**Artificial intelligence document**

**Submitted to: Dr. Sajida**

BSSE-6-A

Abubakar Nadeem 21-NTU-CS-1291

Kainat Murtaza 21-NTU-CS-1323

Mahnoor Ijaz 21-NTU-CS-1329

**Imagine ai**

**INTRODUCTION**

ImagineAI is an innovative application harnessing AI capability for Android, web, and desktop platforms. Utilizing advanced deep learning algorithms, ImagineAI seamlessly converts text inputs into striking visual compositions. This fusion of artificial intelligence and creativity empowers users to effortlessly generate captivating images from simple text prompts.. Whether on their Android device, web browser, or desktop, users can explore limitless creative possibilities with ImagineAI.

**Problem background**

1. **Precision in Image Search:**

Traditional image search engines often struggle to provide precise results based on specific textual prompts. This limitation arises from the reliance on keyword matching and metadata, which may not capture the nuanced context or desired elements of an image. As a result, users may encounter frustration and inefficiency when attempting to find visuals that accurately reflect their needs or preferences.

1. **Image Quality:**

Images obtained through search engines or user-generated content platforms may suffer from deficiencies in quality or composition, impacting their suitability for professional or commercial use. Factors such as poor lighting, improper framing, or low resolution can detract from the overall appeal and usability of an image, diminishing its effectiveness in conveying intended messages or concepts.

**Proposed Solutions**

1. **AI-Powered Image Generation:**

Our solution leverages artificial intelligence (AI) algorithms to generate images based on specific textual prompts provided by users. By training machine learning models on vast datasets of images and associated text, we enable our system to understand and interpret complex semantic relationships, resulting in more accurate and relevant image generation. This approach circumvents the limitations of traditional keyword-based search methods, offering users a precise and tailored visual experience.

1. **Image Enhancement Techniques:**

To address issues related to image quality and composition, we employ advanced image enhancement techniques within our platform. These techniques encompass various computational photography and image processing algorithms designed to correct common imperfections such as exposure discrepancies, noise reduction, and geometric distortions. By automatically optimizing images for clarity, color fidelity, and aesthetic appeal, we ensure that users receive high-quality visuals that meet professional standards, regardless of the original capture conditions.

**Objectives**

1. **Image Generation:**
   * Develop capabilities for generating high-quality images based on specific textual prompts, leveraging AI algorithms to accurately interpret and represent the given descriptions.
   * Ensure that the generated images are diverse, creative, and aligned with the user's input, providing a wide range of visual options for various scenarios and preferences.
2. **Attention-Grabbing Enhancement:**
   * De-emphasize distracting background elements to direct focus towards the subject, enhancing its visual prominence and differentiation.
   * Apply contrast, color enhancement, and selective focus effects to increase the subject's visual appeal and captivate viewers' attention effectively.

# Functional Requirements

## **User Authentication:**

Users must be able to create accounts and log in securely.

## **Text-to-Image Generation:**

Users can input text prompts to generate images.

## **Image Enhancement:**

Users can enhance existing images by improving quality and composition.

## **Image Expansion**:

Users can expand images to improve subject centrality and focus.

## **Platform Compatibility:**

The application must function seamlessly across Android, web, and desktop platforms.

# Non-Functional Requirements

## **Performance:**

The application should load quickly and process image generation and enhancement tasks efficiently.

## **Scalability:**

The system should be able to handle an increasing number of users and data without performance degradation.

## **Security:**

Ensure the security of user data and prevent unauthorized access.

## **Usability:**

The application should have an intuitive and user-friendly interface.

# Quality Attributes

## **Reliability:**

The application should function reliably without crashes or errors.

## **Maintainability:**

The codebase should be maintainable and well-documented to facilitate updates and bug fixes.

## **Portability:**

The application should be portable across different operating systems and devices.

## **Interoperability:**

The application should be able to integrate with other systems and platforms if needed.

**Scope**

1. **Artistic Generation and Design:**
   * Generating artworks and exploring their application in design and other creative processes.
   * Providing tools for artists and designers to experiment with AI-generated visuals in their work.
2. **Educational and Creative Tools:**
   * Developing educational tools that utilize AI-generated images to facilitate learning and creativity.
   * Creating interactive platforms for users to explore and experiment with generative models in a user-friendly manner.
3. **Research on Generative Models:**
   * Conducting research on the capabilities, limitations, and biases of generative models.
   * Investigating novel techniques and approaches to improve the performance and robustness of AI-generated images.
4. **Safe Deployment:**
   * Ensuring the safe deployment of generative models to prevent the generation of harmful content.
   * Implementing safeguards and controls to mitigate the potential risks associated with AI-generated imagery.

