ÇANKAYA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

CENG 407 – Software Requirements Specification

Artificial Intelligence Based Art Gallery Mobile Application

Team Members:

Mehmet Emre KILINÇ 202111058

Mertcan ZAFER 202011076

Batuhan ÖZER 202011072

Emre Şahin DEMİRBAŞ 202011020

Tunahan GÜLTEKİN 202011005

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed overview and description of the artificial intelligence-based art gallery mobile application. This document includes detailed information about the requirements of the project. It reflects the identified constraints and proposed software architecture.

1.2 Scope

This mobile application, whose main feature is to enable users to create artworks and story for artworks using artificial intelligence models and share them on their profiles, also aims to increase the use of the basic features of the application with the gamification feature.

2. General Description

2.1 Glossary

Definitions	
Application Programming Interface (API)	An API is a set of rules, tools, and protocols that allow different software applications to communicate with each other.
Content Delivery Network (CDN)	A Content Delivery Network (CDN) is a system of servers located in different parts of the world that deliver content like images, videos, and website files to users more quickly.
Hamburger Menu	A Hamburger Menu is a navigation menu that is typically hidden and represented by an icon with three horizontal lines (≡).
Negative Prompt	A negative prompt is a concept used in text-to-image AI applications and allows users to specify what should not appear in the generated image.
Firebase Services	Firebase services is a collection of cloud-based tools and services offered by GOOGLE to assist developers in building, improving, and scaling mobile and web applications.
HyperText Transfer Protocol Secure (HTTPS)	HTTPS is an extension of HTTP protocol that is used to securely exchange data over the internet.

2.2 User Characteristics

The application is designed to be highly accessible, enabling use by a broad audience. However, the primary target users are individuals who enjoy creating and sharing images using artificial intelligence and those with limited artistic skills who seek to express themselves visually through the app. This demographic is expected to form the core user base due to the app's ability to empower users to create visually appealing content effortlessly.

2.3 Overview of The Document

This Software Requirement Specification (SRS) document provides a comprehensive description of the requirements for the development of AI Based Art Gallery Mobile Application. It is intended to align all stakeholders, including developers and project managers, with a clear understanding of the system's objectives and functionalities. The document outlines both functional and non-functional requirements, as well as detailed interface specifications, to ensure the system meets the expectations of its users and stakeholders.

The first section covers General Constraints and Assumptions, addressing any limitations or conditions under which the system is expected to operate.

The second section focuses on Specific Requirements and is divided into multiple subsections. The Interface Requirements detail the interactions between the system and its environment, including the user interface, hardware interface, software interface, and communication interfaces. These are followed by the Functional Requirements, which describe the core features and operations of the system. The section also includes Use Cases, which provide practical scenarios illustrating how users will interact with the system and help validate its design by mapping requirements to real-world situations. Lastly, the Non-functional Requirements define essential system attributes such as performance, response time, reliability and security.

2.4 General Constraints and Assumptions

- ✓ The application's artwork generation functionality cannot take more than 500 characters as its prompt.
- ✓ The application could only have a maximum of 50 concurrent users at a time.
- ✓ The application's default language is English.
- ✓ The application has only one user type.
- ✓ The application will not have any sponsored advertisements.
- ✓ The application does not have a paid subscription option.
- ✓ The application will not share any information with 3rd party companies.
- ✓ The application's challenges are periodically updated.
- ✓ The application will not collect current location information.
- ✓ The application has the right to ban any malevolent user without a warning.

3. Specific Requirements

3.1 Interface Requirements

3.1.1 User Interface

1. **Login Page**
   Login page is the first screen that appears when the application is first opened or the first screen that appears after user signs out. There are input text fields for users to enter their usernames and passwords. There is a "Login" button that will log the user, in if the information entered is correct and direct the user to the "Home" page, there is a "I forgot my password" button, which will direct the user to the "Confirmation" page, if the user has forgotten his/her password and if the user does not have an account, there is a "Sign Up" button that will redirect the user to the "Sign Up" page. A paper prototype of this page can be seen in Figure X.

2. **Sign Up Page**
   On this page, there are input text fields for username, email, name, surname and password, which are the information required for users to sign up, and a "Sign Up" button to complete the process. A paper prototype of this page can be seen in Figure X.

3. **Confirmation Code Page**
   This page is an intermediary page for redirecting to the "Change Password" Page, when the user forgets their password or wants to change their password. On this page, users enter their registered e-mail addresses into the input text field and can receive a confirmation code in their e-mails with the "Send Information Code" button, and after entering this code into the other input text field, they are redirected to the "Change Password" page using the "Submit" button. A paper prototype of this page can be seen in Figure X.

4. **Change Password Page**
   On this page, there are input text fields where users can enter their new password and a "Change Password" button. A paper prototype of this page can be seen in Figure X.

5. **Upper Navigation Bar**
   There are 2 pages navigation buttons in the navigation bar. The left button in the navigation bar takes you to the "Challenges and Achievements" page, and clicking the right button takes you to the "Hamburger Menu" Page.

6. **Bottom Navigation Bar**
   There are 5 navigation buttons, which take the user to the related pages, in the navigation bar. Clicking the button in the middle takes the user to the "Home" page, which is also on our main page. When the button to the left of the middle is clicked, it takes the user to the "Discover" page, and the button to the right of the middle takes the user to the "Generate Artwork" page. The leftmost button in the navigation bar takes the user to the "Notifications" page, and clicking the rightmost button takes the user to the "Profile" page.

7. **Home Page**
   Home page is the first screen that appears when the application is opened after the user logs in. On the Home page, posts shared by other users the user follows are listed below each other. This list is sorted by the "Newest" filter by default, and this filter can be changed to "Newest, Top Commented, Daily Best, Weekly Best, Monthly Best" options with the dropdown menu at the top left. The posts on this page have the artwork created by the artificial intelligence at the top, the story for this image created by the artificial intelligence

below it, the prompt entered by the user to create this image below it, and at the bottom of the post, the profile picture and username of the user who shared this post on the left, and the "Comments", "Share" and "Like" buttons on the right, respectively. When the "Like" button is clicked, that is, when the post is liked, the background color of the button becomes red. When the "Comments" button is clicked, this post expands towards the bottom, with a text field at the top where the user can enter their comment, a button for sharing this comment, and all the comments made to this post listed below it. A paper prototype of this page can be seen in Figure X.

8. **Discover Page**

On the Discover page, unlike the "Home" page, users can see posts shared by other users other than the users they follow. On this page, instead of seeing the post in large size, 9 artworks (3x3) can be seen on a page and more can be discovered by scrolling left and right. This page also has the same filtering options as the "Home" page and the number of comments and likes received by the posts is displayed under them. To see the post in more detail, the user can click on the post to make it stand out as it was on the "Home" page (that is, artwork, story, prompt, profile picture and username of the user who shared this artwork, comment, share and like buttons below each other) and blur the background, the user can return to the normal state of this page by touching the blurred part. Users can also search by username and tags with the search bar at the top, and as a result of this search, users are directed to the "Search Results" page. A paper prototype of this page can be seen in Figure X.

