

Artworks Hub

Software Requirement Specification

– Ho Chi Minh City, January, 2024 –

Table of Contents

| | |
|---|----|
| Artworks Hub..... | 1 |
| I. Record of changes..... | 3 |
| II. Software Requirement Specification..... | 5 |
| 1. Product Overview..... | 5 |
| 1. The user Requirements..... | 5 |
| 1.1 Actors..... | 5 |
| 1.2 Use Cases..... | 6 |
| 1.2.1 Diagram(s)..... | 6 |
| 1.2.2 Descriptions..... | 7 |
| 3. Functional Requirements..... | 40 |
| 4. Non-functional requirements..... | 43 |
| 4.1 External Interfaces..... | 43 |
| 4.2 Quality Attributes..... | 43 |
| 4.2.1 Usability..... | 43 |
| 4.2.2 Reliability..... | 43 |
| 4.2.3 Performance..... | 43 |

I. Record of changes

| Date | A* M, D | In charge | Change Description |
|--------|------------|-----------|-------------------------------------|
| 10/Jan | A | | |
| 11/Jan | A | HuyTTM | Added Non-functional Requirements |
| 11/Jan | A | NhanTT | Added functional requirements |
| 12/Jan | A | HuyTTM | Added actors and actors description |
| 12/Jan | A | AnhLQ | Added use case diagram |
| 13/Jan | M | HuyTTM | Updated non-functional requirements |
| 13/Jan | M | HuyTTM | Updated use case description |
| 13/Jan | M | AnhLQ | Updated use case diagram |
| 13/Jan | A | AnhLQ | Updated use case description |
| 13/Jan | M | NhanTT | Updated functional requirements |
| 19/Jan | A | AnhLQ | Added Use Case Specification |
| 19/Jan | M | AnhLQ | Updated Use Case Diagram |
| 23/Jan | M | HuyTTM | Updated Use Case Specifications |
| | | | |
| | | | |

*A - Added M - Modified D - Deleted

II. Software Requirement Specification

1. Product Overview

The Art Sharing Platform is a new software system where creators represent their captivating new ideas to the public and the viewers discover stunning art works. The platform also enables employers to make offers to creators, and creators sell their originals, offer services. It is expected that the platform will evolve after several releases, and make the communication between audiences and creators on the internet easier.

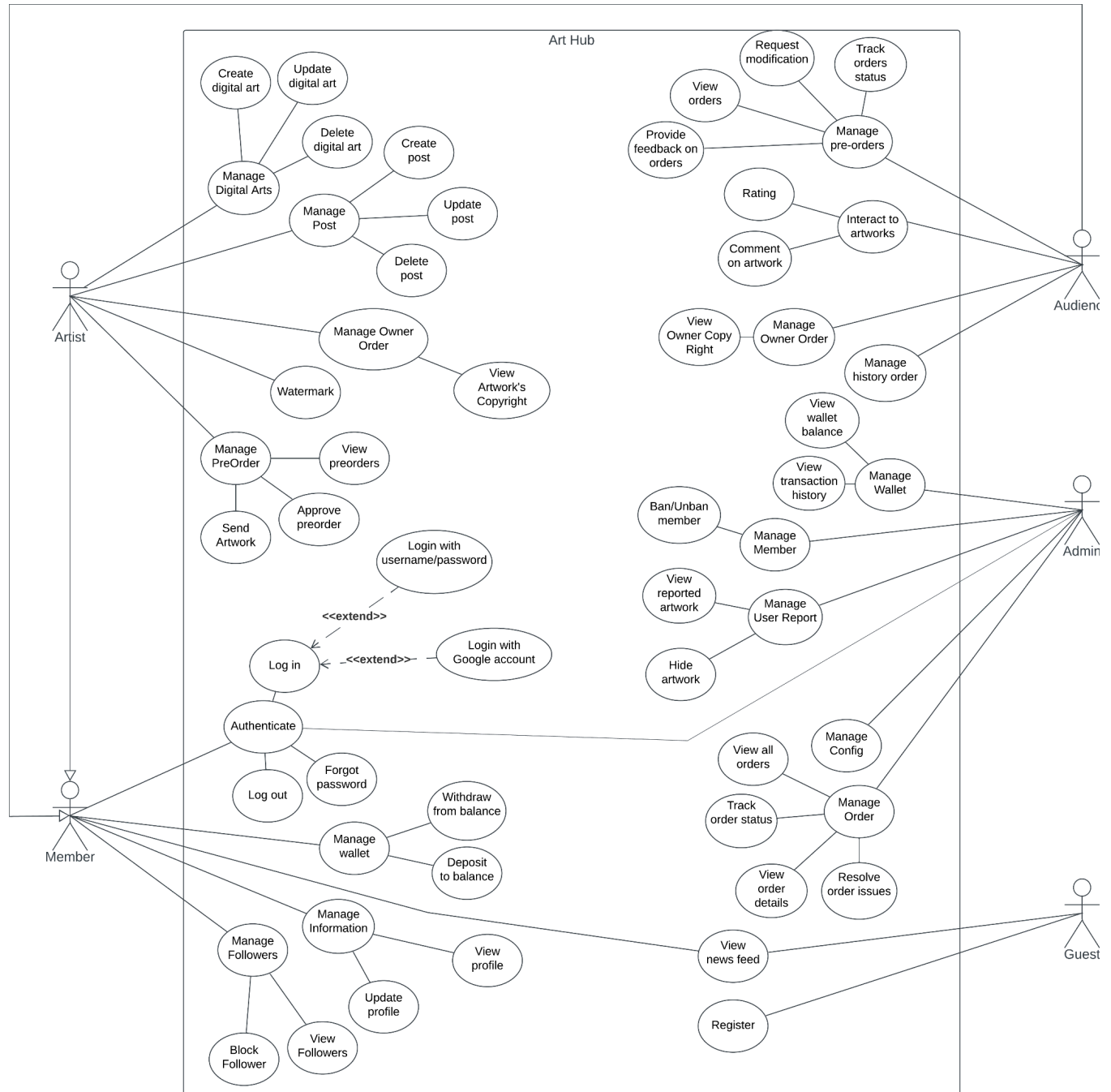
1. The user Requirements

1.1 Actors

| # | Actor | Description |
|---|---------------|--|
| 1 | Administrator | The administrators who are responsible for managing the system. To be more detailed, they can manage users, artwork. |
| 2 | Audience | The audiences who are responsible for interacting to the artwork, and the artists |
| 3 | Artist | The artists who are responsible for managing their artwork and interact to other artwork |
| 4 | Member | |

