Support Who You Love - SWYL

Software Requirements Specification (SRS) Template v1.0

This document is an annotated outline intended for specifying software requirements and is adapted from IEEE 29148-2018.

Version v1.0
Prepared By: Nam (Logan) Nguyen
Prepared For CSC 495
SUNY OSWEGO
Sep 18, 2022

Table of Contents

Change Log

Date	Reason For Changes	Version
Sep 18th, 2022	First SRS initialization - 1. Introduction - 3. Requirement	v1.0
Sep 20th, 20222	- Updated 3. Requirement - Updated Appendix A – Tailoring Policies - Replace Appendix B Copyright to Appendix C - Added Appendix B Analysis Models	v1.1
Sep 22nd, 2022	- Section 4 – Verification first draft - Added Use Case Diagram	v1.2
Sep 23rd, 2022	- Touched up 1.2 - Touched up 1.3.2 - Updated 3.5 – Logical Database	v1.2.1
Sep 24th, 2022	- Added activity diagrams - Added Class Diagram	v2.0
Sep 27th, 2022	- Added Platform Overview - Added revised Moore's vision statement - Added revised user stories - Added revised personas - Updated Product Functions - Revised Product features - Updated Glossary table	v3.0

1. Introduction

1.1 Purpose

[Identify the primary reason for the SRS, the system, and the primary audience of the SRS and the system, if they are different. Keep it as short as possible, you will elaborate in future sections.]

a. **Primary reason**: "Support Who You Love" (referred as **SWYL**) is a web-based music donation/NFT marketplace platform
b. **Primary audience of the SRS**:
   - Professor Bastian Tenbergen
   - Nam (Logan) Nguyen
c. **Primary audiences of the system:**
   - Musicians/artists who want to create music products and sell music NFTs (referred as **CREATORS, ARTISTS, AUTHORS**)
   - Users who want to buy music NFTs (referred as **BUYERS**)
   - Fans/supporters who simply just want to enjoy and/or donate rewards (referred as **SUPPORTERS**)
   - Anyone who oversees implementing, deploying and maintaining the project

1.2 Scope

[Identify the scope of the SRS document, what will be covered, what will be excluded? Additionally, identify the general scope of the system you are building. Keep these briefs, use bullet points to help organize the information.]

- **SWYL** covers the activities of:
  - creating and posting music NFTs

  - sending and receiving donations
  - buying and selling music NFTs.
  - transferring the NFTs' ownerships from one party to another
  - hosting a digital community with membership plans
  - Interacting with community
- **SWYL** excludes the activities of:
  - interacting with other intellectual products that are **NOT** music
  - transferring the **ownership** of the music from one party to another
  - auctioning music NFTs

1.3 Product Overview

1.3.0 Platform Overview

I. Moore's Vision Statement

SWYL is a web-based **music donation/NFT marketplace platform** that is compatible with mobile devices THAT provides artists and their fans an e-performance where artists can show off their artworks and the fans can enjoy them then either tip the artists or **offer to buy the digital copies of the artworks in form of NFTs.** UNLIKE other platforms, OUR platform is powered on top of the Ethereum blockchain (or any other blockchain networks) which means the artists can copyright their music for free, **automatically receive donations from fans, be able to sell limited-edition versions of their music in form of NFTs and instantly receive royalty rewards every time their digital copies go on sale again in future**. Also, blockchain technology will make all the transactions, ownerships and copyrights are transparent and immutable on the blockchain network.

II. Personas

1. Music creators

a. **Kitonashi**, age 25, is a music freelancer and small youtuber who is a music enthusiast. He lives in Shimoda which is less than half-an-hour away from Tokyo. He was born in a small family in which his single mom sells vegetable in the rural market. He wants to branch out more and to create an online community where he can showcase his music so his products can come to bigger audiences and hopefully to **earn donation**s from the **fans** and **supporters**. To gain more income, Kitonashi wants to be able to make **digital copies** out of his music products but still be able to **maintain the ownerships/rarity** so he can decide to digitally sell **either only one or multiple** copies of the authenticated and high-quality music to his fans all over the world. He also wants to add to his digital copies a **royalty percentage** which will give him **instant rewards** whenever his digital copies go on sale again in the future. More than that, he wants to **setup his own membership plans** so people can have the

option to **donate** him **monthly** and **yearly**. He also wants to **keep track of** the list of his **supporters** so that he can **offer** some **exclusive content** to those members.

b. Trang, age 27, is a successful crypto trader and a part-time indie music composer who lives in a central countryside province in Vietnam. She was born in a rural family where both of her parents are farmers. Trang has been creating indie music for six years and never went to any music school. She has her own little studio setup with many necessary music-making tools and a mac book pro to produce music. Working with computers and technical devices for many years granting her the technical skills making her confident when it comes to producing and publishing her music. She's seeking for a playground on the internet where she use her **crypto wallet** to represent her identity then create digital copies of her awesome artworks to copyright them on a **blockchain network** just so she can **stream them online or opt to sell them**. She also wants to be able to **configure her own user profile** by decorating it in the way that people would be interested in browsing around her profile page. More than that, she wants to **attach her social media links** to her user profile so other people can find her on other platforms.

c. Linh, age 41, is a part-time bartender, a music instruments teacher and he makes music for sale. He is a resident in one of the biggest cities in Vietnam, Saigon, where he can meet and connect with many people from all backgrounds. His father is a government officer, and his mother is a businesswoman, both retired. He got his music bachelor from Vietnam National Academy of Music. With the huge music background, he loves creating and teaching music but never gains any interest in computer or technologies in general. Frankly, he is not good with computers but still want to create an online community where people from other cities or countries can hang out and enjoy his music. He wants the mentioned community to be **interactive** in which **he can communicate with his fans** by **posting online posts**, **answering questions** in the **comment sections** or simply just to **update his status** or music progress. He also wants to create and share **digital copies of the artwork** in the community so that other people can offer to buy his products. **Linh also wishes to describe his artworks via a form of metadata which includes all the necessary information about the music he produces.** With that being said, he wants a platform that don't require much.

2. Music buyers

a. **Emily**, age 18, just finished high school in England who doesn't want to go to college but open her own business. She was born in a business family with where her father is a business advisor at Kinico Cooperation and his mother is a business analyst at the same company. His business mindset is influenced by both parents since he was still little. Emily doesn't have any talents or ability to make music, but she loves music enough to want to start her career in music industry. She wants a platform or NFT marketplace where she can easily **buy digital**

music copies of the music from other talented artists then. After she buys the music copies, Emily wants the **ownership** of the **copies** get **transferred** to her **crypto wallet**.

b. Trung, age 33, a crypto and decentralized finance enthusiast. He was born in a normal family in the middle of Hanoi, Vietnam where both of his parents are retired musicians. He is a music lover himself and more like a semi-tech savvy since he has been playing video games and study blockchain and crypto trading for quite some time. He wants a marketplace platform where he can easily buy music NFTs. Then he can either **save it in her collection** or update the **copies' metadata then re-list** it to make revenue on it.

3. Donators

a. Zoey, age 49, is a normal English teacher from the U.S. She is from a rural famer family in Iowa. Even though, she is not any near being a tech savvy in fact she is quite bad at computers, she loves media socializing on her phone and watch YouTube videos in her free time. She happens to come across Linh's channel where she enjoys Linh's music. She wants to show her some support but due to her limited personal finance, she just wants to quickly tip her a small amount of money or "one-time donate" the artwork without any account registration involved or membership required. She also wants to have an option to share Linh's music on her other social media platforms.

b. Carla, age 28, is a successful Software Engineer and living in Italy. She was born in a wealthy family where her parents are successful in their own business. She is a tech savvy and really enjoys Japanese culture. She has been a big fan of Kitonashi for a long time and she always find a way to support him and his family. With her freedom of finance and the love she has for Kitonashi's music, she wants to be his monthly/yearly backers to keep supporting him and get the exclusive content from him. She also follows and subscribes to Kitonashi's online community so she can interact with his products through the online community activities.

III. User Stories

- As a SWYL user, I need to login to my crypto wallet - **Trang**
- As a SWYL user, I want to configure my user profile - **Trang**

1. Music creator
   - As a creator, I want to copyright my products by registering it to a blockchain - **Trang**
   - As a creator, I want to upload my music to blockchain - **Trang**
   - As a creator, I want to add metadata to my products - **Linh**
   - As a creator, I want to create either only 1 or multiple limited-edition versions of my song - **Kitonashi**

- As a creator, I want to decide if I want to sell my digital music copies or not - **Trang**
- As a creator, I want to add a royalty fee with my choice of rate to my digital copies of my song so I can get rewards after a secondary sale on my copies - **Kitonashi**
- As a creator, I want to create my own community – **Linh, Kitonashi**
- As a creator, I want to like my fan's comments - **Linh**
- As a creator, I want to reply to my fan's comments - **Linh**
- As a creator, I want to inform all my followers about my newest status/products - **Linh**
- As a creator, I want to keep track of all my followers and supporters who tip me - **Kitonashi**
- As a creator, I want to attach my social media links in my bio profile page - **Trang**
- As a creator, I want to receive donations from my fans/supporters – **Linh, Kitonashi, Trang**
- As a creator, I want to offer exclusive content to my supporters - **Kitonashi**

2. Music Buyers
   - As a buyer, I want to be able to buy digital copies from the artists on the platform – **Emily, Trung**
   - As a buyer, I want the ownerships of the digital copies are transferred to my wallet after I buy it – **Emily**
   - As a buyer, I want to save my buys in my collection tab - **Trung**
   - As a buyer, I want to re-list the copies for sale to make financial profits on it **- Trung**

3. Fans/backers
   - As a fan, I want to one-time donate to the artist - **Zoey**
   - As a fan, I want to share the products on my social medias - **Zoey**
   - As a fan, I want to register for monthly/yearly membership - **Carla**
   - As a fan with membership, I want to like the products my artist posts - **Carla**
   - As a fan with membership, I want to leave a comment on the products - **Carla**
   - As a fan with membership, I want to get exclusive content from my artists - **Carla**

1.3.1 Product Perspective

SWYL is a standalone full-stack web application and is not a part of any larger system.

1.3.2 Product Functions

[This section of the document is intended to describe all the major functions the product will be able to perform. Functions should be organized in such a way that the list of

functions is understandable to anyone who is reading through the document for the first time.]

- **SYSTEM** features:
  - A feature that allows users to sign-in with crypto-wallet
  - A feature that allows users to configure their own accounts
  - A feature that automatically executes NFT transactions
  - A feature to transfer music NFTs from one crypto wallet to another
  - A feature allows creators to receive donations
  - A feature allows creators to receive royalty rewards instantly and automatically after every transaction
  - A feature that can stream music on the platform

- **CREATORS** accessible features:
  - A feature that allows creators to configure the metadata for the NFTs
    - MP3 files
    - Cover images
    - Name
    - Description
    - Quantity
    - Price
  - A feature that allows creators to mint music NFTs with its metadata
  - A feature that allows creators to mint only one music NFT
  - A feature that allows creators to mint multiple music NFTs
  - A feature that allows **creators** add royalty fees to their NFTs
  - A feature that allows **creators** to choose if they want to list their music NFTs for sale
  - A feature that allows **creators** to create a digital community
  - A feature that allows creators to create, manage and interact with posts in the community
  - A feature that allows **creators** to keep track of all their followers and supporters
  - A feature that allows **creators** to attach their social media links in their bio profile page
  - A feature that allows **creators** to offer exclusive content to supporters who subscribe the membership

- **BUYERS** accessible features
  - A feature that allows **buyers** to buy the music NFTs
  - A feature that allows **buyers** to save their buys in their collection page

- A feature that allows **buyers** to edit the copies' metadata
- A feature that allows **buyers** to re-list the NFTs

- **FANS** accessible features
  - A feature that allows **fans** to one-time donate the product & artists
  - A feature that allows **fans** to share the products on other social media platforms
  - A feature that allows **fans** to register for monthly/yearly memberships
  - A feature that allows **fans** interact in the community
  - A feature that allows **fans** with membership to interact with exclusive contents

1.3.3 User characteristics

[Describe the characteristics of the product's intended group of users. Include things such as technical expertise, and any information that may impact usability or accessibility to the product.]