9. **Search Results Page**

On this page, users see the results of their searches via the search bar on the "Discover" page. Users with usernames containing the word entered into the search bar appear at the top, and posts with tags matching the word entered into the search bar appear below them. At the top of this page, there is a search bar, just like on the "Discover" page. A paper prototype of this page can be seen in Figure X.

10. **Notifications Page**

The notifications page displays notifications that will come when other users the user follows share something, earn an achievement, or complete a challenge. On the left side of each notification, there is the profile photo of the user that the related with that notification, followed by the content of the notification. When a notification is clicked, the user is directed to the profile page of the other user included in that notification. A paper prototype of this page can be seen in Figure X.
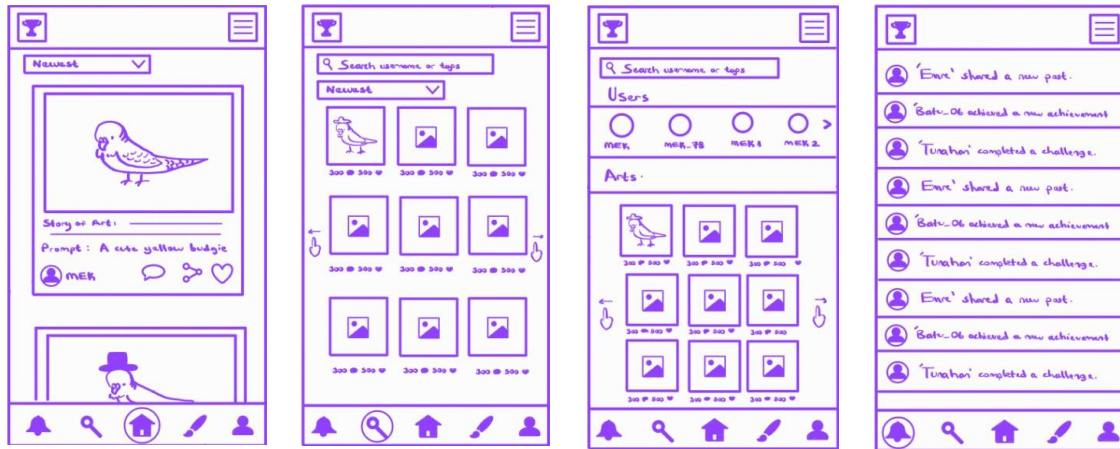
11. Challenges and Achievements Page

Users are directed to this page by clicking the "Achievements" button under the "Hamburger" button at the top right of various pages. The Challenges section is first opened on this page and in this section, under the title "Challenge of the Day", there is a checkbox indicating whether this challenge has been completed and the content of the challenge, and below it, there is "Challenges of the Week" with same order. The paper prototype of this page can be seen in Figure X. The user can go to the Achievements section of this page by clicking the "Achievements" link at the top. In this section, achievements are shown one below the other. The checkboxes of completed achievements are checked, and the uncompleted achievements appear as a progress bar in the background.

12. Profile (User's and Other Users') Page

On the profile page, in the top middle, there is the user's profile photo and username, below them are the number of likes the user has made so far, the number of users this user follows and the number of comments this user has made so far, and below them are the number of people following this user. By clicking on the "Follows" or "Followers" links, the user can be directed to a page where the profile photos and usernames of the users being followed or following are listed. At the bottom of the page, the artworks previously created and shared by this user are located in a similar order to the "Discover" page. Here, as on the "Discover" page, the user can click on the artwork to bring that post to the front for more detailed viewing and blur the background. Users can view the "Profile" pages of other users by clicking on the username that shared the post at the bottom left of a post or on the "Notifications" page. The profile page of other users has the same order and functionality as described above, the only difference is the "Follow" button under the profile picture and username, which allows the logged-in user to follow this other user whose profile they are viewing.

13. Hamburger Menu Page

Users are directed to this page by clicking the "Hamburger" button at the top right of various pages. On this page, there is a "Settings" button that directs the user to the "Settings" page and a "Sign Out" button for the user to log out.

14. Settings Page

This page contains a "Change Password" button that directs the user to the "Confirmation" page, a "Language" dropdown menu that allows changing the language of the application, and a "Dark Mode" checkbox that changes the theme of the application.

15. Generate Artwork Page

On this page, which is one of the most important pages of the application, the user enters the prompt in the first input text field to be able to generate artwork, the negative prompt in the second input text field to indicate that they do not want it to be in the artwork to be created, if there is a preset style (cartoon, realistic, 1950s, anime...) that they want to choose under these, they can select it and under that, they can mark the checkbox whether a story should be created for this artwork to be created or not. After these sections are filled in, when the "Generate" button at the bottom of the page is clicked, the application is directed to the "Artwork Result and Publish" page where it will create and display the artwork.

16. Artwork Result and Publish Page

On this page, the user sees the artwork that will be produced by the system as a result of what s/he entered on the "Generate Artwork" page. Depending on the cloud service, the artwork is not created immediately, and during the time it takes to create it, the loading symbol appears where the artwork would normally appear, and other buttons are disabled during this time. Once the artwork has been created successfully and the buttons are enabled, the user can download the artwork to their device with the button at the bottom right of the area where the artwork is displayed. Depending on the "Add Story" checkbox on the "Generate Artwork" page, there will be a story generated by the system under the area where the artwork is displayed on this page. The user chooses who can see this artwork when he shares it with the checkboxes under the "Story" section by checking the "Public" and "Just Followers" checkboxes. The user enters tags in the "Add Tags" input text field located below this section. After the user enters the required fields, s/he can share his/her artwork and story by clicking the "Share" button at the bottom right of the page.

When artworks are shared, they are added to the "My Artworks" section of the user's profile.
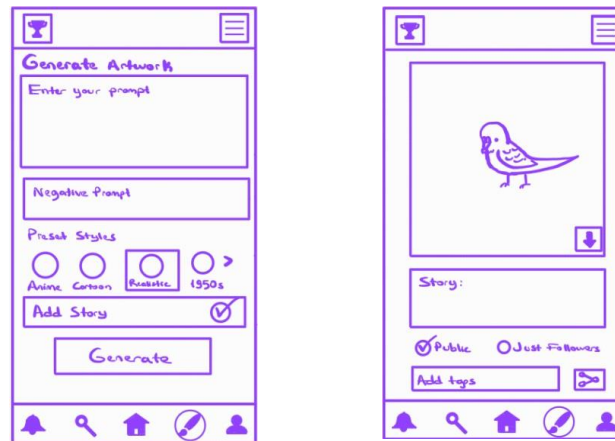
17. Message/Error Pop-Up

When an exception occurs while using the application or when the application needs to give any message, the application shows a pop-up message stating the reason for the error/message. The pop-up disappears when the user clicks on a part of the screen that does not contain a pop-up window.

