ARTficial – Neural Style based image manipulation

Final Year Project – Mid Report

Session 2019-2023

A 4th Year Student

A project submitted in partial fulfilment of the
COMSATS University Degree
of
BSc. (Hons.) BS in Computer Science / Software Engineering (CUI)

Department of Computer Science

COMSATS University Islamabad, Lahore Campus

Project Detail

Type (Nature of project)	[ ] **D**evelopment [ ] **R**esearch [✓] **R&D**			
Area of specialization	Artificial Intelligence and Computer Vision			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	FA19-BCS-016	Abdul Rehman Aziz	Fa19-bcs-016@cuilahore.edu.pk	AbdulRehman
(ii)	FA19-BCS-061	Muhammad Daniyal Javed	Fa19-bcs-061@cuilahore.edu.pk	Daniyal

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I am **Abdul Rehman Aziz** S/D/o **M. Anjum Naveed**, group leader of FYP under registration no CIIT/ **FA19-BCS-016**//LHR at Computer Science Department, COMSATS University Islamabad, Lahore Campus. I declare that my FYP proposal is checked by my supervisor and the similarity index is ________% that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: __1/4/2023__ Name of Group Leader: **Abdul Rehman Aziz** Signature: AbdulRehman

Name of Supervisor: **Dr. Aksam Iftikhar** Co-Supervisor (if any) : **Dr. Zeeshan Gillani**

Designation: Assistant Professor Designation: Assistant Professor

Signature: ______________________ Signature: ______________________

Abstract:

Our project, ARTficial, is a Neural Style Transfer based Image manipulation and generation platform for creating artworks. The art and design industry are one of the most time inefficient industries because it takes numerous hours to create even simple artworks. This leads to artists compromising over their creativity in order to meet deadlines for their projects. In addition to that, they are undercompensated for hours they put in. Furthermore, individuals with non-artistic background have hard time getting into the art and design industry because of their less prior experience and talent. Our project will solve these problems by aiding in generating artworks in a timelier manner using neural-networks techniques. This way, even non-artistic people can create amazing artworks with ease and confidence. Moreover, the blockchain industry is heavily influenced by NFTs (Non-Fungible Tokens), which are basically ERC-721 tokens created on top of Ethereum blockchain. Once, an NFT is created it cannot be altered since they are non-fungible, so artists prefer to create and sell their artwork in the form of NFTs. Our project will aid NFT artists by assisting them in cases such as when they want their artwork to be influenced by certain famous artist, they can use our Neural Style Transfer-based technology to create their artwork accordingly. This project will also allow users to play around with their custom images in different themes for promotional or personal usage. So, ARTficial will also be used as an assistive technology for artists in the development cycle of their artwork that will help them in their creative process.

Table of Content

Table of Figures

List of Tables

CH 01

1 Introduction:

1.1 Introduction:

The use of artificial intelligence is prevalent in almost all he fields and aspects of our lives. It is being used in medicine, finance, accounting, agriculture and several other fields. But the most exiting area where Artificial Intelligence has gained a lot of immersion is the arts industry. AI in arts is a growing area of research and development that explores the use of artificial intelligence to create or assist in the creation of art. This includes a wide range of applications, such as using machine learning algorithms to generate visual art, music, or literature, as well as using AI to help artists with tasks such as image manipulation and styling.

So, our project is regarding the use of Artificial Intelligence based technique, Neural Style Transfer, for image manipulation. This technique is still in research phases and there aren't much platforms developed to facilitate users to utilize this technique for artwork generation and manipulation. So, our platform will enable artists and other users to generate their artworks using Neural Style Transfer technique.

1.2 Objectives:

- To create an application that uses Artificial Intelligence based technique, Neural Style Transfer, for manipulation and generation of artworks.
- This is still an emerging field and there isn't much platform build to facilitate AI based Image manipulation and artworks generating. This platform will be really helpful for artists and other users to generate artworks of their desired style
- The key benefits of using AI for image manipulation is that it can save artists and photographers time and effort to manually stylize or edit the images.
- This platform will automatically stylize a large number of photos in a fraction of the time it would take a human to do the same task manually. This will be very useful for professional photographers who need to quickly and efficiently process large volumes of images.
- Users will be able to store their generated artworks in the galleries. These galleries can be edited, deleted and renamed. This will allow the users to easily manage and organize their created artworks.
- Users can share their created artworks and galleries on social media platforms i.e., Twitter, WhatsApp and Facebook.
- Users can view trending artworks generated through the application for inspiration purpose.

1.3 Problem statement

Artificial intelligence has outreached to all the fields including finance, healthcare, music generation and self-driving cars. But the most exciting application of artificial intelligence is in the art generation field. But still, most of the work being done in this area is in research phase, there aren't much applications developed on top of Artificial Intelligence based research publications. We keep hearing about the wonder of Artificial Intelligence and how it will empower all segments of the society in their day-to-day tasks.

But the issue in this regard is, these AI based technologies are either not accessible or are limited to very few people who know how to operate them. This is primarily due to the fact that AI is not as easy to embed in our day chores, since one needs to be technically aware to utilize these emerging technologies. Secondly, there aren't much applications built on top of these research works that allows general public to make use of these emerging technologies for their benefit and to facilitate them the aid of Artificial intelligence in their day-to-day tasks.

Specifically in the field of Neural Style Transfer there aren't much platforms that exists for the artists to generate or manipulate their artworks. So, the aim of our project is the develop an easy-to-use application that is built on top of Neural Style Transfer technique that will enable all users to this technology for AI based artwork generation, irrespective of their expertise in Artificial Intelligence and Computer Vision.

1.4 Assumptions & constraints:

Assumptions:

- The user must have a computer or smartphone device to use our platform.
- The user must have a web browser that can run the web application.
- The user must have an active internet connection while the processing of model to generate the stylized image.

Constraints:

- The user should be registered on the platform to further use it.
- The user must provide relevant images to generate a stylized image.
- The user must provide relevant images in JPG or PNG format.
- The platform should use the NST model to generate images rather than primitive computer vision techniques.

1.5 Project scope (what and what not to consider):

The scope of this project is to develop a neural style transfer-based image manipulation platform that allows users to apply the visual style of one image to another image. The software will utilize a machine learning algorithm to understand the style of an image and how it can be transferred to another image.

The platform will allow users to select a "style" image and a "content" image, and the algorithm will analyze both images to create a new, merged image that combines the style of the style image with the content of the content image.

The software will have additional functionalities such as storing the generated artworks in a separate collection in gallery so that they can share their collection with others. The platform will allow users to generate images by specifying its textual description, and the application will generate the relevant image which could be used as a content image.

This application has a number of potential use cases in the industries of art, design, advertisement and entertainment. It can be used to create unique and visually striking images for use in advertisement, social media and influencer presence, and visually enticing scenery in gaming industry. It can also be utilized by artists and designers as a tool to experiment with different nuances and generate new forms of art.

Overall, the goal of this project is to create a platform for artists that utilizes neural style transfer technology to allow users to apply the visual style of one image to another with minimal effort.

CH 02

2 Requirements Analysis

2.1 Literature review / Existing system study

A lot of work has been done in the domain of style transfer in the last few decades. Researchers have proposed different architectures which have had their pros and cons. One of the examples may include that most of the ideas proposed for image-to-image translation used paired images for training of the generator in the GAN-based architectures. This means for each object as an image, it must have its image taken in one setting, and other in another setting. This helps the generator learn the idea for stylizing. But this means the gathering of dataset would be quite gruesome and hectic. Cycle-GANs or Cycle-Consistent GANs solved this problem by proposing two GAN architecture. It allows training of unpaired images thus, allowing more flexibility in the data gathering process. But Cycle-GAN is more focused on translating images from one domain to another instead of gearing towards transferring style from one image to another.

Thus, there has always been preceding architectures being improved as well as new architectures being proposed for this task of style transfer. Neural transfer is a technique that is used for transferring the style of one image to the content of another image, while also maintaining the content of the original image. We do this by training a convolutional neural network that maximizes the function of transferring content of the content image to a noise image, and the function of transferring style of the style image to that same image. Following is some of the research papers that discuss various techniques and applications related to Neural Style Transfer.

1. **Image Style Transfer with Generative Adversarial Networks [1]:**

Previous techniques of style transfer using GANs have come a long way. Different types of GANs have their advantages, per se, Cycle-GAN can work without paired images, thus making the process of training exponentially convenient particularly in case of gathering dataset. Generative Adversarial Networks (GAN) have showed their great potential of translating images from one domain to another, in absence of paired images. But this approach cannot guarantee the expected results all the time. These existing style transfer techniques using GANs often times fail to capture all the design patterns like the structure and geometry that demonstrates the quality of the output image. This research deals with investigation of these image style transfer mechanisms, in order to design methods to solve the discussed the problems regarding them. Through the authors experiments and hyper-parameter tuning, they have showed their further insights into these approaches, and the directions they may pursue in the future.

2. **Neural Style Transfer: A Critical Review [2]:**

The Neural Style Transfer has been proven to be the best technique for style transfer beating the style GANs but still has a lot of room for improvement. The author of this research paper have discussed how the Neural Style Transfer have impacted multiple forms of media, inclusive to transformation of scenes and manipulating environments, particularly in videos, images and even game development. These use cases of NST have made it popular in the entertainment industry as this has made the hectic process of stylization immensely convenient for the designers. This paper deals with the comparison of different architecture implementations of NST, along with the limitations and advantages of using them. Along with that, the authors have looked at the results of utilizing NST for image and video style transfer, particularly using mobile devices. The problems faced using NST in real-time video and image generation were all reviewed by the authors in this paper.

3. **Rethinking and Improving the Robustness of Image Style Transfer [3]:**

This research papers explains a simple yet affective method to generalise the implementation of neural style transfer using other architectural networks, particularly the Resnet family such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. The standardized way of implementing the NST is using the VGG for the development of convolution network, as proposed and pioneered by Leon Gatys et al but using the same approach and replacing the VGG for ResNet, the result is not very robust because the feature maps produced in ResNet are of small entropy, which does not work well for the style transfer part of the NST. Thus, this paper has improved the result for style transfer in case of ResNet by using a softmax transformation activation function of the feature maps at each layer that improves their entropy and thus improves the result of style transfer in ResNet.

4. **Neural Style Transfer for image within images and conditional GANs for destylization [4]:**

This research paper shows another application of Neural Style Transfer: Steganography. The author proposes the idea of hiding the secret image into cover image. The underlying mechanism is to take an input of secret image, which will be the image that the user may want hide or encrypt, and another input image of the cover image, which will be used to the hide the secret image. The general gist of it is very similar to that of working of a norm NST. The content from the secret image, will be hidden using the style from the cover image. Then for future when the user needs to reshow the image and decrypt the secret image, the author proposes the idea of destylization using Conditional GANS using Residual in Residual Dense Blocks (RRDBS).

5. **Comic style transfer based on generative confrontation network [5]:**

In the modern times, utilizing the deep learning methods for style transfer is trending in the domain of image research. Many different researchers have proposed their own architectures and techniques to accomplish this task of transferring style from a particular style image prototype, and applying these extracted stylized properties to the input image. This research also proposes a GAN-based solution for style transferring. The author suggests using a cyclic consistent GAN, which can also be interpreted as cyclic consistent confrontation network, and utilizing a dense CNN. The author claims that the test results demonstrate that style is more prominent using this architecture, and the quality and visual properties of the image have enhanced.

6. **Rethinking Style Transfer: From Pixels to Parameterized Brushstrokes [6]:**

In this research paper, the authors have dealt with a problem in the standard NST that is the limited stylization of pixels. Generally, the artistic property of the artworks is dependant more on the flow of brushstrokes instead of pixels. This kind of representation in which the stylization is occurring through the brushstrokes instead of individual pixels feels more natural. Thus the authors have proposed a method for the stylization part of the NST in which they have optimized parameterized brushstrokes instead of pixels and furthermore introduced a rendering method.

7. Improving the Latent Space of Image Style Transfer [7]:

The standard Neural Style Transfer has a lot of arbitrariness around it in context of final output. The style and content extraction by the VGG network may not be consistent with the visual style we visualize. Such a case would be when the style distance between images of different styles may be quite less than that of the same style. So, the objective function may behave wrongly in this case for which the author has proposed the idea of two training schemes. These training schemes are contrastive in which one is style contrastive and the other content contrastive. They work in such way that the style contrastive loss pulls the stylized result closer to the same visual style image and pushes it away from content image. Whereas the content contrastive loss enables the encoder to retain more available details.

We will construct a custom Neural Style Transfer that will be trained on a dataset different from ImageNet dataset. In addition to that, we will utilize a custom architecture instead of the standard VGG-19. The standard proposed makes use of the layer 4_2 (second part of the fourth convolution layer), for the extraction of content, whereas we may utilize a different layer. Lastly, we will make use of the weights that maximises or minimises the content and style affect, for the hyper-parameter tuning of our model.

Existing Applications:

Apart from the research side, there are only few applications that have made use of the Neural Style Transfer technique as their underlying core technique for image manipulation as per our knowledge. This is because the Neural Style Transfer technique is still in improvement. Researchers have pointed out the arbitrary and unexpected results that may not be very useful in the production for the fast-paced world today. However, some of the applications that have made custom NST for their specific use cases are as followed:

1) Adobe Photoshop [8]:

The Adobe creative cloud has a number of applications to offer to their users for different purpose. Per se, Adobe Illustrator is used by cartoonists and designers in order to create aesthetic designs and comics conveniently. In addition to that, Adobe Reader is utilized by millions of users for their daily activities, such as reading books, study articles and many more activities related to the documents. Adobe Photoshop is another one of the applications by Adobe which is used for image and video enhancement. This tool is used by professional editors to rework and modify media creatively. Adobe photoshop offers a feature called “Style Transfer” as a standalone feature to their users, that allows users to transfer the style of one image to another.

2) Prisma [9]:

Prisma is one of the famous applications that utilizes the Neural Style Transfer as their core algorithm for stylization. It allows users to play around with static filters, that once selected are applied to the user input image. However, Prisma does not allow users to input their own style reference image. In addition to that, Prisma is a paid application, which restricts and confines users to a limited filter.

3) DeepArt [10]:

DeepArt is another one of the web-applications that had made Neural Style Transfer with the combination of another algorithm as their core underlying technique. In addition to that, DeepArt allowed users to enhance their images using basic image manipulation techniques such grey scaling, custom orientations and cropping as well as other color-related enhancements. However, DeepArt have halted operations and stopped offering their services.

None of these applications provide a social platform, where users can share, aspire, and create artworks as a community. Our application would be a user-driven platform which would utilize a custom NST, with more than just artistic filters. In addition to that, we will allow users to input their style references, generate their custom content image. The possibilities for creation of artworks would be endless and the only limitations would be the user's imaginations. Each and every service of our application would be cost-free.

2.2 Stakeholders list (Actors):

1) Project Supervisors

The complete project will be developed and architected under the supervisor of Project supervisors. They are the most crucial stakeholder in the development life cycle of this project.

2) Developers

The complete development circle of this project will be done by developers under the supervision of Project supervisors.

3) Users/Artists

As this project is all about artworks generation and collection. The primary stake holder of this project will be the artists registered of the platform who will be using the Neural Style Transfer technique to generate their artworks.

4) Admin

The admins' role will be critical in the checking and overall monitoring of the application and all the content available on it. He will ensure the best user experience by entertaining the user requests and queries regarding different aspects.

2.3 Requirements elicitation

2.3.1 Functional Requirements

FR01: Signup

Req. No. Functional Requirements

FR01-01 The application allows users to sign up.

FR01-02 User shall add all necessary details for signup including email, password, name and phone number

FR01-03 User's credentials are maintained in the database.

FR01-04 The application allows users to sign up using OAUTH (login via 3rd Party accounts like google, Instagram etc.,)

FR02: Login

Req. No. Functional Requirements

FR02-01 The user shall be able to login.

FR02-02 Only users with correct email and password can login

FR02-03 After logging in, the user can access the system

FR02-04 The application allows users to Log in using OAUTH (login via 3rd Party accounts like google, Instagram etc.,)

FR03: User Driven Artwork Feed

Req. No. Functional Requirements

FR03-01 User will be able to explore different artworks created by other inspiring artists using the platform conveniently.

FR03-02 The user will be able to filter out and search by different artwork categories, artist styles and era.

FR03-03 The user will be able to view the details page of a particular artwork.

FR04: AI based Artwork generation and manipulation.

Req. No. Functional Requirements

FR04-01 User will be able to upload content image for styling purpose

FR04-02 User has the choice to upload style reference as well.

FR04-03 The style from the style reference (if provided) will be merged to the content (user input) image and the resultant generated image will be available for the user to be further manipulated.

FR05: Users Artwork Collection (Galleries)

Req. No. Functional Requirements

FR05-01 User will be able to view their overtime generated artworks and edit/delete their collection of artworks.

FR05-02 User will be able to review their favorited artworks that they starred generated by different users/artists.

FR06: Trending Artworks

Req. No. Functional Requirements

FR06-01 The Users will be able to view the trending artworks page by clicking the respective linked icon.

FR06-02 The users will be able to view the details page of specific trending artwork by clicking that artwork.

FR07: Users Profile

Req. No. Functional Requirements

FR07-01 Each user will have a profile page where they can manage their profile information.

FR07-02 Users will be able to showcase their portfolio of created artworks through the platform.

FR08: Sharing the Artworks

Req. No. Functional Requirements

FR08-01 The Users will be able to share any specific artwork page by clicking the share icon button.

FR08-02 The shared artworks shared link will be available for social media platforms like Twitter, WhatsApp and Facebook.

FR09: Admin Panel

Req. No. Functional Requirements

FR09-01 Admin/Moderator will be able to monitor different analytics and statistics of users on their application.

2.3.2 Non-Functional Requirements

R01: Accuracy of Model

Req. No. Non-Functional Requirements

FR01-01 The accuracy of the images generated through the NST model will be ensured while the training of the model to the best possibility. The model generated image would be the gratifying amalgamation of content and style images.

R02: Secure Authentication Process

Req. No. Non-Functional Requirements

FR02-01 The Security of the users will be protected to its best by using Secure encryption methods for storing users' credentials in the database. The passwords will first be hashed and then they will be stored on the database, to secure them from malicious attacks.

R03: Ease of use

Req. No. Non-Functional Requirements

FR03-01 The User Interface and User Experience of the application will be designed considering the design principles like consistency, perceptibility, simplicity and discoverability for making it easier for the user to explore the application.

R04: Application Deployment

Req. No. Non-Functional Requirements

FR04-01 The application will be deployed on the reliable and efficient cloud-based deployment platforms (i.e., Heroku, AWS, Azure). These platforms are most trusted service providers globally for their deployment of applications.

R05: Improvisations based on Users Feedback

Req. No. Non-Functional Requirements

FR05-01 Users will be able to share their experience/feedback with the administrators, and based upon their feedback the admin will moderate the application. This will ensure the better and reliable user experience.

R06: Scalability

Req. No. Non-Functional Requirements

FR06-01 The application will be scalable as it starts to attract large number of users, like the database which will initially be used of free tier but as the application grows, a premium membership of cloud database will be used.

R07: Rate Limiting

Req. No. Non-Functional Requirements

FR07-01 The NST model will be deployed via an API which will be protected for DDoS attacks by using the rate limiting technique for protecting the API. This way, users can only send limited number of requests to the API for NST implementation from the application.

R08: Responsiveness

Req. No. Non-Functional Requirements

FR08-01 The application will be completely responsive. That means it can be viewed on the mobile devices as well. This will give our users a better interactive experience with the application.

R09: Dynamic Image Handling

Req. No. Non-Functional Requirements

FR09-01 The input images will be dynamically handled. That means that users can provide input images of any resolution and format to our application and the model will process images of all resolutions.