- **CREATORS, ARTISTS, AUTHORS**: Musicians/artists who create music products and/or sell music NFTs
- **BUYERS**: Users who want to buy and sell music NFTs
- **FANS**: Fans/supporters who simply just want to enjoy and/or donate rewards

1.3.4 Limitations

[Identify any limitations that will impact the development. Include things such as hardware limitations, safety and security considerations, quality requirements, regulatory requirements and/or policies, etc.]

- **TBD**

1.4 Definitions

[Include any terms and definitions needed to understand the SRS or the System Under Development (SUD), terms placed here should also be placed in the Appendix. Terms should use the full name and the general definition of that term, any abbreviations that will be used in the document and the source should be placed in Appendix 5.2]

Term	Definition

Blockchain	a type of Digital Ledger Technology that consists of growing list of records, called blocks, that are securely linked together using cryptography.
Ethereum	a decentralized, open source blockchain with smart contract functionality
Polygon	an Ethereum layer-2 protocol and framework for building interconnected blockchain ecosystems
Smart Contract	simply programs stored on a blockchain that run when predetermined conditions are met
Music non-fungible token (NFT)	is a music digital asset that can be identified through its unique qualities held within its metadata
NFT marketplace	A marketplace which houses all for-sale NFTs on the platform
SWYL Service Fee (SSF)	SSF is a service fee that only applies to buying NFT and selling NFT activity. It states that a 2% fee based on the total transaction amount will be deducted from each party and get transferred to SWYL Service Wallet Address
SWYL Service Wallet Address (SSWA)	SSWA is an Ethereum crypto wallet address which oversees receiving SSF from NFT transactions
Royalty Fees (RF)	RF is a fee that get transferred to original authors instantly and automatically on the secondary sales
TBD	**TBD**

2. References

[Include citations to external sources and resources in this section. References to other internal documents can be placed here but should also be referenced in the appendix.
Example: ISO/IEC/IEEE 29148.2018, Systems and software engineering
Life cycle processes — Requirements engineering]

TBD

3. Requirements

[This section should contain all the software requirements at a level of detail sufficient enough to enable designers to design a system, and for testers to test that system, in a way that satisfies the requirements. Each requirement should be perceivable by users, operators, or other external systems. At minimum, the description should include the inputs and outputs of the system, and all functions performed by the system in response to an input or in support of an output. Specific requirements should include the following characteristics:

- *Correct*
- *Unambiguous*
- *Complete*
- *Consistent*
- *Verifiable*
- *Modifiable*
- *Traceable*
- *Ranked for importance and/or stability*
- *uniquely identifiable (usually by numbering)*
- *organized in a way that allows for maximum readability*

The purpose of the requirement is not to dictate design, but rather to guide designers to make the safest, most correct version of the system possible. Do not attempt to build solutions to your written requirements]

3.1 Functions

[Define the fundamental actions that the system must stake in order to accept inputs and generate outputs. It may make sense to organize or partition the functional requirements into sub-functions or sub-processes, do not expect development to mimic this organization.

- Validity checks on the inputs
    - The system shall check the validity of a crypto wallet when users try to sign in
    - The system shall check the validity of the input music files when creators want to mint NFTs (shall limit to use only .mp3 file type)

    - The system shall check the validity of the input image files during the process of filling metadata for the NFTs (shall be in the type of .png, .jpeg, .svg, .jpg, etc.)
    - The system shall check the validity of the input metadata for a community post
    - TBD
- Exact sequence of operations
    - The system shall receive the NFT metadata then generate a new NFT on a blockchain network
    - The system shall receive transaction information then generate success/fail notifications
    - The system shall receive the posts' metadata then generate new posts to the digital communities
    - The system shall receive the input from users to generate comments to posts
    - The system shall receive the input from users to generate reactions to posts
    - TBD
- Responses to abnormal situations including:
    - Overflow
        - If the system experiences data overflow, it shall temporarily stop accepting new inputs
    - Error Handling and Recovery
        - The system shall prompt Page Not Found error message if experiences unregistered routes
        - The system shall prompt Unauthorized error message if users are not authorized to access resources
        - The system shall prompt Bad request error message if users inputs are not valid
    - Database gateway
        - The system shall prompt Server error message if the backend is not function properly
- TBD

3.2 Performance Requirements

[In measurable terms, specify the numerical requirements of the system. Include static performance requirements such as the number of terminals, simultaneous users, etc. As well as dynamic performance requirements such as the number of tasks able to be completed in a set period of time.]

- The system shall be available and compatible with many web broswers
- The system shall be capable of supporting at least 1,000 users concurrently
- The system shall be able to handle multiple tasks (minting NFTs, listing NFTs, buying NFTs, setting up transaction reports, etc.) during a user session

- The system shall be able to make sure that all visible pages of the system respond in timely manner
- The system shall be able to store more than 100, 000 music NFTs (approximately 10mb/file)
- The system shall be available online for 24/7
- **TBD**

3.3 Usability Requirements

[Define usability and quality requirements that are measurable in effectiveness, efficiency, satisfaction, and in avoidance of harm that could arise from specific use cases.]

- The system shall conveniently and neatly prompt helpful information about all the abbreviations or technical terms used within the site
- The system shall have a succinct and transparent forms including descriptive content to guide users what shall be done while trying to submit any action
- The system shall have self-descriptive buttons to help users navigate through the site
- **TBD**

3.4 Interface Requirements

[List all inputs and outputs from the system. It should mirror but not repeat the information found in sections 4.2- 4.6. For each defined interface, be sure to include:

- *the name of the item*
- *description of the purpose of the interface*
- *source of input OR output destination*
- *range, accuracy and/or tolerance*
- *units of measurement*
- *timing*
- *I/O relationships*
- *data formats*
- *command formats*

any information included within the I/O.]]

3.4.1 Input Interface Requirements

- Social media interface:
    - Name: community
    - Purpose: offers an e-performance where artists and fans can digitally hang out
    - Source of input: platform users
    - Data formats: JSON

- Registration interface:
    - Name: Registration page
    - Purpose: guides users on how to register for a page
    - Source of input: platform users
    - Data formats: JSON
- NFT registering interface:
    - Name: NFT registering
    - Purpose: guides users to fill out the metadata so that NFTs can be minted
    - Source of input: platform users
    - Data formats: JSON
- NFT buying interface
    - Name: NFT Buying
    - Purpose: provides the guides on how to execute transactions to buy NFT
    - Source of input: platform users
    - Data formats: JSON
- Marketplace interface:
    - Name: NFT marketplace
    - Purpose: offers a marketplace where seller and buyer can interact to buy and sell music NFTs
    - Source of input: platform users
    - Data formats: JSON
- Membership setting interface
    - Name: NFT marketplace
    - Purpose: offers a page where artists can set up membership plans
    - Source of input: platform users
    - Data formats: JSON
- Membership registering interface
    - Name: NFT marketplace
    - Purpose: offers a page where users can register for membership
    - Source of input: platform users
    - Data formats: JSON
- Profile interface
    - Name: profile page
    - Purpose: offers an area for users to showcase their profile
    - Source of input: platform users
    - Data formats: JSON
- Profile settings interface
    - Name: profile page
    - Purpose: offers a wizard where users can configure their information
    - Source of input: platform users

    - Data formats: JSON
- Searching interface
    - Name: search interface
    - Purpose: search for users on the platform
    - Source of input: platform users
    - Data formats: JSON
- NFT history transactions interface
    - Name: NFT history
    - Purpose: show a list of transactions of the NFTs
    - Source of input: platform users
    - Data formats: JSON

3.5 Logical Database Requirements

[Identify the logical requirements for information that will be place in a database.

- Two main databases || storages:
    - On-chain data (i.e., NFT metadata, marketplace transactions) will be stored on a blockchain network
    - Off-chain data (i.e., users' data, community data, memberships data, etc.) will be stored on a NoSQL database
- One helper storage
    - Audio files and cover image files are retained in NFT.storage & IPFS
- Types of information that will be used:
    - User data
    - Membership plans data
    - Community posts data
    - Community supporter lists data
    - NFTs' metadata
    - Marketplace transactions data
- Frequency of use: High
- Accessibility: Always available
- Security:
    - Users' information stays private and secured inside the database
    - Membership plans data is public to the platform users
    - Community posts' information is public to the platform users
    - Community supporter lists' information is public to the platform useters
    - NFTs' metadata information maintains public and transparent on the blockchain
    - Marketplace transactions' information stay public and transparent on the blockchain
- data retention

- On-chain data (NFTs, marketplace transactions, supporter lists, etc.) will be retained firmly and immutably on blockchain network
- Off-chain data (Users' information, Community's information, etc.) will be retained in a NoSQL database
- Audio files and cover image files are retained in NFT.storage & IPFS

3.6 Design Constraints

[List any constraints on the system. These constraints should be from external sources such as regulatory standards, legal, or project limitations.]

- The system smart contracts must be compatible with Ethereum Virtual Machine
- The system must be deployed to one of the Layer 2 blockchain network (Polygon)
- The system must not violate Customer and Privacy Act.
  - The system must ensure the user information is private and secured
  - The system must ensure to publish only information that is disclosed with the consent of the user
- The system must be completed within the Fall Semester of 2022
- Only users with crypto wallet can perform NFT-related actions
- NFT music files are not heavier than 10mb
- **TBD**

3.7 Software System Attributes

[For each of the attributes of the software system (Reliability, Availability, Security, Maintainability, Portability, etc.), list the factors that will establish functionality or stability. For example, when establishing requirements for the Security attribute, you may include one that restricts communication between two one area of the program and another.]

3.7.1 Reliability:

- The system performs correctly in case of failures. Self-detecting and self-healing in case of infrastructure problems, malicious attacks

3.7.2 Availability:

- TBD

3.7.3 Security:

- The system guarantees confidentiality, integrity and protection against malicious attacks

3.7.4 Maintainability:

- The system is simple to operate and evolve over time without major code refactors

TBD

3.8 Supporting Information

[Add any additional information needed to understand the SRS, include things like background information, problem descriptions, packaging instructions for code, sample input/output formats, etc.]

4. Verification

[List all inputs and outputs from the system. It should mirror but not repeat the information found in sections 3.1- 3.8. For each defined interface, be sure to include:

- *the name of the item*
- *description of the purpose of the interface*
- *source of input OR output destination*
- *range, accuracy and/or tolerance*
- *units of measurement*
- *timing*
- *I/O relationships*
- *data formats*
- *command formats*
- *any information included within the I/O.]*

4.1 Functions

[See sections 4.0, 3.1 for specific directions about what outputs should be included here.]