3.1.2 Hardware Interface

Devices:

- **Client Devices:** The application will run on Android-based smartphones and tablets. It requires access to hardware components such as CPU, GPU, memory, and storage for efficient operation.
- **Cloud Resources:** The backend services, including AI model inference and storage, rely on scalable cloud infrastructure with high-performance GPUs, CPUs, and distributed storage systems.

Specifications:

- **Minimum Client Device Requirements:**
  - o **Operating System:** Android 7.0 or later.
  - o **Processor:** Quad-core CPU with a clock speed of 1.4 GHz or higher.
  - o **Memory:** At least 4GB of RAM.
  - o **Storage:** Minimum of 16GB available storage.
  - o **Graphics:** GPU supporting OpenGL ES 3.0 or higher.

- **Recommended Client Device Requirements:**
  - o **Processor:** Octa-core CPU with a clock speed of 2.0 GHz or higher.
  - o **Memory:** 6GB of RAM or more for smooth multitasking.
  - o **Storage:** 32GB or more internal storage.
  - o **Graphics:** Dedicated GPU or modern integrated graphics capable of rendering real-time previews.

Cloud Hardware Resources:

Compute Instances:

- High-performance GPUs (e.g., NVIDIA A100, Tesla V100, or similar) for AI inference and rendering tasks.
- CPUs with high-core counts (e.g., AMD EPYC or Intel Xeon processors) for API processing and other backend services.

Memory and Storage:

- Large-scale memory configurations for handling concurrent user requests.
- Distributed storage systems (e.g., Firebase Storage, Firebase Firestore, Firebase Authentication) for storing user prompts, generated images, and stories.

Ports and Connectors:

- No specific physical port requirements default ports will be used.
- Network connections via Wi-Fi and mobile data are sufficient.

Additional Features:

- Hardware Acceleration: Utilize device GPU for local processing, such as image rendering previews or animation effects.
- Offline Capabilities: Basic functionality, like story browsing or cached images, available without an internet connection.

3.1.3 Software Interface

Operating Systems: This application supports devices having Android Operating Systems with version 7.0 or higher.

Third-party Services: The application integrates with external services like Microsoft Azure, Google Cloud Platform (GCP) for generating images using Artificial Intelligence. Furthermore, the application uses Firebase Services (Firebase Firestore, Firebase Storage, Firebase Realtime Database, and Firebase Authentication) for storing data. The application uses services such as OneSignal or Firebase Cloud Messaging (FCM) for comments, likes, and updates.

Databases: The application uses Firebase Realtime Database with NoSQL data model or SQL databases such as Amazon RDS, Google Cloud SQL, and PostgreSQL.

Libraries and Frameworks: For Text to image generation with AI and cloud computing tasks, the application uses AWS SDK for Kotlin, Hugging Face Transformers, Retrofit or OkHttp. In Authentication, Firebase Authentication or Auth0 Kotlin SDK are used by the application.

3.1.4 Communication Interfaces

Network Protocols:

- Secure data exchange over HTTPS.
- WebSocket connections for real-time updates (e.g., showing progress during image generation).

Data Formats:

- **Request Formats**: JSON for prompt submission and parameter configurations.
- **Response Formats**:
  - JSON for story content.
  - Image files in PNG, JPEG, or PDF for visuals.

API Specifications:

- **Image Generation**: APIs such as Stable Diffusion, Imagen, DALL-E 3, or StyleGAN3 for visual content.
- **Story Generation**: GPT-based APIs like Gemini, ChatGPT for generating textual narratives based on image context.

Cloud Resource Requirements for Communication:

- **Data Bandwidth**: High-speed internet connection for transmitting large images and real-time API communication.
- **Server Instances**: Optimized for low-latency responses to user interactions.
- **Caching**: Use CDN (Content Delivery Network) for delivering static content (e.g., frequently accessed prompts or templates).

Security Features:

- Encryption algorithms for all data transmissions.
- OAuth 2.0 or similar frameworks for user authentication and authorization.
- Periodic audits of API endpoints and server infrastructure to prevent vulnerabilities.

Additional Communication Features:

- Push notifications for completed image and story generations.
- Support for fallback protocols, ensuring minimal downtime during connectivity issues.

3.1.5 Functional Requirements Specification

Reference No	R1
Functional Requirement	Login
Description	The registered users log into the application by entering the required information correctly.
Input(s)	Username and password received from the user.
Processing	1. The application asks the user for username and password before clicking the login button. 2. The user enters the necessary information to these text fields and presses the *"Login"* button. 3. Login database checks whether the inputs are matched or not. 4. If the user does not have an account, the user presses the *"Sign Up"* button to create an account. 5. If the user forgets the password, the user presses the *"forgot password?"* to create a new password.
Exception(s)	• If there is no match username or the password in database, the application displays a pop-up message regarding this error. • If unregistered users press *"forgot password?"*, they receive an error code as *"You must have an account"*.
Output(s)	The application directs the user to the main menu.

Reference No	R2
Functional Requirement	Forgot Password
Description	The registered users create a new password to be able to log into the application.
Input(s)	The application asks the user to enter registered email and confirmation code. The user enters a new password and confirms the password.
Processing	1. The user enters the mail and clicks the *"Send confirm code"* button to retrieve a confirmation code from the mail. 2. The user enters the code into confirmation text box and then presses the *"Change password"* button. 3. User creates a new password based on the password rules and confirms the password by entering it into the *"confirm password"* textbox.
Exception(s)	• The user must enter a valid email address, and the entered email must be the same as the registered email address. • If the confirmation code is not received by the user, then the user must click on the "Resend Confirmation Code Again" button. • If user entered the correct email and then presses the *"Send confirm button"* get the confirmation code. • If the user presses the change password button without entering the confirmation code, the application displays an error as *"Confirmation code is missing"*. • The user must create a new password according to password rules given by the application. If the new password and confirm password are not the same, the application displays an error message as *"The passwords are different!"*.
Output(s)	The user's password is updated on the database.

Reference No	R3
Functional Requirement	Sign Up
Description	To use the application, the user must create an account.
Input(s)	Username, mail, name, surname, and password are received by the user.
Processing	1. The user inputs username, email address, name, surname, and password information to given textboxes. 2. At the end of the sign-up page, the user must confirm the password by entering the same password 3. Once all fields are filled, the user presses the *"Sign Up"* button to save the information. 4. Application's database creates a new account and stores it into the database.
Exception(s)	• If the user enters an email which is already registered, the application warns the user to enter another mail. • If the user enters any of the information incompletely then, the application warns the user with error messages. • The password, name, surname, and password inputs should satisfy certain rules such as name, surname must consist of at least 5 letters, etc.
Output(s)	The user's account is pushed to the database.