1.2 Use Cases

1.2.1 Diagram(s)



1.2.2 Descriptions

| ID | Use Case | Actors | Use Case Description |
|----|----------------------------|----------|--|
| 01 | View Wallet Balance | Admin | |
| 02 | View Transaction History | | |
| 03 | View Reported Artwork | | |
| 04 | Ban Artist | | |
| 05 | Hide Artwork | | |
| 06 | Ban Audience | | |
| 07 | Track Order Status | | |
| 08 | View Order Details | | |
| 09 | Resolve Order Issues | | |
| 10 | Manage Config | | |
| 10 | Track orders status | Audience | |
| 11 | Request modification | | |
| 12 | View orders | | |
| 13 | Provide feedback on orders | | |
| 14 | Comment on artwork | | |
| 15 | Rating | | |
| 16 | Buy Artwork | | Audience members complete a purchase of artwork directly through the platform. |
| 17 | View Owner Copy Right | | |
| 18 | Manage history order | | |
| 19 | Create digital art | Artist | |
| 20 | Update digital art | | |

| | | | |
|----|--------------------------|---------------|--|
| 21 | Delete digital art | | |
| 22 | Create post | | |
| 23 | Update post | | |
| 24 | Delete post | | |
| 25 | View Artwork's Copyright | | |
| 26 | Watermark | | |
| 27 | View preorders | | |
| 28 | Approve preorder | | |
| 29 | Send Artwork | | |
| 30 | Log in | Admin, Member | Existing users log in to their account using credentials. |
| 31 | Sign Up | | New user creates an account to access the platform's features. |
| 32 | Forgot Password | | The users retrieve their forgotten password through email or other verification. |

1. Use Case Specification

| | |
|----------------------------|---|
| Use Case ID | UC-1.1 |
| Use Case Name | Log in |
| Description | As a user, I want to login to use more functions of website |
| Actor(s) | Auth The user, Artis, Audience |
| Priority | Must Have |
| Trigger | The user wants to login our website |
| Pre-Condition(s) | <ul style="list-style-type: none"> - The user already had a account - The user has a permission to login - The user's laptop or phone must have internet |
| Post-Condition(s) | <ul style="list-style-type: none"> - The user login successfully |
| Basic Flow | <ol style="list-style-type: none"> 1. The user goes to the Art Hub website. 2. The user chooses to log in with Art Hub account. 3. The user enters username and password and click login. 4. Application authentication information successfully and redirect the user to the home page. |
| Alternative Flow | <ol style="list-style-type: none"> 2a. The user chooses to login with a Google account. 3a. The user is redirected to the Google login page. 4a. The user enters their Google account and password. 5a. Google verifies the account and sends the account information to the Art Hub system. 6a. The Art Hub system verifies the account successfully and redirects the user to the Art Hub home page. |
| Exception Flow | <ol style="list-style-type: none"> 3a. Application authentication information fails and alerts the user. <ol style="list-style-type: none"> 3a1. The user cancels login.(Stop use case here) 3a2. The user clicks the forgot password. (Continue with use case UC1-2) |
| Business Rules | BR1.1-1: If the user enters wrong credentials for the 6th consecutive time, the account is disabled for 30 minutes. |
| Non-Functional Requirement | |

| | |
|-------------------|--|
| Use Case ID | UC-1.2 |
| Use Case Name | Forgot password |
| Description | As a user, I want to reset my password as i forgot it |
| Actor(s) | Auth The user, Artis, Audience |
| Priority | Must Have |
| Trigger | The user clicks on the forgot password button |
| Pre-Condition(s) | <ul style="list-style-type: none"> • The user has an account of the application • The user's laptop or phone must have internet |
| Post-Condition(s) | - The user reset their password successfully |
| Basic Flow | <ol style="list-style-type: none"> 1. The user clicks on the forgot password button 2. The user enters the correct email to receive the OTP to reset their password. 3. The user is directed to the reset password page if the input email is correct 4. On the reset password page, the user is requested to enter the received OTP and clicks continue. 5. If the OTP is correct, the user is redirected to the change password page 6. The user enters a new password that meets the complexity requirements, click the Change password button 7. System validates the user's new password and updates the user's account with the new password. 8. System displays a success message and redirects the user to the login page. |
| Alternative Flow | |
| Exception Flow | <p>2a. Inputted email does not exist.</p> <p>2a1. The user cancel reset password.(Stop use case here)</p> <p>2a2. The user re-enters the email. (Restart from step 2)</p> <p>4a. Inputted OTP is incorrect.</p> <p>4a1. System displays the error and request the user to re-enter the OTP (Restart from step 4)</p> <p>4b. OTP expired.</p> <p>4b1. System displays the error and request the user to revoke the</p> |

| | |
|----------------------------|--|
| | <p>OTP again (Restart from step 2).</p> <p>7a. New password does not meet system validation</p> <p>7a1. System displays the error and requests the user to re-enter the new password (Restart from step 6).</p> |
| Business Rules | <p>BR1.2-1: Password must be at least 8 characters, including lowercase letters, uppercase letters, numeric characters, special characters.</p> <p>BR1.2-2: If the user fails from entering the correct OTP for the 6th consecutive time, disable the forgot password function of the account for 1 hour.</p> <p>BR1.2-3: The OTP must be valid in 15 minutes.</p> |
| Non-Functional Requirement | |

| | |
|----------------------------|--|
| Use Case ID | UC-2 |
| Use Case Name | Log out |
| Description | As a user, I want to log out from the website after using it. |
| Actor(s) | Auth The user, Artis, Audience |
| Priority | Must Have |
| Trigger | The user wants to log out from our website. |
| Pre-Condition(s) | <ul style="list-style-type: none"> - The user has logged in to the system. - The user's laptop or phone must have the internet. |
| Post-Condition(s) | <ul style="list-style-type: none"> - The user logs out successfully. |
| Basic Flow | <ol style="list-style-type: none"> 1. The user clicks on the log out button. 2. The browser sends a request to the system indicating that the user wants to log out. 3. The system validates the request. 4. The user is logged out from the application. 5. The browser redirects the user to another page: home page, login page. |
| Alternative Flow | |
| Exception Flow | <p>2a. The browser can not send the request to the system because of the network error:</p> <p>2a1. Display an error message indicating network problems. (Stop the use case here).</p> <p>2a2. Prompt the user to check their connection and retry logout. (Restart from step 1).</p> |
| Business Rules | |
| Non-Functional Requirement | |