4.1.1 System features

4.1.1.1 **Sign into the platform**

a. HIGH priority

- A-1: System shall provide users a sign-in with crypto-wallet wizard to let users sign-in

b. Stimulus/response sequence

- B-1: User action: Users will be able to sign into the platform with the help of industry-standard crypto currency wallet

- B-2: System response: The system shell let users sign into the platform to execute transactions

c. Functional requirements

- C-1: The system shall let users sign into the platform with a valid wallet account
- C-2: The system shall not let users sign into the platform with an invalid wallet account
- C-3: The system shall let users to log out of the platform

4.1.1.2 **Configure their own user profile**

a. HIGH priority

- A-1: System shall provide users an ability to configure their own user profile

b. Stimulus/response sequence

- B-1: User action: Users will be able to configure their own user profile
- B-2: System response: The system shell let users update their user profile

c. Functional requirements

- C-1: The system shall let users submit a JSON file which contains new profile information
- C-2: The system shall process the JSON file and update the user profile with the new information
- C-3: The system shall show to the user new updated information on user's profile page

4.1.1.3 **Executing NFT transactions**

a. HIGH priority

- A-1: System shall be able to process and execute NFT transactions

b. Stimulus/Response Sequence

- B-1: User action:
  - Buyers click on the "buy" button to start NFT buying process
  - Provide valid inputs for buying transaction form
  - Buyers click on confirm button to confirm the transactions
- B-2: System response: The system shall respond with failed or successful messages

c. Functional requirements

- C-1: The system shall be able to process the buying inputs and ensure that the input is valid in terms of matching data fields, valid wallet address, valid amount, etc.
- C-2: The system shall be able to calculate a total 4% SWYL service fee (SSF) based on the total transaction amount

- C-2: The system shall transfer the NFT price amount – SSF directly to creators
- C-3: The system shall transfer the SSF to SWYL Service Wallet Address (SSWA)

4.1.1.4 **Transfer music NFTs from one crypto wallet to another**

a. HIGH priority
- A-1: System shall transfer NFTs' ownership from sellers to buyers

b. Stimulus/response sequence
- B-1: System response: The system shall transfer the NFTs' ownership from sellers to buyers

c. Functional requirements
- C-1: The system shall process executed transaction and get the information from sellers and buyers
- C-2: The system shall transfer the ownership from seller to buyers

4.1.1.5 **Executing donating transactions**

a. HIGH priority
- A-1: System shall be able to process and execute donating transactions

b. Stimulus/Response Sequence
- B-1: User action:
  - Click on the "donate" button to start donating process
  - Provide valid inputs for donating transaction form
- B-2: System response: The system shall respond with transaction failed or successful message

c. Functional requirements
- C-1: The system shall be able to process the donating inputs and ensure that the input is valid in terms of matching data fields, valid wallet address, valid amount, etc.
- C-2: The system shall transfer the donating amount directly to creators
- C-3: The system shall distinguish direct donating amount and NFT donating amount

4.1.1.6 **Transfer royalty rewards instantly to original authors**

a. HIGH priority
- A-1: System shall be able to transfer the RF to original authors

b. Stimulus/Response Sequence
- B-1: System response: The system shall transfer the RF to original authors instantly and automatically

c. Functional requirements
- C-1: The system shall be able to process the donating inputs and ensure that the input is valid in terms of matching data fields, valid wallet address, valid amount, etc.
- C-2: The system shall transfer the donating amount directly to creators
- C-3: The system shall distinguish direct donating amount and NFT donating amount

4.1.1.6 **Music streaming**

a. HIGH priority
- A-1: The system shall be able to stream the music online on the platform

b. Stimulus/Response Sequence
- B-1: System response: The system shall stream the music online

c. Functional requirements
- C-1: The system shall process the NFT metadata from blockchain to retrieve the audio file
- C-2: The system shall play the audio sound online when a user click play button

4.1.2 Creator Features

4.1.2.1 **Mint Save products' metadata as NFTs on chosen blockchain**

a. HIGH priority
- A-1: System shall be able to receive metadata input from users and integrate it to the chosen blockchain

b. Stimulus/Response Sequence
- B-1: User action: Provide valid metadata input
- B-2: System response: The system shall be able to process the metadata and mint it to the chosen blockchain as NFTs

c. Functional requirements
- C-1: The system shall provide a form where users can type in the metadata inputs
- C-2: The system shall ensure that the metadata received is valid in terms of matching data fields correctly, required fields, etc.
- C-3: The system shall offer users an option to mint only 1 unique NFT
- C-4: The system shall offer users an option to mint multiple unique NFTs
- C-5: The system shall offer users an option to modify and add a royalty fee to their NFTs
- C-6: The system shall be able to process the metadata, create and save NFTs on the chosen blockchain and record the original creator

- C-7: The system shall be able to generate a share link and attach it to the NFT's metadata after it gets minted

4.1.2.2 **Listing NFT to showcase and/or for sale to the marketplace**

a. HIGH priority
- A-1: System shall offer users an option to list new created NFTs to showcase and/or for sale

b. Stimulus/Response Sequence
- B-1: User action: Provide valid listing choices
- B-2: System response: The system shall be able to list the NFTs based on listing choices

c. Functional requirements
- C-1: The system shall list the new created NFTs to creators' gallery page by default
- C-2: The system shall be able to process the users' listing choices then list the NFT either on marketplace or profile gallery page based on listing choices
- C-3 The system shall list the NFTs along with the received metadata

4.1.2.3 **Creating community**

a. LOW priority
- A-1: System shall offer a digital community for artists to gather their fans

b. Stimulus/Response Sequence
- B-1: User action:
  - Creators can create a community and its materials
  - Fans can interact with the community's materials
- B-2: System response: The system shall create a community under creators' account

c. Functional requirements
- C-1: The system shall let creators create one and only one digital community
- C-2: The creators shall be able to configure their own community
- C-3: The creators shall be able to attach their gallery page to the community
- C-5: Fans can interact in community

4.1.2.4 **Creating manage and interact with community posts**

a. LOW priority
- A-1: System shall allow users to interact with community post

b. Stimulus/Response Sequence

- B-1: User action:
  - Creators can create post with metadata
  - Fans can interact with post (i.e., comment, like, etc.)
- B-2: System response: The system shall create community posts creators' account and let fans interact with it

c. Functional requirements

- C-1: Creators shall be able to create posts along with its metadata
- C-2: Creators shall be able to update their posts
- C-3: Creators shall be able to receive comments on their posts from fans
- C-4: Creators shall be able to reply to their comments
- C-4: Creators shall be able to delete their posts

4.1.2.5 **Keep track of all their followers and supporters**

a. LOW priority

- A-1: System shall allow users to keep track of their followers and supporters

b. Stimulus/Response Sequence

- B-1: System response: The system shall add the supporters' wallet address to the supporters list

c. Functional requirements

- C-1: The system shall automatically add the supporters' wallet address to the creators' supporters list on the blockchain
- C-2: The system shall be able to retrieve the list from blockchain and show it to creators
- C-3: Creators shall be able to show the supporters list publicly

4.1.2.6 **Attach links to other social media platforms in profile page**

a. LOW priority

- A-1: System shall allow users to attach links to other social media platforms

b. Stimulus/Response Sequence

- B-1: User action: Submit a JSON file including the information about links to other social media platforms
- B-1: System response: The system shall process the JSON file then add the links to the user model database

c. Functional requirements

- C-1: The system shall provide a form for users to submit the JSON file of social media links
- C-2: The system shall process the JSON file then add the links to the user model in database

- C-3: The system shall let creators to choose if they want to show their social media links publicly

4.1.2.7 **Offer exclusive content to supporters who subscribe the membership**

a. LOW priority
- A-1: System shall allow users to offer exclusive content to supporters who subscribe their membership

b. Stimulus/Response Sequence
- B-1: User action: Submit a JSON file includes information about the content
- B-1: System response: The system shall be able to show the exclusive content to only subscribed members

c. Functional requirements
- C-1: The system shall offer a form which allows users to restrict the exclusive content to only subscribed members
- C-2: The system shall be able to show the exclusive content to only subscribed members

4.1.3 Buyer Features

4.1.3.1 **Buy the NFTs**

a. HIGH priority
- A-1: System shall allow users to buy other users' NFTs

b. Stimulus/Response Sequence
- B-1: User action: Click buy button and confirm transaction
- B-1: System response: The system shall be able to execute the buy transactions

c. Functional requirements
- C-1: The system shall offer a form for users to input in their buy transactions
- C-2: The system shall execute the buy transactions if the input is valid
- C-3: The system shall be able to transfer the right price amount from buyers to sellers
- C-4: After the transaction is done executed, the system shall be able to remove the NFT from marketplace

4.1.3.2 **Save their buys in their collection page**

a. HIGH priority
- A-1: System shall allow users to save their buys into their collection page

b. Stimulus/Response Sequence
- B-1: System response: The system shall add the new bough NFTs into buyers' collection page automatically after a sale

c. Functional requirements
- C-1: The system shall add the new bough NFTs into buyers' collection page automatically after a sale

4.1.3.3 **Re-list the NFTs**

a. HIGH priority
- A-1: System shall allow buyers to re-list the new bought NFTs for sale

b. Stimulus/Response Sequence
- B-1: User action: Edit NFT's metadata, then click re-list button
- B-1: System response: The system shall add the NFT back to marketplace with new information

c. Functional requirements
- C-1: The system shall offer a form where users can add and edit metadata
- C-2: The system shall process the new information and add the NFT back to marketplace

4.1.4 Fans Features

4.1.4.1 **One-time donate**

a. HIGH priority
- A-1: The system shall allow fans to one-time donate the product and/or artist

b. Stimulus/Response Sequence
- B-1: User action: submit donation form then click acknowledge button
- B-1: System response: The system shall process the donation information and execute the donation

c. Functional requirements
- C-1: The system shall offer a form where fans can input in the donation information
- C-2: The system shall be able to process the donation information and execute the donation
- C-3: The system shall be able to transfer the donation from fans' wallet address to creators' wallet address

4.1.4.2 **Share the products on other social media platforms**

a. HIGH priority
- A-1: The system shall allow fans to copy share links of the NFTs

b. Stimulus/Response Sequence
- B-1: User action: copy share links and share it on other platforms
- B-1: System response: The system shall be able to offer user the share links

c. Functional requirements

- C-1: The system shall be able to let user copy the share links on each NFTs

4.1.4.3 **Subscribe membership plans**

a. HIGH priority
- A-1: The system shall allow fans to subscribe their favorite artists membership plans

b. Stimulus/Response Sequence
- B-1: User action: Click subscribe membership plan
- B-1: System response: The system shall add the user wallet address to supporter list

c. Functional requirements
- C-1: The system shall be able to process information about membership subscription
- C-2: The system shall be able to notify artists about new subscription
- C-3: The system shall be able to add the fans' wallet address to the supporters list
- C-3: Once a user register for a membership plan, the user shall be able to cancel the membership right away if they wish but the user must not be able to get the refund

4.1.4.4 **Comment community posts as users**

a. LOW priority

b. Stimulus/Response Sequence
- B-1: User action: Comment on community posts

c. Functional requirements
- System shall have a text area box for fans to input in comments
- Fans shall be able to click comment button to post the comments to the posts
- Fans shall be able to click edit button to edit the comments
- Fans shall be able to click delete button to delete the comments
- System shall have a list of emoji for fans to choose to react to posts
- Fans shall be able to pick an emoji from the list
- Fans shall be able to change their emoji to another emoji in the list
- Fans shall be able to delete the reaction by click to the same emoji they already chose

4.1.4.5 **Interact with exclusive contents**

a. HIGH priority

- A-1: The system shall allow fans with membership to access to exclusive contents set by the artists

b. Stimulus/Response Sequence

- B-1: System response: The system shall allow users with membership to access exclusive contents and block users with no membership from accessing exclusive content

c. Functional requirements

- C-1: The system shall be able to distinguish which users have memberships
- C-2: The system shall be able to allow users with memberships to access exclusive content set by artists
- C-3: The system shall be able to block users with no memberships from accessing exclusive content set by artists

4.1.5 Extra Features

4.1.5.1 **Searching feature**

a. MEDIUM priority

- A-1: System shall be able to allow users to search for other users and NFTs

b. Stimulus/Response Sequence

- B-1: User action: Type in the input needed to look up and click search button
- B-2: System response: The system shall respond with a list of result(s)

c. Functional requirements

- System shall offer a text box for Searching component
- System shall receive the input from users, process it and look up the search in the database
- System shall return a list of result(s) if the input is valid
- System shall return an empty list if the input is invalid/not found

4.1.5.2 **Add NFT to favorite**

a. LOW priority

- A-1: System shall be able to allow users to save NFT as favorites

b. Stimulus/Response Sequence

- B-1: User action: Click add to favorite button
- B-2: System response: The system shall add the NFT to favorited list

c. Functional requirements

- System shall offer a add to favorite button
- System shall make sure the item is not in “favorited” to add to “favorited”

- System shall remove NFT if the item is already in “favorited”

4.1.5.3 **NFT history transactions**

a. MEDIUM priority
- A-1: System shall be able record NFTs’ history transactions

b. Stimulus/Response Sequence
- B-1: System response: The system shall return a list of NFTs’ history transactions

c. Functional requirements
- C-1: The system shall keep recording the NFTs’ transactions
- C-2: The system shell return a list of NFTs’ history transactions when needed

4.2 Performance Requirements

[See sections 4.0, 3.2 for specific directions about what outputs should be included here.]