Reference No	R4
Functional Requirement	Like an Artwork
Description	Allows users to like a specific artwork.
Input(s)	Touch input from the device's touchscreen
Processing	1. The user touches on the heart icon, 2. The related artwork gets added to the user's "liked artworks" page.
Exceptional Situation(s)	-
Output(s)	The heart icon that the user has touched changes color from white to red.

Reference No	R5
Functional Requirement	Share an Artwork
Description	Allows the user to share a specific artwork on other possible social media platforms.
Input(s)	Touch input from the device's touchscreen
Processing	1. The user touches on the *"share"* icon 2. Then the list of social media platforms that are available on the device on which this artwork could be shared appears 3. The user chooses a platform, and the artwork is shared on the respective platform.
Exceptional Situation(s)	-
Output(s)	A success message will be displayed, which indicates that the artwork has successfully shared on the desired platform.

Reference No	R6
Functional Requirement	Filter
Description	Allows the users to filter the artworks according to categories (Newest, Top Commented, Daily/Weekly/Monthly Best).
Input(s)	Touch input from the device's touchscreen
Processing	1. The user touches on the filter bar (by default the artworks are filtered by "Newest") 2. Then the drop-down list with all the options is shown to the user, according to the user's choice, the artworks would be filtered by that parameter.
Exceptional Situation(s)	-
Output(s)	The display order of the artworks is changed according to the filtering option.

Reference No	R7
Functional Requirement	Comment
Description	Allows the users to post comment(s) on artworks.
Input(s)	Touchscreen keyboard input from the device's touchscreen
Processing	1. The user touches on comment bubble icon 2. Then the device's keyboard becomes usable, and the user types the desired thoughts 3. The user presses on the "post" button to post his/her comment.

Exceptional Situation(s)	-
Output(s)	The comment would be displayed in the comment section.

Reference No	R8
Functional Requirement	Visit another User's Profile
Description	Allows the users to view another user's profile.
Input(s)	Touch input from the device's touchscreen
Processing	1. The user clicks on the profile picture of the user that he/she would like to visit 2. Then the desired user's profile content is presented.
Exceptional Situation(s)	-
Output(s)	The desired user's profile content would be displayed on the current user's screen.

Reference No	R9
Functional Requirement	Follow Another User
Description	Allows the users to follow another user.
Input(s)	Touch input from the device's touchscreen
Processing	1. The user clicks on the "+" icon which he/she wants to follow 2. Then the user is added to user's followers list, 3. The user would be able to get notifications when the user that he/she follows performs a public action.
Exceptional Situation(s)	-

Output(s)	• The number of followers of the followed user would be increased. • The user's perspective, the *"+"* icon replaced with a *"-"* icon, indicating that as of that moment he/she is following that user. • When the *"-"* button is clicked, the current user can unfollow the previously followed user.

Reference No	R10
Functional Requirement	Homepage navigation
Description	Allows the users to navigate between the artworks.
Input(s)	Touch input from the device's touchscreen
Processing	1. The user swipes the screen to the left or to the right, as it is indicated on the screen with a finger icon 2. Then the upcoming or previous set of artworks becomes visible to the user.
Exceptional Situation(s)	-
Output(s)	The next/previous set of artworks are displayed on the user's screen.

Reference No	R11
Functional Requirement	Complete achievements
Description	Triumph of the current predetermined achievements given by the application.
Input(s)	-
Processing	The application monitors the user's use of the application and checks if the conditions of the achievements are completed.
Exceptional Situation(s)	-

Output(s)	Change of the completion status of the current achievements.

Reference No	R12
Functional Requirement	More options
Description	Allow the users to change the settings of the application.
Input(s)	The user clicks on the buttons.
Processing	1. The user clicks hamburger icon and open the more option page 2. Then the user can click one of the buttons (Settings or Sign Out)
Exceptional Situation(s)	-
Output(s)	• Users view more options page • If Settings or Sign Out buttons clicked, then the current page changes

Reference No	R13
Functional Requirement	Search
Description	Allows the users to search by username or tag name of the artwork
Input(s)	User enters either username or tag name
Processing	1. The user enters username or tag name

	2. Then the user click search button 3. Application shows similar usernames and images with the given tag name.
Exceptional Situation(s)	-
Output(s)	Similar usernames and images are displayed which contains sub words/words given user query
Reference No	R14
Functional Requirement	Inspect photo
Description	The user view additional information about selected artwork
Input(s)	The user clicks on the desired image
Processing	1. The user clicks on the desired image 2. The application opens an artwork detail page including story of the artwork, creator, follow, share options by blurring rest of the page content
Exceptional Situation(s)	-
Output(s)	Application shows artwork detail page

Reference No	R15
Functional Requirement	View Notifications
Description	Allow the users to view their notifications.
Input(s)	The user clicks buttons and scrolls down the page.
Processing	1. The user navigates to the *"Notifications"* page using the button on the far left of the bottom navigation bar 2. The notifications are listed one under the other, with the profile photo of the user on the left side and the content of the notification message on the right side.
Exceptional Situation(s)	-
Output(s)	User views notifications.

Reference No	R16
Functional Requirement	Click on Notifications
Description	Allow the users to view the profiles of the other users to whom the notification appeared.
Input(s)	The user clicks on a notification.
Processing	1. The user clicks on a notification he/she chooses on the *"Notifications"* page 2. The user is directed to the *"Other User Profile"* page of the other user that the notification concerns.
Exceptional Situation(s)	If the other users' profile to whom the notification concerns cannot currently be viewed due to an error, the application displays the pop-up error message *"This user cannot currently be viewed"*.
Output(s)	The user views the other user's profile page, where he/she can see their artwork, likes, comments and follower counts.

Reference No	R17
Functional Requirement:	View challenges
Description	Viewing the current challenges.
Input(s)	The user clicks the *"Challenge & Achievements"* icon.
Processing	1. The user clicks the *"Challenge & Achievements"* icon on the upper right side.

	2. The user views the current challenges on this page.
Exceptional Situation(s)	-
Output(s)	View of current challenges

Reference No	R18
Functional Requirement:	View achievements
Description	Viewing the achievements.
Input(s)	1. The user clicks on the *"Challenge &Achievements"* icon. 2. The user clicks on the achievement button
Processing	1. The user clicks the *"Challenge & Achievements"* symbol on the upper right side. 2. The user clicks on the achievement button and achievements appear on the page instead of the current challenges.
Exceptional Situation(s)	-
Output(s)	View of achievements

Reference No	R19
Functional Requirement:	Complete challenges
Description	Completion of the current predetermined challenges given by the application.
Input(s)	-
Processing	The application monitors the user's use of the application and marks it as completed if the user completes one of the current challenges.
Exceptional Situation(s)	-

Output(s)	Change of the completion status of the current challenges.

Reference No	R20
Functional Requirement	View User's Own Profile
Description	Allow the users to view his/her own profile, where the user can see the number of likes, comments, followers and views previously created artworks.
Input(s)	User click buttons and scrolls down the page.
Processing	1. The user navigates to the *"My Profile" page* using the button on the far right of the bottom navigation bar 2. The application displays the user's profile page.
Exceptional Situation(s)	-
Output(s)	The user views their own profile page.