**Excluded Uses**

The project does not aim to produce factual or true representations of people or events. Therefore, using the platform to generate such content is beyond its intended scope.

Limitations:

1. The platform may not achieve perfect photorealism in generated images.
2. Generating legible text may pose a challenge for the platform.
3. Complex compositional tasks, such as rendering specific scenes or objects, may be challenging for the platform.
4. Faces and human figures may not be accurately generated due to limitations in the model.
5. The auto encoding part of the platform may result in loss in image compression.

**Methodology**

1. **Development Using Flutter Framework:**
   * Utilize the Flutter framework to develop ImagineAI, ensuring cross-platform compatibility and consistent user experience across Android, web, and desktop platforms.
   * Leverage Flutter's flexibility and performance capabilities to create an intuitive and responsive user interface for interacting with the image generation functionality.
2. **Integration of Deep Learning Model:**
   * Integrate a pre-trained deep learning model into the Flutter application to power the core image generation functionality.
   * Implement the model seamlessly within the application architecture to ensure reliable performance across different platforms.
   * Utilize an ensemble of experts pipeline for latent diffusion, where the base model generates noisy latent, which are then refined using a separate refinement model.
3. **Diffusion-based Text-to-Image Generative Model:**
   * Employ a diffusion-based text-to-image generative model to generate and modify images based on text prompts.
   * Utilize two fixed, pertained text encoders (OpenCLIP-ViT/G and CLIP-ViT/L) to encode textual descriptions into latent representations that can be used by the generative model.
4. **Model Description:**
   * Implement a Latent Diffusion Model that utilizes the pretrained text encoders to encode textual prompts into latent representations.
   * Use the latent representations to generate images that correspond to the provided text prompts, leveraging the capabilities of the diffusion-based generative model.
   * Allow users to modify generated images based on additional text prompts, enabling iterative refinement and customization of the generated visuals.
5. **Testing and Evaluation:**
   * Conduct thorough testing of the ImagineAI application to ensure its functionality, usability, and performance across different platforms.
   * Evaluate the accuracy and quality of the generated images produced by the deep learning model, comparing them against the provided text prompts.
   * Gather user feedback to identify areas for improvement and refinement in both the application interface and the image generation process.

**API Implementation**

if (response.statusCode == 200) { // Check if the content type indicates binary data (image)

if (response.headers['content-type']?.contains('image/jpeg') == true || response.headers['content-type']?.contains('image/png') == true)

{ return response.bodyBytes; }

else { print( 'Unexpected response format: ${response.headers["content-type"]}');

print('Response Body: ${response.body}'); } }

else if (response.statusCode == 503) { // Handle the service unavailable error

final Map<String, dynamic> responseBody = json.decode(response.body); final double estimatedTime = responseBody['estimated\_time'];

print( 'Model is loading. Estimated time: ${estimatedTime.toStringAsFixed(2)} seconds.');

print('Please retry after the estimated time.'); }

else {

print('HTTP Error: ${response.statusCode}'); #

print('Response Body: ${response.body}'); } }

catch (e) { // Handle network and other unexpected errors

print('Network or unexpected error: $e'); } throw Exception('Failed to query model'); }}

**API Code Description**

**Http Request Handling:**

* The code is likely part of a function or method that takes an HIIP response object (response) as input.
* If checks the status code of the response using response.statusCode.

**Response Handling based on Status Code:**

* If the status code is 200, it further checks the content type of the response to

determine if it's an image (JPEG ur PNG). if it's an image, it returns the body of the response as bytes.

* If it's not an image, it prints out the content type and body of the response, Indicating that the format was unexpected.

**Handling Specific HTTP Status Codes:**

* If the status code is 503 (Service Unavailable), it assumes there's a model loading and extracts the estimated time from the response body, printing it out for the user to see.
* If the status code is anything else, it simply prints out the HTTP error code and the response body.