- The system shall be available and compatible with all modern webs browser
- The system shall be capable of supporting at least 10,000 users concurrently
- The system shall be able to handle multiple tasks (minting NFTs, listing NFTs, buying NFTs, setting up transaction reports, etc.) during a user session
- The system shall be able to make sure that all visible pages of the system respond in timely manner
- The system shall be able to store more than 100, 000 music NFTs (approximately 10mb/file)
- The system shall be available online for 24/7

4.3 Usability Requirements

[See sections 4.0, 3.3 for specific directions about what outputs should be included here.]

- The system shall conveniently and neatly prompt helpful information about all the abbreviations or technical terms used within the site
- The system shall have a succinct and transparent forms including descriptive content to guide users what shall be done while trying to submit any action
- The system shall have self-descriptive buttons to help users navigate through the site

4.4 Interface Requirements

[See sections 4.0, 3.4 for specific directions about what outputs should be included here.]

4.4.1 social media (Community) interface

- TODO: insert community mockup

4.4.2 Registration interface
- TODO: insert Registration mockup

4.4.3 NFT registering interface
- TODO: insert NFT registering mockup

4.4.4 NFT buying interface
- TODO: insert NFT buying mockup

4.4.5 NFT marketplace interface
- TODO: insert marketplace mockup

4.4.6 Membership setting interface
- TODO: insert Membership setting mockup

4.4.7 Membership registering interface
- TODO: insert Membership registering mockup

4.4.8 Profile interface
- TODO: insert Profile mockup

4.4.9 Profile setting interface
- TODO: insert Profile setting mockup

4.4.10 Searching interface
- TODO: insert Searching mockup

4.4.11 NFT history interface
- TODO: insert NFT history mockup

4.5 Logical Database Requirements

[See sections 4.0, 3.5 for specific directions about what outputs should be included here.]

- Two main databases:
    - On-chain data will be stored on a blockchain network
    - Off-chain data will be stored on a NoSQL database
- One helper storage

    - Audio files and cover image files are retained in NFT.storage & IPFS
- Types of information that will be used:
    - User data
    - Membership plans data
    - Community posts data
    - Community supporter lists data
    - NFTs' metadata
    - Marketplace transactions data
- Frequency of use: High
- Accessibility: Always available
- Security:
    - Users' information stays private and secured inside the database
    - Membership plans data is public to the platform users
    - Community posts' information is public to the platform users
    - Community supporter lists' information is public to the platform useters
    - NFTs' metadata information maintains public and transparent on the blockchain
    - Marketplace transactions' information stay public and transparent on the blockchain
- data retention
    - On-chain data (NFTs, marketplace transactions, supporter lists, etc.) will be retained firmly and immutably on blockchain network
    - Off-chain data (Users' information, Community's information, etc.) will be retained in a NoSQL database
    - Audio files and cover image files are retained in NFT.storage & IPFS

4.6 Design Constraints

[See sections 4.0, 3.6 for specific directions about what outputs should be included here.]

- The system smart contracts must be compatible with Ethereum Virtual Machine
- The system must be deployed to one of the Layer 2 blockchain network (Polygon)
- The system must not violate Customer and Privacy Act.
    - The system must ensure the user information is private and secured
    - The system must ensure to publish only information that is disclosed with the consent of the user
- The system must be completed within the Fall Semester of 2022
- Only users with crypto wallet can perform NFT-related actions
- TBD

4.7 Software System Attributes

[See sections 4.0, 3.7 for specific directions about what outputs should be included here.]

3.7.1 Reliability:

- The system performs correctly in case of failures. Self-detecting and self-healing in case of infrastructure problems, malicious attacks

3.7.3 Security:

- The system guarantees confidentiality, integrity, and protection against malicious attacks

3.7.4 Maintainability:

- The system is simple to operate and evolve over time without major code refactors

5. Appendix A – Tailoring Policies

5.1 Assumptions and dependencies

[Identify any and all factors that may impact the implementation and execution of the requirements written below. These factors do not add a constraint but may impact development if they are changed. Example: a major update to an operating system(OS) on which the SUD is intended to run impacts the implementation of one of the core features. The version of the OS that the system had intended to run on should be listed in this section.]

5.2 Acronyms and Abbreviations

Term	Definition	Abbreviation
Support Who You Love	The final version of the standalone web application of the system under development.	*SWYL*
Blockchain	a type of Digital Ledger Technology that consists of growing list of records, called blocks, that are securely linked together using cryptography.	
Ethereum	a decentralized, open source blockchain with smart contract functionality	ETH/eth

Polygon	an Ethereum layer-2 protocol and framework for building interconnected blockchain ecosystems	
Smart Contract	programs stored on a blockchain that run when predetermined conditions are met	
Music non-fungible token	is a music digital asset that can be identified through its unique qualities held within its metadata	Music NFT
NFT marketplace	A marketplace which houses all for-sale NFTs on the platform	
SWYL Service Fee	SSF is a service fee that only applies to buying NFT and selling NFT activity. It states that a 2% fee based on the total transaction amount will be deducted from each party and get transferred to SWYL Service Wallet Address	SSF
SWYL Service Wallet Address	SSWA is an Ethereum crypto wallet address which oversees receiving SSF from NFT transactions	SSWA
Royalty Fees	RF is a fee that get transferred to original authors instantly and automatically on the secondary sales	RF
NFT minting page	A page available to creators allowing creators to fill in NFT metadata then mint it to the blockchain	
Mint an NFT	System process the input metadata and create an NFT to the chosen blockchain	
Donations	The number of crypto/money fans donate to artists	
Crypto transactions	The transactions that are made by interacting with smart contracts (selling & buying NFTs or sending donations)	
Creators	Musicians/artists who create music products and sell music NFTs	
Buyers	Users who want to buy music NFTs	
Supporters	Fans/supporters who simply just want to enjoy and/or donate rewards	
Register/profile page	The page allows users to configure their personal database	
Digital Community	A digital community acts as a social media where Creators and Supporters can hang out	
Community Posts	Social media posts in community	
NFT minting page	A page available to creators allowing creators to fill in NFT metadata then mint it to the blockchain	

Marketplace page	A page available to anyone on the platform offers a marketplace where showcase all the NFTs and allows users to buy and sell NFTs	
Membership page	A page to showcase membership plans	
Membership configuration page	A page allows artists to manage their own membership plans	
Searching function	A function allows users on the platform search to find other users or NFTs	
NFT history transactions page	A page showcase a list of transaction of the NFTs	

5.3 Tailoring Policies

Tailoring is not a requirement to bring the document into compliance with the standards set by IEEE 29148-2018. Tailoring should only occur when conformance to the standard is not possible or practical. The act of tailoring is the modification and/or removal of one of the content sections outlined in this document, adding additional information items for organization is not considered tailoring. Tailoring should only occur when factors or circumstances:

- surround an organization that is using the document
- influence a project using this document to meet an agreement
- reflect the needs of an organization.

When tailoring the document, the following activities shall be implemented:

- Identify and document the circumstances that may influence tailoring.
    - novelty, size and complexity
    - stability of operating environments
    - variety in operating environments
    - starting date and duration
    - emerging technology
    - availability of services of enabling systems
    - other standards with which the document needs to conform.
- Identify and get input from all parties impacted by the tailoring process.
    - Such as stakeholders, contributors, and other interested parties
- Delete the information contents that require tailoring.