2.3.3 Requirement traceability Matrix:

Test Case-ID	Functional Requirements-ID	Description of Requirement	Objective	Priority
01	FR-01	This requirement allows users to Sign up.	Users Registrations	High
02	FR-02	This requirement allows users to Login.	User Login	High
03	FR-03	This requirement allows users to explore different artworks.	Exploration of artworks	High
04	FR-04	This requirement allows users to generate artworks.	Artwork generation	High
05	FR-05	This requirement allows users to create collections for artworks	Galleries/Collections creation	High
07	FR-06	This requirement allows users to view tending artworks	To view trending artworks	Medium
08	FR-07	This requirement allows users to view their profile.	To view the user's profile	Medium
10	FR-08	This requirement allows users to share the artwork on social platforms.	To share artwork with others.	Medium
11	FR-09	This requirement allows admin to monitor the application	Monitoring the platform	High

2.4 Use case descriptions:

Table 1 Signup:

Use case ID: **Name:**	**UC-01** **Signup**

Actors:	**User**
Description:	**This use case allows users to Sign up to the application by registering them on the platform by providing required credentials.**
Pre-Conditions:	
The user has not registered on the platform.	
Post-Condition:	
Success:	The user will be successfully signed up.
Failure:	The user with given username already exists.
Main Flow:	
	1) The user reaches signup page by clicking signup button **2) The user fills all the mandatory fields** **3) The user clicks register button**
Alternate Flow:	
	1) The user with entered username already exists **2) The user has entered wrong text fields**

Table 2 Login:

Use case ID:	**UC-02**
Name:	**Login**
Actors:	**User**
Description:	**This use case allows users to Log in to the application by providing required credentials.**
Pre-Conditions:	
The user should first be registered on the platform.	
Post-Condition:	
Success:	The user will be successfully Logged in.
Failure:	The user with given username does not exists.
Main Flow:	
	1) The user reaches log in page by clicking log in button **2) The user fills all the mandatory fields** **3) The user clicks log in button**
Alternate Flow:	
	1) The user with entered username does not exist **2) The user has entered wrong text fields.**

Table 3 Artwork Feed:

Use case ID:	**UC-03**
Name:	**Artwork Feed**
Actors:	**User**
Description:	**This use case allows users to explore different artworks created by artists to the application by registering them on the platform by providing required credentials.**

Pre-Conditions:	
The user must be logged in.	
Post-Condition:	
Success:	The user will be able to explore Artwork feed page.
Failure:	The user cannot view this page unless he/she is logged in.
Main Flow:	
	1) The user signs up before continuing to view this page. **2) The user logs in if they are already a registered user.** **3) The user reaches Artwork feed page to explore artworks.**
Alternate Flow:	
	1) The user did not Register/Logged In to the application. **2) The user could not view Artwork feed page**

Table 4 AI based Artwork generation and manipulation:

Use case ID:	**UC-04**
Name:	**Artwork generation**
Actors:	**User**
Description:	**This use case allows users to Generate and Manipulate artworks using the Neural style transfer model.**
Pre-Conditions:	
The user must be registered on the platform.	
Post-Condition:	
Success:	The user will be able to successfully generate artwork of his choice.
Failure:	The user cannot generate artwork unless he is logged in.
Main Flow:	
	1) The user reaches create artwork page from home page **2) The user either provides content image or textual description of content image.** **3) The user either provide style image or selects a preset.** **3) The user clicks generate button.**
Alternate Flow:	
	1) The user couldn’t provide necessary images or textual input. **2) The user didn’t selected style preset.**

Table 5 User artwork Collection(galleries):

Use case ID:	**UC-05**
Name:	**Artwork Collections**
Actors:	**User**
Description:	**This use case allows users to have their collection of favorite artworks .**
Pre-Conditions:	
The user must be registered/logged in on the platform.	
Post-Condition:	

Success:	The user will be able to successfully generate artwork using the style filters/presets.
Failure:	The user cannot use style filters for artwork generation unless he is logged in or he provides required content image.
Main Flow:	
1) The user registers/logs in to the platform. **2) The user explores different artworks** **3) The user adds a certain artwork to his collection that he admires.**	
Alternate Flow:	
1) The user didn’t register or logged in to the platform. **2) The user could not add any artwork to his collection.**	

Table 6 Contemporary Trending Artwork

Use case ID:	**UC-06**
Name:	**Trending Artworks**
Actors:	**User**
Description:	**This use case allows users to view all the trending artworks.**
Pre-Conditions:	
The user must be registered/logged in on the platform.	
Post-Condition:	
Success:	The user will be able to view the trending artworks generated by artists overtime using the platform.
Failure:	The user cannot use style filters for artwork generation unless he is logged in or he provides required content image.
Main Flow:	
1) The user registers/logs in to the platform. **2) The user reaches Trending page from Feed page.**	
Alternate Flow:	
1) The user didn’t register or logged in to the platform. **2) The user could not view the trending artworks page.**	

Table 7 User Profile

Use case ID:	**UC-07**
Name:	**User Profile**
Actors:	**User**
Description:	**This use case allows users to view the Profile Pages of registered users.**
Pre-Conditions:	
The user must be registered/logged in on the platform.	
Post-Condition:	
Success:	The user will be able to successfully view the profile information of registered users.
Failure:	The user would not be able to view the profile information unless they are logged in.

Main Flow:	
	1) The user clicks view profile button **2) The user reaches profile information page.**

Table 8 Share Artwork

Use case ID:	**UC-08**
Name:	**Artwork Sharing**
Actors:	**User**
Description:	**This use case allows users to**
Pre-Conditions:	
The user must be registered/logged in on the platform and he should have created an artwork.	
Post-Condition:	
Success:	The user will be able to successfully share their artwork on the social platforms.
Failure:	The users and admin couldn’t agree upon terms for promotion.
Main Flow:	
	1) The user reaches Artwork details page from Feed page. **2) The user clicks share artwork icon/button.** **3) The user selects the social platform where he wants to share artwork.** **4) The artwork gets shared.**
Alternate Flow:	
	1) The users did not selected social platform to share. **2) The user did not get his artwork promoted.**

Table 9 Admin Panel

Use case ID:	**UC-09**
Name:	**Admin Panel**
Actors:	**Admins**
Description:	**This use case allows admin to moderate and monitor the platform.**
Pre-Conditions:	
The admin must be logged in on the application.	
Post-Condition:	
Success:	The admin will be able to view and monitor user’s requests.
Failure:	The admin could not be logged in because he provided wrong credentials.
Main Flow:	
	1) The user logs in as admin from Login page. **2) The admin views and operates user’s requests and queries.**
Alternate Flow:	
	1) The admin provided wrong credentials in the login page. **2) The admin cannot access the admin panel.**

2.5 Use case design:

Signup:

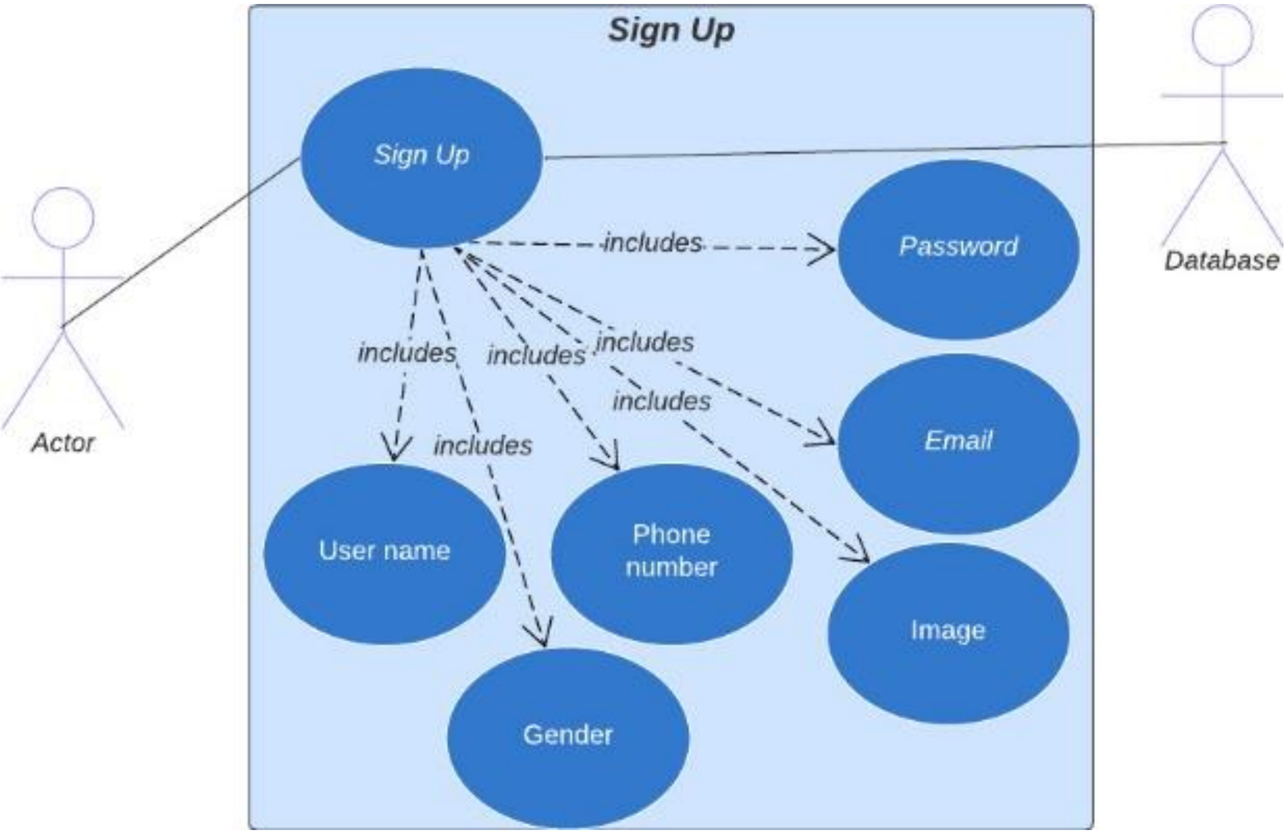


Figure 1 Use case Diagram Sign Up

Login:

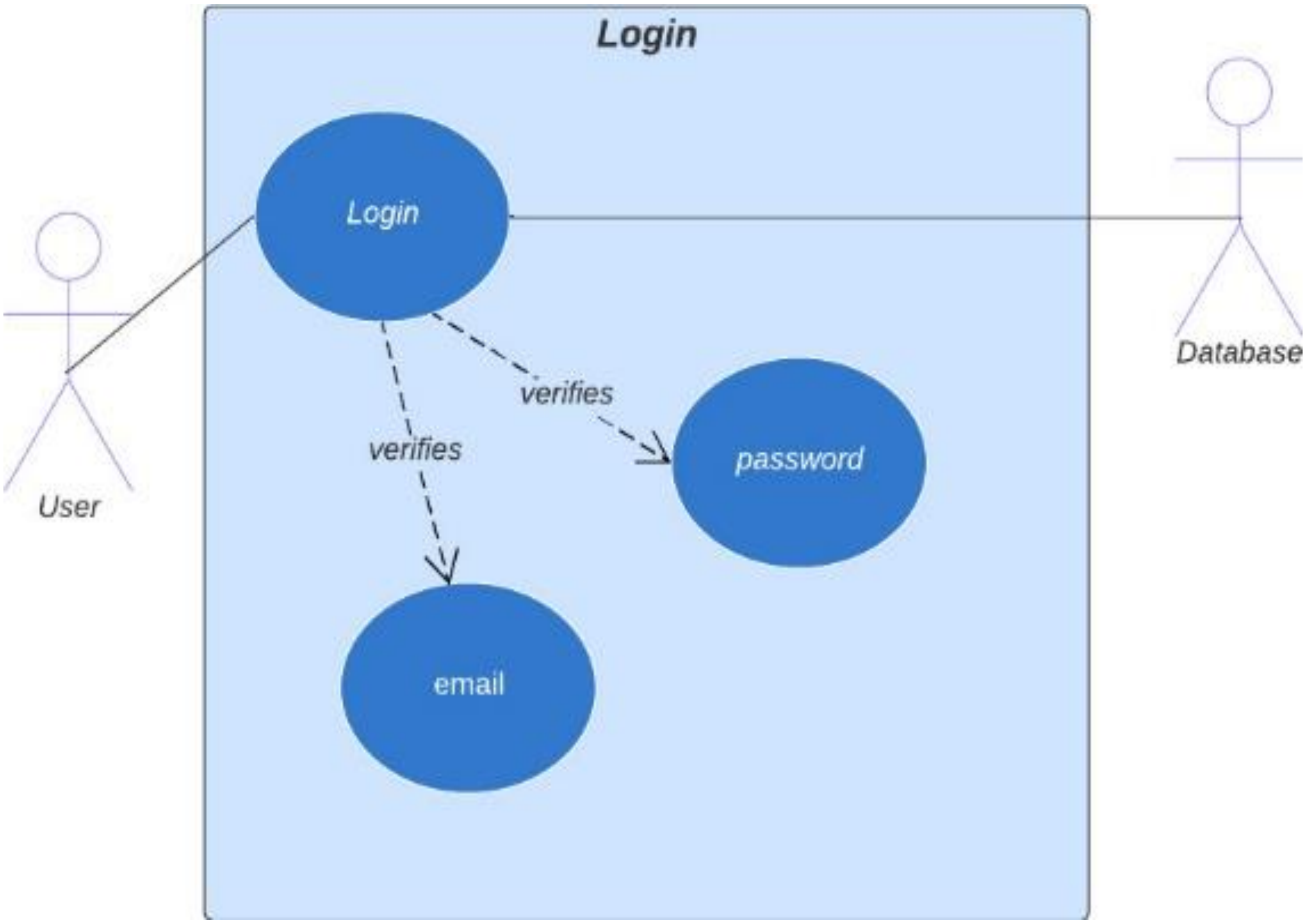


Figure 2 Use case Diagram Login

Artwork Feed:

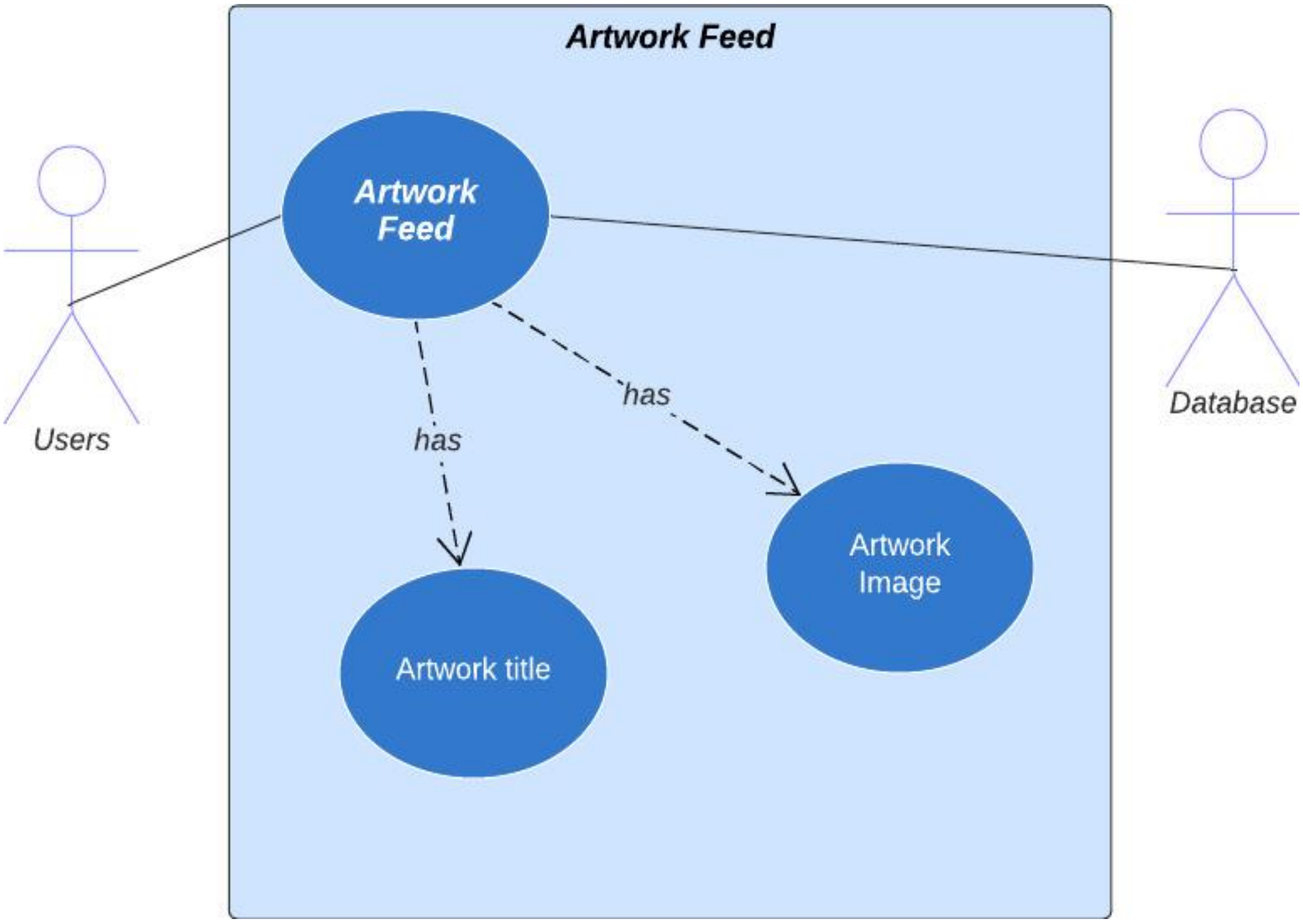


Figure 3 Use case Diagram Artwork Feed

Artwork Generation:

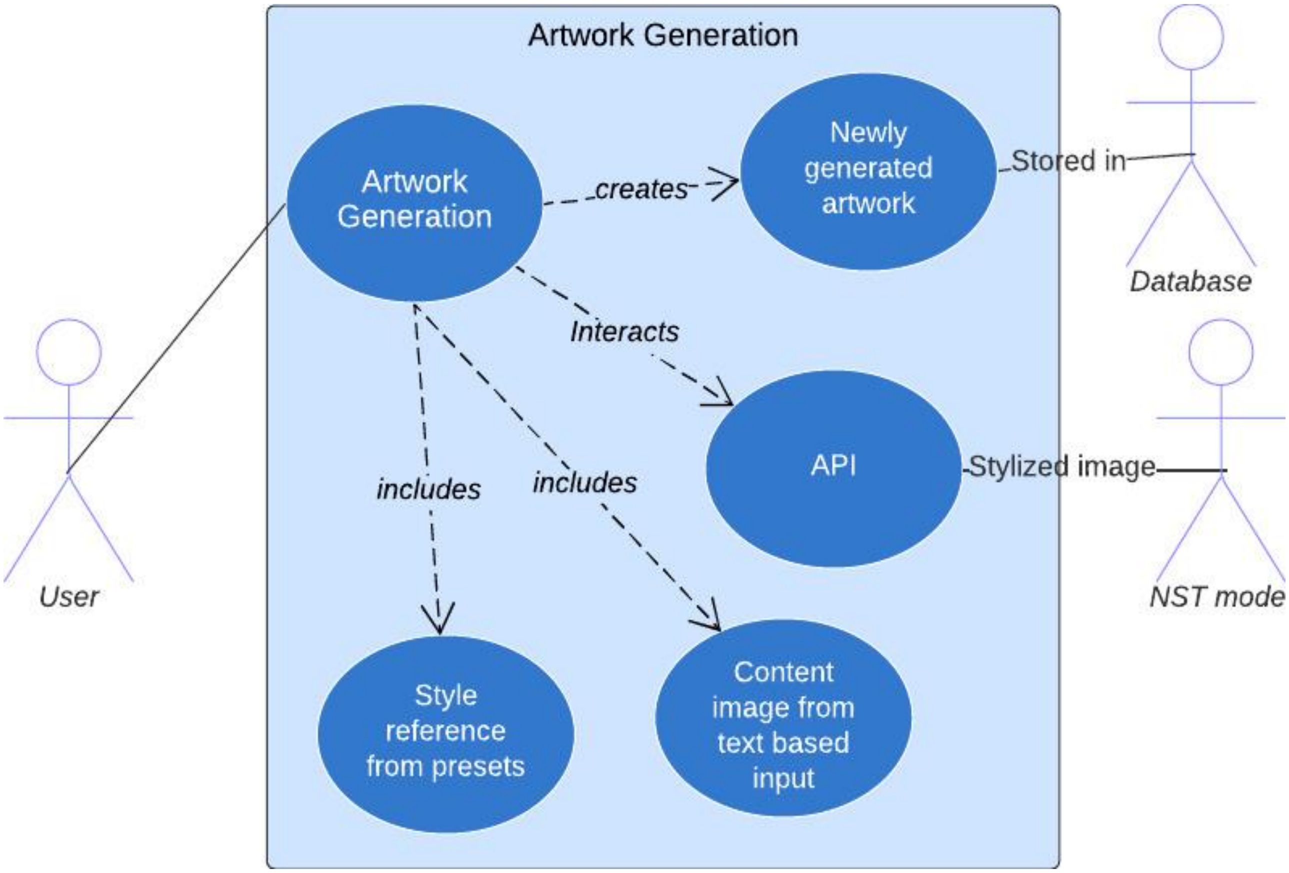


Figure 4 Use case Diagram Artwork Generation

Artwork Collections:

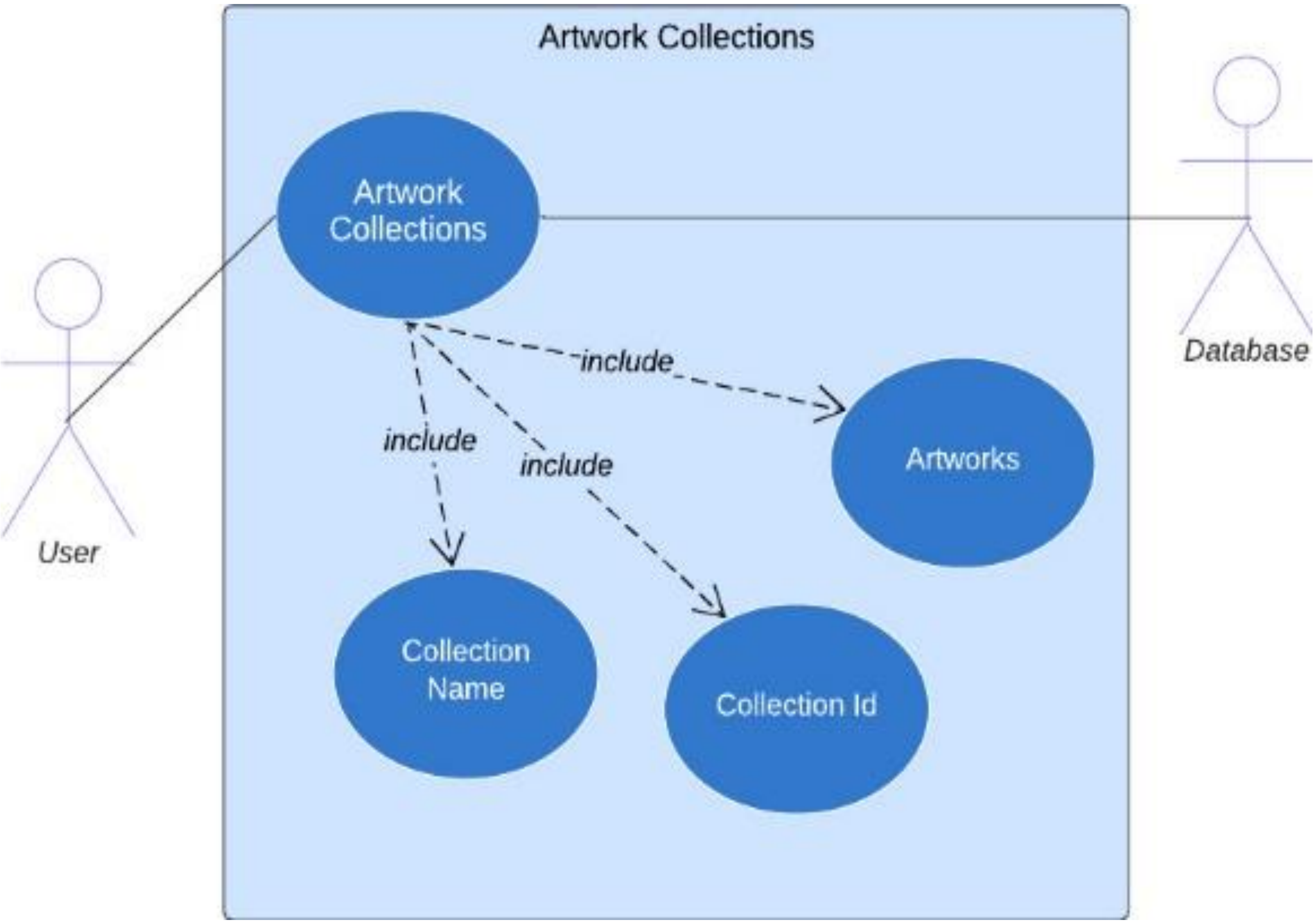


Figure 5 Use case Diagram Artwork Collection

Trending Artworks:

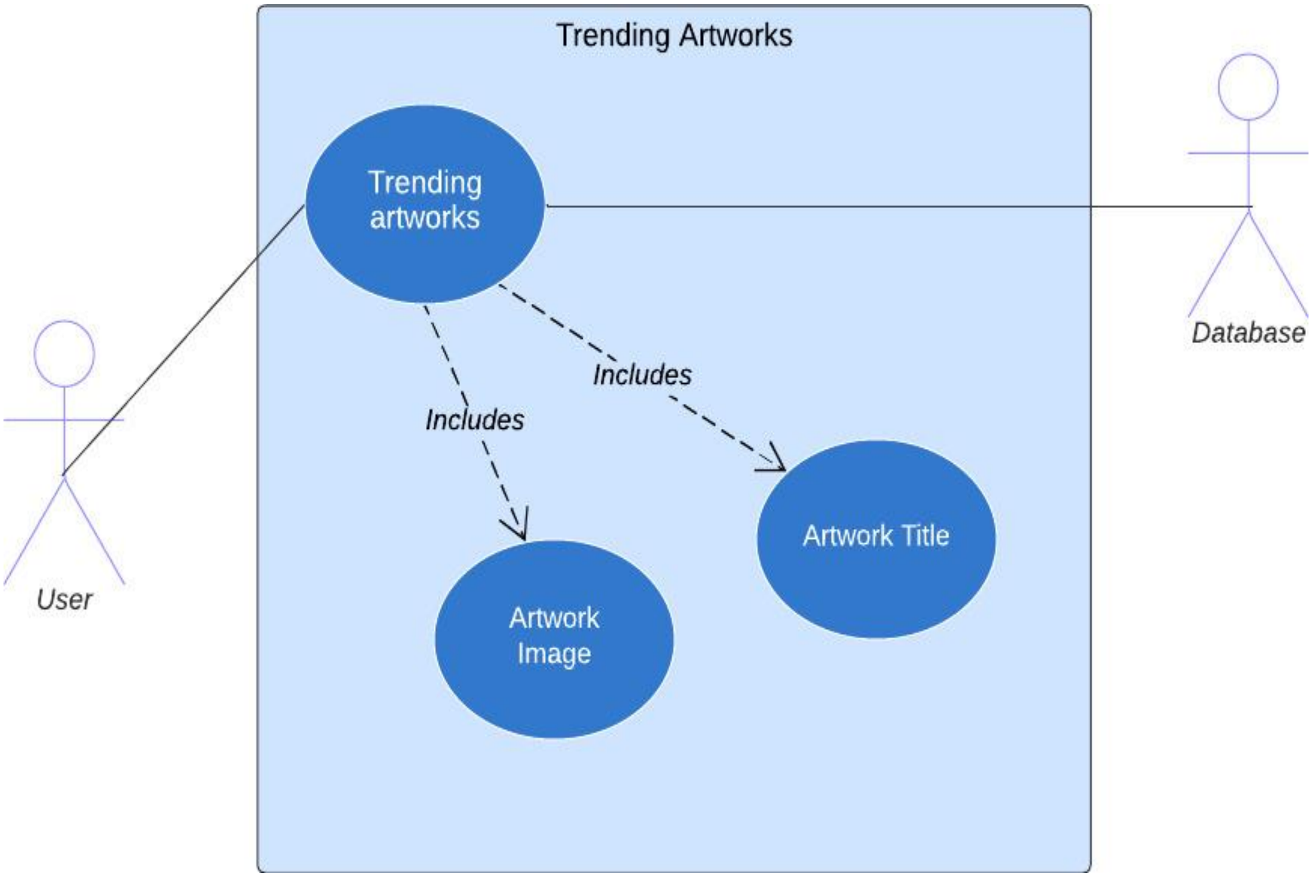


Figure 6 Use case Diagram Trending Artworks

User Profile:

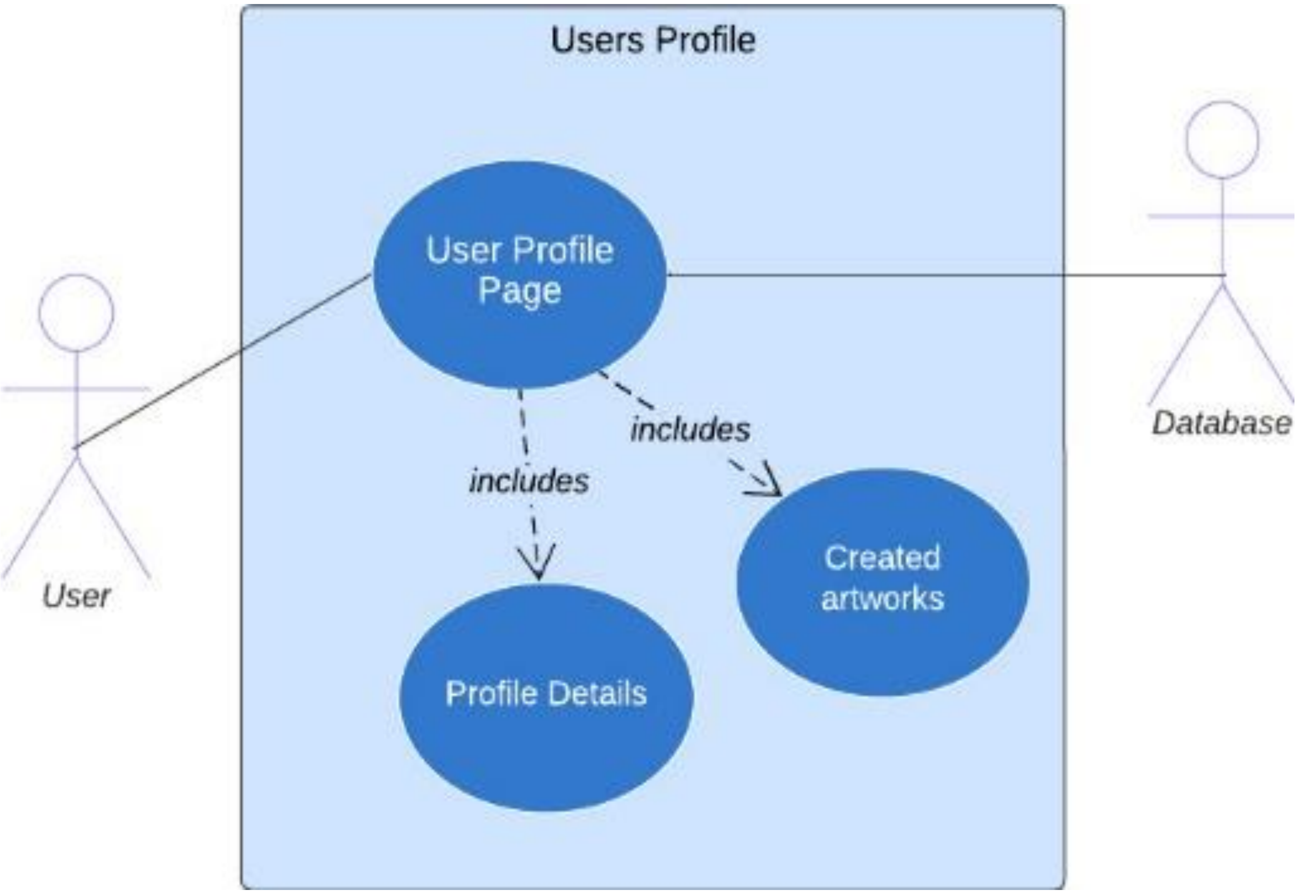


Figure 7 Use case Diagram User Profile

Sharing the Artwork:

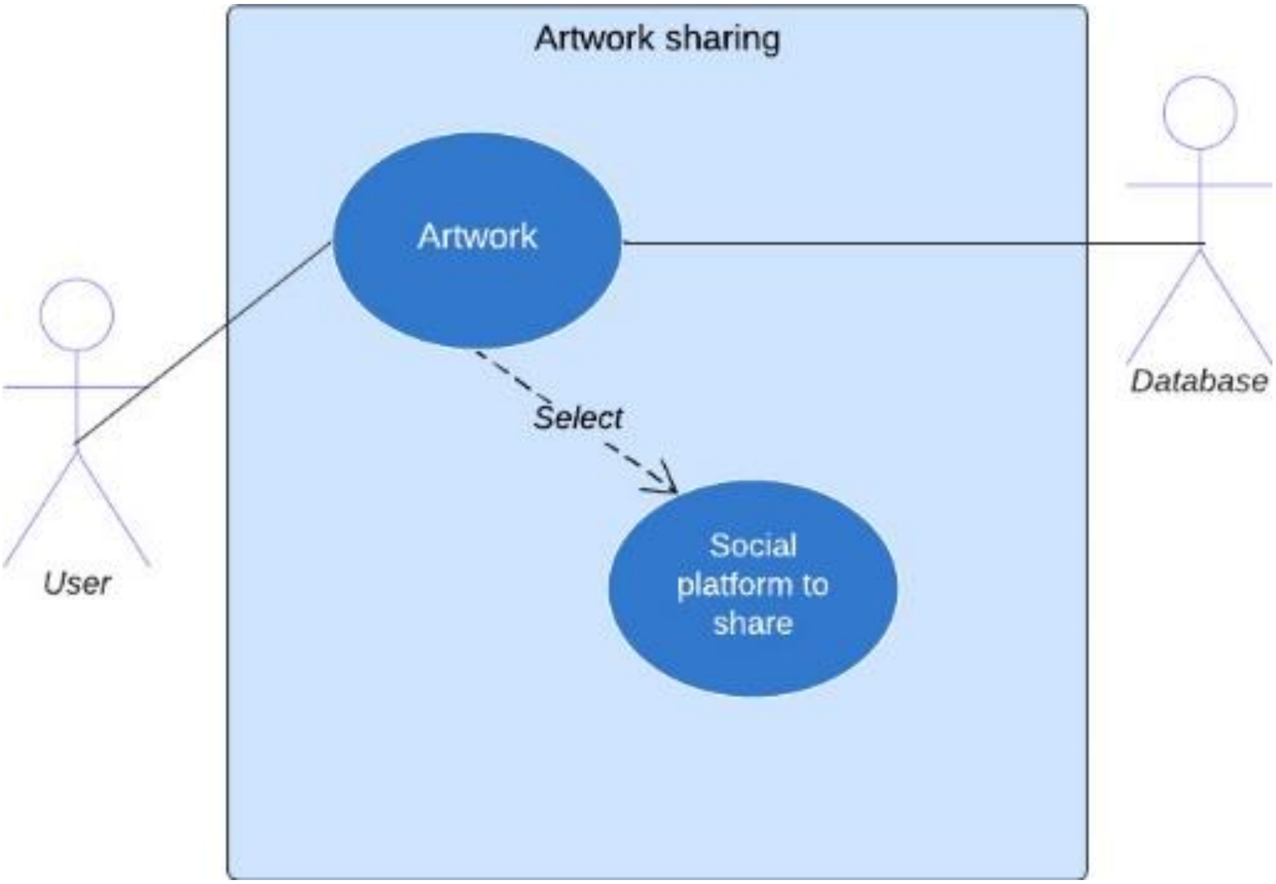


Figure 8 Use case Diagram Artwork Sharing

Admin Panel:

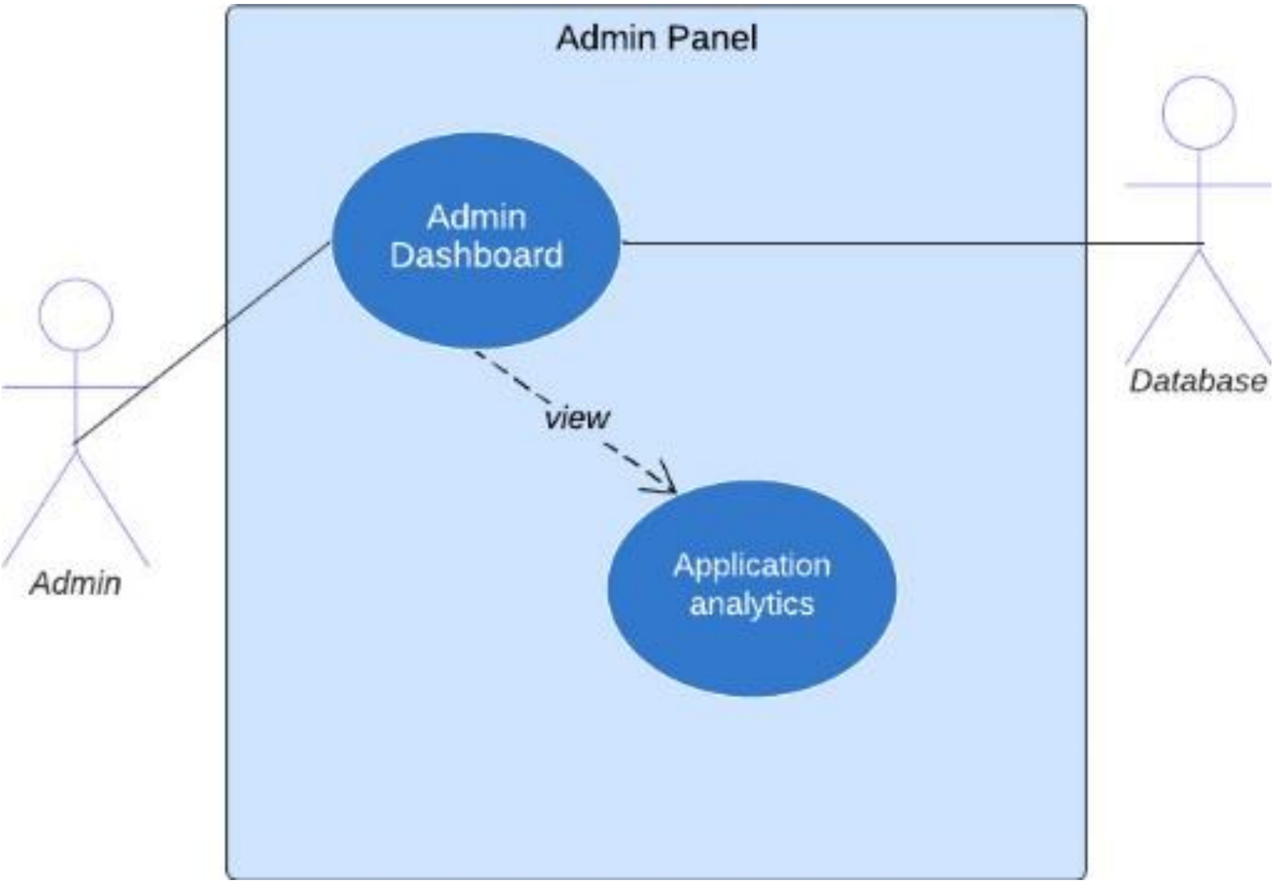


Figure 9 Use case Diagram Admin Panel

2.6 Software development life cycle model (justification on why this particular model is considered):

Waterfall model:

The Software development Life cycle we will be utilizing will be the simple and old-fashioned Waterfall Model. The waterfall model works on preceding phase that acts as input for every next phase because of the sequential design process where progress will be seen as steadily flowing through each phase of requirements, analysis, design, coding, testing and operations.

