation & Validation**

- * Evaluate model performance using content loss, style loss, and perceptual loss.
- * Perform cross-validation on separate datasets to ensure generalizability.

– **API Implementation (FastAPI)**

- * Define endpoint `/apply-style/` for handling image processing requests.
- * Process images using the `Inferencer.eval_image()` method.
- * Return the stylized image as output.

– **Visualization & Reporting**

- * Generate real-time previews of the stylized output.
- * Provide execution reports including time, applied styles, and loss values.

– **Deployment & Continuous Learning**

- * Deploy the model as an API for seamless integration with client applications.
- * Collect user feedback to fine-tune and improve style transfer results.

7.11 Pseudocode

The Pseudo code for the proposed system:

Pseudocode for Neural Style Transfer API]

BEGIN

Step 1: Initialization

Import necessary libraries (FastAPI, Torch, PIL, etc.)

Initialize FastAPI application

Set `device` to "cuda" if GPU is available, else use "cpu"

Step 2: Model Preparation

Define model parameters:

`num_styles = 16`

```
image_size = 512
```

```
model_dir = "style16/"
```

Load pre-trained model using Torch and Inferencer

Step 3: Image Processing

Define FastAPI endpoint for image processing:

if User uploads an image **then**

 Convert image to tensor

 Apply neural style transfer using the loaded model

 Return stylized image

end if

Step 4: API Deployment

Run FastAPI server on a specified host and port

END

Bibliography

Books

- [1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. A foundational text on deep learning, relevant to TensorFlow-based AI model implementation in your project.
- [2] Flanagan, D. (2020). *JavaScript: The Definitive Guide*. O'Reilly Media. Useful for understanding React.js, the front-end framework used in your tool.

Journal Articles

- [3] Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423. This is the seminal paper on style transfer, directly applicable to your tool.
- [4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. Describes ResNet architectures, which may be part of pre-trained models used in your application.

Online Resources

- [5] TensorFlow. (n.d.). *TensorFlow Documentation*. Retrieved from <https://www.tensorflow.org> Official documentation for TensorFlow, the framework you are using for style transfer models.
- [6] React.js. (n.d.). *React: A JavaScript Library for Building User Interfaces*. Retrieved from <https://reactjs.org> Documentation for React.js, the frontend framework.
- [7] Amazon Web Services (AWS). (n.d.). *AWS Documentation*. Retrieved from <https://aws.amazon.com/documentation> For hosting and scaling your application.