6. Appendix B – Analysis Models

6.1 Use Case Diagram

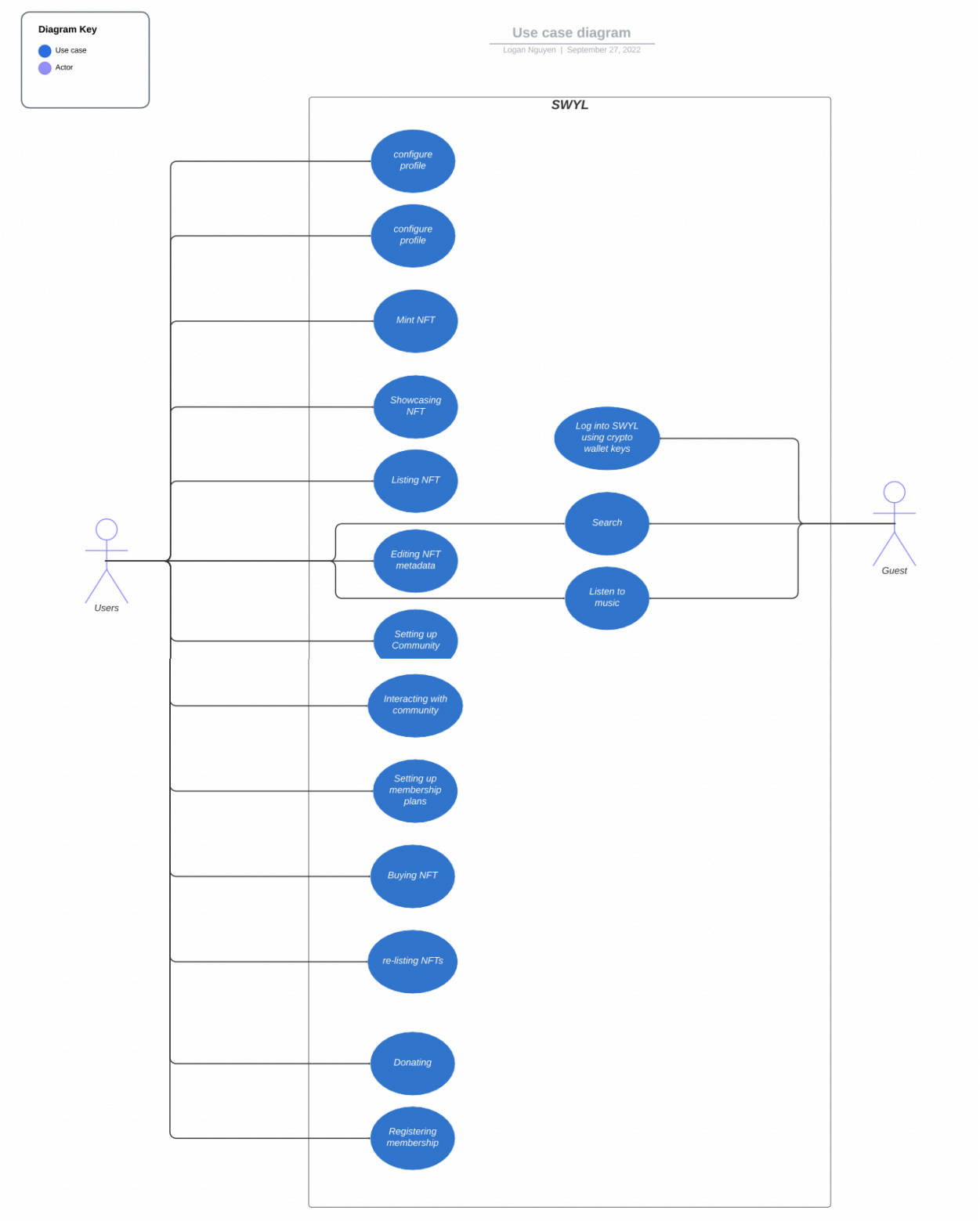

Figure 1.0: Use Case Diagram

6.2 Activity Diagrams (AD)

6.2.1 Login AD

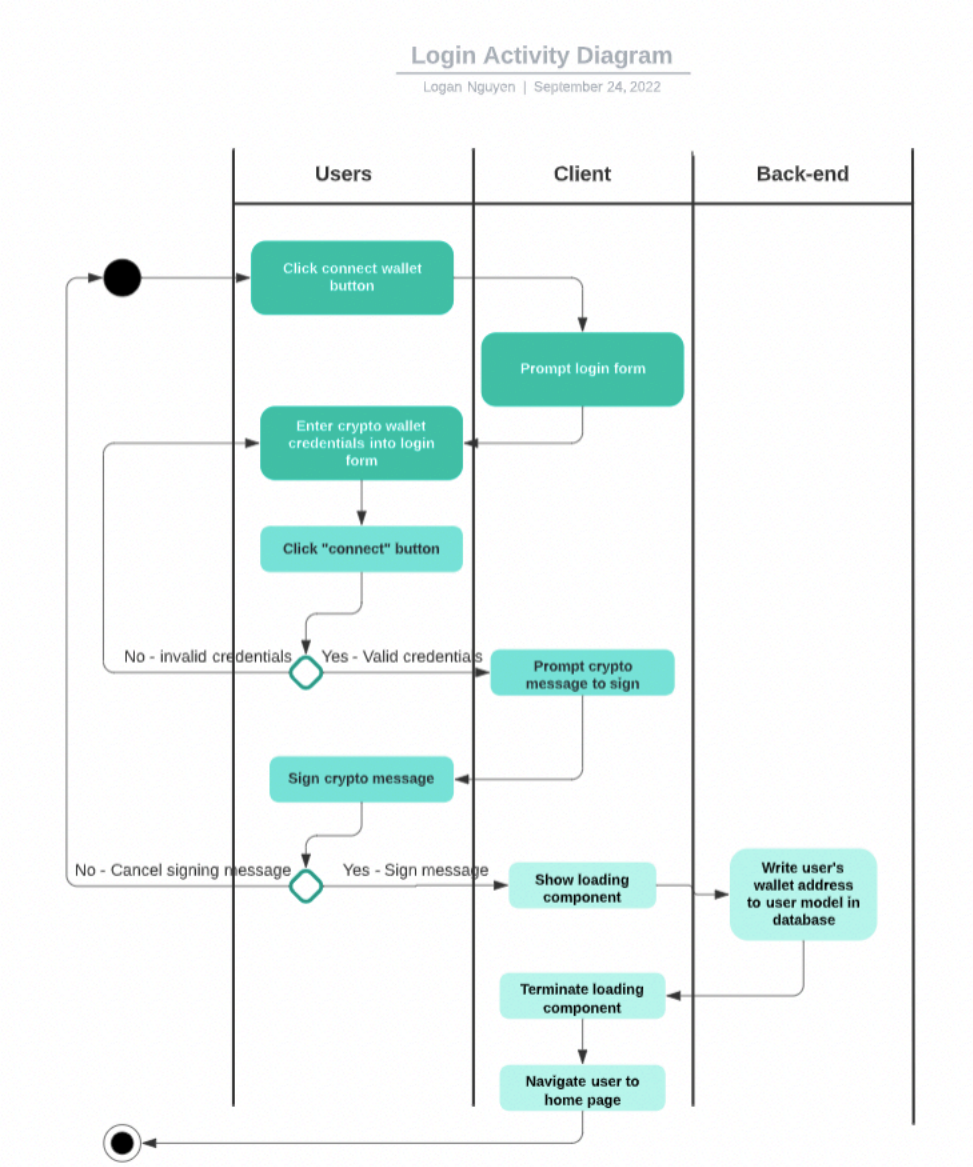

Figure 2.0: Login activity diagram

6.2.2 Minting NFT AD

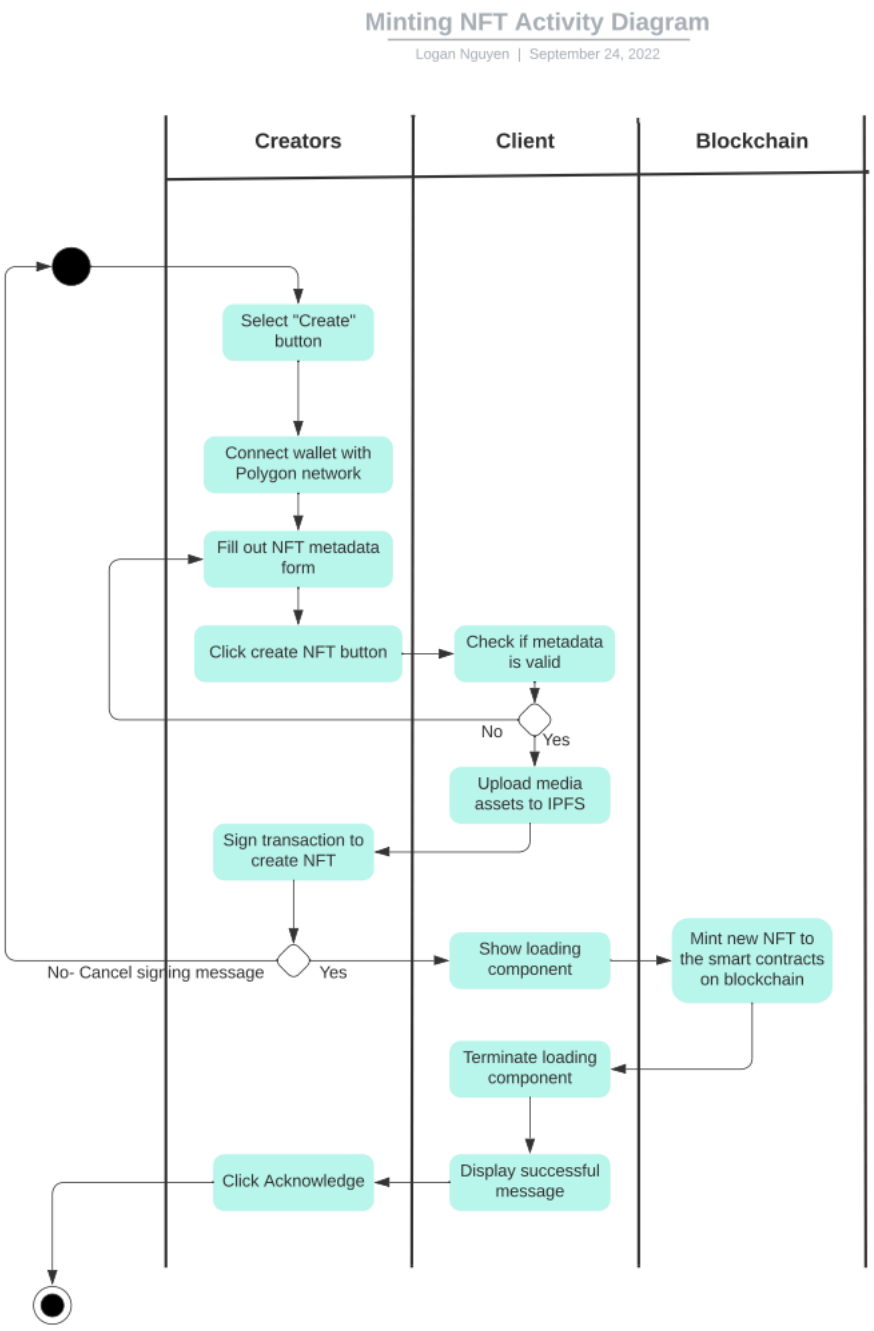

Figure 3.0: Minting NFT activity diagram

a