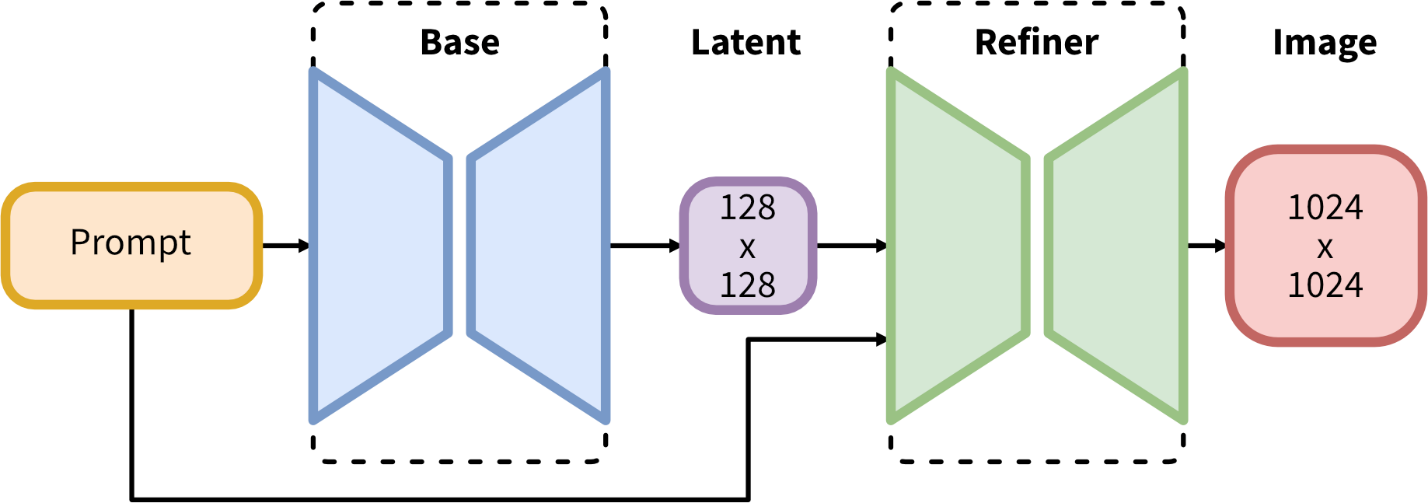
**Error Handling:**

* The code is wrapped in a try-catch block to handle any network- related or unexpected errors that might occur during the III IP request.
* If an error occurs, it prints out the error message.

**Throwing an Exception:**

* At the end of the function, there's a line that throws an exception with the message 'Failed to query model. This suggests that if none of the conditions above are met (i.e., the response status code is neither 200 nor 503, or there's an error during the request), the function will throw this exception.

**Framework**



**Latent Diffusion Model**

* **Latent Space**: This refers to a compressed representation of data. Instead of working directly with high-dimensional

images, the model transforms the images into a lower-dimensional latent space using an auto encoder.

This makes the learning and generation processes more efficient.

* **Diffusion Process**: This involves a forward and reverse process. In the forward process,

data is gradually turned into noise. In the reverse process, the model learns to denoise this noisy data back to its

original form.

**Training Procedure**

* **Auto encoder**: The auto encoder encodes images into latent space (a lower-dimensional space) and then decodes them

back to the original image space. This helps in efficiently learning the patterns and structures in the images.

* **Diffusion Process in Latent Space**: Instead of applying the diffusion process directly to high-dimensional image data,

Stable Diffusion applies it to the latent representations created by the auto encoder. This reduces computational

complexity and focuses on the most important features of the data.