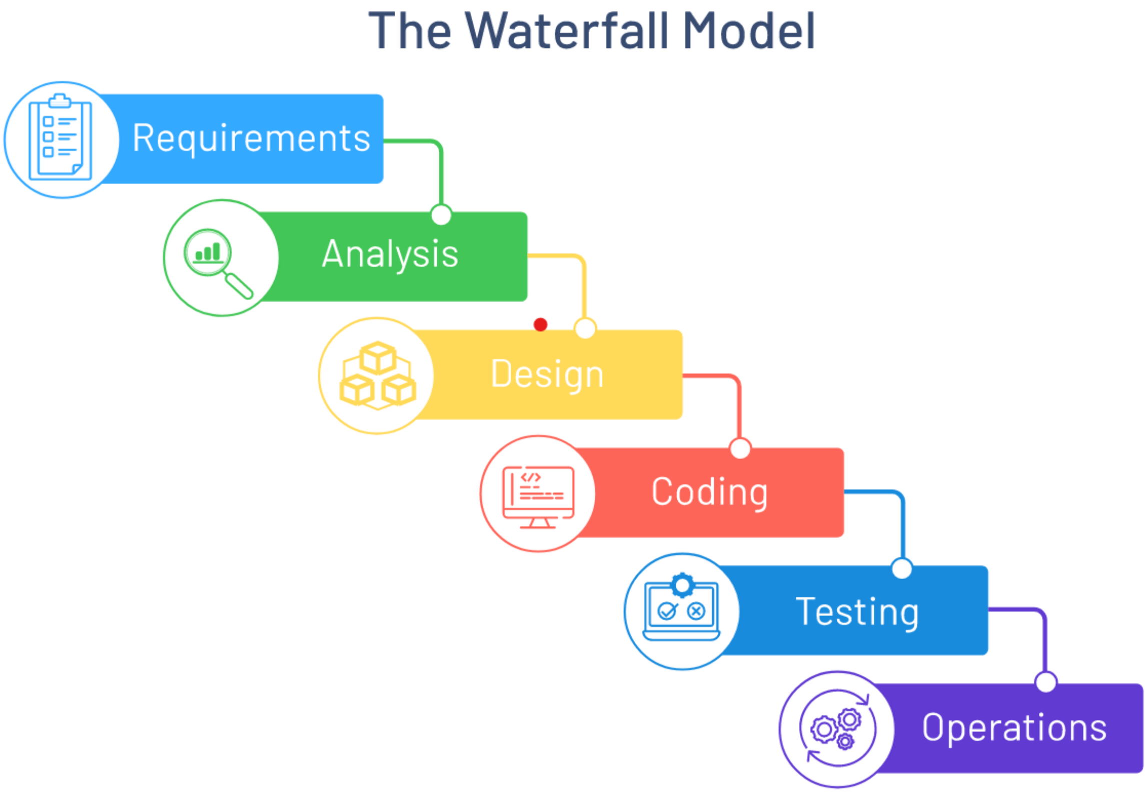


Figure 10 Waterfall Model

Requirements:

The requirements phase in context to our project would be the functional requirements and non-functional requirements designing and discussion with our respective supervisor. As our analysis on the requirements is dependent on completion of this phase, we won’t move on until the scope of functionality of our project is finalized.

Analysis:

In the analysis phase, we will reflect on our functional as well and non-functional requirements and perform formal analysis by observing competency required for using the software tools including the frameworks and technologies we will utilize to develop our frontend client for user, set up servers to handle requests as well the API we will create to deploy our model.

Design:

After the completion of analysis phase, we will begin designing our project. This designing will cover each and every aspect of our project. From designing the schema for our backend database to designing high fidelity prototypes for user interface, all of the designing and wireframing will be done in this phase so the development can be started.

Coding:

Once the preceding activity of designing all components of the project is finished, we can start working on the development. In this phase, we will begin developing our frontend which will be convenient as we will have prototypes prepared, followed by setting up our server for interaction with database (which will also follow the schema we would have designed in design phase), and finally we start with the development of our model, which will go through every step of developing workflow of a machine learning model. After our model is trained and tested, we will move on to the deployment part of it, which we will execute by converting it to an API.

Testing:

Once our project is fully development, we will begin testing it using different types of testing to fully make it bug free. We will begin by unit testing, where we will test individual methods and functions, components, or modules we will have in our project. Followed by unit testing, we will have integration testing that will substantiate that different auxiliary modules or services being used by our projects are performing well together. Finally, we will have acceptance testing that will verify if our project satisfies our business perspective requirements (our monetization plans) and system testing in which we will conduct complete integrated testing of the entire system, along with its specified requirements. So, that our final project will be a debugged version of the original, presumably free of any possible defects.

Operations/Maintenance:

After the complete development and testing of our project, it will be ready for deployment and maintenance. We will begin by utilizing a deploying service (like Azure, AWS) and set up all components of our project put up globally. With all our servers and frontend running live, we will transition into maintenance to work on the future bugs and errors that may occur.

Justification for using this model:

Our project follows a sequential following of steps that are dependent of their preceding phases without needing to recur on the previous step ever again. In context of our project, we will begin by defining our requirements, which once finalized won’t require us to come back again to change them again. Similarly, the analysis phase will also occur only once and all of the analysis of the requirements will be done in this phase. The same pattern will follow in other phases of designing, coding, testing, and operations/maintenance. In every phase, that relevant activity will be completed and won’t require us to ever involve in it again. Thus, there won’t ever be any need for any kind of iterative go through over all the activities again as these activities would only happen once and for all. Due to these reasons, any kind of incremental or iterative process model like agile or hybrid won’t be required and just would be an overkill for this kind of project. The simple waterfall will do the trick for us. The waterfall model is a type of software development process that is linear and sequential. In this model, each stage of the development process is completed before moving on to the next stage. This means that there is no overlap or iteration between the different stages (just like our project). Because of this, the waterfall model is best suited for projects where the requirements are well understood and relatively unchanging.

Other process models including iterative and incremental based approaches like agile won’t be necessary here as they would be an overkill. These techniques shine in cases where the requirements are volatile and changing rapidly. Some cases also include in which the client is asking for improvement for customer satisfaction. None of these scenarios apply to us. Our FYP will be developed in a linear flow, and once requirements are finalized, they won’t be changed again. Thus, waterfall model would be more sufficient than agile or any other process model.

CH 03

3 System Design

3.1 Work breakdown structure (WBS)

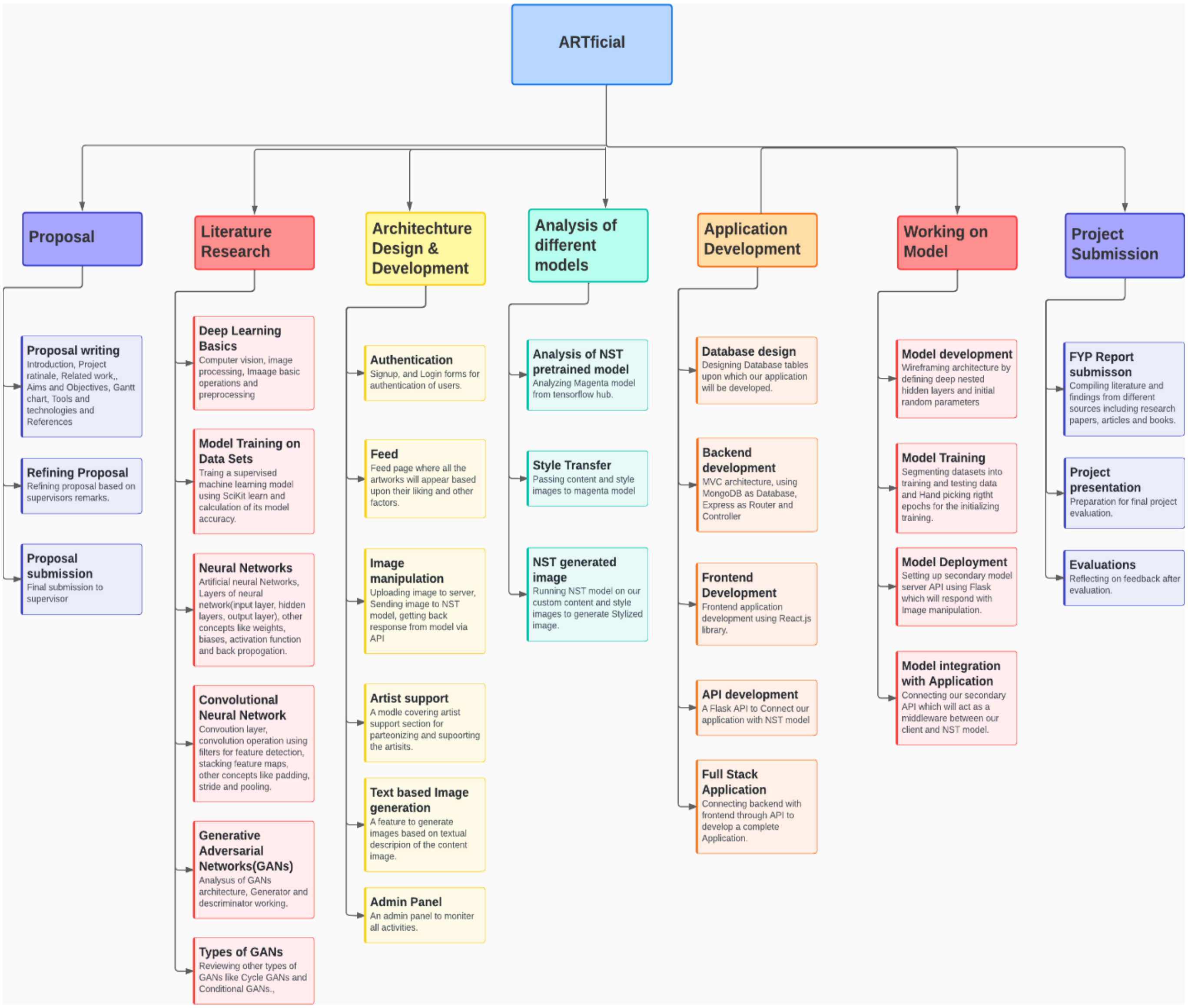

ARTficial
Proposal
Proposal writing
Introduction, Project ratinale, Related work,, Aims and Objectives, Gantt chart, Tools and technologies and References
Refining Proposal
Refining proposal based on supervisors remarks.
Proposal submission
Final submission to supervisor
Literature Research
Deep Learning Basics
Computer vision, image processing, Imaage basic operations and preprocessing
Model Training on Data Sets
Traing a supervised machine learning model using SciKit learn and calculation of its model accuracy.
Neural Networks
Artificial neural Networks, Layers of neural network(input layer, hidden layers, output layer), other concepts like weights, biases, activation function and back propogation.
Convolutional Neural Network
Convoution layer, convolution operation using filters for feature detection, stacking feature maps, other concepts like padding, stride and pooling.
Generative Adversarial Networks(GANs)
Analysus of GANs architecture, Generator and descriminator working.
Types of GANs
Reviewing other types of GANs like Cycle GANs and Conditional GANs.,
Architechture Design & Development
Authentication
Signup, and Login forms for authentication of users.
Feed
Feed page where all the artworks will appear based upon their liking and other factors.
Image manipulation
Uploading image to server, Sending image to NST model, getting back response from model via API
Artist support
A modle covering artist support section for parteonizing and supoorting the artisits.
Text based Image generation
A feature to generate images based on textual descripion of the content image.
Admin Panel
An admin panel to moniter all activities.
Analysis of different models
Analysis of NST pretrained model
Analyzing Magenta model from tensorflow hub.
Style Transfer
Passing content and style images to magenta model
NST generated image
Running NST model on our custom content and style images to generate Stylized image.
Application Development
Database design
Designing Database tables upon which our application will be developed.
Backend development
MVC architecture, using MongoDB as Database, Express as Router and Controller
Frontend Development
Frontend application development using React.js library.
API development
A Flask API to Connect our application with NST model
Full Stack Application
Connecting backend with frontend through API to develop a complete Application.
Working on Model
Model development
Wireframing architecture by defining deep nested hidden layers and initial random parameters
Model Training
Segmenting datasets into training and testing data and Hand picking rigtht epochs for the initializing training.
Model Deployment
Setting up secondary model server API using Flask which will respond with Image manipulation.
Model integration with Application
Connecting our secondary API which will act as a middleware between our client and NST model.
Project Submission
FYP Report submisson
Compiling literature and findings from different sources including research papers, articles and books.
Project presentation
Preparation for final project evaluation.
Evaluations
Reflecting on feedback after evaluation.

Figure 11 Work Breakdown Structure

3.2 Activity diagram

Signup:

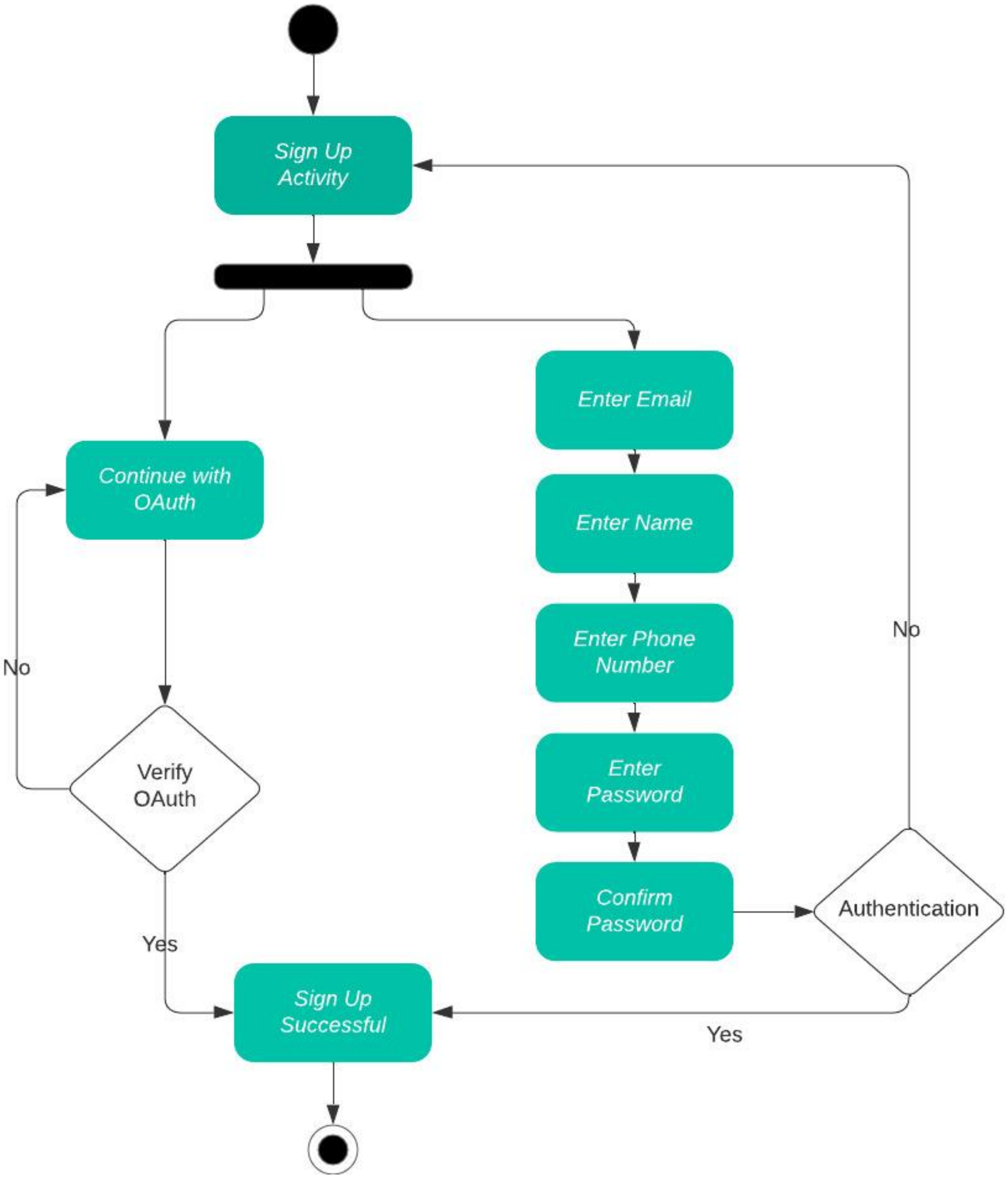


Figure 12 Activity diagram Signup

Login:

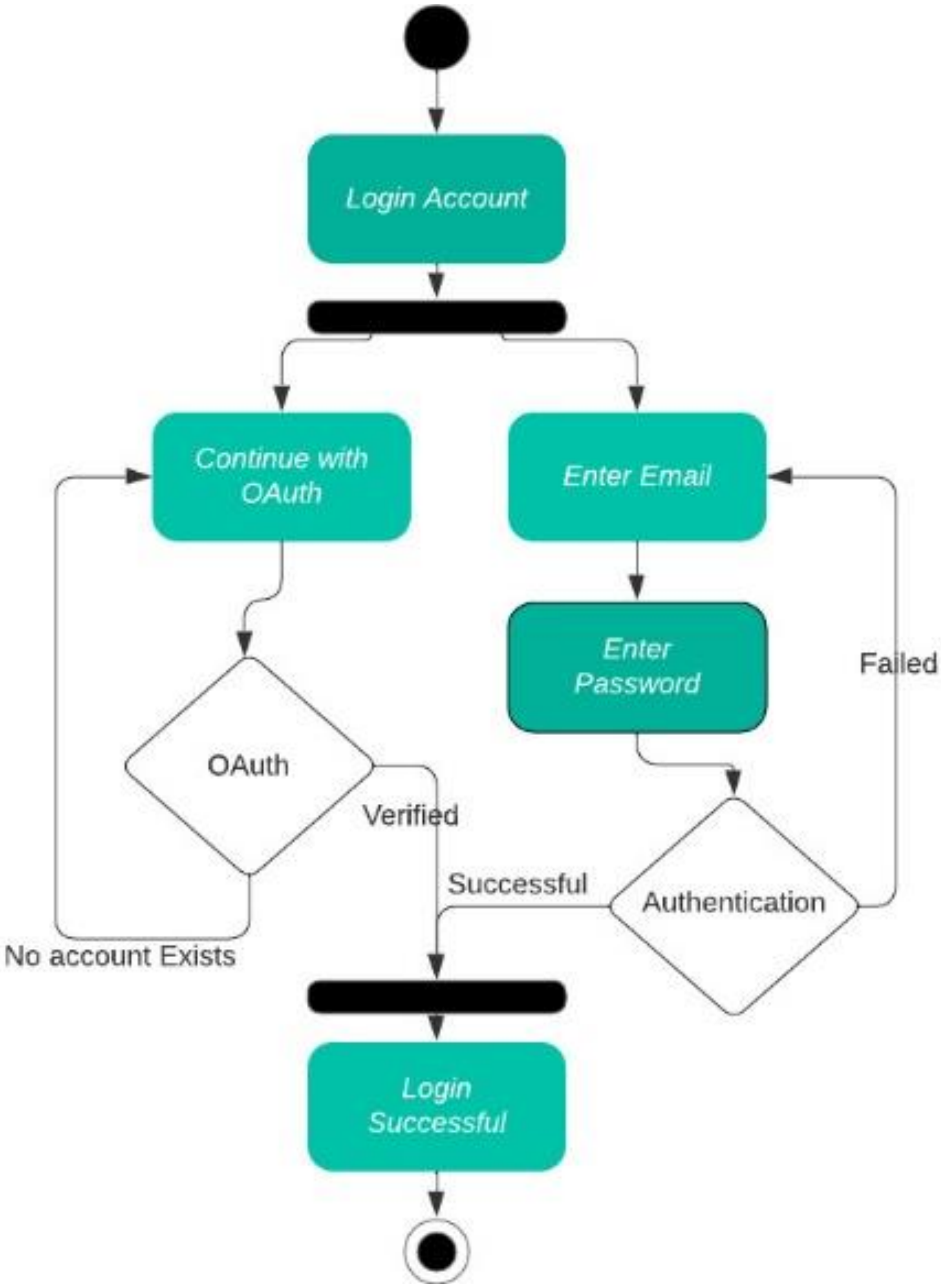


Figure 13 Activity diagram Login

Artwork Feed:

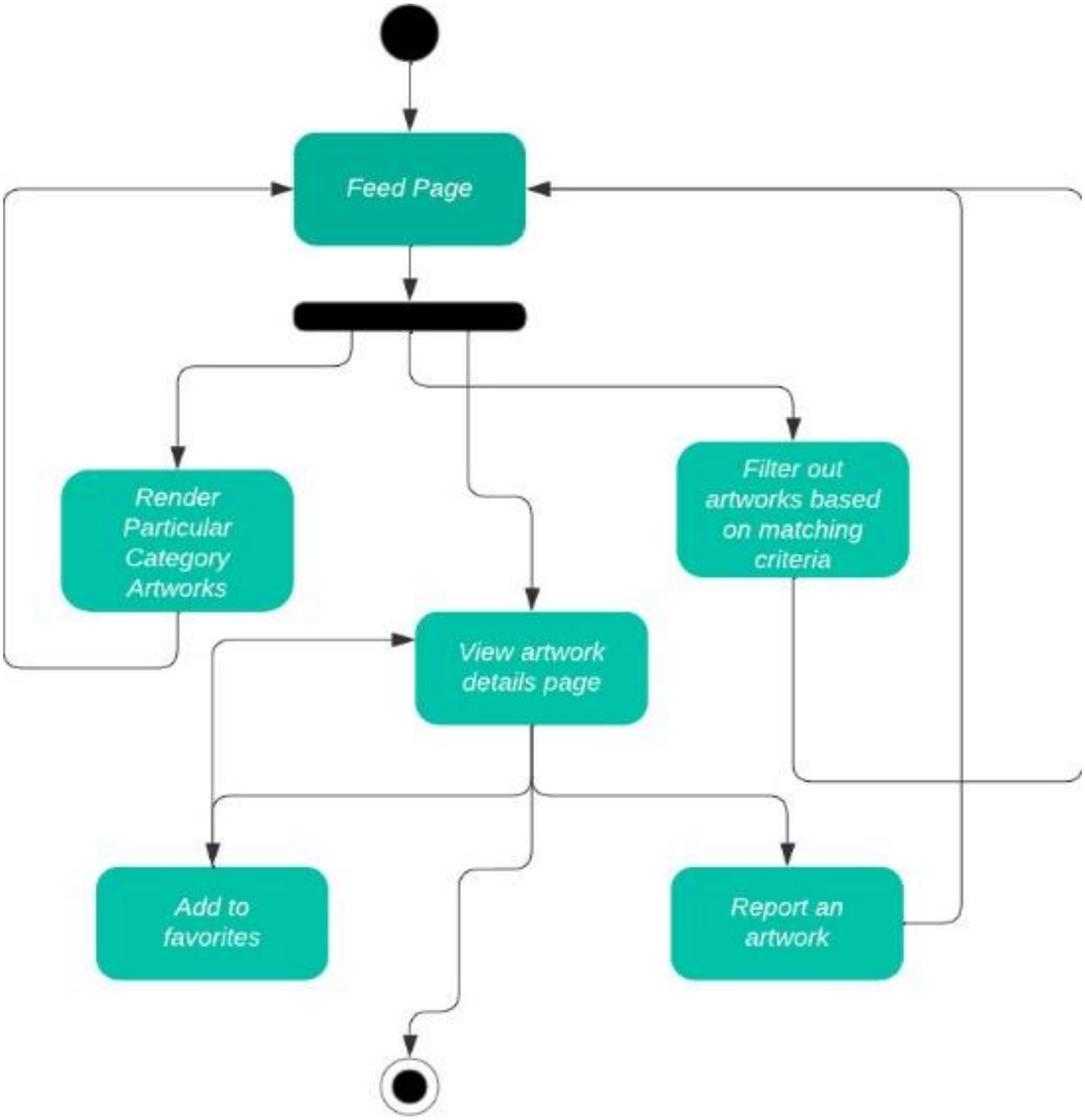
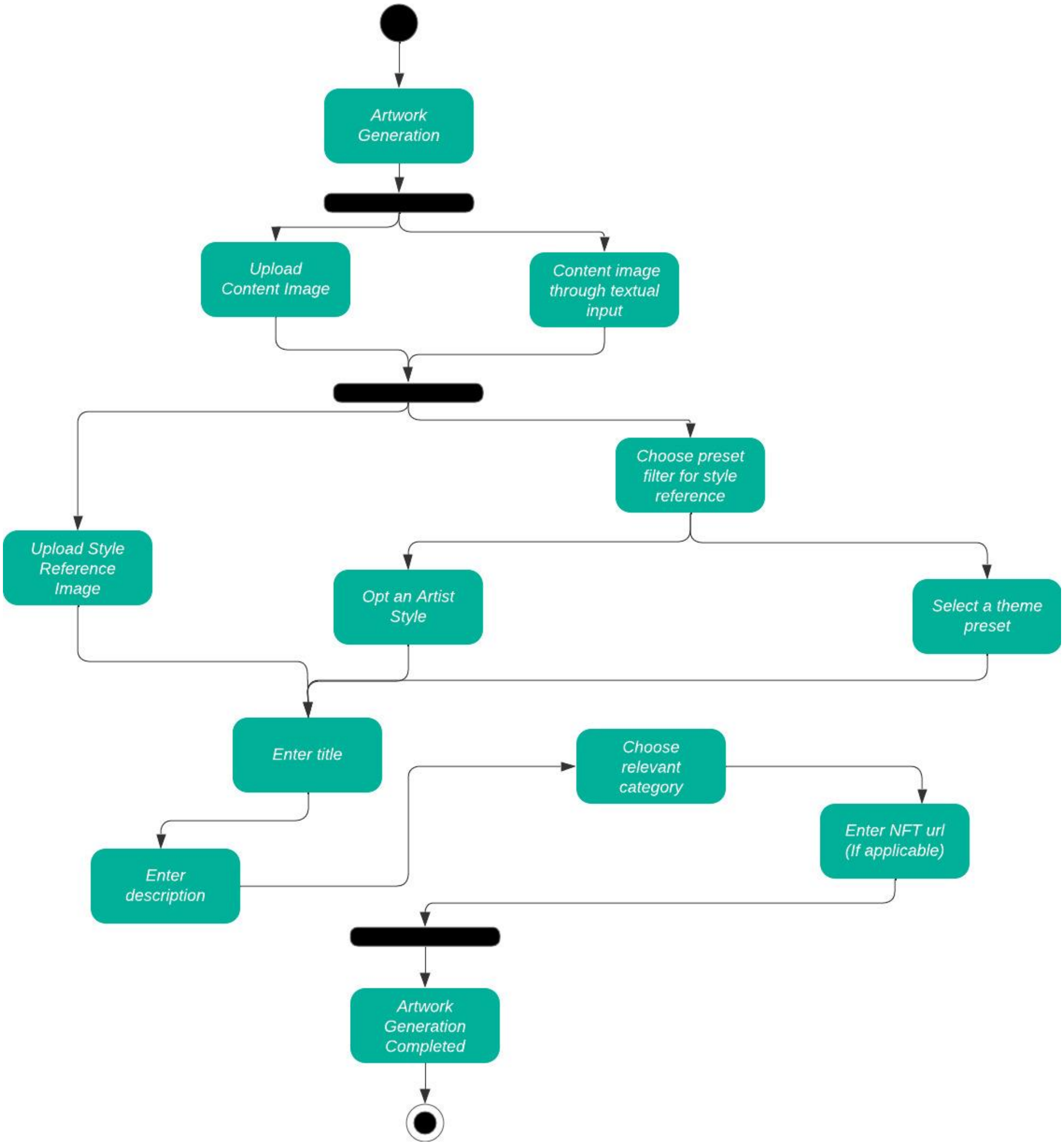

Feed Page
Render Particular Category Artworks
Filter out artworks based on matching criteria
View artwork details page
Add to favorites
Report an artwork

Figure 14 Activity diagram Artwork Feed

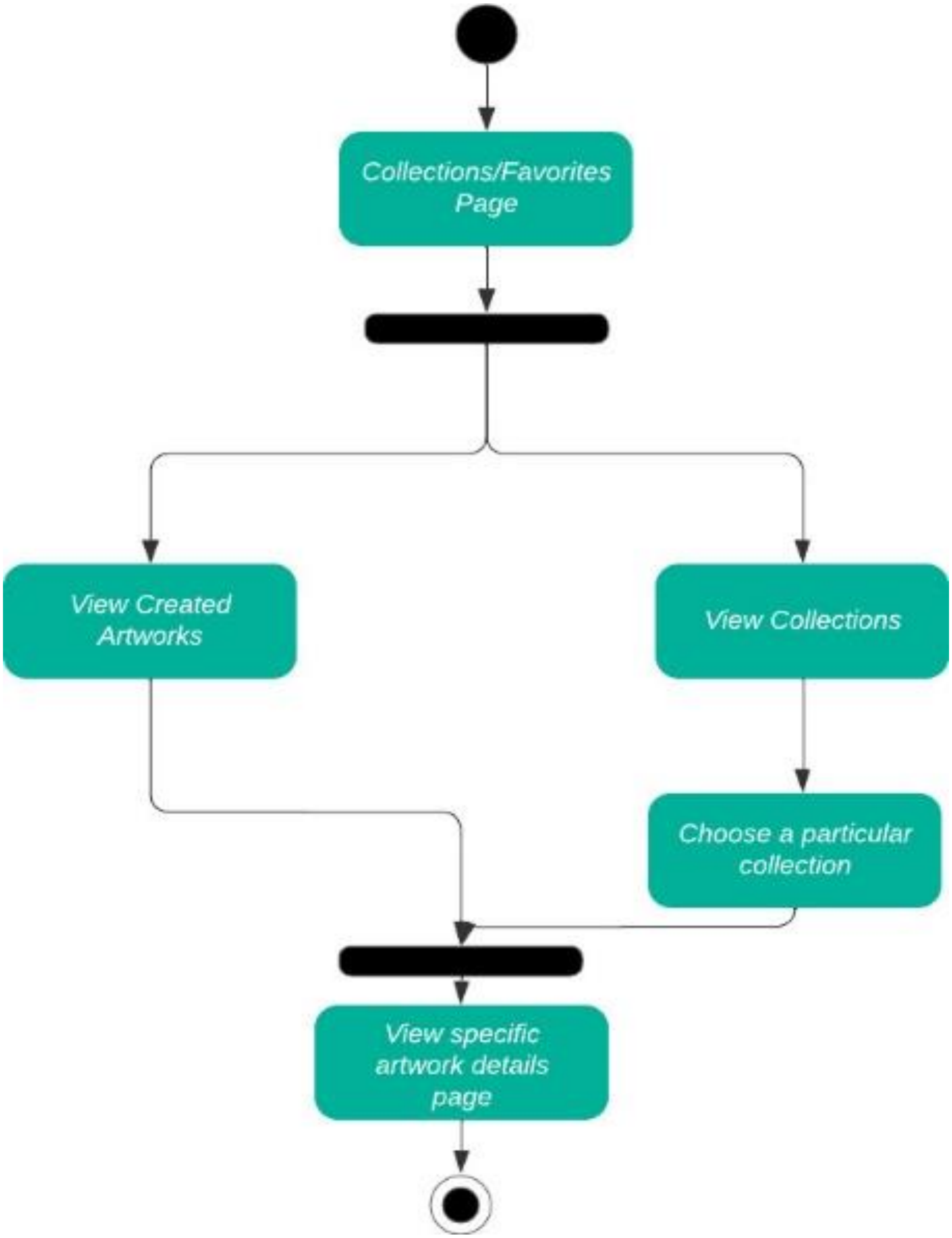
Artwork generation:



Artwork Generation
Upload Content Image
Content image through textual input
Choose preset filter for style reference
Upload Style Reference Image
Opt an Artist Style
Select a theme preset
Enter title
Choose relevant category
Enter NFT url (If applicable)
Enter description
Artwork Generation Completed

Figure 15 Activity diagram Artwork Generation

Artwork collection:



Collections/Favorites Page
View Created Artworks
View Collections
Choose a particular collection
View specific artwork details page

Figure 16 Activity diagram Artwork Collection

Trending Artworks:

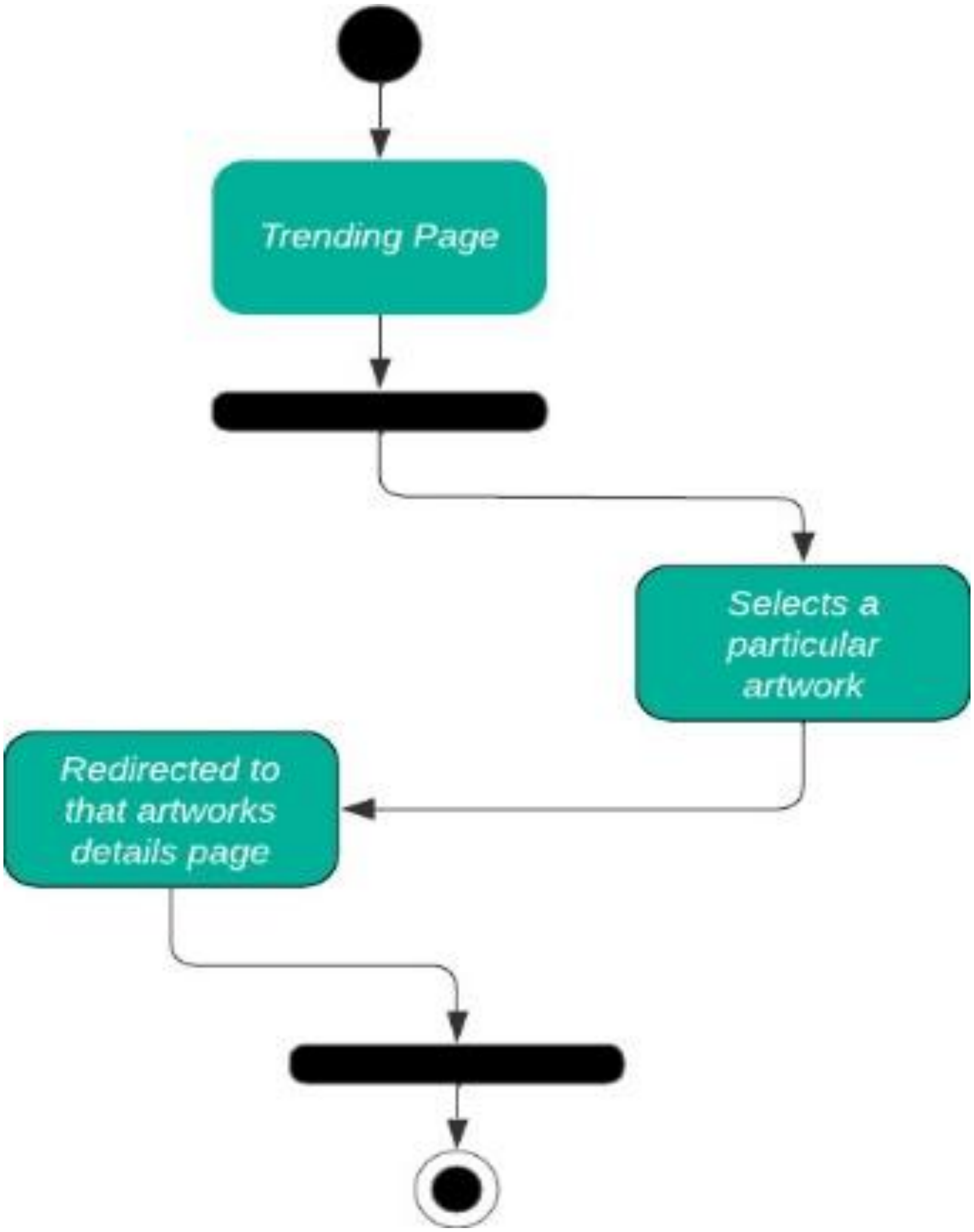

Trending Page
Selects a particular artwork
Redirected to that artworks details page

Figure 17 Activity diagram Trending Artworks

Users Profile:

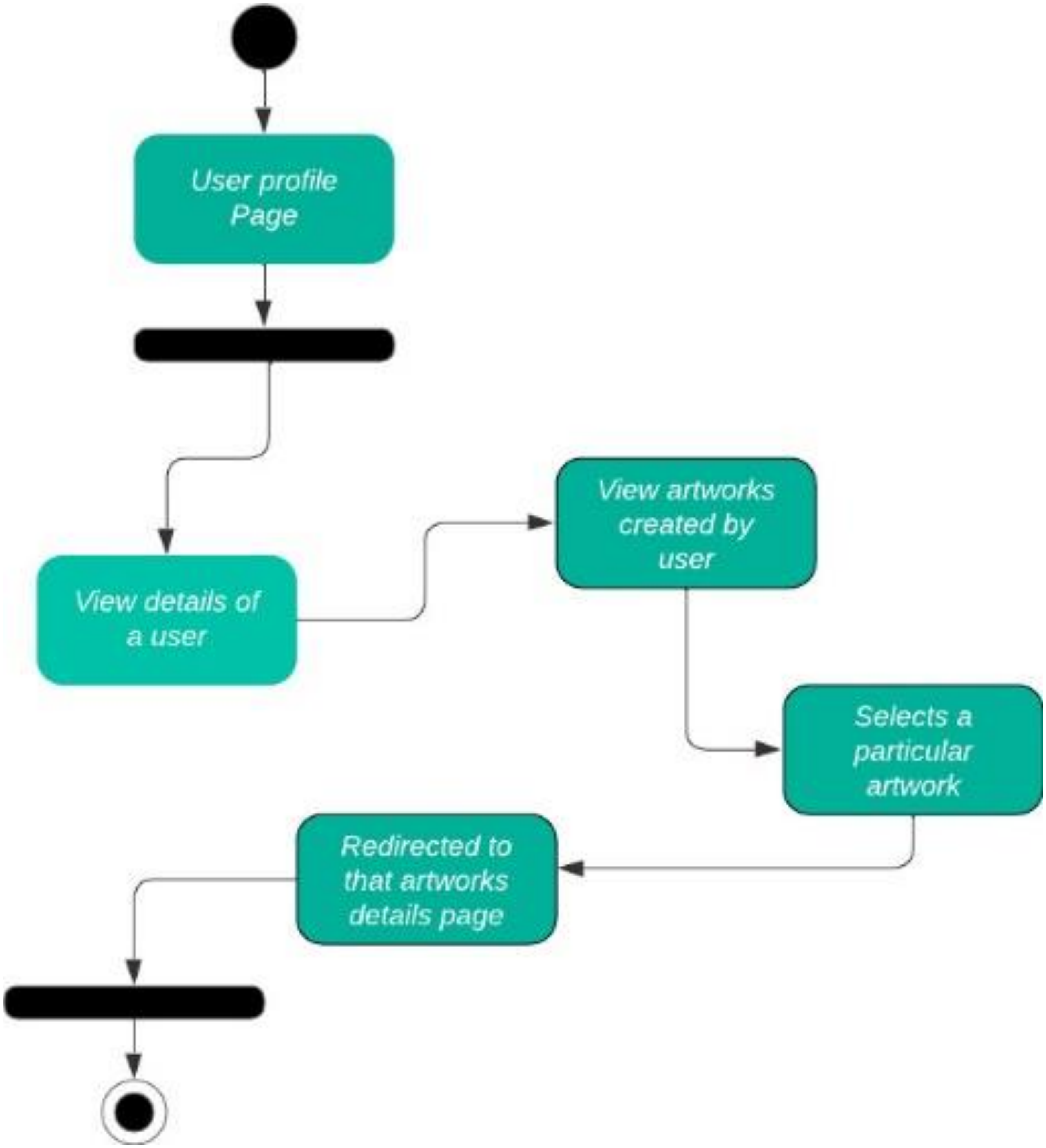

User profile Page
View details of a user
View artworks created by user
Selects a particular artwork
Redirected to that artworks details page

Figure 18 Activity diagram Users Profile

Sharing the Artwork:

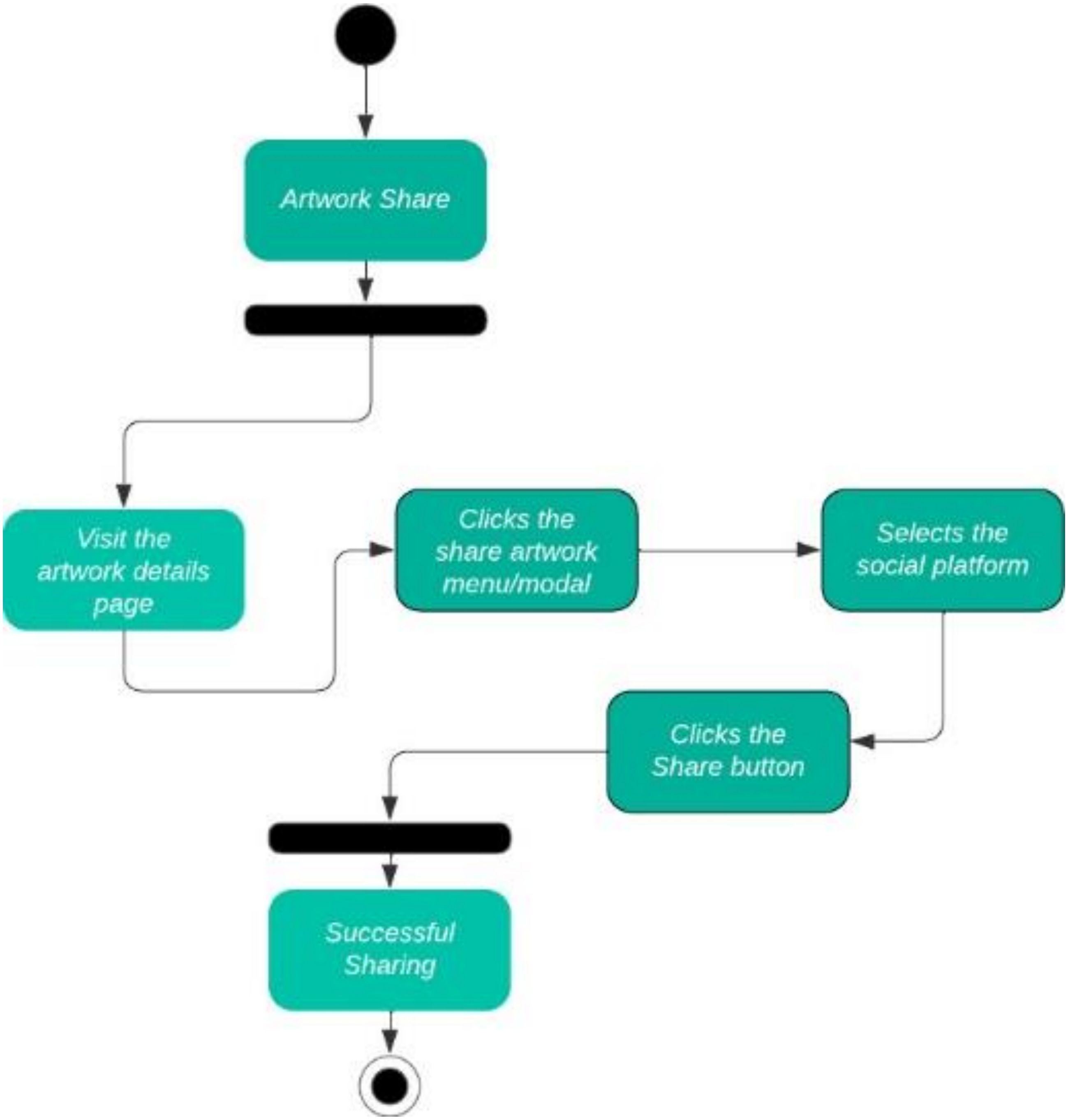

Artwork Share
Visit the artwork details page
Clicks the share artwork menu/modal
Selects the social platform
Clicks the Share button
Successful Sharing

Figure 19 Activity diagram Artwork Sharing

Admin panel:

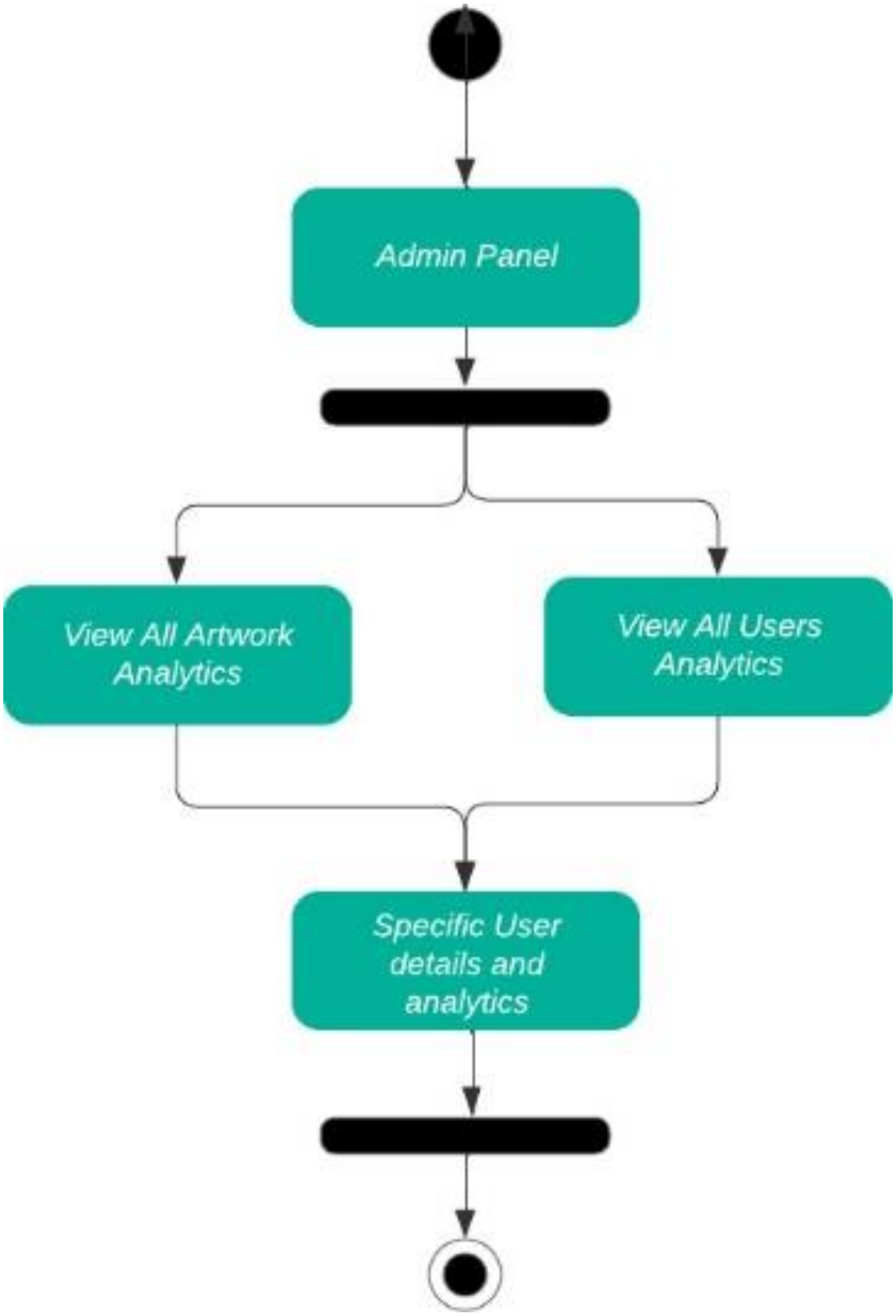

Admin Panel
View All Artwork Analytics
View All Users Analytics
Specific User details and analytics

Figure 20 Activity diagram Admin Panel

3.3 Sequence diagrams

Signup

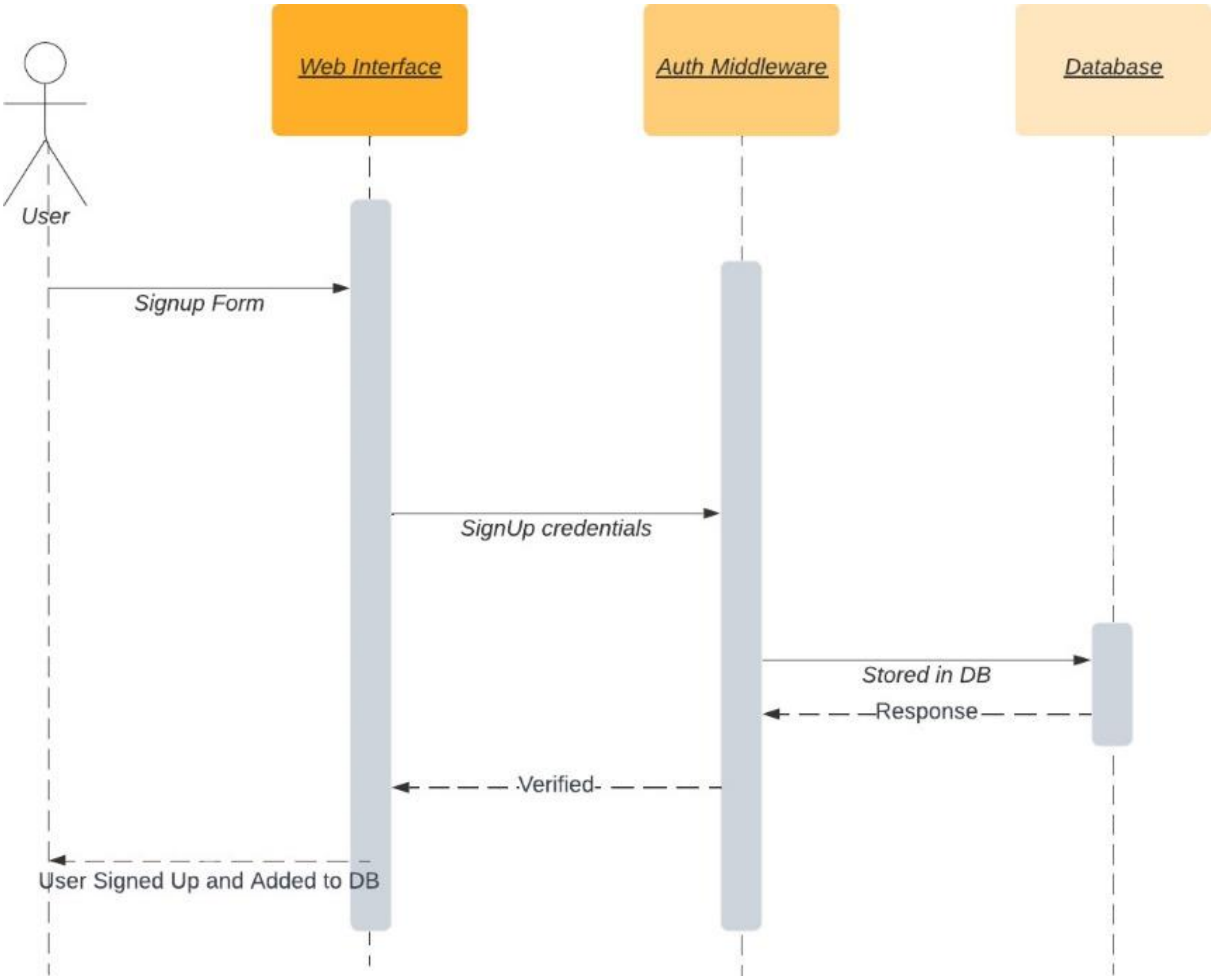


Figure 21 Sequence diagram Signup

Login

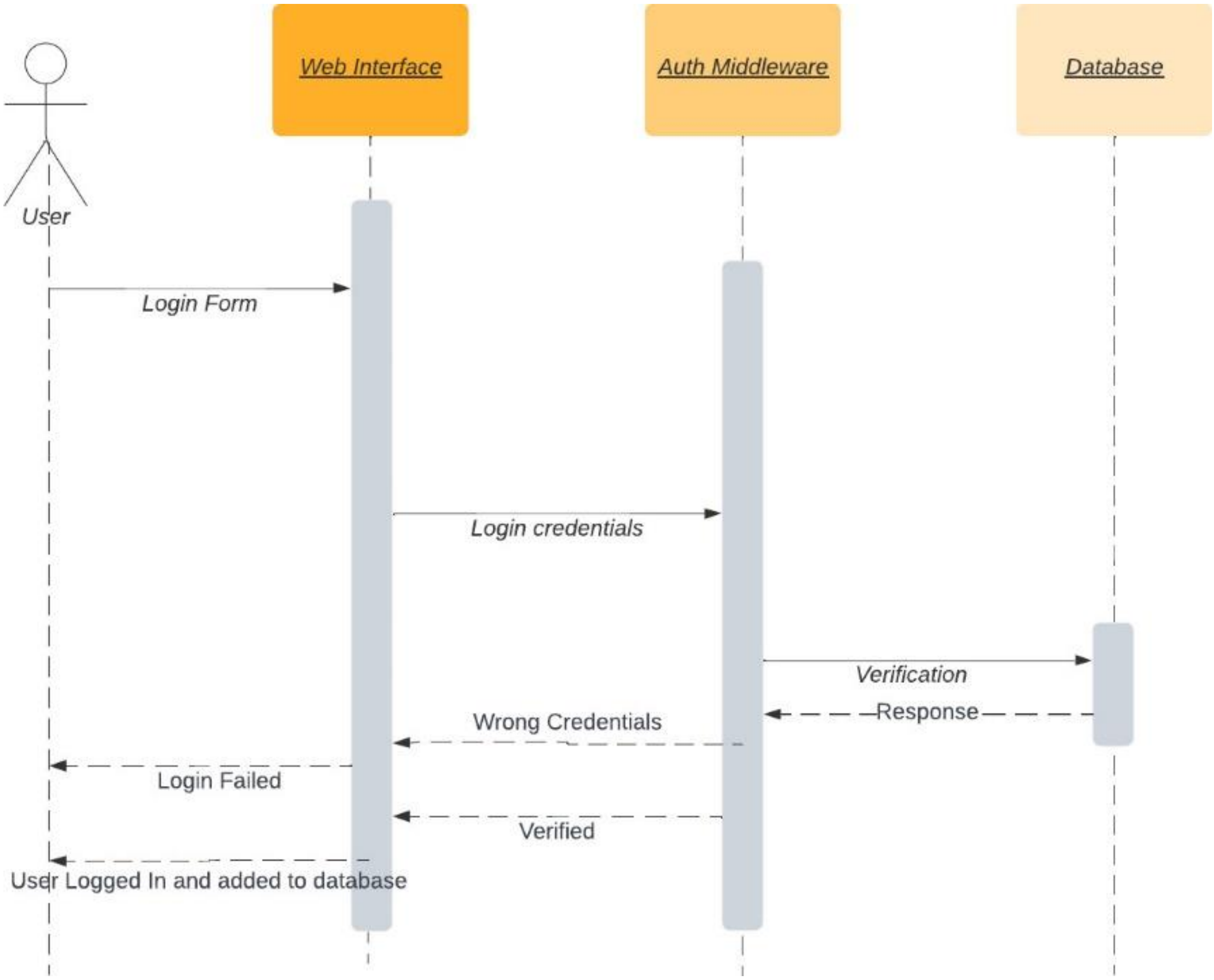


Figure 22 Sequence diagram Login

Artwork feed

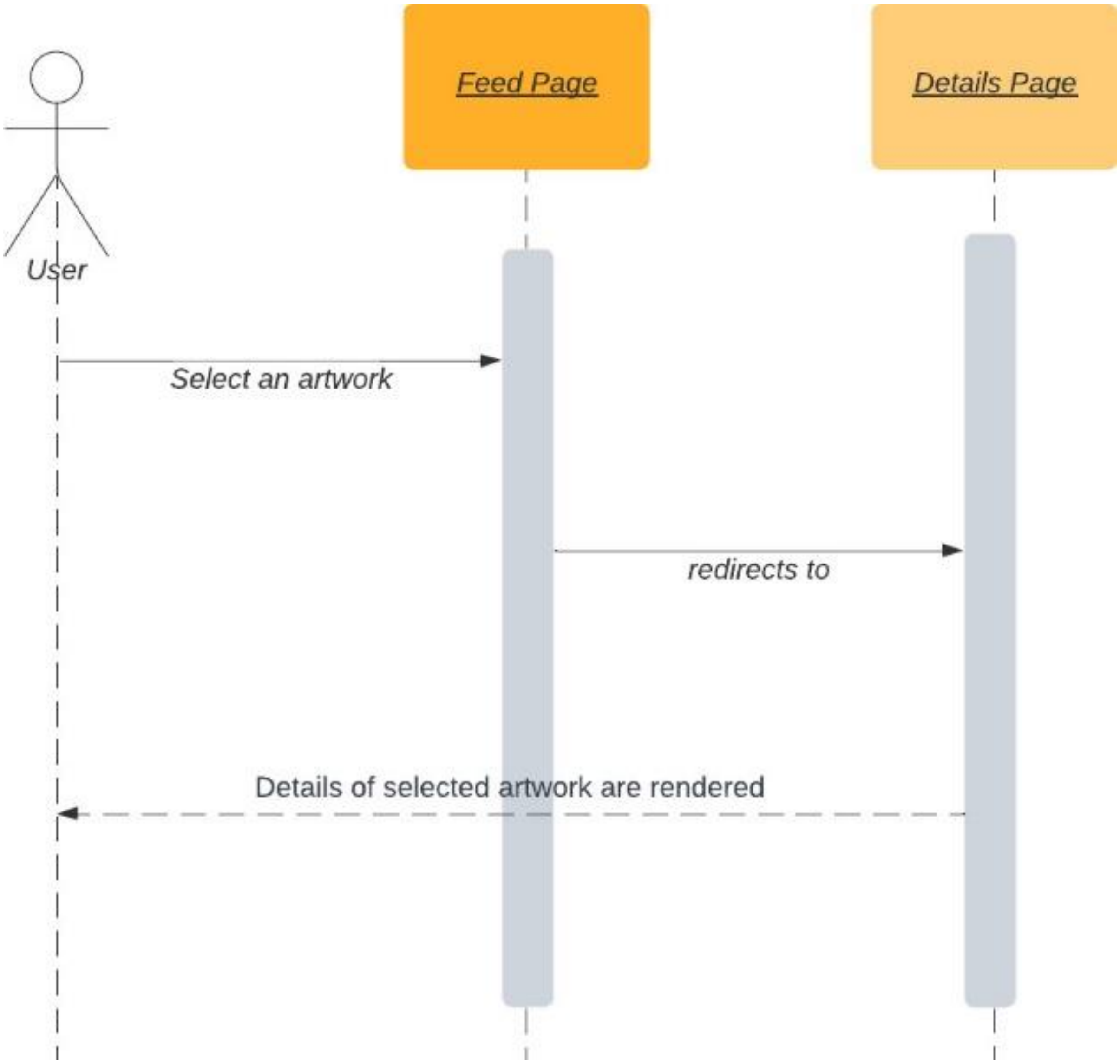


Figure 23 Sequence diagram Artwork Feed

Artwork generation

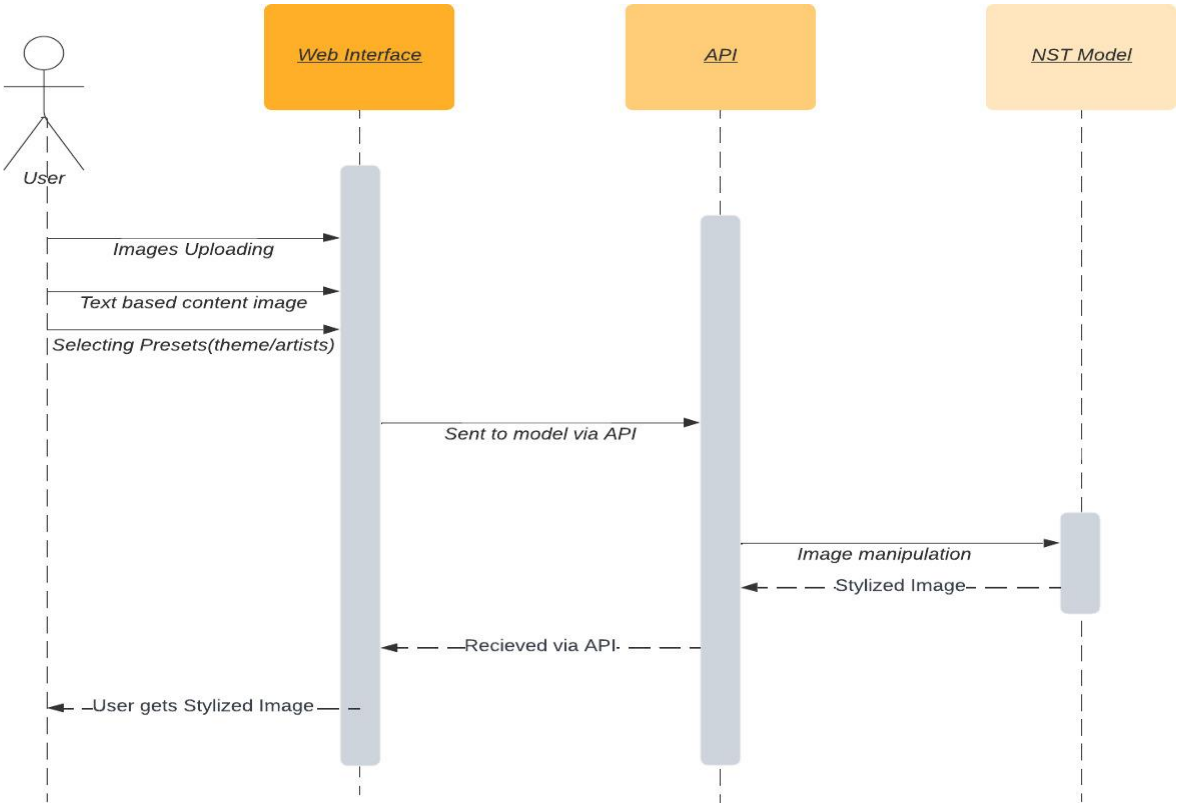


Figure 24 Sequence diagram Artwork Generation

Artwork collection:

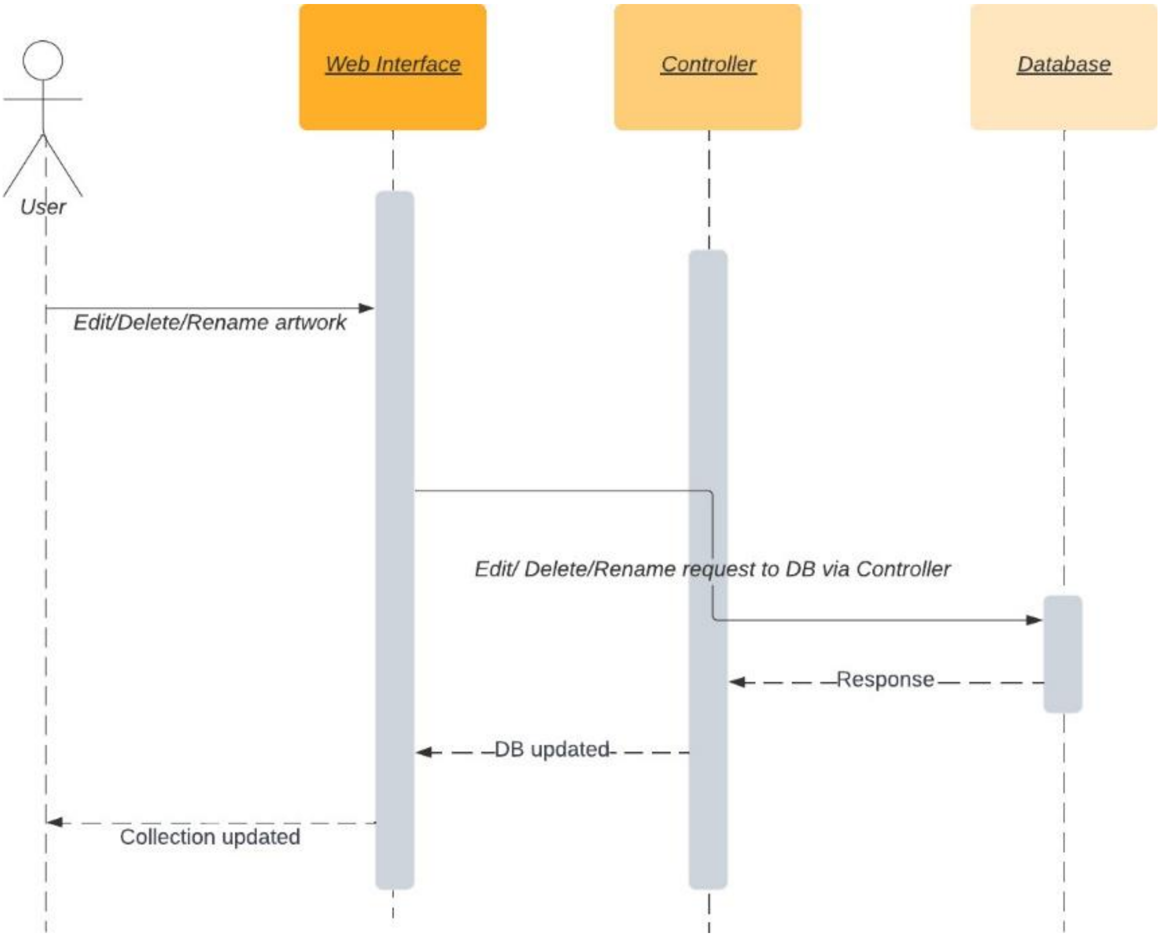

Web Interface
Controller
Database
User
Edit/Delete/Rename artwork
Edit/ Delete/Rename request to DB via Controller
Response
DB updated
Collection updated

Figure 25 Sequence diagram Artwork Collection

Trending Artworks:

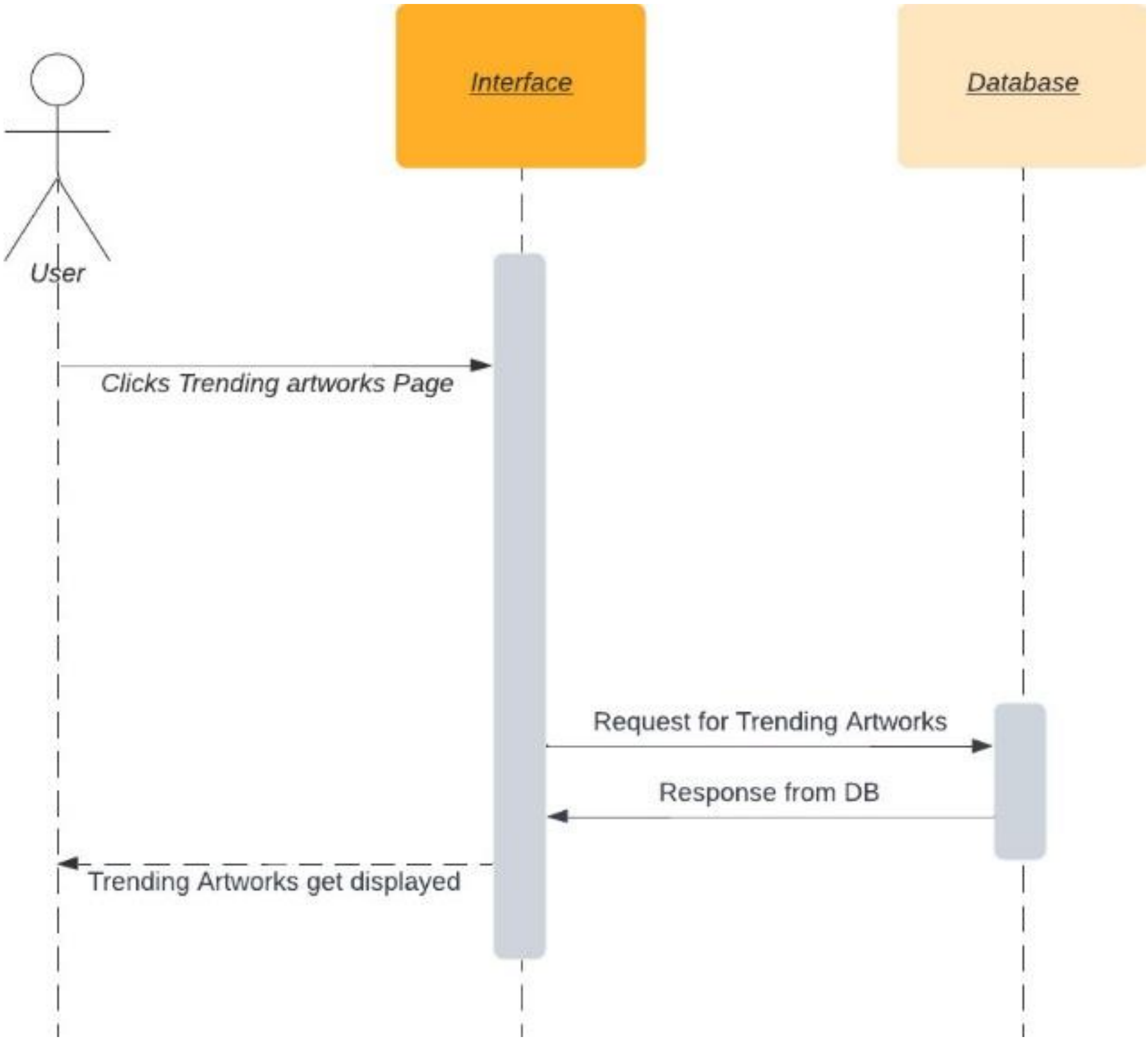

Interface
Database
User
Clicks Trending artworks Page
Request for Trending Artworks
Response from DB
Trending Artworks get displayed

Figure 26 Sequence diagram Trending Artworks

Users Profile:

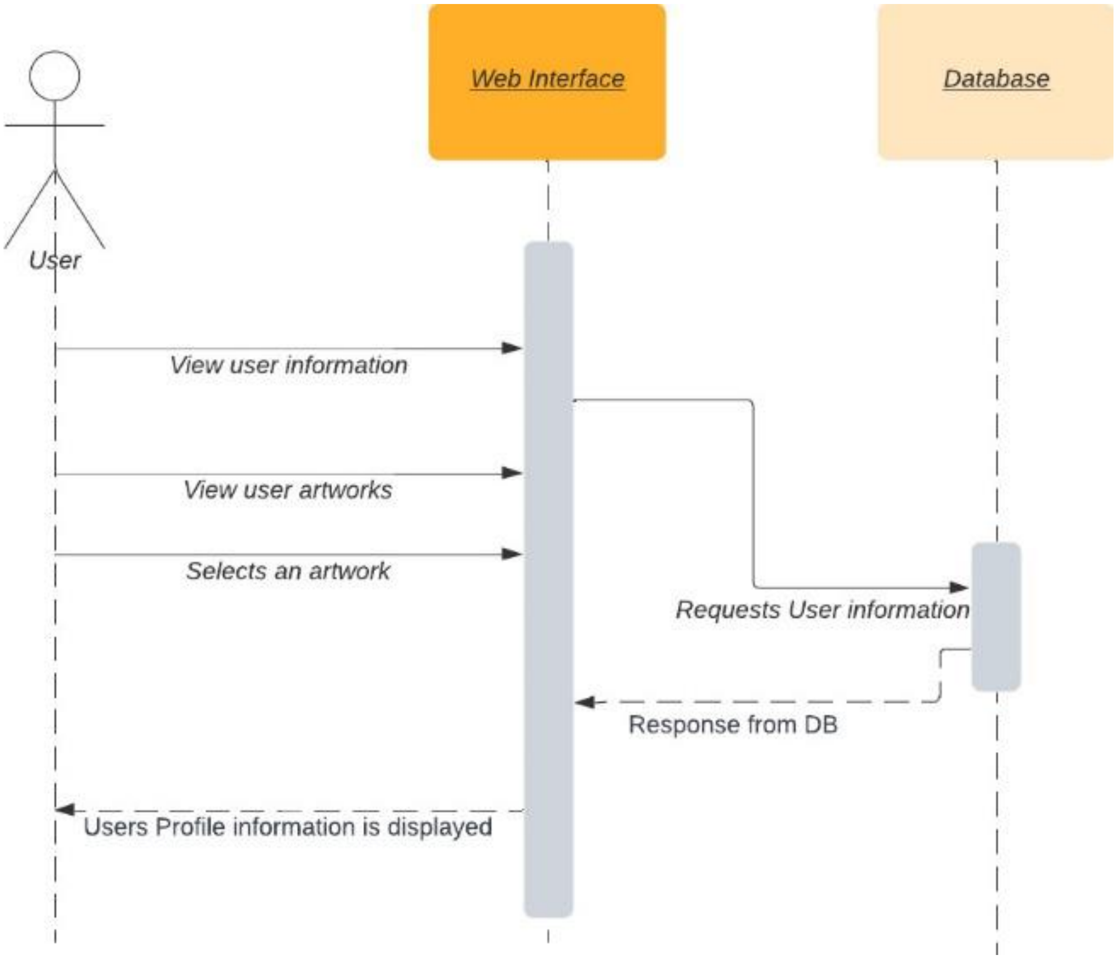


Figure 27 Sequence diagram Users Profile

Sharing the Artwork:

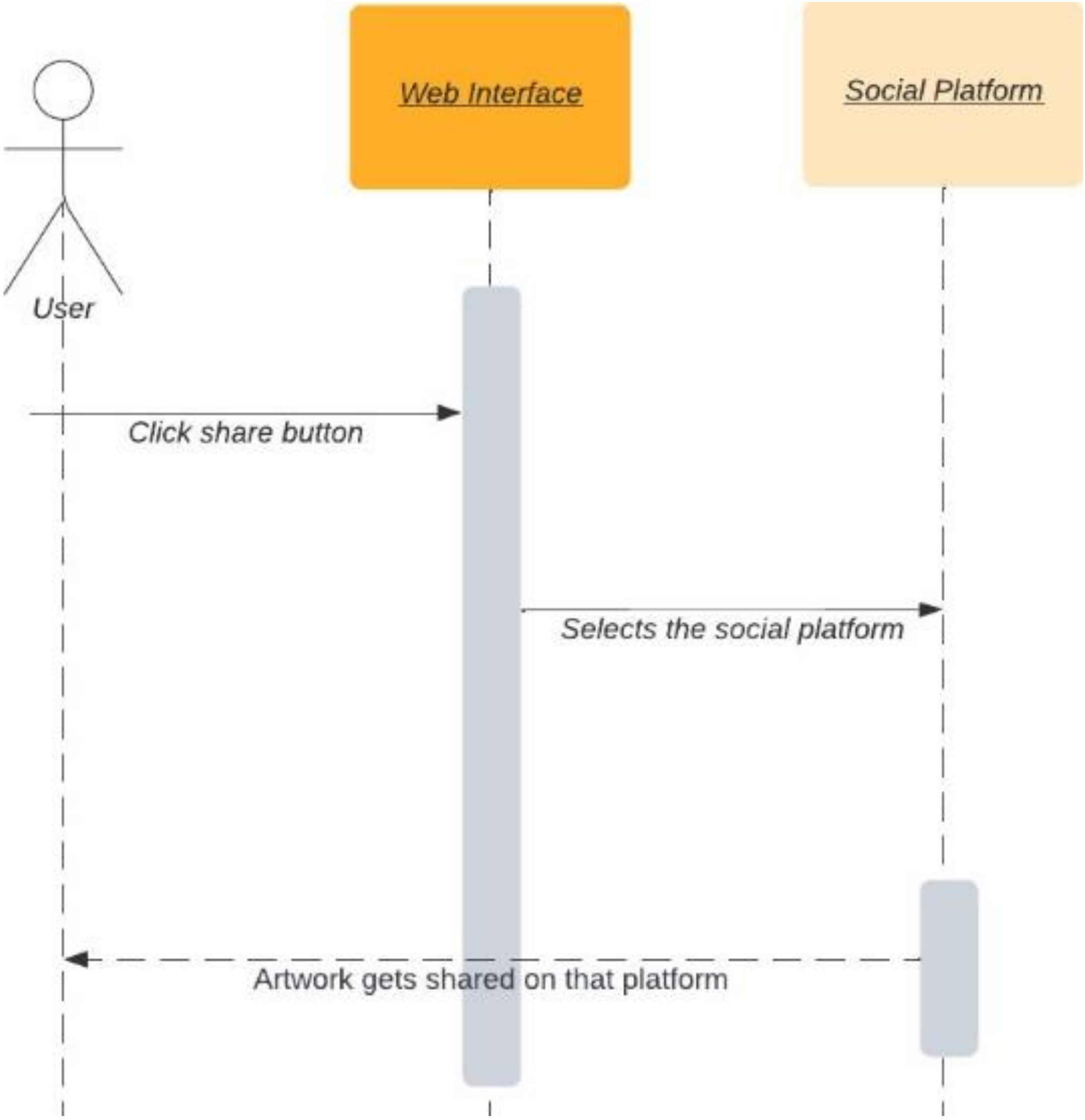

Web Interface
Social Platform
User
Click share button
Selects the social platform
Artwork gets shared on that platform

Figure 28 Sequence diagram Artwork Sharing

Admin panel:

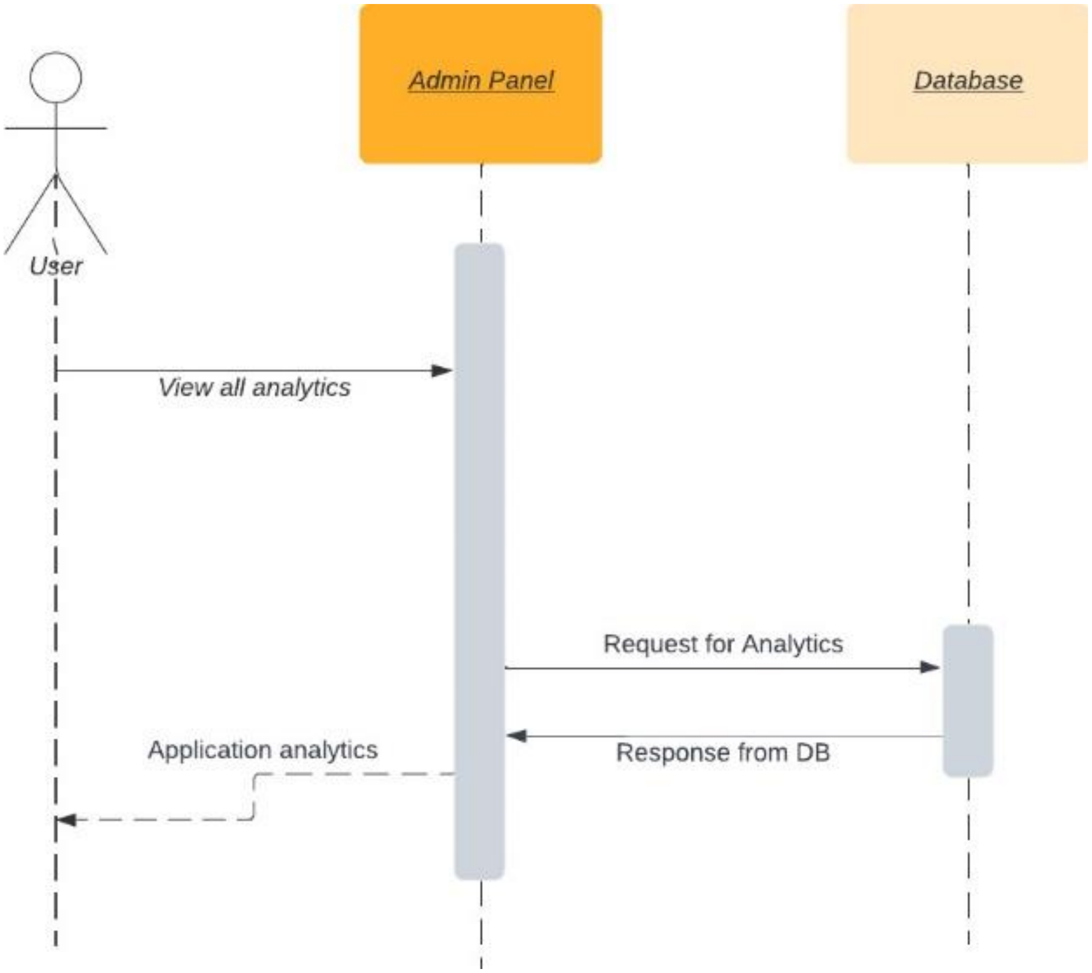


Figure 29 Sequence diagram Admin Panel

3.4 Software architecture

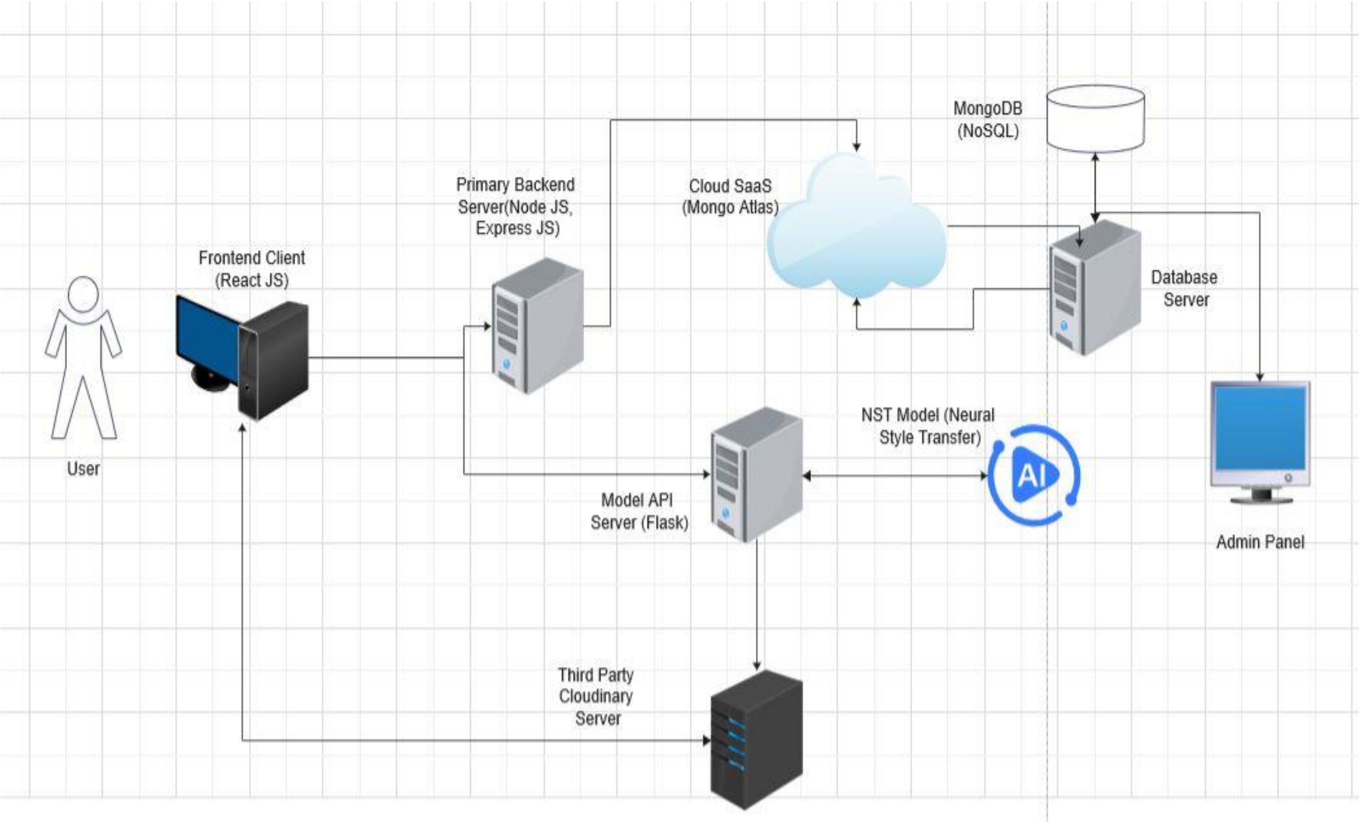


Figure 30 Software Architecture

3.5 Class diagram

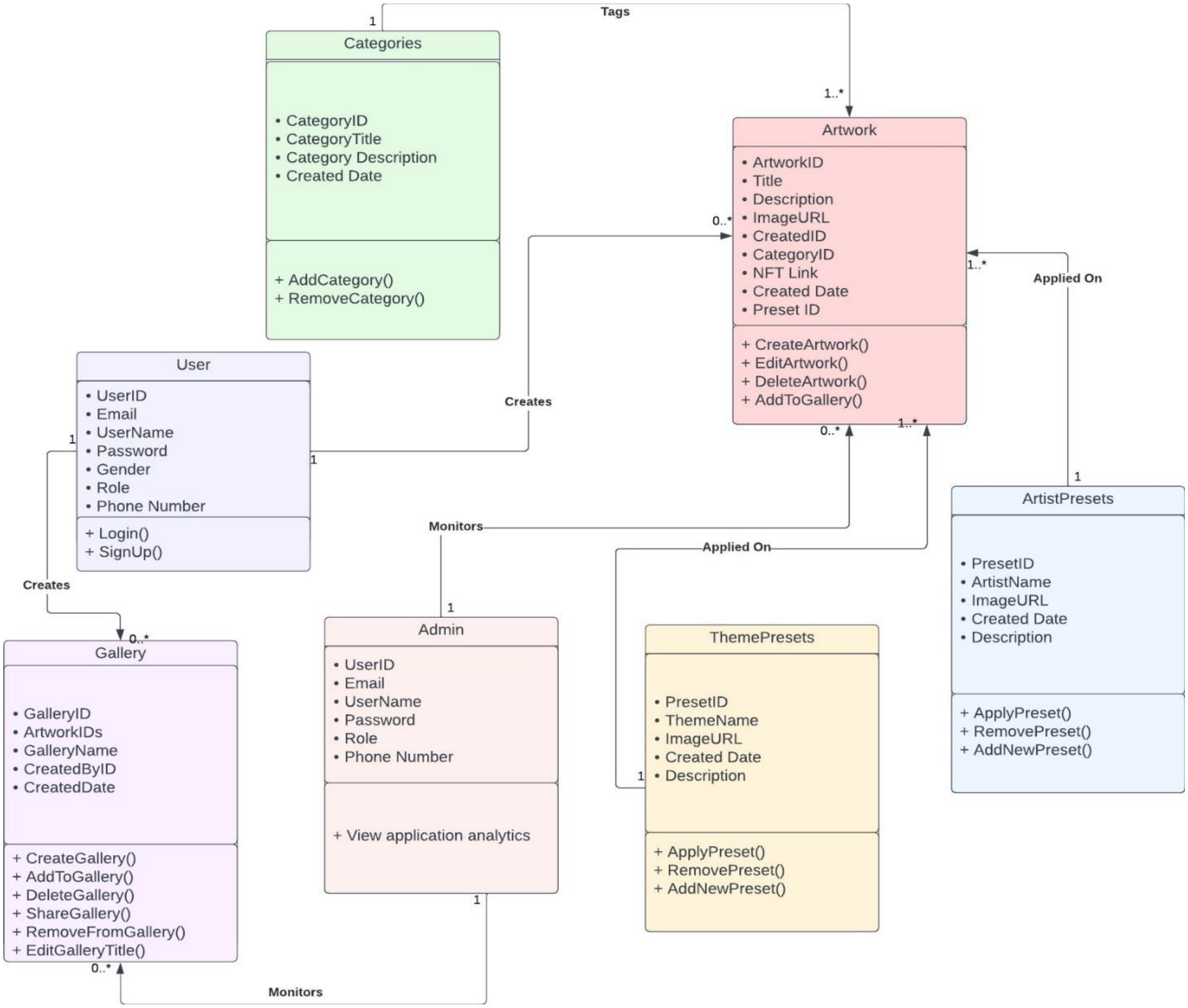


Figure 31 Class Diagram

3.6 Database diagram

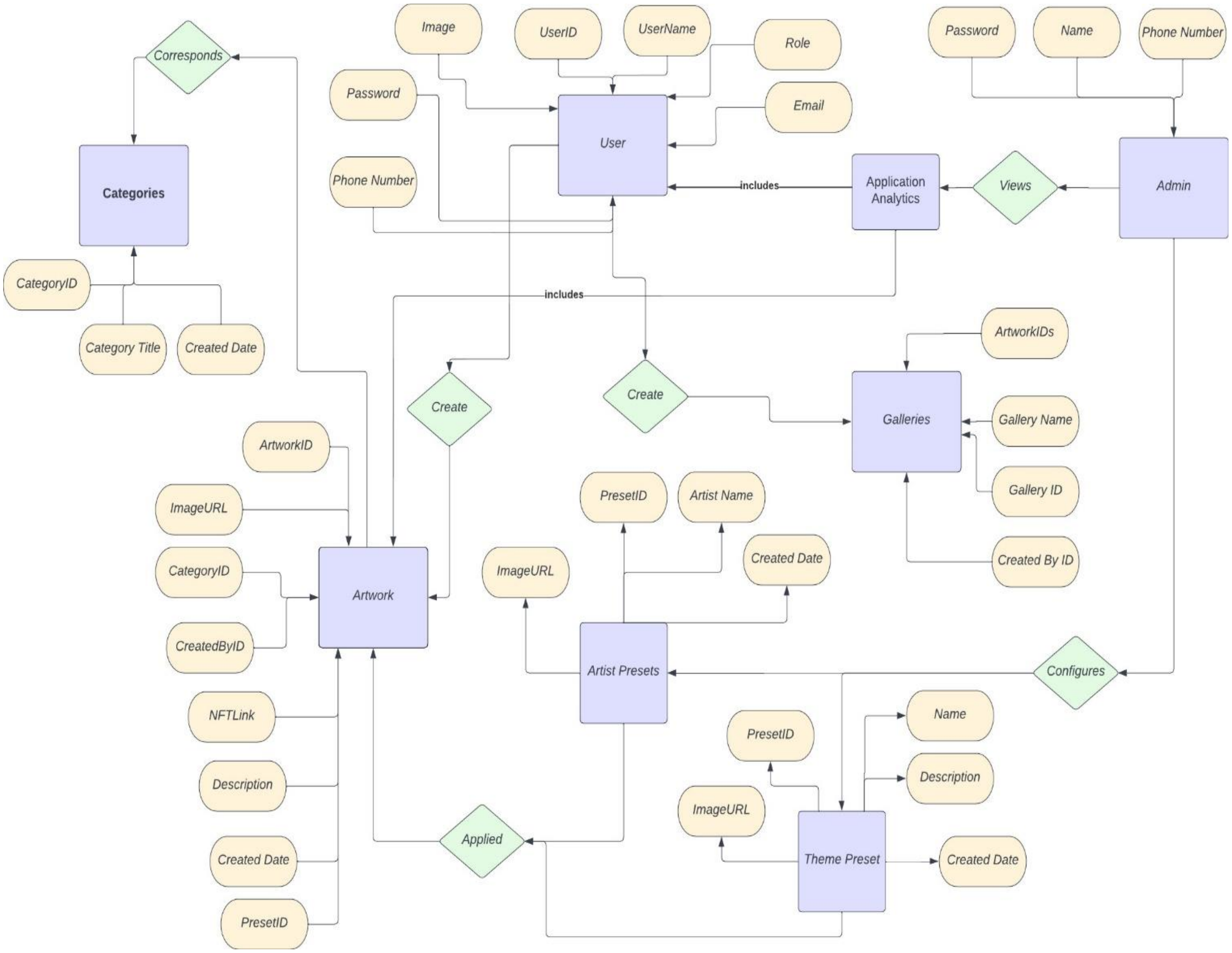
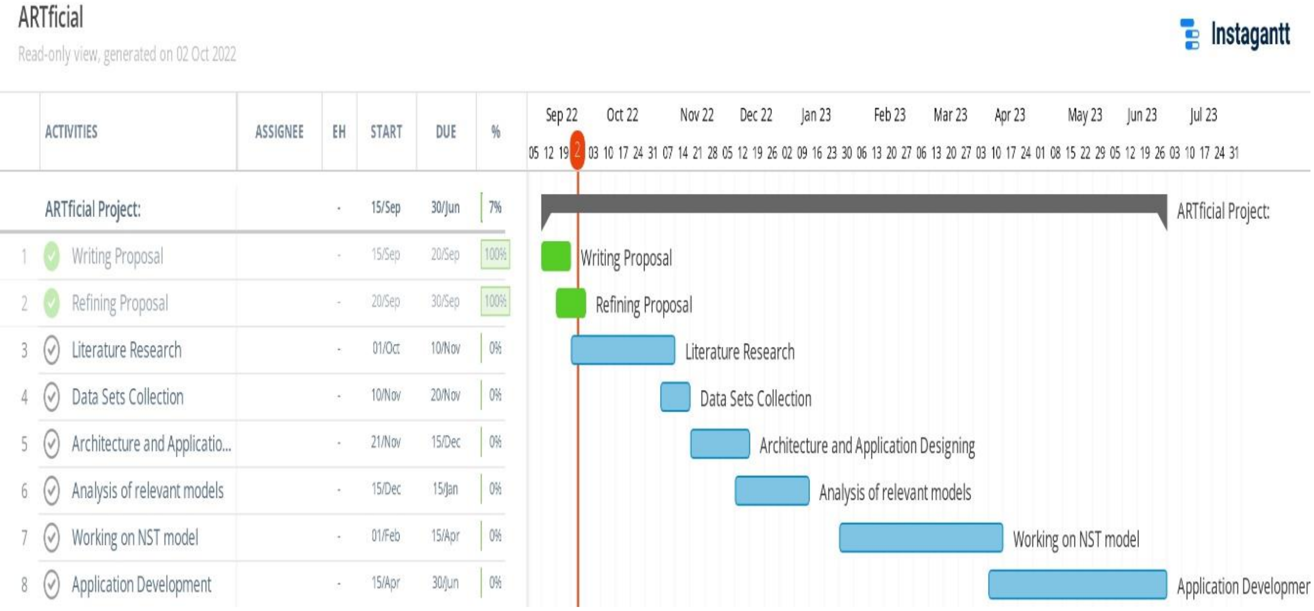


Figure 32 Database diagram

3.7 Network diagram (Gantt chart)



ARTficial

Read-only view, generated on 02 Oct 2022

	ACTIVITIES	ASSIGNEE	EH	START	DUE	%
	ARTficial Project:		-	15/Sep	30/Jun	7%
1	✓ Writing Proposal		-	15/Sep	20/Sep	100%
2	✓ Refining Proposal		-	20/Sep	30/Sep	100%
3	✓ Literature Research		-	01/Oct	10/Nov	0%
4	✓ Data Sets Collection		-	10/Nov	20/Nov	0%
5	✓ Architecture and Applicatio...		-	21/Nov	15/Dec	0%
6	✓ Analysis of relevant models		-	15/Dec	15/Jan	0%
7	✓ Working on NST model		-	01/Feb	15/Apr	0%
8	✓ Application Development		-	15/Apr	30/Jun	0%

Figure 33 Network Diagram

CH 04

4 System Testing

4.1 Test Cases

Table 10 Test case 01: Signup

Steps	Actions	Goals
1	User clicks signup button.	Signup form is opened
2	User provides all mandatory valid input fields.	New user is successfully registers.
3	User enters wrong/ incomplete text fields.	Users is not registered

Table 11 Test case 02: Login

Steps	Actions	Goals
1	User clicks login button.	Opens login form
2	User provides all the mandatory valid input fields.	User is successfully logged in to the platform.
3	User enters wrong/ incomplete credentials in the text fields.	Users is not logged in to the platform.

Table 12 Test case 03: Artwork Feed page

Steps	Actions	Goals
1	User opens Artwork Feed Page.	Array of artworks should be displayed
2	User clicks specific Artwork component.	Clicked artwork's details page should be opened.

Table 13 Test case 04: Artwork Generation

Steps	Actions	Goals
1	User provides content and style reference image	Both images are sent to the NST model for image manipulation.

2	User selects a style reference from different artists presets.	Selected style reference is applied to the content image.
3	User provides incomplete images to NST model.	NST model doesn’t generates any stylized image.

Table 14 Test case 05: Artwork Collection

Steps	Actions	Goals
1	The user makes an artwork Collection.	New artworks collection should be created.
2	User should be able to rename a collection	The collection should be renamed successfully.
3	The user deletes a collection	The collection should be successfully deleted.

Table 15 Test case 06: Text based image generation

Steps	Actions	Goals
1	The user describes required image in textual format in the text input field.	Requested image by user should be generated by Open Ai API.
2	User does not specify the proper textual description for image in text field.	User’s requested image that is generated would be vague.

Table 16 Test case 07: Admin Panel

Steps	Actions	Goals
1	The admin enters his credentials in the form	He should be able to login as an admin.
2	The admin opens the application statistics dashboard.	The admin should be able to view the application statistics.

4.2 Unit / integration / acceptance testing

Unit Testing

1. In unit testing procedure, all of the functionalities are treated as different units of the application.
2. The test cases were designed based on the individual units and functionalities of a software application.
3. This is smallest level of testing where each individual unit of the system is tested for the proper functioning and consequently staged to the next testing phase.

4. If any unit has failed while unit testing procedure, it is easier to reconsider the development aspect of that particular unit rather than review the code of whole application which is a long procedure.

Integration Testing

1. Integration testing involves testing the application after all the individual units are integrated into a single unit which is the final application.

2. In our application, the backend is tested via postman for all the API requests, the frontend is tested for valid rendering of the components and pages and the model is tested for its integration with the application via the API.

3. Finally, the complete platform is tested by running it through all the use cases.

Acceptance testing

1. The application is deployed in a test environment, and the end users are given access to the system.
2. The end users test the system by performing various tasks, such as registering on the platform, uploading images for manipulation, artwork generation.
3. The end users report the issues or problems that they faced while the testing process.
4. The development team reviews those issues and resolves them in this phase.
5. Once all issues have been resolved and the end users are satisfied with the system, application is deployed in a production environment and made available to all users.

CH 05

5 Conclusion

5.1 Problems faced and lessons learned:

- Creating a schema for such a large-scale application was a difficult task since the scale of project was pretty large and the relation between all entities was a challenging task to design.
- Doing research work was also challenging since we hadn't worked on such type of research project ever before. So, gathering relevant research papers and reviewing their contribution in the NST space was also a big challenge.
- Another challenge was the integrating of NST model with the application since both were implemented in different languages and frameworks.
- Finally, the training of the model was another challenge since our goal was to train the model for best possible accuracy so that users can get their desired resultant stylized image.
- Most of the lessons learned are in the research area, like finding the most relevant research papers and extracting their contributions from it, but our supervisors were extremely helpful in this phase and it is only possible due to their guidance that we are developing a Research and Development project.

5.2 Project Summary:

The project will be a complete platform where users can register for an account by signing in. After successfully registering, they will be able to use the Neural Style Transfer technique for image manipulation. Firstly, they will provide a content image (an image on which the style will be eventually transferred) and a style reference (whose style will be applied on the content image). For Style Reference, they can either specify their own style reference or they can also use any of the platform provided style presets which will be based on different artists and themes.

These two images will be sent to the NST model to analyze the style and content of both images and generate a new image that combines the content of the first image with the style of the second image. The stylized image, generated by the model will a merger of both content and style reference images having style features of the style refence to be mapped on the user provided content image.

Furthermore, the platform will provide additional feature such as creation of artwork galleries which could be edited, deleted and renamed. The platform will provide the sharing feature to the users so that they can share their artwork and gallery with others on the social platforms like Twitter, WhatsApp and Facebook.

5.3 Future work:

The application is currently developed with transferred learning approach where we are using a pretrained model “magenta”, from TensorFlow hub for the purpose of transferring style using the NST technique. In FYP 2 we plan to train our own NST model which will eventually be replaced with the pretrained model. So, the final application after the completion of FYP 2 will use our own trained model for style transferring. Furthermore, on the development side some additional features will be introduced like Admin Panel for monitoring the analytics of the platform. After Fyp2, we plan to develop the mobile application of this platform for make it available for mobile users for installing it from the play store/apple store. This mobile application will have a better trained model than existing and will include all the revisions from the feedback of web platform.

6 References:

1. Li, R. (2021, October). Image Style Transfer with Generative Adversarial Networks. In Proceedings of the 29th ACM International Conference on Multimedia (pp. 2950-2954).
2. Singh, A., Jaiswal, V., Joshi, G., Sanjeeve, A., Gite, S., & Kotecha, K. (2021). Neural Style Transfer: A Critical Review. *IEEE Access*.
3. Wang, P., Li, Y., & Vasconcelos, N. (2021). Rethinking and improving the robustness of image style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 124-133).
4. Ubhi, J. S., & Aggarwal, A. K. (2022). Neural Style Transfer for image within images and conditional GANs for destylization. *Journal of Visual Communication and Image Representation, 85,* 103483.
5. Wei, T., & Zhu, L. (2021, April). Comic style transfer based on generative confrontation network. In *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)* (pp. 1011-1014). IEEE.
6. Kotovenko, D., Wright, M., Heimbrecht, A., & Ommer, B. (2021). Rethinking style transfer: From pixels to parameterized brushstrokes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12196-12205).
7. Bai, Y., Wang, C., Yuan, C., Fan, Y., & Wang, J. (2022). Improving the Latent Space of Image Style Transfer. *arXiv preprint arXiv:2205.12135*.
8. "Adobe Photoshop" (Adobe Inc, 1990). Windows/MacOS
9. "Prisma" (Prisma Labs Inc, 2016). Android/iOS
10. "DeepArt.io" (DeepArt UG, 2015). Web