6.2.3 Listing NFT AD

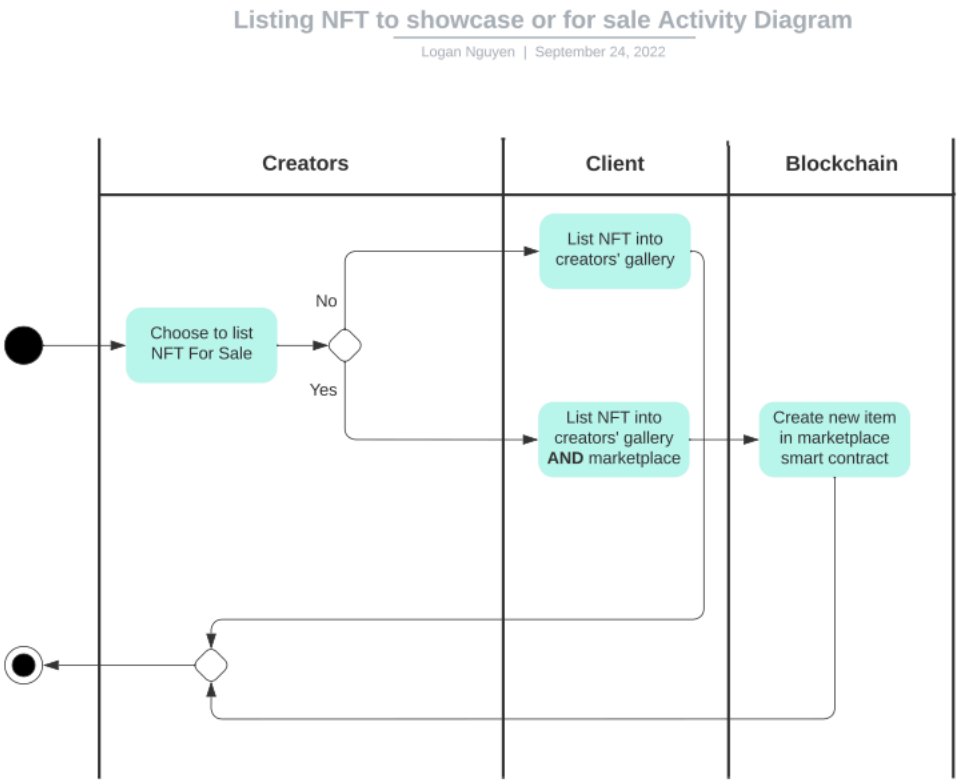

Figure 4.0: Listing NFT activity diagram

6.2.4 NFT Transactions AD

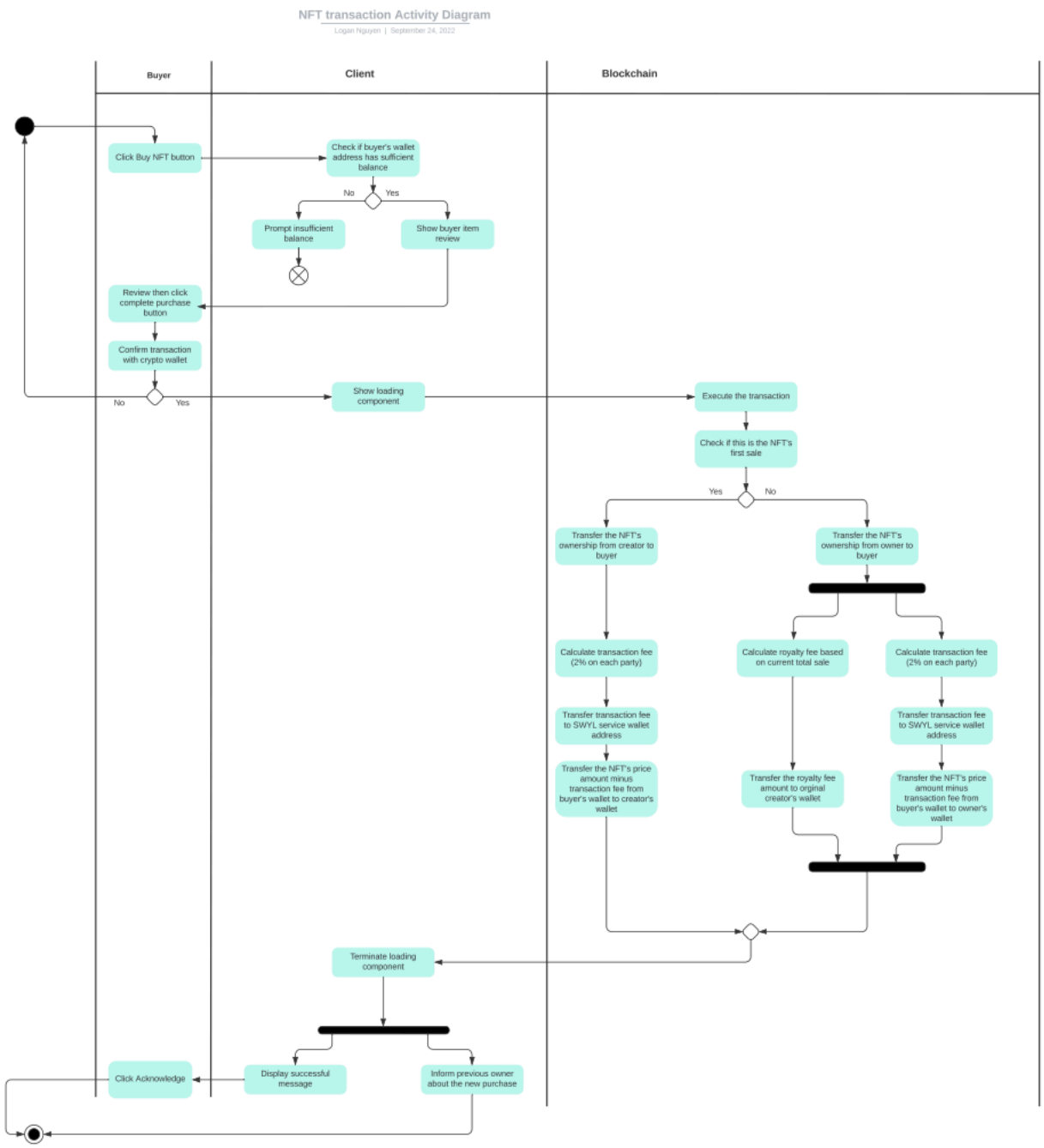

Figure 5.0: NFT transactions activity diagram

6.2.5 Donation AD

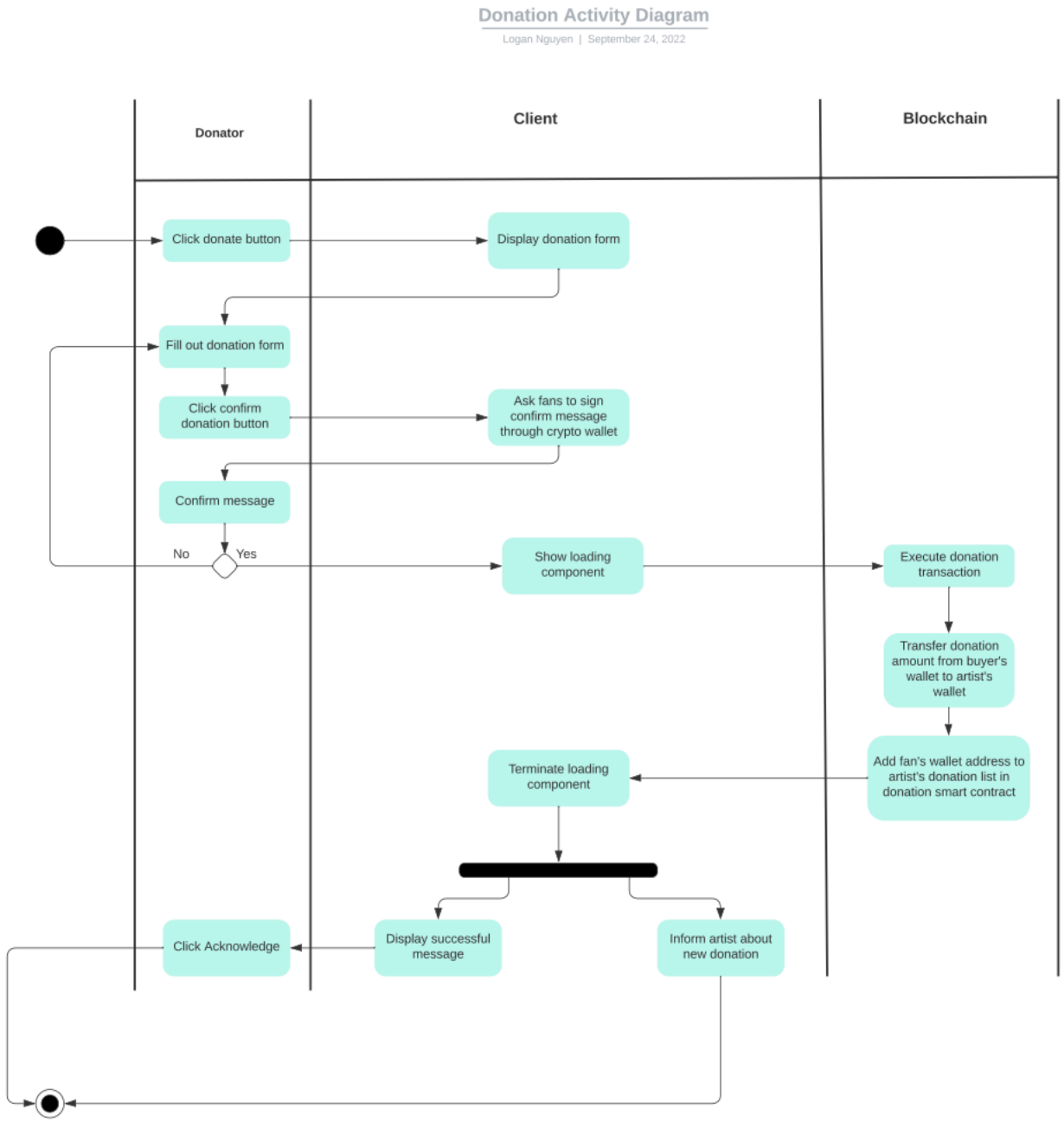

Figure 6.0: Donation activity diagram

6.2.6 Membership AD