Reference No	R21
Functional Requirement	See Follows/Followers List
Description	Allow the users to view other users follows.
Input(s)	The user clicks buttons and scrolls down the page.
Processing	1. The user clicks on the *"X (number) Followers"* text/link located under the profile photo 2. Then the followers' usernames display on the *"My Profile"* page

	3. The application displays the profile photos and usernames of follows/followers one under the other.
Exceptional Situation(s)	-
Output(s)	The application displays the other users as a list form.

Reference No	R22
Functional Requirement	See Comments
Description	Allow the users to view comments made on an artwork.
Input(s)	The user clicks buttons and scrolls down the page.
Processing	1. The user clicks on the *"comments"* button at the bottom right of a post on the "*Home*" page. Alternatively on the post detail, which opens as a result of clicking on an artwork seen in the feed on the *"Discover"* page. 2. The comments made are listed one under the other in a scrollable manner towards the bottom of the post.
Exceptional Situation(s)	-
Output(s)	The user views the comments made to this artwork.

Reference No	R23
Functional Requirement	View User's Own Previous Artworks
Description	Allow the users to view artworks that they have previously created and shared.

Input(s)	The user clicks buttons and scrolls down the page.
Processing	1. Using the button at the far right of the bottom navigation bar, the user navigates to the "My Profile" Page 2. The application displays the artworks of the user that are previously created in the *"My Artworks"* section on this page and the user can swipe that part of the screen left or right to view more. 3. By clicking on the artworks here, the user can navigate to the *"Artwork Details" p*age of that artwork.
Exceptional Situation(s)	-
Output(s)	The user views the artworks he/she has previously created.
Reference No	R24
Functional Requirement:	Sign out
Description	The user signs out of his/her account.
Input(s)	The user clicks the sign out button
Processing	1. The user enters the side menu using the hamburger icon on the top right. 2. The user clicks the *"Sign out"* button on this page.
Exceptional Situation(s)	-
Output(s)	The application signs out of the user's account and redirects to the login page.

Reference No	R25
Functional Requirement:	Change password
Description	The user changes the password of the account.
Input(s)	User's existing password User's new password

Processing	1. The user enters the side menu using the hamburger icon on the top of right then clicks the *"Settings"* button on this page.
	2. The user clicks the *"Change Password"* button on this page and redirects to the password change screen.
	3. User enters the existing password then enters the new password two times to confirm.
	4. User clicks the *"Save"* button to save the new password.
Exceptional Situation(s)	-
Output(s)	Change of the password.

Reference No	R26
Functional Requirement:	Change color theme
Description	The user changes the theme of the application between light mode/dark mode.
Input(s)	The theme chosen by the user.
Processing	1. The user enters the side menu using the hamburger icon on the top of right then clicks the *"Settings"* button on this page.
	2. The user changes between themes using the radio button on this page.
Exceptional Situation(s)	-
Output(s)	Change of application theme.
Reference No	R27
Functional Requirement:	Change language
Description	The user changes the language of the application.
Input(s)	The language chosen by the user.
Processing	1. The user enters the side menu using the hamburger icon on the top of right then clicks the *"Settings"* button on this page.
	2. The user clicks the *"Change Language"* button and selects a language from the dropdown menu.

Exceptional Situation(s)	-
Output(s)	Change of application language

Reference No	R28
Functional Requirement	Generate Artwork
Description	The user creates the image with specified features using prompts and Artificial Intelligence.
Input(s)	Prompt text, preset style and add story option chosen by the user.
Processing	1. The user provides a text prompt describing the desired image. The user can specify a certain style for their image by selecting a specified style under the "preset style" category. 2. The user can select *"Add story"* button to let the application generate a story based on the generated image. 3. The user presses *"Generate"* button to generate the image. 4. The application collects the user's input and generates the image using an Artificial Intelligence model in cloud service.
Exception(s)	• If the user enters an invalid prompt, the application validates the prompt and gives feedback message to the user. • If the image is not generated due to mistake of user or application backend, then the image page won't pop-up.

	Also, the application will give an error message to the user with detailed information.
Output(s)	The generated image is sent back to the application from the cloud service.

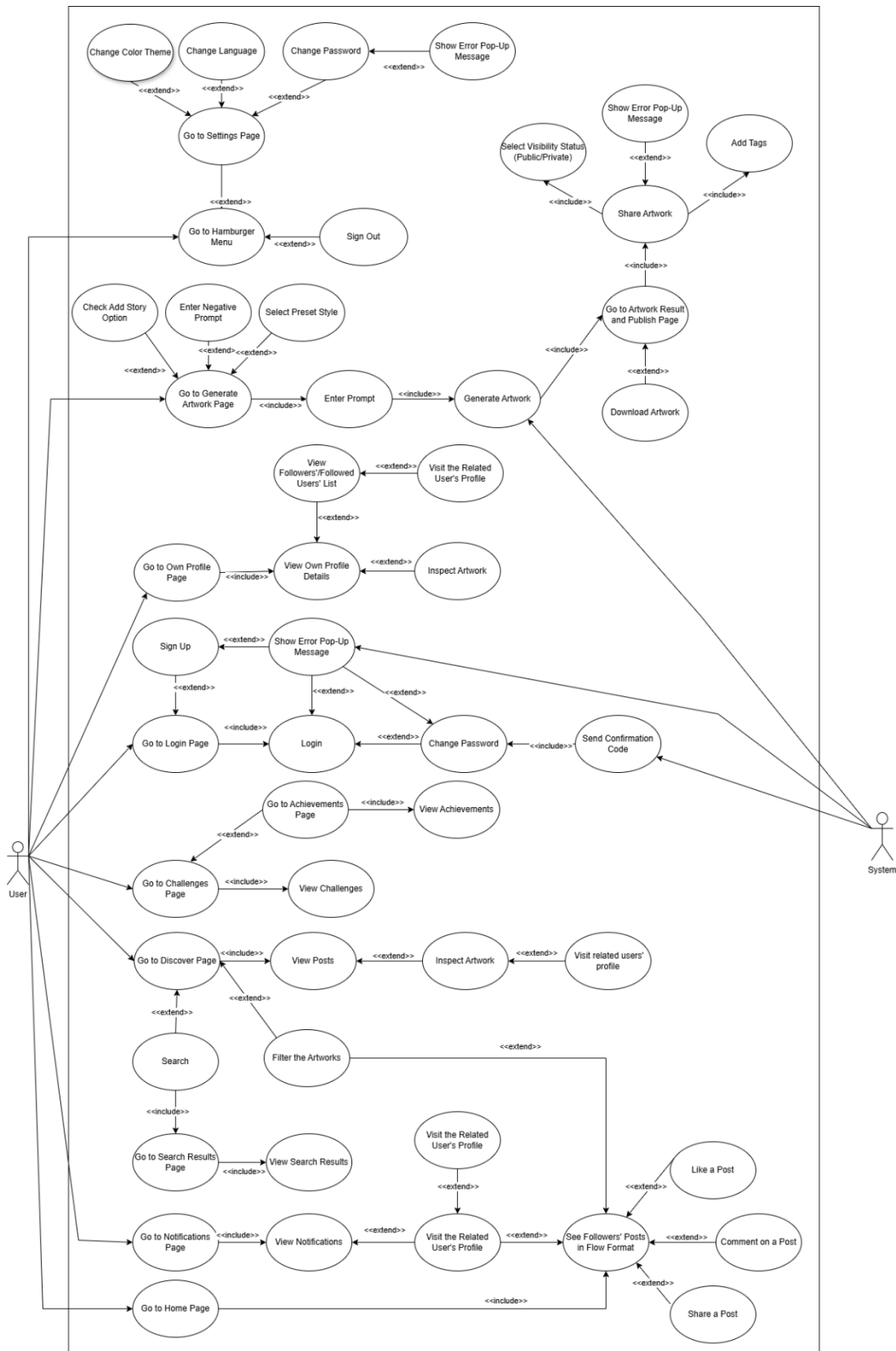|	
Reference No	R29
Functional Requirement	Generate Story
Description	The application generates a story based on the generated image
Input(s)	The application takes the generated Image file as input.
Processing	1. The application analyzes the context of the image using specific models. 2. Then the application extracts additional contextual information from the image. 3. After that the application converts the information to text.
Exception(s)	1. If generating the story takes too much time, the application gives an error message to the user. 2. If the user doesn't provide a tag, then the application gives a warning message "tag must not be empty".
Output(s)	The generated story is retrieved by the application and displayed under the image.