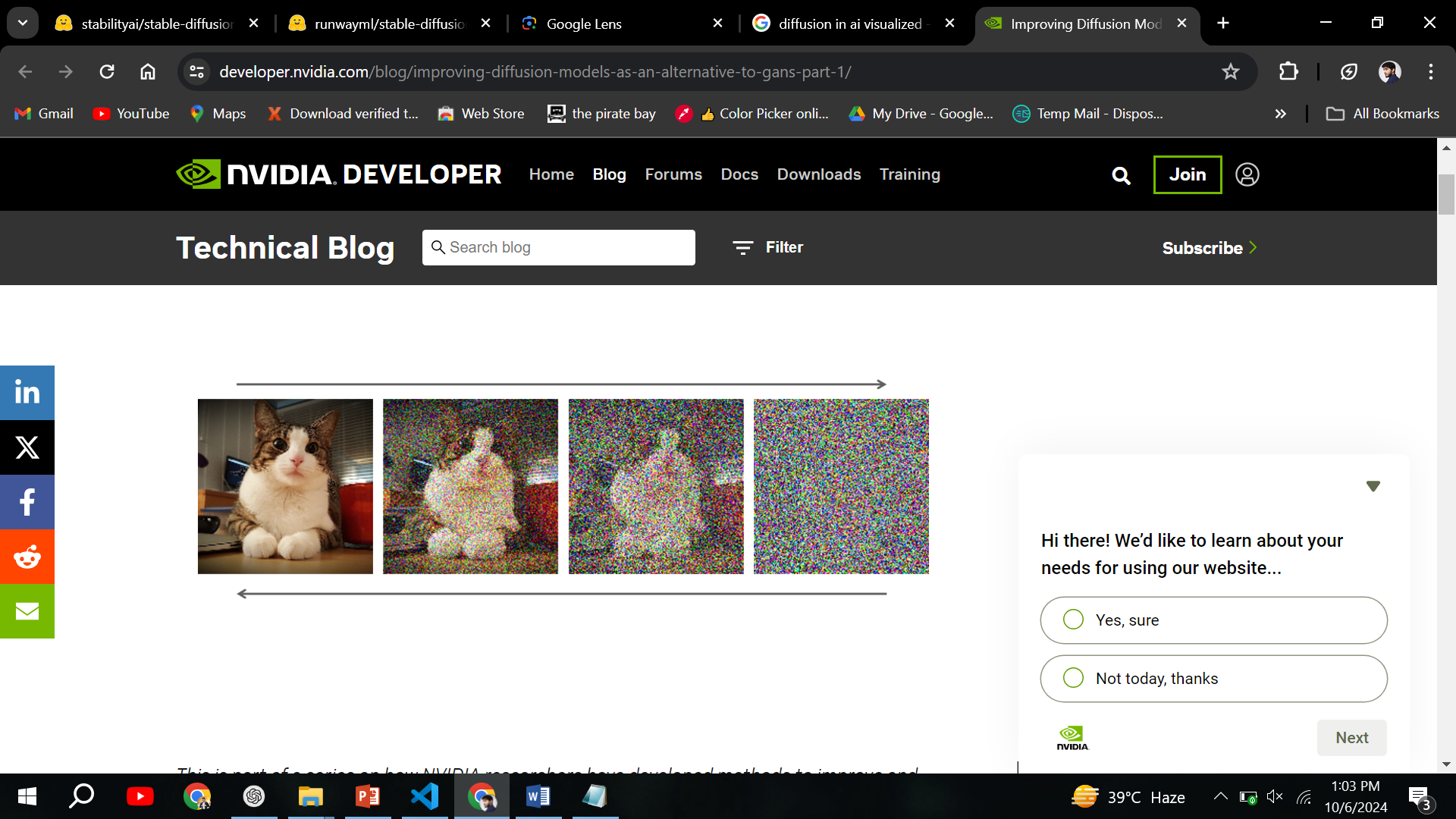
**Steps of Diffusion**

* **Forward Process**: The original image is incrementally corrupted by adding noise at each step, resulting in a fully noisy

image.

* **Reverse Process**: The model is trained to remove this noise step-by-step, reconstructing the image from the noise.

This denoising is guided by the input text prompt, ensuring that the generated image matches the desired description.



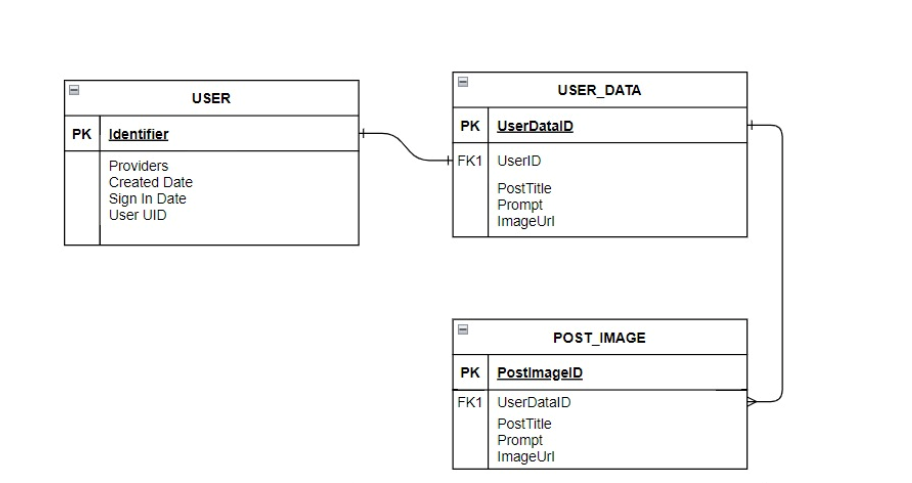
**Dataset Description**

The dataset utilized for image generation and expansion in ImagineAI comprises text prompts paired with corresponding images. This dataset serves as the foundation for training and evaluating the deep learning models responsible for generating and modifying images based on textual descriptions.