| | |
|-------------------|---|
| Use Case ID | UC-3 |
| Use Case Name | Register account |
| Description | As a new user, I want to create an account on this website. |
| Actor(s) | Guest |
| Priority | Must Have |
| Trigger | The user wants to create an account on the application. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • The user has an account of the application • The user's laptop or phone must have internet |
| Post-Condition(s) | - The user can create an account successfully. |
| Basic Flow | <ol style="list-style-type: none"> 1. The user enters the Art Hub website. 2. The user chooses to register an Art Hub account with their email and password. 3. The user enters an email and password to create the account. 4. System validates email and password. 5. System sends verification email including to the user 6. The user clicks on the verification link in the email to verify the email 7. The account is created successfully, and the server saves the account. 8. The user is redirected to the home page. |
| Alternative Flow | <ol style="list-style-type: none"> 2a. The user chooses to register an account with their Google account. 3a. The user is redirected to the Google login page then enters the google account and password. 4a. Google verifies the Google account. 5a. System saves the user's Google account information if the Google account is correct. 6a. The user is redirected to the home page. |
| Exception Flow | <ol style="list-style-type: none"> 3.1 The inputted email already existed: <ol style="list-style-type: none"> 3.1.1 System displays the error and requests the user to re-enter another email. (Restarts from step 3) 3.2 The inputted password does not meet complexity requirements. <ol style="list-style-type: none"> 3.2.1 System displays the error and requests the user to re-enter the password. (Restarts from step 3) 3.3 The verification link expired: <ol style="list-style-type: none"> 3.3.1 System displays the error and shows button to let the user get a new verification link. |

| | |
|----------------------------|---------|
| Business Rules | BR1.1-1 |
| Non-Functional Requirement | |

| | |
|-------------------|---|
| Use Case ID | UC-4 |
| Use Case Name | View newsfeed |
| Description | As a guest, |
| Actor(s) | Member , Guest |
| Priority | Must Have |
| Trigger | The actors access the Art Hub website. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • User's laptop or phone must have the internet. • Actors have access to the main page. |
| Post-Condition(s) | <ul style="list-style-type: none"> • The news feed is displayed on the homepage. |
| Basic Flow | <ol style="list-style-type: none"> 1. All actors landing on the homepage. 2. The news feed is displayed. 3. Content displayed on the newsfeed depends on the actor: <ol style="list-style-type: none"> 3.1. Audience: <ul style="list-style-type: none"> • New artwork uploads from followed artists. • Trending artwork based on platform activity. • Artist announcements and updates. • Platform news and events. • Content curated by the platform based on user preferences. 3.2. Artist: <ul style="list-style-type: none"> • Engagement with their own artwork (likes, comments, purchases). • Activity of their followers (artwork interactions, new followers). • Relevant platform news and events for artists. 3.3. Guest: <ul style="list-style-type: none"> • See a general feed featuring popular artwork, platform highlights, and curated content. 4. Interactions on posts also vary depending on the actors: <ol style="list-style-type: none"> 4.1. Audience, Artist: <ul style="list-style-type: none"> • Like, comment, share artwork posts. |

| | |
|----------------------------|--|
| | <ul style="list-style-type: none"> Discover new artists and artwork through feed recommendations. Click on posts to view the featured artwork in detail. Follow artists directly from the news feed. 4.2. Guest: <ul style="list-style-type: none"> Discover new artists and artwork through feed recommendations. Click on posts to view the featured artwork in detail. |
| Alternative Flow | |
| Exception Flow | <ol style="list-style-type: none"> Actor attempts unauthorized action: <ul style="list-style-type: none"> System restricts access and prompts actors to log in or register. (Continues with use case UC1-1) Actor encounters network issues: <ul style="list-style-type: none"> System displays an error message and retries loading the news feed. |
| Business Rules | |
| Non-Functional Requirement | <ol style="list-style-type: none"> Performance: News feed content should load quickly and efficiently. Personalization: The news feed should adapt to individual actor preferences and interests. |

| | |
|---------------|---|
| Use Case ID | UC-5 |
| Use Case Name | Create post |
| Description | As an artist, I want to create a post to share my artworks to the audience. |
| Actor(s) | Artist |
| Priority | Must Have |

| | |
|----------------------------|--|
| Trigger | The artist clicks on the add post button. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • User has logged in to the website. • User's laptop or phone must have the internet. |
| Post-Condition(s) | <ul style="list-style-type: none"> • The post is created successfully and added to the artist's timeline. |
| Basic Flow | <ol style="list-style-type: none"> 1. The artist clicks on the Create Post button. 2. Insert content for the post. 3. Insert artworks. 4. Click Post. 5. The post is added to their timeline and shown on the news feed. 6. The artist is redirected to the home page. |
| Alternative Flow | |
| Exception Flow | <p>5a. The post can not be created due to network error:</p> <p>5a.1. The system displays the error and asks the artist to add the post again. (Restarts from step 4)</p> |
| Business Rules | |
| Non-Functional Requirement | |

| | |
|----------------------------|--|
| Use Case ID | UC-6 |
| Use Case Name | Manage Wallet |
| Description | This use case allows the admin to manage wallets, which includes view transaction history, view wallet balance of a user. |
| Actor(s) | Admin |
| Priority | Must Have |
| Trigger | Admin click link or button redirect to manage wallet page |
| Pre-Condition(s) | <ul style="list-style-type: none"> - Admin have to log in to the application with an account administrator role. |
| Post-Condition(s) | <ul style="list-style-type: none"> - The user's wallet information is modified according to the admin's actions. |
| Basic Flow | <ol style="list-style-type: none"> 1. Admin selects the option to manage wallets from the administration panel. 2. The system displays an input to input user email. 3. The admin clicks the view button. 4. The system displays the user account balance and transaction history. 5. The admin views the account balance and transaction history. |
| Alternative Flow | <ol style="list-style-type: none"> 1. If no wallets exist, the system displays an appropriate message. 2. If the system cannot update the wallet information due to an error, it notifies the admin. |
| Exception Flow | <p>2a. The system can not show the results because of the network error:</p> <p>2a1. Display an error message indicating network problems. (Stop the use case here).</p> <p>2a2. Prompt the user to check their connection and retry. (Restart from step 1).</p> <p>3a. The system can not show the wallet's info:</p> <p>3a1. Display a message showing no records found. (Stops the use case here).</p> <p>3a2. Prompt the user to check the user id and retry. (Restart from step 2).</p> |
| Business Rules | <ul style="list-style-type: none"> - All changes must be logged for audit purposes. - The system must ensure data integrity during the update process. |
| Non-Functional Requirement | |