Reference No	R30
Functional Requirement	Download

Description	The user downloads the generated image to his/her local mobile device.
Input(s)	Touch input from the device's touchscreen
Processing	1. The user presses the download button on the bottom right corner of the image. 2. The user chooses a file format (jpeg, png, pdf) to download. 3. The image is downloaded to the user's downloads folder on the device.
Exception(s)	-
Output(s)	The image file is stored under the *"Downloads"* folder of the user's mobile device.
Reference No	R31
Functional Requirement	Share image on your profile
Description	The user shares the generated image to his/her profile.
Input(s)	The user chooses the visibility of the image as public, just followers, and enters a tag name for the image
Processing	1. The user chooses public or just followers and enters a tag name for the image 2. Then the user presses the share button to publish the image on the profile.
Exception(s)	• All images on user's profile must have a unique tag, so when user enters an existing tag, the application gives an error. • If any of the *"public"* and *"just followers"* options are not selected, then the image is shared publicly as a default option.
Output(s)	The application directs the user to the profile page. The created images are visible on the "My artworks" part.

Reference No	R32

Functional Requirement	Observe search results
Description	The user can observe the searched users and their art works
Input(s)	Touch input from the device's touchscreen
Processing	1. The user can choose other users from the user list by pressing the profile picture of that user 2. The application directs the user to the profile page of the person that the user chose. 3. The user can press an artwork from the Arts list and can observe the art in detail.
Exception(s)	-
Output(s)	The user can view the existing artworks and other users' profiles

3.1.6. Use Cases

Use Case Name: Login

Authors: Mertcan Zafer

Actors: User, System

Overview: A registered user logs into the application by entering the required information.

References: R1, R2

Related Use Cases: Go to Login Page, Show Error Pop-Up Message, Change Password, Send Confirmation Code, Sign Up

Typical Flow Description:

1. The user goes to "Login Page" to log into the application
2. The system asks the user to input username and password
3. The user enters the username and password into the text boxes.
4. The user presses "Login" button to log into the application.
5. The system checks the username and password information inputted by the user, with the user's data in database.
6. The user logs in successfully and goes to the main application.

Alternative Flow Description:

3a.

1. The user presses "I forgot my password" link to change the password.
2. The system asks the user to input mail.
3. The user enters an email into the text box and clicks "Send confirmation code" button.
4. The system validates email and sends a confirmation code to the corresponding email address.
5. The user enters the confirmation code into the text box.
6. The user is directed to password change page.
7. The system asks the user to input the new password.
8. The user inputs the new password into the text box and reenters the password into the "confirm password" text box.
9. The user clicks "Change password" button.
10. The system updates the password and overwrites the information into the database.

3b.

1. The system checks the given password with the password in the confirm password field.
2. The system gives a warning to user as "Confirm password and new Password must be same!!".
3. The user clicks "Change password" button.

4. The system updates the password and overwrites the information into the database

5a.

1. The password or username is not matched with the information on the database. The system gives a warning message to the user
2. The user enters the correct information into the text boxes and presses "Login" button
3. The system checks the username and password information inputted by the user, with the user's data in database.
4. The user logs in successfully and goes to the main application.

Use Case Name: Sign Up

Authors: Mertcan Zafer

Actors: User, System

Overview: The user registers to the application to be able to log into the application

References: R3

Related Use Cases: Go to Login Page, Show Error Pop-Up Message

Typical Flow Description:

1. The user goes to login page.
2. The user clicks "Click here" link to create a new account.
3. The system directs the user to the Account page and wants the user to fill the necessary fields.
4. The user sequentially enters username, mail, name and surname, password, and confirm password.
5. The user presses "Sign Up" button.
6. The system creates a new user account and adds it into the database.
7. The user redirects to the login page.

Alternative Flow Description:

5a

1. The system checks if the mail address is a valid address and not already registered before.
2. The system gives a warning message to user to input a valid email address.

3. The user reenters a correct email address and presses "Sign Up" button.
4. The system creates a new user account and adds it into the database.
5. The user redirects to the login page.

5b

1. The system checks whether any of the given fields are incorrect.
2. The system warns the user to correct the invalid inputs with Pop-up messages.
3. The user reenters the correct inputs to given fields and presses "Sign Up" button.
4. The system creates a new user account and adds it into the database.
5. The user redirects to the login page.

Use Case Name: Go to Generate Artwork Page

Authors: Mertcan Zafer

Actors: User, System

Overview: The user creates an artwork with specified features using prompts and Artificial intelligence.

References: R28, R29, R30, R31

Related Use Cases: Check Add Story Option, Select Preset Style, Enter Prompt, Generate Artwork, Select Visibility Status, Share Artwork, Download Artwork, Add Tags

Typical Flow Description:

1. The user wants to create a new artwork.
2. The user presses "Art" icon on the navigation bar, and the system directs the user to generate Artwork page.
3. The system wants the user to enter specific information before generating the artwork.
4. The user enters prompt into the "Prompt message text box".
5. The user selects one of the Preset styles for his/her artwork.
6. The selects add story button from the "Add story" check circle field.
7. The user presses "Generate" button to get the artwork.
8. The system collects inputs from the user and sends them to the cloud services.
9. The system's artificial intelligence model produces an artwork based on the given information and sends the generated artwork to the application.
10. The system generates a story based on the generated artwork and the user input using artificial intelligence.
11. The system directs the user to another page including the image and detailed information about the image.
12. The user chooses the visibility of the image whether it is public or just followers.
13. The user inputs a tag name for the artwork and presses "Share the artwork" icon to send the artwork to the user's profile.