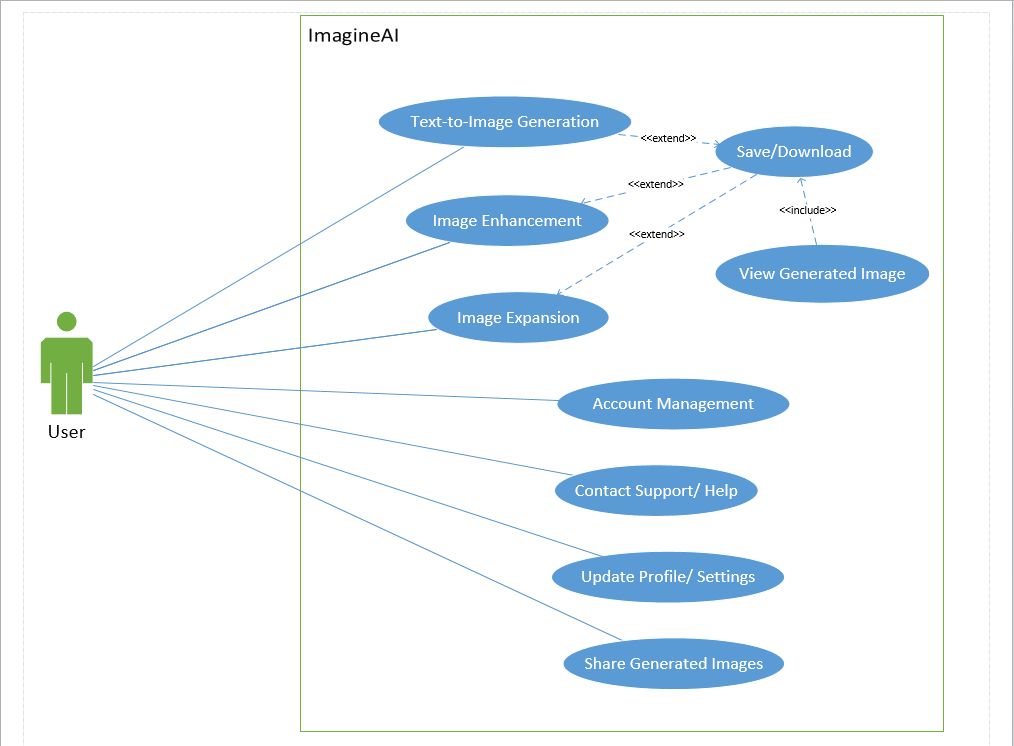
1. **Text Prompts:**
   * The dataset includes a diverse range of textual descriptions or prompts that serve as input for the image generation and expansion processes.
   * Text prompts may describe various scenes, objects, concepts, or scenarios, providing detailed instructions or creative inspirations for generating corresponding images.
   * Prompts are carefully curated to encompass a broad spectrum of themes, styles, and contexts, ensuring the versatility and adaptability of the generative models.
2. **Paired Images:**
   * Each text prompt in the dataset is associated with one or more corresponding images generated or expanded based on the provided description.
   * Images may vary in style, composition, and content, reflecting the richness and complexity of the textual prompts they represent.
   * Paired images are meticulously curated and annotated to facilitate model training, validation, and evaluation processes.
3. **Metadata:**
   * The dataset may include additional metadata such as image labels, annotations, or categorizations to provide context and insights into the content and characteristics of the paired images.
   * Metadata attributes may include image dimensions, file formats, timestamps, or other relevant information to support data analysis and management tasks.
4. **Data Preprocessing:**
   * Prior to model training, the dataset undergoes preprocessing steps to ensure data consistency, quality, and compatibility with the deep learning models.
   * Textual prompts may be tokenized, cleaned, and encoded into numerical representations suitable for input into the generative models.
   * Images may be resized, normalized, or augmented to standardize their dimensions and enhance model performance and robustness.