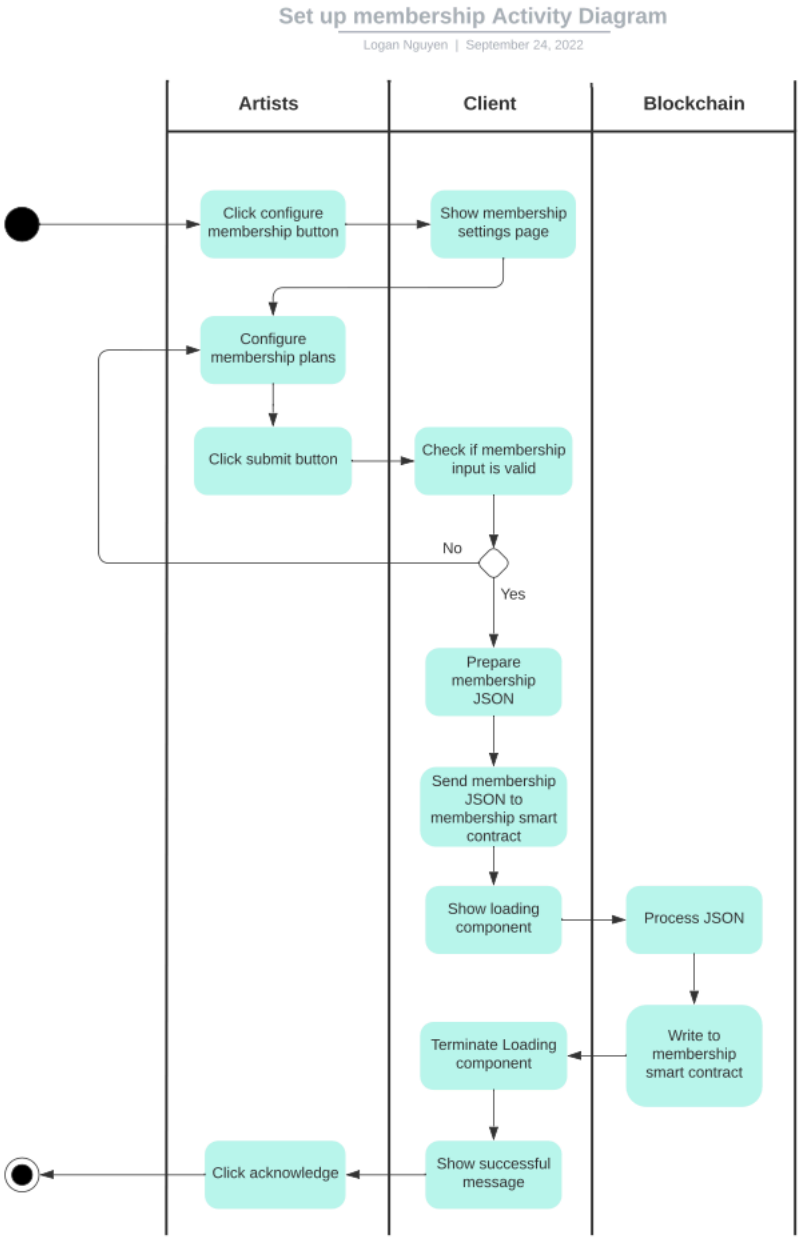

Figure 7.1: Set up Membership activity diagram

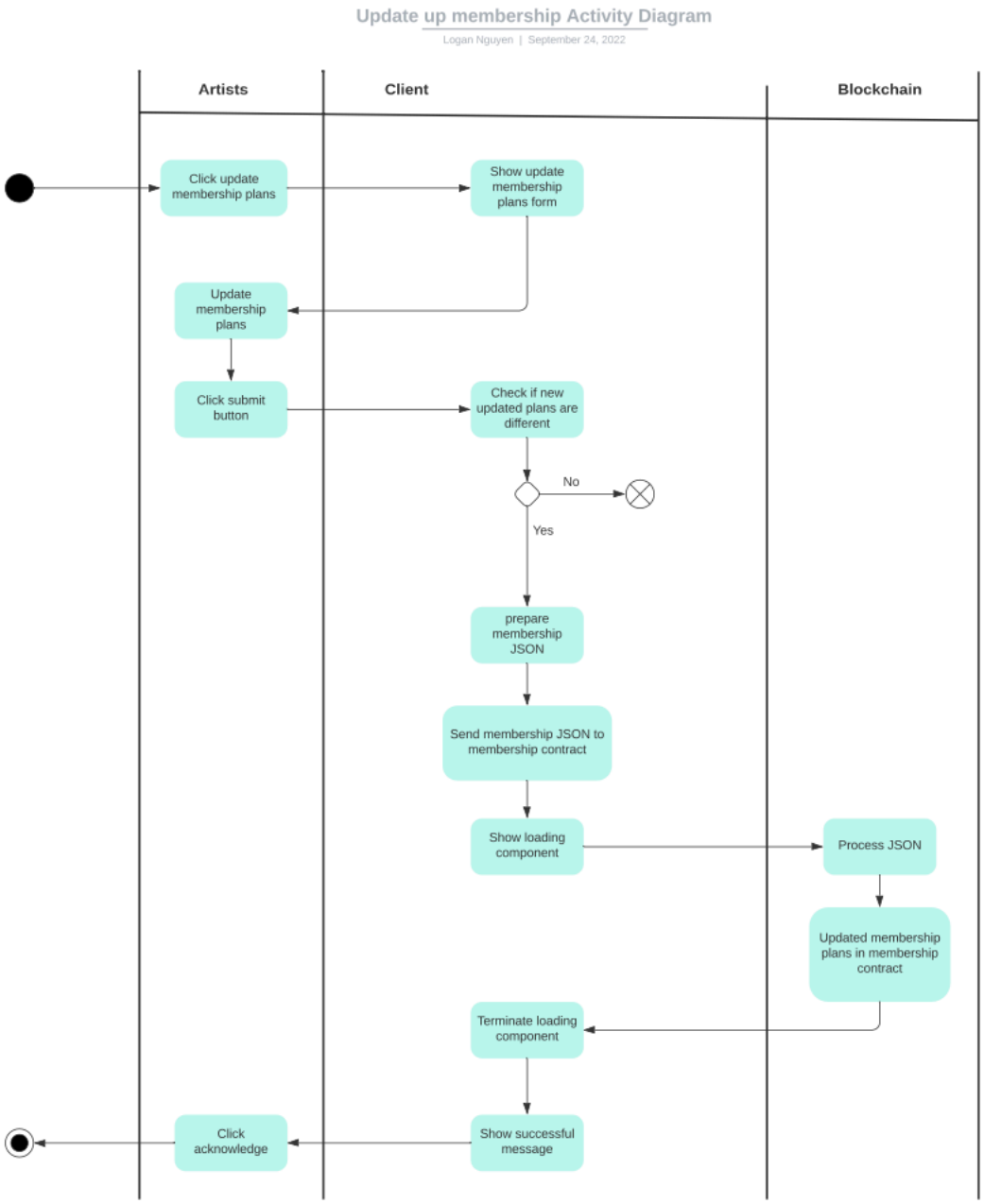

Figure 7.2: Update Membership activity diagram

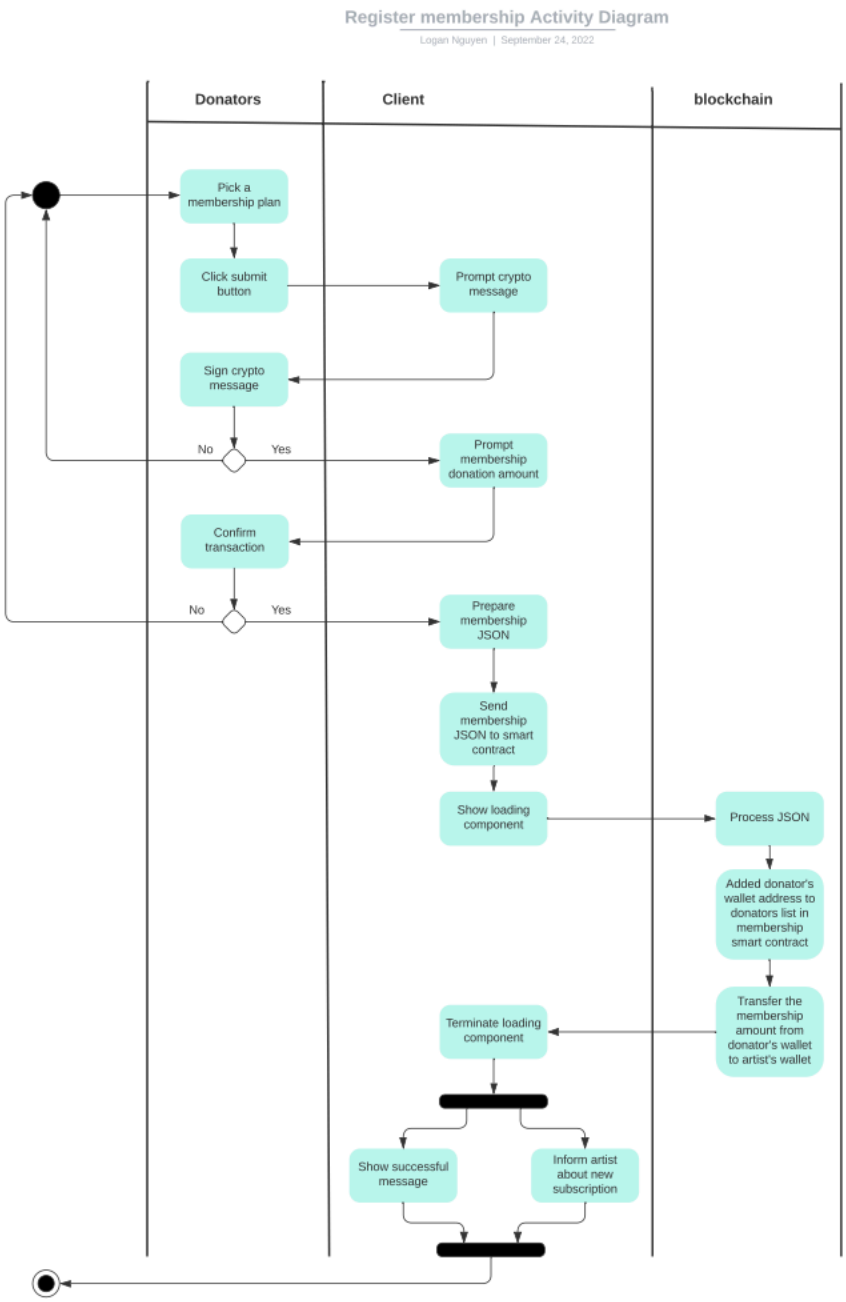

Figure 7.3: Register Membership activity diagram

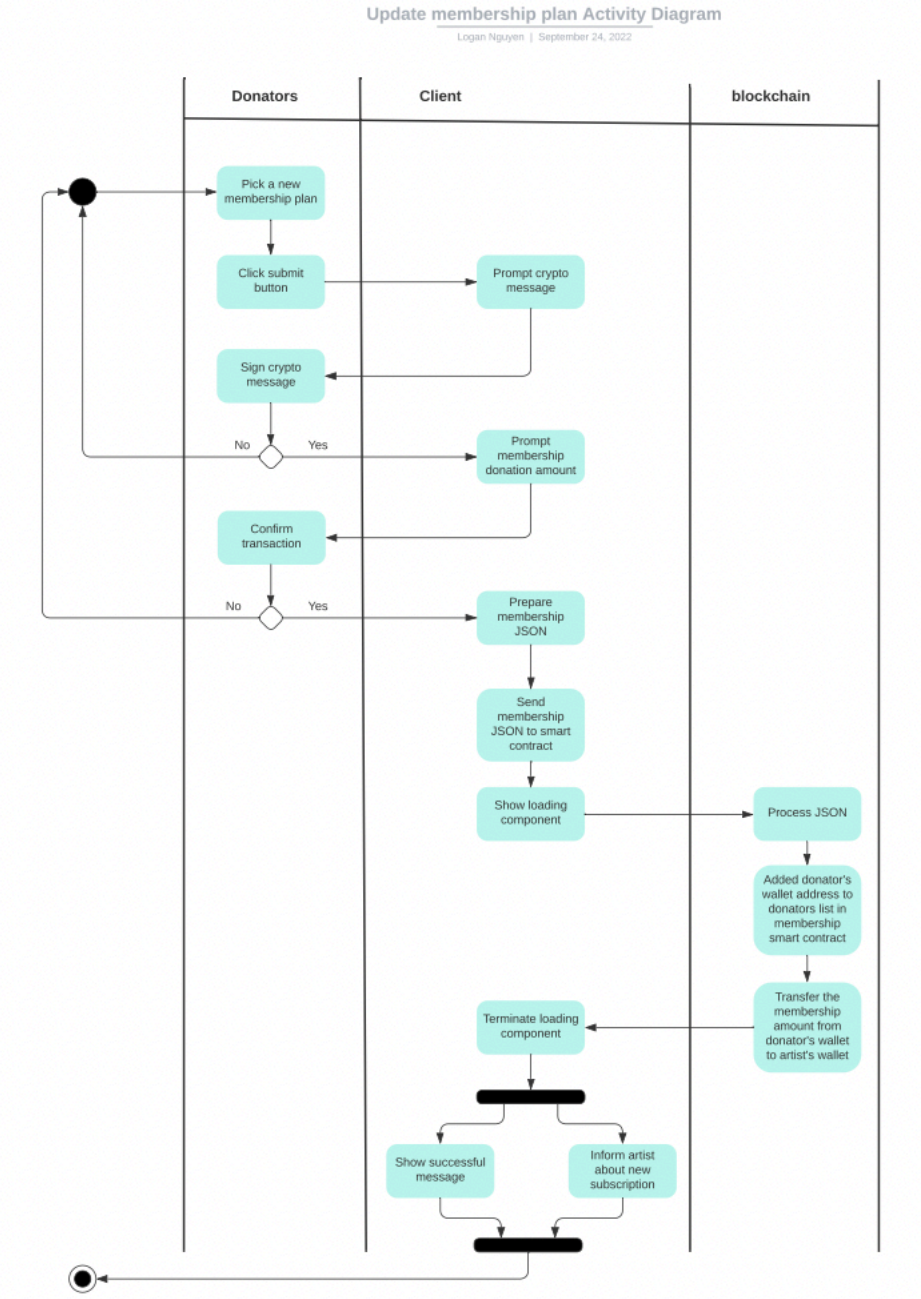

Figure 7.4: Update Membership plan activity diagram