14. The system directs the user to his/her profile page.

Alternative Flow Description:

6a.

1. The user doesn't select "add story" option for his/her artwork.
2. The user presses "Generate" button to get the artwork.
3. The system collects inputs from the user and sends them to the cloud services.
4. The system's artificial intelligence model produces an artwork based on the given information and sends the generated artwork to the application.
5. The system directs the user to another page including the image and detailed information about the image.
6. The user chooses the visibility of the image whether it is public or just followers.
7. The user inputs a tag name for the artwork and presses "Share the artwork" icon to send the artwork to the user's profile.
8. The system directs the user to his/her profile page.

7a.

1. The system validates the user inputs and gives a warning pop up message that one of those information is either missing or not suitable with the system's generating artwork rules (The user prompts might include incorrect or inconvenient information).
2. The user reenters these fields properly.
3. The user repeats the same steps from the Typical Flow Description.

9a.

1. The system's artificial intelligence model might not generate the artwork due to an undefined behavior of the system's backend (Internet connection might be not stable or generating image process can exceed the model's limits).
2. The system never directs the user to the artwork page and gives an error message to the user with error information and details.

11a.

1. The user wants to download the generated artwork
2. The user clicks download button on to bottom right of the corner of the artwork
3. The user chooses a file format (jpeg, png, pdf) to download the artwork.
4. The image is downloaded to the user's downloads folder on the device.

5. The user follows the same steps in the Typical Flow Description.

12a.

1. The user chooses the "public" option for visibility of the artwork.
2. The user inputs a tag name for the artwork and presses "Share the artwork" icon to send the artwork to the user's profile.
3. The system directs the user to his/her profile page.
4. The system makes the artwork visible to all users of the application

12b.

1. The user chooses the "just followers" option for visibility of the artwork.
2. The user inputs a tag name for the artwork and presses "Share the artwork" icon to send the artwork to the user's profile.
3. The system directs the user to his/her profile page.
4. The system makes the artwork visible to only followers of the user.

13a.

1. The user does not provide a tag name for the artwork or give a unique tag name.
2. The system gives a warning message to the user to input a valid tag name.
3. The system directs the user to his/her profile page.
5. The system makes the artwork visible to all users of the application or only followers of the user based on the visibility option of the artwork.

Use Case Name: View Own Profile Page

Authors: Mertcan Zafer

Actors: User

Overview: The user views his/her own profile, where user can see the number of likes, comments, followers and views previously created artworks.

References: R23

Related Use Cases: Go to Own Profile Page, View Followers/Followed List Users' List, Inspect Artwork, Visit Related User's Profile

Typical Flow Description:

1. The user navigates to the "My Profile" page using the button on the far right of the bottom navigation bar
2. The user views his/her own profile page.

3. The user wants to inspect his/her artwork by clicking onto them from "My Artworks" part.
4. The user views the details of their artworks such as number of likes, comments, and the users who liked or commented on their artworks.
5. The user wants to see her/his followers/followed list.
6. The user clicks "X (number) Follows" or "X (number) Followers" text (link) located under the profile photo and username on the "My Profile" Page.
7. The user views all her/his followers with their username and profile photos. The user scrolls down the list to see more people from the list.

Alternative Flow Description:

3a

1. The user wants to see settings and other options by clicking "Hamburger menu" button at the top right of the Profile page.
2. The user presses "Sign out" button to sign out.
3. The user follows the steps from Use case "Sign Out".

3b

1. The user presses "Settings" button to see settings of the application.
2. The user follows the steps from Use case "Go to settings Page".

3c

1. The user wants to see "Challenges and Achievements list" by clicking the "Achievement cup" icon under the Hamburger menu.
2. The user follows the steps from Use case "Go to Challenges Page".

Use Case Name: View Notifications

Authors: Emre Şahin Demirbaş

Actors: User

Overview: This use case allows the user to view his/her notifications.

References: R8, R15, R16

Related Use Cases: Go to Notifications Page, Visit the Related User's Profile

Typical Flow Description:

1. The user wants to see his/her notifications.

2. The user navigates to the notifications page.

3. The user observes his/her notifications in list format, could swipe the screen to right see more.

Alternative Flow Description:

2a.

1. The user wants to visit the profile of the user who is related with that notification.

2. The user touches on the respective artwork creator's profile picture.

3. The user views the related user's profile details.

Use Case Name: See Followers' Posts in Flow Format

Authors: Emre Şahin Demirbaş

Actors: User

Overview: This use case allows the user to view his/her followers' posts in flow format.

References: R4, R5, R6, R7, R8,

Related Use Cases: Go to Home Page, Filter the Artworks, Visit the Related User's Profile, Like a Post, Comment on a Post, Share a Post

Typical Flow Description:

1. The user wants to see the artworks that are created by his/her followers.

2. The user touches on home page button on the navigation bar if he/she is on another page, otherwise on first opening of the app this is the default page.

3. The user views his/her followers' posts one below the other in a free scrolling form, could scroll down to see more artworks.

Alternative Flow Description:

2a.

1. The user wants to filter the artworks' displaying order.

2. The user taps on the filtering bar.

3. The user selects a filtering option from the drop-down list.

4. The user views his/her followers' artworks in a filtered manner.

2b.

1. The user wants to visit the profile of an artwork's creator.

2. The user touches on the respective artwork creator's profile picture.

3. The user views the related user's profile details.

2c.

1. The user wants to like a post.

2. The user touches on the heart icon below the respective post.

3. The user observes that the heart icon changes its color from white to red.

2d.

1. The user wants to comment on a post.

2. The user touches on the comment bubble icon.

3. The user enters his/her desired text with the help of the touchscreen-keyboard.

4. The user clicks on the "Post the Comment" button.

5. The user sees his/her comment in the comment section of the related artwork.

2e.

1. The user wants to share a post.

2. The user clicks on the share icon.

3. The user selects the platform that he/she wants to share the artwork on.

4. The user views a message on the screen, informing the result of the sharing operation.

Use Case Name: Go to Challenges Page

Authors: Emre Şahin Demirbaş

Actors: User

Overview: This use case allows the user to navigate to the challenges page, which includes two sections as "Challenges" and "Achievements", where the user could observe the current challenges and observe his/her achievements.

References: R17, R18, R19

Related Use Cases: View Challenges, Go to Achievements Page, View Achievements

Typical Flow Description:

1. The user wants to check out the current challenges.

2. The user taps on the trophy icon on the screen.

3. The user observes the ongoing challenges.

Alternative Flow Description:

2a.