**Diagrams**

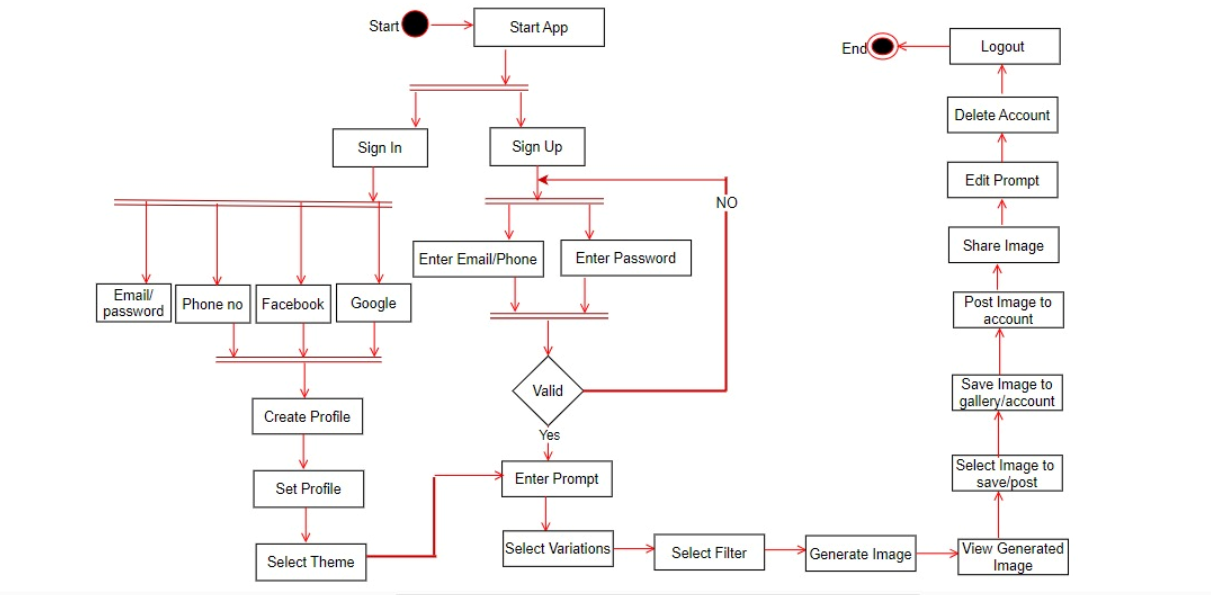
**Entity Relation Diagram**



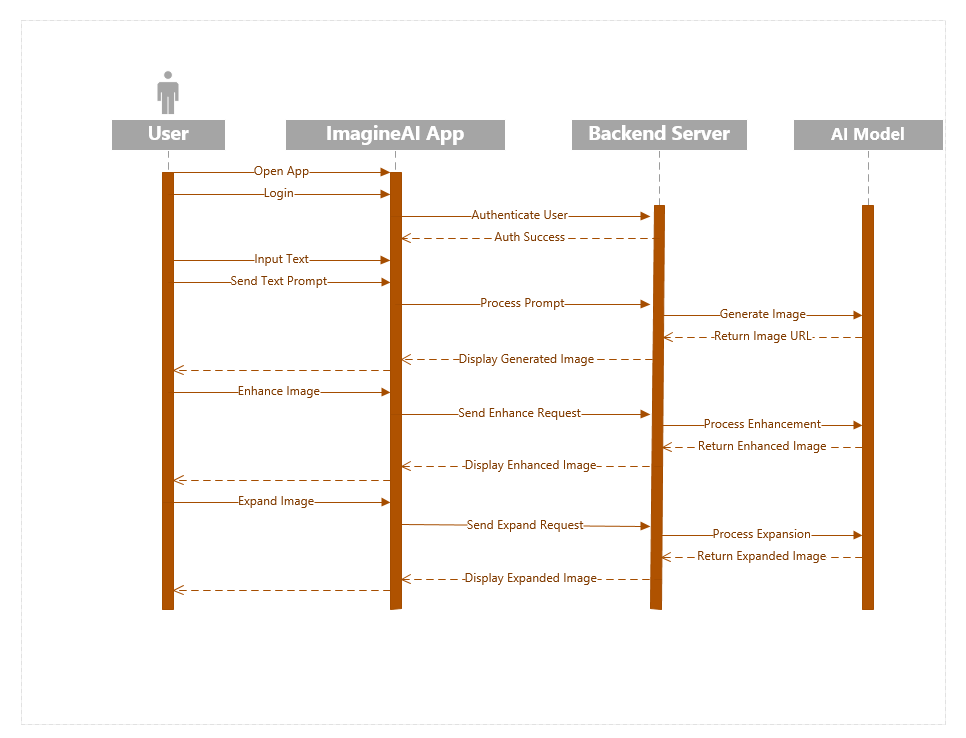