| | |
|-------------------|--|
| Use Case ID | UC-7 |
| Use Case Name | View Owner Orders |
| Description | <ol style="list-style-type: none"> 1. As an artist: <ol style="list-style-type: none"> 1.1. I want to view my orders, so that I know who bought the owner right. 2. As an audience: <ol style="list-style-type: none"> 2.1. I want to know what owner's rights I have. |
| Actor(s) | Member |
| Priority | Must Have |
| Trigger | Actor navigates to the Owner page. |
| Pre-Condition(s) | <ul style="list-style-type: none"> ● User has logged in to the website. ● User's laptop or phone must have the internet. |
| Post-Condition(s) | <ul style="list-style-type: none"> ● The user's Owner orders are displayed on the web page. |
| Basic Flow | <ol style="list-style-type: none"> 1. The user clicks on the Owner Orders button. 2. The orders are displayed on the web page. 3. Next actions based on the actor <ol style="list-style-type: none"> 3.1. Artist: <ul style="list-style-type: none"> ● The artist can see who bought the copyright. 3.2. Audience: <ul style="list-style-type: none"> ● The audience can view which artwork copyright they have. |
| Alternative Flow | |
| Exception Flow | <ol style="list-style-type: none"> 2a. The orders can not be displayed due to network error: <ol style="list-style-type: none"> 2a.1 The system displays the error and asks the actor to reload the page. (Restarts from step 2) |
| Business Rules | |

| | |
|----------------------------|--|
| Non-Functional Requirement | |
|----------------------------|--|

| | |
|----------------------------|---|
| Use Case ID | UC-8 |
| Use Case Name | View Wallet Balance |
| Description | This use case describes the admin's ability to view the current balance of application. |
| Actor(s) | Admin |
| Priority | Must Have |
| Trigger | Admin click link or button redirect to manage wallet page |
| Pre-Condition(s) | <ul style="list-style-type: none"> - Admin must be authenticated and authorized to view wallet balances. - Admin have to success of Use Case UC-6 |
| Post-Condition(s) | <ul style="list-style-type: none"> - Admin has accessed the balance information of the application's wallet. |
| Basic Flow | <ol style="list-style-type: none"> 1. Admin go to Wallet page. 2. Admin click to tab show Balance. 3. Admin enter the pin, If successfully, system wil show the balance. |
| Alternative Flow | <ol style="list-style-type: none"> 3a. If wrong pin, send the email alert to admin's email and log to system. |
| Exception Flow | |
| Business Rules | <ul style="list-style-type: none"> - All changes must be logged for audit purposes. - The system must ensure data integrity during the update process. |
| Non-Functional Requirement | |

| | |
|----------------------------|---|
| Use Case ID | UC-9 |
| Use Case Name | View Transaction History |
| Description | This use case describes the admin's ability to view the current balance of the application. |
| Actor(s) | Admin |
| Priority | Must Have |
| Trigger | Admin click link or button redirect to manage wallet page |
| Pre-Condition(s) | <ul style="list-style-type: none"> - Admin must be authenticated and authorized to view transaction history - Admin have to success of Use Case UC-6 |
| Post-Condition(s) | <ul style="list-style-type: none"> - Admin has accessed the transaction history |
| Basic Flow | <ol style="list-style-type: none"> Admin go to Wallet page. Admin click to tab show transaction history. System show transaction history with paging |
| Alternative Flow | |
| Exception Flow | |
| Business Rules | |
| Non-Functional Requirement | |

| | |
|-------------------|---|
| Use Case ID | UC-10 |
| Use Case Name | View pre-order |
| Description | <ol style="list-style-type: none"> 1. As an audience: <ol style="list-style-type: none"> 1.1. I want to track my order status, to know if my order is completed yet so that I can ask for modifications on my order. 2. As an artist: <ol style="list-style-type: none"> 2.1. I want to view the orders and accept or deny an order. |
| Actor(s) | Member |
| Priority | Must Have |
| Trigger | The actor clicks on the View order button on the order showing on the Orders page. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • User has logged in to the website. • User's laptop or phone must have the internet. • The user is on the Pre-order page. |
| Post-Condition(s) | <ul style="list-style-type: none"> • The user's pre-orders are displayed. |
| Basic Flow | <ol style="list-style-type: none"> 1. The actor navigates to the Orders page. 2. The actor clicks on the View order button on the order. 3. Next actions are based on the actor: <ol style="list-style-type: none"> 3.1. Artist: <ol style="list-style-type: none"> 3.1.1. View the order details. 3.1.2. Accept or deny the order. 3.1.3. Prepare the order if accepted. 3.1.4. Send order. 3.1.5. Mark the order as completed. 3.2. Audience: <ol style="list-style-type: none"> 3.2.1. View the order details. 3.2.2. View product. |
| Alternative Flow | <p>3.2.2a: The audience is not satisfied with the product:</p> <p>3.2.2a.1: They can ask for a modification. (Restarts from step 3.1.1)</p> |
| Exception Flow | 2a. The orders can not be displayed due to network error: |

| | |
|----------------------------|--|
| | 2a.1 The system displays the error and asks the actor to reload the page. (Restarts use case) |
| Business Rules | The number of orders an artist can take at once is limited to 3 orders. |
| Non-Functional Requirement | |

| | |
|----------------------------|--|
| Use Case ID | UC-11 |
| Use Case Name | Feedback on pre-orders |
| Description | As an audience, I want to make feedback on my orders. |
| Actor(s) | Audience |
| Priority | Optional |
| Trigger | The audience is on the Pre-orders page. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • User has logged in to the website. • User's laptop or phone must have the internet. • The user is on the pre-orders page. • The artist sent order. |
| Post-Condition(s) | <ul style="list-style-type: none"> • The feedback is made successfully |
| Basic Flow | <ol style="list-style-type: none"> 1. The audience navigates to the pre-orders page. 2. The audience selects a completed order. 3. The audience then provides feedback on the order. 4. The audience presses the Submit Feedback button. 5. The feedback is sent to the artist and saved to the system. |
| Alternative Flow | |
| Exception Flow | |
| Business Rules | The order is marked as completed only if the audience confirms the order completion |
| Non-Functional Requirement | |