1. The user wants to check out his/her achievements.

2. The user touches on "Achievements" text to go to the achievements page.

3. The user views his/her achievements with their completion progresses if not have fully completed, otherwise the achievements have "tick" symbols on their names' left-hand side.

Use Case Name: Sign Up

Authors: Mehmet Emre Kılınç

Actors: User, System

Overview: This use case captures of the sign-up process of a new user who is not registered to the application.

References: (Requirementler)

Related Use Cases: Go to Login Page, Show Error Pop-Up Message

Typical Flow Description:

1. A new user who does not have an account for the application wants to sign up to the application.

2. User goes to the "Login" page, which is the first page that opens when the user opens the application.

3. The user goes to the "Sign Up" page by clicking on the "Click here" link in the text "Don't have an account? Click here to sign up" on the "Login" page.

4. The user fills in the username, mail, name surname, password and confirm password input text fields on the "Sign Up" page and clicks the "Sign Up" button.

5. The system signs up the new user.

Alternative Flow Description:

4a.

1. The user left one or more of the required input text fields blank.

2. The system shows a pop-up error message regarding filling in mandatory fields.

4b.

1. A user who has signed up with the entered email is already signed up in the system.

2. The system shows a pop-up error message that such a user already exists.

 Use Case Name: Go to Settings Page

Authors: Mehmet Emre Kılınç

Actors: User

Overview: This use case captures the process of going to the "Settings page", which allows the user to make various changes to the application settings.

References: (Requirementler)

Related Use Cases: Go to Hamburger Menu, Change Color Theme, Change Language, Change Password

Typical Flow Description:

1. User is logged in to the system.

2. The user wants to go to the "Settings" page where s/he can change some settings of the application.

3. The user goes to the "Hamburger Menu" page by clicking on the symbol to the right of the top navigation bar.

4. The user goes to the "Settings" page by clicking the "Settings" button on this page.

5. The user sees the application's settings that are allowed to be changed.

Alternative Flow Description:

5a.

1. The user wants to change the language of the application.

2. The user clicks on the "Language" dropdown list and selects the language they want to change.

3. The system changes the language of the application.

5b.

1. The user wants to change the theme of the application.

2. The user checks or unchecks the "Dark Mode" checkbox depending on the theme s/he wants.

3. The system changes the theme of the application.

5c.

1. The user wants to change his/her password.

2. The user clicks on the "Change Password" button and goes to the "Confirmation" page.

3. The user enters the e-mail he/she registered with in the system into the "Enter your e-mail" input text box and clicks the "Send Confirmation Code" button.

4. The system sends a confirmation code to the entered e-mail.

5. The user enters the incoming confirmation code into the "Confirmation Code" input text field on the page and clicks the "Change Password" button.

6. The system checks the confirmation code and if it is correct, it directs the user to the "Password Change" page. If it is incorrect, it shows a pop-up error message.

7. The user fills in the "New Password" and "Confirm Password" input text fields and clicks the "Change Password" button.

8. If the new passwords in the entered fields match, the system changes the user's password and displays a pop-up success message about it. If it does not match, it shows a pop-up error message about it.

Use Case Name: View Posts

Authors: Tunahan Gültekin

Actors: User

Overview: The user sees artworks of user that follows.

References:

Related Use Cases: Go to discover page, Inspect artwork

Typical Flow Description:

1. The user wants to view publicly shared artworks.

2. The user goes to the discover page from the bottom navigation bar.

3. The user can view publicly shared artworks and can change posts by swiping left and right.

Alternative Flow Description: -

Use Case Name: Search

Authors: Tunahan Gültekin

Actors: User

Overview: The user searches for artwork or users.

References: R13

Related Use Cases: Go to discover page, Go to search results page

Typical Flow Description:

1. The user wants to search for a person or artwork.
2. The user goes to the discover page from the bottom navigation bar.
3. The user clicks on the search bar and performs a search.
4. The user sees the search results page prepared by the system on the page that opens.

Alternative Flow Description: -

Use Case Name: Sign Out

Authors: Tunahan Gültekin

Actors: User

Overview: The user signs out of their account.

References: R24

Related Use Cases: -

Typical Flow Description:

1. The user wants to sign out of his / her account.
2. The user clicks on the hamburger menu icon at the top right of the screen.
3. The user clicks on the sign out button on the page that opens and logs out.

Alternative Flow Description: -

Use Case Name: Change Password

Authors: Batuhan Özer

Actors: User, System

Overview: This use case allows the user to change his/her login password

References: R25

Related Use Cases: Go to Settings Page

Typical Flow Description:

1. User navigates to change password page from the settings page
2. User views the current password and new password fields
3. User fills the empty fields
4. Users click the change password button
5. Users direct to the settings page

Alternative Flow Description:

2a.

1. User navigates to change password page from the settings page
2. User views the current password and new password fields
3. User does not fill all the empty fields
4. Users click the change password button
5. System shows an error message pop-up
6. Users direct to the change password page again

Use Case Name: Go to Artwork Result and Publish Page

Authors: Batuhan Özer

Actors: User, System

Overview: This use case allows user to download and share the generated artwork

References: R31

Related Use Cases: Add Tags, Generate Artwork, Select Visibility Status (Public/Private), Share Artwork, Download Artwork

Typical Flow Description:

1. User navigates to Artwork Result and Publish Page
2. User selects visibility status (Public/Private)
3. User adds tags to the artwork
4. User shares the artwork

Alternative Flow Description:

2a.

1. User navigates to Artwork Result and Publish Page
2. User doesn't fill the one/none of the visibility status and tags fields
3. User shares the artwork
4. System shows error messages pop-up
5. User is directed to the Artwork Result and Publish Page

2b.

1. User navigates to Artwork Result and Publish Page
2. User fills the none/one/both visibility status and tags fields
3. User downloads the artwork

Use Case Name: Go to Achievements Page

Authors: Batuhan Özer

Actors: User, System

Overview: This use case allows user to see his/her achievement process

References: R18

Related Use Cases: Go to Challenges Page, View Achievements

Typical Flow Description:

1. User navigates Achievements Page
2. User view achievement process (completed/In progress) of each achievement

3.1.7. Non-Functional Requirements

Response Time: The application has different acceptable response times according to the tasks. For search result return, response time should be at most 10 seconds, for image load, response time should be at most 5 seconds, for image and story generation, response time should be at most 10 minutes, according to the prompt and user's choice to create a story or not.

Security: The application has various security measures to prevent possible malevolent usage, namely; the prompt provided by the user cannot be more than 500 characters and it would be searched for most frequently used inappropriate keywords, that would cause images which are not suitable for every user, the user has only 3 consecutive chances to attempt to login in a minute, in case the person attempting to login is an unauthorized user, the user passwords are stored in a hashed manner.

Performance: The application should be able to handle 50 concurrent users with the promised response times.

Reliability: The application's main server should be available at all times except the pre-announced, at most half an hour, maintenance periods.