**Use Case - Diagram**

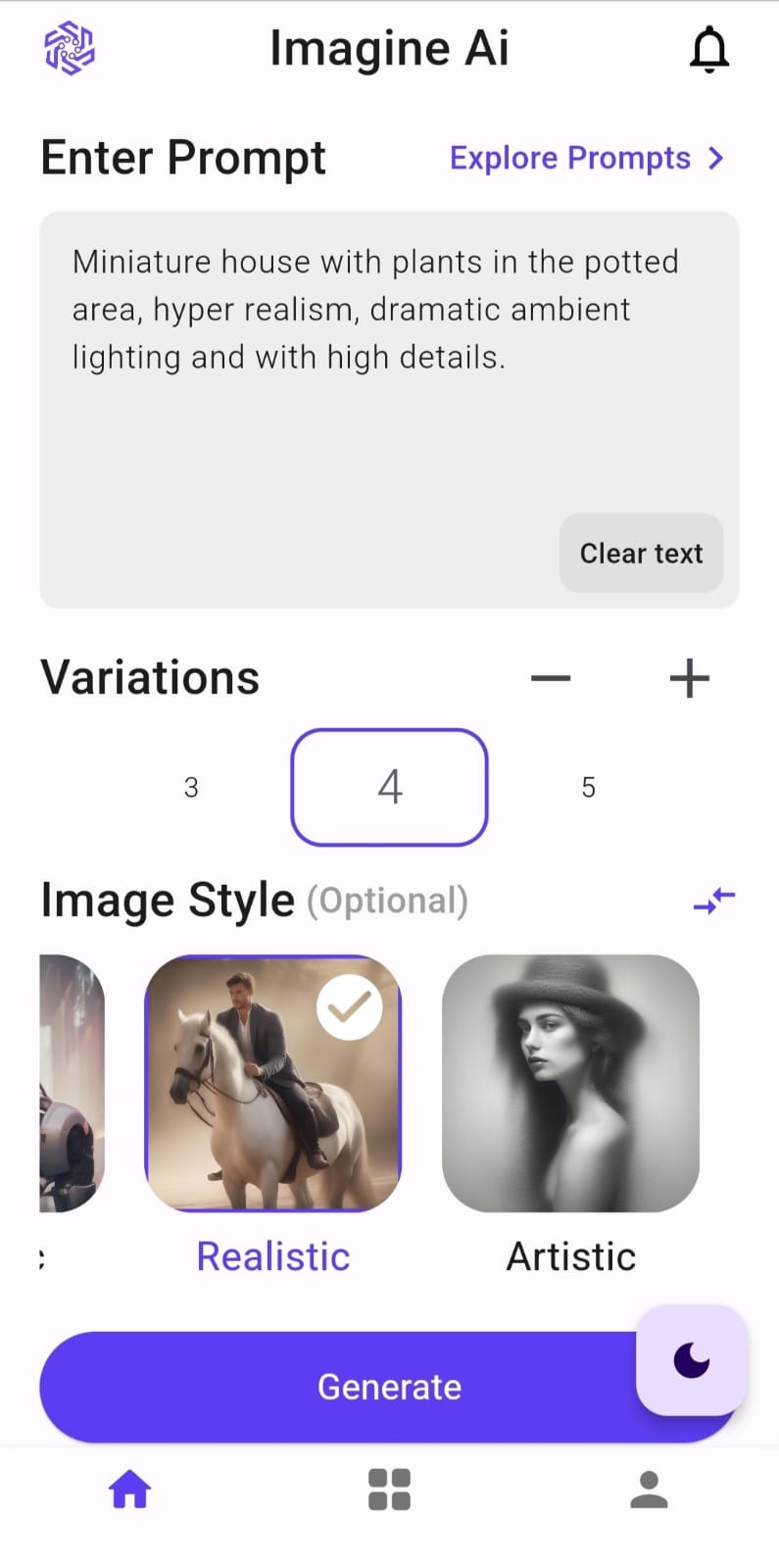


**Activity Diagram**



**Sequence Diagram**



**Results**