| | |
|-------------------|--|
| Use Case ID | UC-12 |
| Use Case Name | Withdraw from wallet |
| Description | As a user, I want to withdraw my money from the residue balance to my bank account. |
| Actor(s) | Member |
| Priority | Must Have |
| Trigger | The artist clicks the Withdraw button on the Wallet page. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • User has logged in to the website. • User's laptop or phone must have the internet. • The user's balance must be more than \$20 to withdraw |
| Post-Condition(s) | <ul style="list-style-type: none"> • The user can withdraw successfully. |
| Basic Flow | <ol style="list-style-type: none"> 1. The user logs in to the Art Hub system. 2. The user navigates to the Wallet section 3. The user clicks on the Withdraw button. 4. User chooses "Withdraw to Bank Account" on the platform. 5. User enters the amount to withdraw. 6. User is redirected to the payment gateway. 7. User fills in their bank account information. 8. The user clicks withdraw. 9. The user's bank account receives the withdrawal amount successfully. 10. User returns to the platform automatically and receives a confirmation notification. |
| Alternative Flow | |
| Exception Flow | <p>5a1. If the user's balance on the platform is less than \$20, 5a.1.1: Cancel withdrawal.(Use case ends). 5a.1.2: Deposit to the account so that the account's balance is more than \$20.(Continues with use case 15)</p> <p>5a2. The user's withdrawal amount is more than the user's balance: 5a.2.1: Cancel withdrawal. (Use case stops).</p> |

| | |
|----------------------------|---|
| | <p>5a.2.2: Deposit to the account so that the account's balance is more than \$20.(Continues with use case 15)</p> <p>8a. If the user's bank account information is incorrect or invalid:</p> <p>6a.1: Cancel. (Use case stops).</p> <p>6a.2: Select another account (Restarts the use case from step 7)</p> <ul style="list-style-type: none"> • If the user loses connection to the internet during the redirect, they can resume the withdrawal process later by reviewing their transaction history. |
| Business Rules | <p>The account's balance must be equal to or more than \$20 to be able to withdraw.</p> <p>The withdrawal amount must be equal to or more than the account's balance.</p> |
| Non-Functional Requirement | <p>The money must be withdrawn to the selected account safely and successfully.</p> |

| | |
|----------------------------|--|
| Use Case ID | UC-13 |
| Use Case Name | View reported artwork |
| Description | Allows the admin to view artworks that have been reported by users for potential policy violations and take appropriate action |
| Actor(s) | Admin |
| Priority | Must Have |
| Trigger | Admin go to the user report page |
| Pre-Condition(s) | <ul style="list-style-type: none"> Admin must be authenticated and authorized to view reported artwork |
| Post-Condition(s) | <ul style="list-style-type: none"> Admin can view report from user |
| Basic Flow | <ol style="list-style-type: none"> Admin go to the user report page Admin click to the view report tab System show report from user |
| Alternative Flow | <ol style="list-style-type: none"> Admin continue with use case UC-14 |
| Exception Flow | 3a. System can not load report, system send toast to notification. |
| Business Rules | |
| Non-Functional Requirement | |

| | |
|----------------------------|--|
| Use Case ID | UC-14 |
| Use Case Name | Hide artwork |
| Description | Enables the admin to temporarily hide artwork from public view, which may be necessary for reviewing content or addressing reports. |
| Actor(s) | Admin |
| Priority | Must Have |
| Trigger | Admin go to the user report page |
| Pre-Condition(s) | <ul style="list-style-type: none"> • Admin must be authenticated and authorized to view reported artwork • Admin continues at UC-13. |
| Post-Condition(s) | <ul style="list-style-type: none"> • Artwork will be hidden. |
| Basic Flow | <ol style="list-style-type: none"> 1. After the use case UC-13, admin decide to hide or not a artwork to be reported. 2. Hide artwork by clicking the hide button. 3. Enter a pin to verify action. 4. If correct, pin to hide artwork and toast the notification to the artist. |
| Alternative Flow | |
| Exception Flow | 3a. If admin enter pin wrong 6 times to stop action and send email to administrator. |
| Business Rules | <ul style="list-style-type: none"> - All changes must be logged for audit purposes. - All changes to artwork must be notified to the artist. |
| Non-Functional Requirement | |

| | |
|-------------------|--|
| Use Case ID | UC-15 |
| Use Case Name | Deposit to the wallet |
| Description | As a user, I want to deposit money into my wallet to be able to pay for my pre-orders and owner orders. |
| Actor(s) | Authenticated |
| Priority | Must Have |
| Trigger | The user clicks on the Deposit button. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • User has logged in to the website. • User's laptop or phone must have the internet. • The user must have a bank account to take the money from |
| Post-Condition(s) | <ul style="list-style-type: none"> • The money must be deposited to the wallet safely and successfully. |
| Basic Flow | <ol style="list-style-type: none"> 1. The user logs in to the Art Hub system. 2. The user navigates to the Wallet section 3. The user clicks on the Deposit button. 4. The user enters the deposit amount. 5. The system redirects the user to the website's third party payment gateway to do the deposit. 6. The user enters the bank account information. 7. The user clicks Deposit. 8. The deposit amount is added to the user's balance. 9. The user is redirected to the platform's website. |
| Alternative Flow | |
| Exception Flow | <p>7a.1: Insufficient bank account balance</p> <p>7a.1.1: Cancel the deposit. (Use case ends)</p> <p>7a.1.2: The user selects another bank account (Restarts from step 6)</p> <p>7a.2: Incorrect bank account information</p> <p>7a.2.1: Cancel the deposit. (Use case ends)</p> <p>7a.2.2: The user selects another bank account (Restarts from step 6)</p> |

| | |
|----------------------------|--|
| Business Rules | |
| Non-Functional Requirement | |

| | |
|-------------------|---|
| Use Case ID | UC-16 |
| Use Case Name | Ban/Unban member |
| Description | The admin has the authority to ban members of the audience, disallowing their access to the platform, often as a result of violating community guidelines. |
| Actor(s) | Admin |
| Priority | Must Have |
| Trigger | An admin identifies a member who has violated the community guidelines or decides to restore access to a previously banned member. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • Admin must be logged into the admin panel with sufficient permissions to ban or unban members. • The member's account in question must exist within the platform. |
| Post-Condition(s) | <ul style="list-style-type: none"> • If banned, the member's account is inactive and they cannot log in or access the platform. • If unbanned, the member's account is reactivated and they regain their access privileges. |
| Basic Flow | <ol style="list-style-type: none"> 1. Admin navigates to the user management section. 2. Admin searches for a specific member by username or ID. 3. Upon selection, the admin chooses to ban or unban the member. 4. The system prompts for confirmation and reason for banning or unbanning. 5. Admin confirms the action and submits the reason. 6. System processes the request and changes the member's account status. 7. System logs the action taken by the admin and sends a notification to the affected member if appropriate. |
| Alternative Flow | 3a. If the member's account is already in the desired state (already banned/unbanned), notify the admin. |

| | |
|----------------------------|---|
| Exception Flow | <p>5a. If the admin does not have permission to ban/unban, the system denies the action and logs the attempt.</p> <p>7a. If the system fails to process the request due to a technical issue, the admin is notified and asked to try again later.</p> |
| Business Rules | <ul style="list-style-type: none"> • Only admins with specific user management roles can ban or unban members. • A reason for banning or unbanning must be provided to maintain an audit trail. • Members have the right to be informed of the reason for the ban. |
| Non-Functional Requirement | |

| | |
|----------------------------|--|
| Use Case ID | UC-17 |
| Use Case Name | Upload Digital Art |
| Description | This use case details the process by which an artist can upload their digital artwork to the platform. It includes steps for file selection, categorization, and submission for review or direct publishing. |
| Actor(s) | Artist |
| Priority | Must Have |
| Trigger | Artist selects the option to upload new artwork within the platform. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • The artist must be registered and logged into the platform. • The artist's account must be in good standing without any restrictions on uploading content. |
| Post-Condition(s) | <ul style="list-style-type: none"> • The digital art is uploaded to the platform and is either pending review or is published, depending on the platform's policy. • The artist's portfolio is updated with the newly uploaded artwork. |
| Basic Flow | <ol style="list-style-type: none"> 1. The artist navigates to the 'Upload Artwork' section of the platform. 2. The artist selects the digital art file from their device. 3. The artist enters details about the artwork such as title, description, tags, and pricing (if applicable). 4. The artist chooses the appropriate category or categories for the artwork. 5. The system performs a validation check on the file type and size. 6. The artist submits the artwork for upload. 7. The system confirms the successful upload and provides a preview to the artist. |
| Alternative Flow | <ol style="list-style-type: none"> 1a. If the artist chooses to save the upload as a draft instead of submitting, the system saves the information and the file for later completion. |
| Exception Flow | <ol style="list-style-type: none"> 2a. If the file type or size is incorrect, the system notifies the artist and does not proceed with the upload. 2b. If the upload fails due to a system error, the artist is prompted to try again later. |
| Business Rules | <ul style="list-style-type: none"> • Only supported file types and sizes can be uploaded. • Artwork must comply with the content guidelines of the platform. • The artist must have all necessary rights to the artwork they are uploading. |
| Non-Functional Requirement | |

| | |
|-------------------|---|
| Use Case ID | UC-18 |
| Use Case Name | Update post |
| Description | As a member, I want to edit my posts so that I can keep them accurate and relevant, improving my content and my audience engagement. |
| Actor(s) | Admin |
| Priority | Must Have |
| Trigger | Member clicks the "Edit" button on a post. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • Member is logged in. • Member has created the post or has editing permissions. |
| Post-Condition(s) | <ul style="list-style-type: none"> • Post content is updated and displayed on the timeline and news feed. |
| Basic Flow | <ol style="list-style-type: none"> 1. Member clicks "Edit" button. 2. System displays editing options. 3. Member modifies content. 4. Member clicks "Save" button. 5. System saves changes and updates post. |
| Alternative Flow | <ul style="list-style-type: none"> • Member cancels editing (returns to post without saving). |
| Exception Flow | <ol style="list-style-type: none"> 1. Network errors 2. Invalid content 3. Insufficient permissions |
| Business Rules | <ul style="list-style-type: none"> • Only the post creator or authorized editors can modify content. • Editing is limited to text and media within the post. • Deleted posts cannot be recovered. |

| | |
|----------------------------|---|
| Non-Functional Requirement | <ul style="list-style-type: none">• Performance: Updates should reflect quickly.• Security: Prevent unauthorized modifications.• Version control: Track changes to posts. |
|----------------------------|---|

| | |
|----------------------------|--|
| Use Case ID | UC-19 |
| Use Case Name | Update Digital Art |
| Description | This use case allows an artist to make updates to digital art that has already been uploaded to the platform. Updates may include changing the art file itself, modifying metadata, or adjusting categorization and pricing. |
| Actor(s) | Artist |
| Priority | Must Have |
| Trigger | Artist chooses to update an existing digital art piece from their portfolio. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • The artist must be logged in to the platform. • The artwork to be updated must be previously uploaded and exist in the artist's portfolio. • The artist has the right to edit the artwork in question. |
| Post-Condition(s) | <ul style="list-style-type: none"> • The digital artwork is updated on the platform. • Any changes are reflected immediately if the artwork is live, or submitted for review if the platform requires it. |
| Basic Flow | <ol style="list-style-type: none"> 1. The artist navigates to their portfolio and selects the artwork they wish to update. 2. The artist clicks on the 'Update Artwork' option. 3. The artist makes desired changes to the artwork file, title, description, tags, categories, and pricing. 4. Click Save changes. 5. The system validates the updated information and file format. 6. The artist submits the updates. 7. The system confirms the successful update and displays the updated artwork. |
| Alternative Flow | 1a. If the artist only wants to update metadata (e.g., title, description, tags), they make the desired changes and submit without uploading a new file. |
| Exception Flow | <p>2a. If the new file format or size is not supported, the system notifies the artist and rejects the update.</p> <p>6a. If the system encounters an error during the update process, it notifies the artist and suggests retrying.</p> |
| Business Rules | <ul style="list-style-type: none"> • Artists must comply with the platform's content guidelines when making updates. • Artists should be able to revert to previous versions if the platform supports version control. |
| Non-Functional Requirement | <ul style="list-style-type: none"> • The update process should be straightforward and not require extensive technical knowledge. • Updates should be processed and reflected in real-time or within a reasonable time frame. |

| | |
|-------------------|--|
| Use Case ID | UC-20 |
| Use Case Name | Delete post |
| Description | As a member, I want to delete my posts when they are no longer relevant or if I change my mind, so that I can maintain control over my content and online presence. |
| Actor(s) | Member |
| Priority | Must Have |
| Trigger | Member clicks the "Delete" button on a post. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • Member is logged in. • Member has created the post or has deletion permissions. |
| Post-Condition(s) | <ul style="list-style-type: none"> • Post is removed from the timeline and news feed. |
| Basic Flow | <ol style="list-style-type: none"> 1. Member clicks "Delete" button. 2. System displays confirmation message. 3. Member confirms deletion. 4. System removes post. |
| Alternative Flow | 4a.1 Member cancels deletion (returns to post). |
| Exception Flow | <ul style="list-style-type: none"> • Network errors • Insufficient permissions |
| Business Rules | <ul style="list-style-type: none"> • Only the post creator or authorized users can delete posts. • Deleting removes all associated comments and reactions. |

| | |
|----------------------------|--|
| Non-Functional Requirement | <ul style="list-style-type: none"> • Security: Ensure proper authentication for deletion • Data integrity: Maintain data consistency after deletion. |
|----------------------------|--|

| | |
|----------------------------|---|
| Use Case ID | UC-21 |
| Use Case Name | Block Follower |
| Description | As a member, I want to block other members to prevent them from seeing my posts and interacting with me. |
| Actor(s) | Member |
| Priority | High |
| Trigger | Member clicks the "Block" button on their or another member's profile or post. |
| Pre-Condition(s) | Member is logged in. |
| Post-Condition(s) | Blocked member can no longer see the blocking member's posts, comments, or profile. |
| Basic Flow | <ol style="list-style-type: none"> 1. Member clicks "Block" button. 2. System confirms blocking action. 3. Member confirms blocking. 4. System blocks the other member. |
| Alternative Flow | Member cancels blocking. |
| Exception Flow | - Network errors |
| Business Rules | - Blocked members cannot see the blocking member's content. - Blocked members cannot send messages or comments to the blocking member. |
| Non-Functional Requirement | - Security: Ensure proper authentication for blocking. |

| | |
|----------------------------|---|
| Use Case ID | UC-22 |
| Use Case Name | View Followers |
| Description | As a member, I want to see a list of my followers. |
| Actor(s) | Member |
| Priority | Medium |
| Trigger | Member clicks the "Followers" button on their profile page. |
| Pre-Condition(s) | Member is logged in. |
| Post-Condition(s) | List of followers is displayed. |
| Basic Flow | <ol style="list-style-type: none"> 1. Member clicks "Followers" button. 2. System retrieves list of followers. 3. System displays list of followers. |
| Alternative Flow | None specified. |
| Exception Flow | - Network errors |
| Business Rules | - Only the member can view their own follower list. |
| Non-Functional Requirement | - Performance: List should load quickly. |

| | |
|----------------------------|---|
| Use Case ID | UC-23 |
| Use Case Name | View Profile |
| Description | As a member, I want to view other members' profiles. |
| Actor(s) | Member |
| Priority | High |
| Trigger | Member clicks on another member's name or profile picture. |
| Pre-Condition(s) | Member is logged in. |
| Post-Condition(s) | Other member's profile is displayed. |
| Basic Flow | <ol style="list-style-type: none"> 1. Member clicks on another member's profile link. 2. System retrieves profile information. 3. System displays profile information. |
| Alternative Flow | None specified. |
| Exception Flow | - Network errors |
| Business Rules | - Members can control what information is displayed on their profiles. |
| Non-Functional Requirement | - Performance: Profile should load quickly. |

| | |
|-------------------|--|
| Use Case ID | UC-24 |
| Use Case Name | Update Profile |
| Description | As a member, I want to update my profile information. |
| Actor(s) | Member |
| Priority | High |
| Trigger | Member clicks the "Edit Profile" button on their profile page. |
| Pre-Condition(s) | Member is logged in. |
| Post-Condition(s) | Profile information is updated. |
| Basic Flow | <ol style="list-style-type: none"> 1. Member clicks "Edit Profile" button. 2. System displays profile editing options. 3. Member modifies profile information. 4. Member clicks "Save" button. 5. System saves changes and updates profile. |
| Alternative Flow | Member cancels editing (returns to profile without saving). |
| Exception Flow | - Network errors |
| Business Rules | - Members can update their profile picture, bio, contact information, and other settings. |

| | |
|----------------------------|---|
| Use Case ID | UC-25 |
| Use Case Name | Upload Digital Art |
| Description | This use case details the process by which an artist can upload their digital artwork to the platform. It includes steps for file selection, categorization, and submission for review or direct publishing. |
| Actor(s) | Artist |
| Priority | Must Have |
| Trigger | Artists select the option to upload new artwork within the platform. |
| Pre-Condition(s) | <ul style="list-style-type: none"> • The artist must be registered and logged into the platform. • The artist's account must be in good standing without any restrictions on uploading content. |
| Post-Condition(s) | <ul style="list-style-type: none"> • The digital art is uploaded to the platform and is either pending review or is published, depending on the platform's policy. • The artist's portfolio is updated with the newly uploaded artwork. |
| Basic Flow | <ol style="list-style-type: none"> 8. The artist navigates to the 'Upload Artwork' section of the platform. 9. The artist selects the digital art file from their device. 10. The artist enters details about the artwork such as title, description, tags, and pricing (if applicable). 11. The artist chooses the appropriate category or categories for the artwork. 12. The system performs a validation check on the file type and size. 13. The artist submits the artwork for upload. 14. The system confirms the successful upload and provides a preview to the artist. |
| Alternative Flow | <ol style="list-style-type: none"> 1a. If the artist chooses to save the upload as a draft instead of submitting, the system saves the information and the file for later completion. |
| Exception Flow | <ol style="list-style-type: none"> 2a. If the file type or size is incorrect, the system notifies the artist and does not proceed with the upload. 2b. If the upload fails due to a system error, the artist is prompted to try again later. |
| Business Rules | <ul style="list-style-type: none"> • Only supported file types and sizes can be uploaded. • Artwork must comply with the content guidelines of the platform. • The artist must have all necessary rights to the artwork they are uploading. |
| Non-Functional Requirement | |

3. Functional Requirements

- Authorization:
 1. SignIn/SignUp: After viewing or scrolling through our amazing gallery, they can choose to sign in using their registered account or email. Or they just need to create a fresh account on the registration page.
 2. Forgot Password: Helping users to retrieve their credentials if their account has already been registered in the system. After entering an appropriate email address, an automatic email containing an OTP code will be sent to the user, type it to the text box, if it's correct, they will be allowed to enter new credentials.
- Audience:
 1. Emote/ Comment on Artwork: Allow users in our system to like and comment on their favourite artwork, if they change their mind, they can easily undo the previous actions such as stop liking the artwork or delete their comment, etc.
 2. Report Artwork: If users encounter an inappropriate content when they are scrolling through our web page, they can report that content by choosing a list of options to the system, and it will be handled immediately by the administrator.
 3. View Artwork: When users sign in to our website, a whole new content of artwork will be displayed to them, their job is just to scroll and interact with the piece of content that catches their attention. The default criteria when they first sign in is "Latest". The users can choose two more options like "Popular" and "Genre".
 4. View Marketplace: If users want to support their favourite creator, they can go to the marketplace page to purchase their artwork. On this page, they will see all the artworks that have been set to be sold, and with the Search and Sort functionalities, users can easily choose based on their interests such as genre, price, name, or creator.
 5. Purchase premium artwork: The users can purchase artwork from their favourite artists. In order to complete the purchase, users can choose a third-party application such as PayPal. After our system has confirmed that the order is accepted, a notification email will be sent to that user.
 6. Follow a creator: The users can click the follow button shown on the screen so as to go on an adventure with that specific artist. And if they change their mind, they can just do the same action, the button will now be changed to "Unfollow".
 7. Add to wishlist: When encountering an artwork that catches their attention, and they want to purchase it. Unfortunately, the price is quite unaffordable, the users can click the button "Add to wishlist" so that in the future, if that artwork is on sale, our system will notify users that have had it in the "Wishlist" page. Otherwise, users can just click the same button to remove it from the list.

8. View Wishlist: The users can track how many or what artwork you have added over the year, they can see the current price of it and have the ability to remove that artwork from the list. There are also Search and Sort functionalities, users can just type in the name of that artwork, or click the Sort button, the default will be “Price Descending”.
9. View Order history: The users can view all orders they have purchased in the past with the Search and Sor functionalities, when clicking on the Sort button, the default keyword will be “Price Ascending”.
10. View Order details: The users click on a specific order in the “Order History” page, it will redirect them to the details page of it, the page will contain all information about that order such as code, name, purchased date, from which creator, price, coupon, and amount.
11. View a creator page: The users can access to their favourite creator page just by clicking on their avatar on the artwork, they can see their name, contact information, how many followers they have, and all of their artworks, they can also click the “Follow” button on this page.
12. Propose an artistic service upon a specific creator: Next to the “Follow” button is the “Service” button, it will create a service form for you to propose your artwork ideas to this specific artist. For example, if you want this artist to draw a portrait of yourself, you can choose from the dropdown options and click “Send”. That service will be sent to the creator “My Service” page, if the creator considers your ideas interesting and doable, you will have a notification that your proposed idea has been accepted, else a “Rejected” notification.
13. Service tracking: The users can see how their service progresses through the deadline that the creator set, they can see how many steps until it is finished. For example, your portrait needs five steps to be completed, when the creator completes one step they can just update that step to “Completed”, default it will be “Pending”, and you can see the price the creator proposes.
14. Add to cart
 - Artist:
 1. Upload Artwork: Upload their content to our system, but it must come with some constraints, such as it can not be over x Mb, etc. Before everyone can view it, it must be approved by the moderator department, they will decide if your content has reached our terms of service or not. If said, your artwork will be on the front of our homepage. Otherwise, a notification containing a message that your content is not allowed on our website.
 2. Edit Artwork: The users can modify their uploaded artwork, they can change it to different status, such as public, private or viewed by some specific people. They can change all information, but it must be approved by the moderator department again.

3. **Sell Artwork:** The users can manage and set their artwork for sales, with the price however they want, the revenue will be 90/10 split between all creators and us. Artwork that is set for sales will be uploaded to the marketplace.
 4. **Open artistic services upon request:** Audiences who want their favourite creator to do some artistic work for them, contact via email or a new chat system implementing in the future, creators can view the service, such as due date, price, and description. The request can either be accepted or rejected. If rejected, an email or notification will be sent to the user, otherwise the creator can negotiate with the requesting user about the due date or ask further questions. After finishing the service, creators will receive their rewards.
 5. **View Followers/Blocks:** Creators can view all of their followers and blocked accounts.
 6. **Manage Profile:** Creators can add artistic certs, promo information to their profile page, social pages, and change their information like name, city, country, email, etc.
 7. **Report Artwork:** If creators see an inappropriate piece of content during their scroll, they can just simply hit the “Report” red button and send it straight to the administrator.
 8. **View Sales Profile:** Creators can view their sales status during their activities on our website. For example, Manage Artwork, have all the artworks that have been put on sales this x year, Sales Report, see how well the creator do, how much they gain with the sort options, Ratings & Reviews, creator can see how the audience’s enjoyment when buying their artwork, appreciate all the reviews the audiences have given them in order to improve some aspects in the future.
- **Administrator:**
 1. **Account Management:** Admin can view all accounts with paging, search, sort that have been created throughout the year. They can disable a specific account if it violates the terms of service in our system. This disable functionality just sets the state of the account to “Disabled”, not completely delete it from the database. And if the owner of the account contacts via email to retrieve it, the admin can just enable it again.
 2. **Artwork Management:** Admin can view all artworks with paging, search, sort that have been uploaded to the system. They can view three states of artwork, such as “Pending”, “Accepted”, “Disabled”.
 3. **Pending report Management:** All reports and feedback sent by the audiences will be handled by the Administrator. All reports when sent will be set to “Pending” state, after guaranteeing that it has been completed, it will be set to “Handled”. For example, if a user comments something inappropriate on an artwork, it will be handled, if an artwork violates our terms of service, it will be handled. If users suggest something to improve their experience, it will also be handled.

4. Dashboard: The frontpage of the admin dashboard, it will contain all the statistics throughout the year our system has gone into working, such as users gain in this x year, revenue of Q1, Q2, Q3 and Q4, some charts to vividly demonstrate our sales or growth the entire year.
5. View Order history: Administrator can view all orders that have been completed in our system, such as name of the artwork, price, date purchased, of which creator and by which user.

4. Non-functional requirements

4.1 External Interfaces

1. The user Interfaces (UI):
 - Graphical The user Interface (GUI): Design elements (buttons, icons, layouts), colour schemes, font types and sizes, responsiveness, multi-language support, screen resolution adaptability, device compatibility (desktop, mobile, tablet).
2. Software Interfaces:
 - APIs: Available APIs, data formats (JSON), authentication methods, rate limits, error handling.
 - Databases: Supported database types (SQL), query languages, connection pooling, data synchronisation, transaction management.
 - External systems: Integration protocols (REST), data exchange formats, security measures, error handling, logging.ill communicate properly with users and with external hardware or software/system elements.

4.2 Quality Attributes

4.2.1 Usability

- After a period of 2-hour training, users can interact with the system without obstacles.
- Frequently used features must be easily accessible and reached with minimal navigation.

4.2.2 Reliability

- The system is 98% up-time in the first year.

4.2.3 Performance

- The system must be able to handle 100 concurrent transactions.
- The system should be scalable to accommodate the user increment.