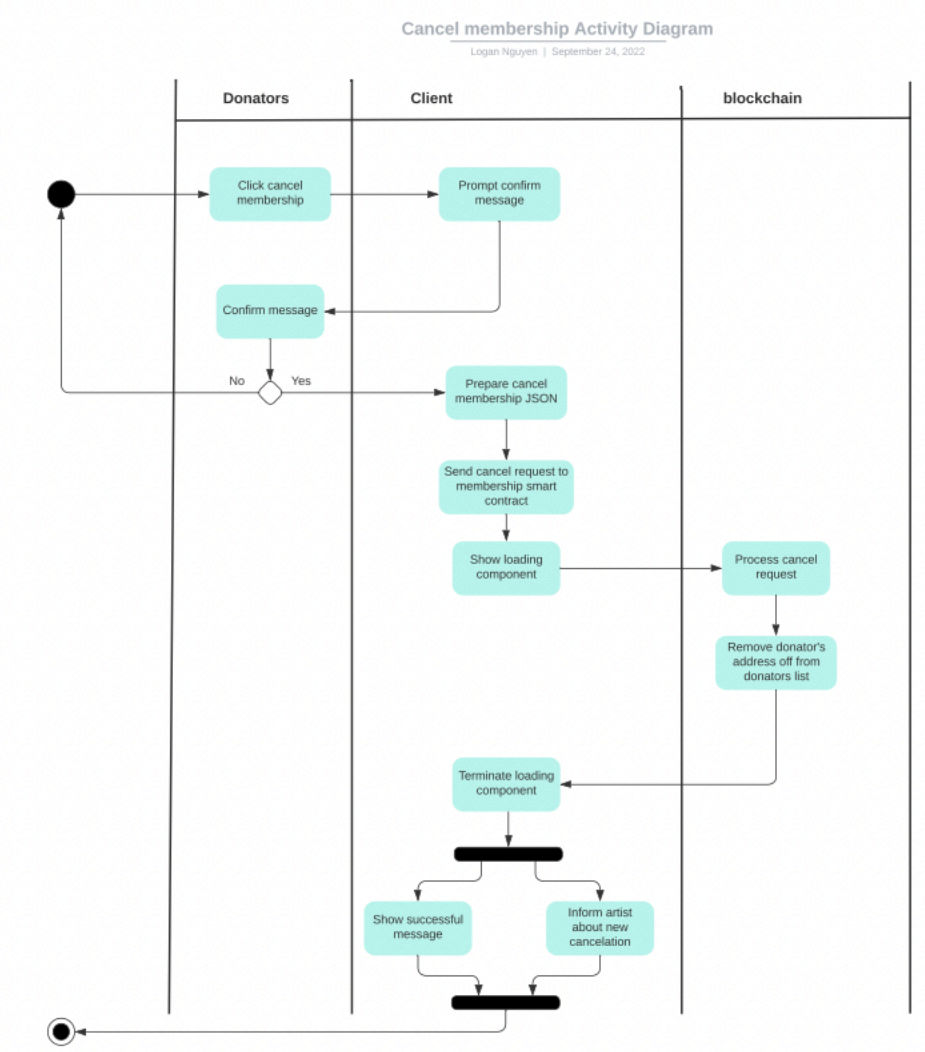

Figure 7.5: Cancel Membership activity diagram

6.2.7 Community AD

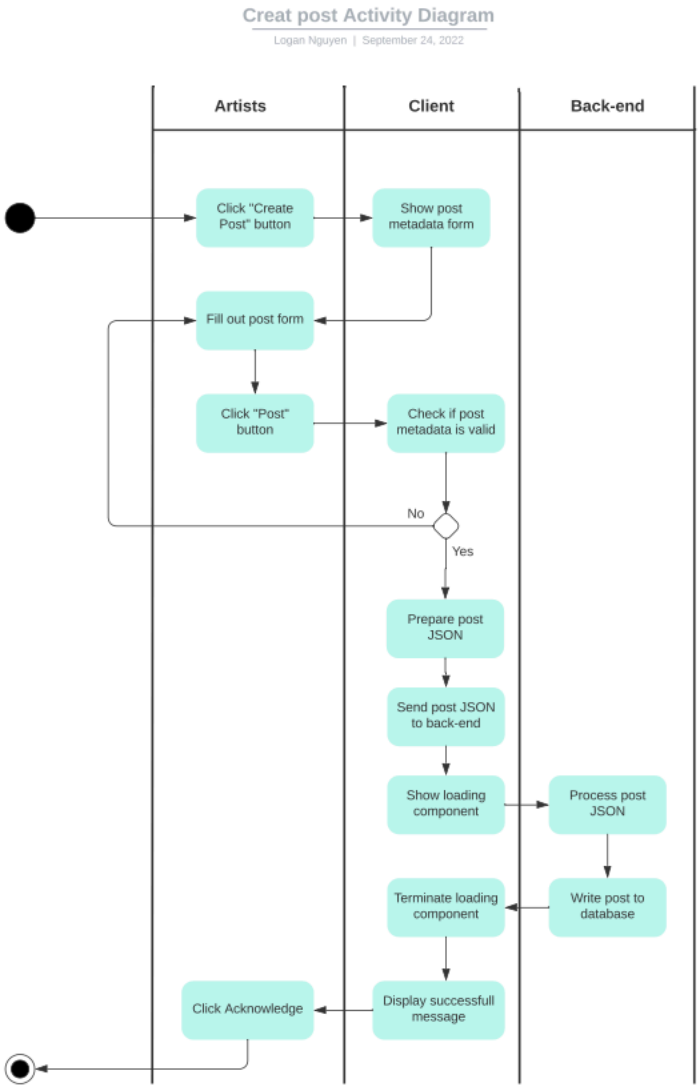

Figure 8.1: Create post activity diagram

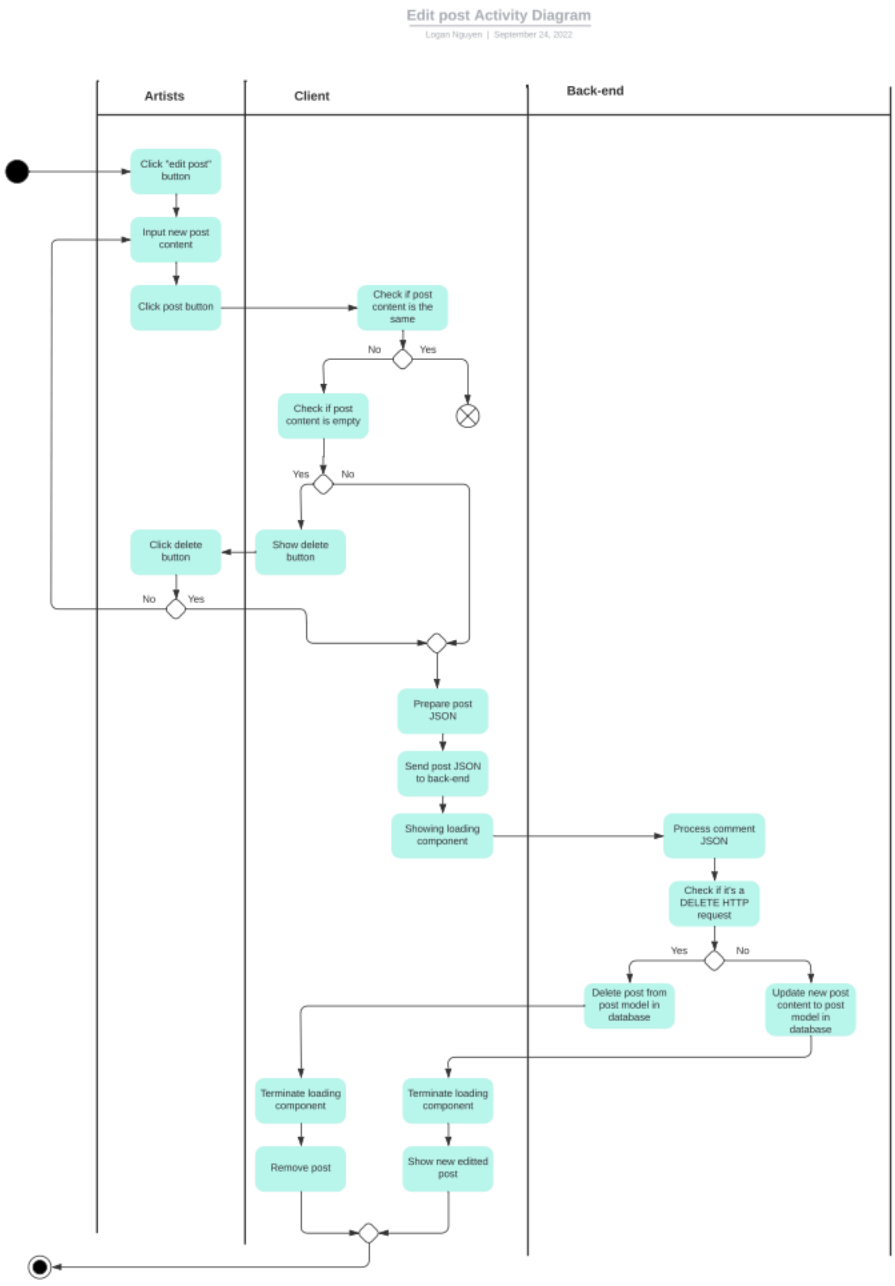

Figure 8.2: Edit post activity diagram

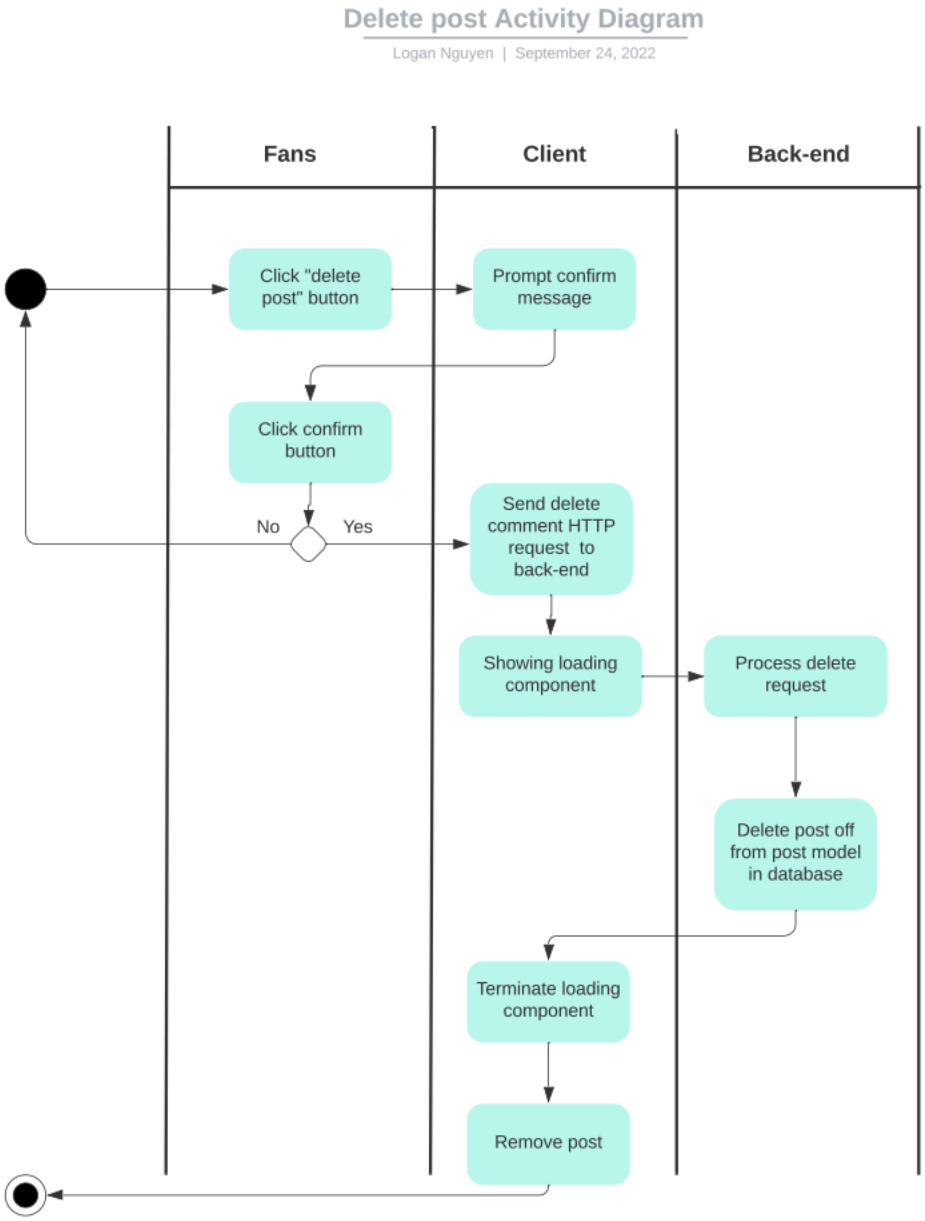

Figure 8.3: Delete post activity diagram

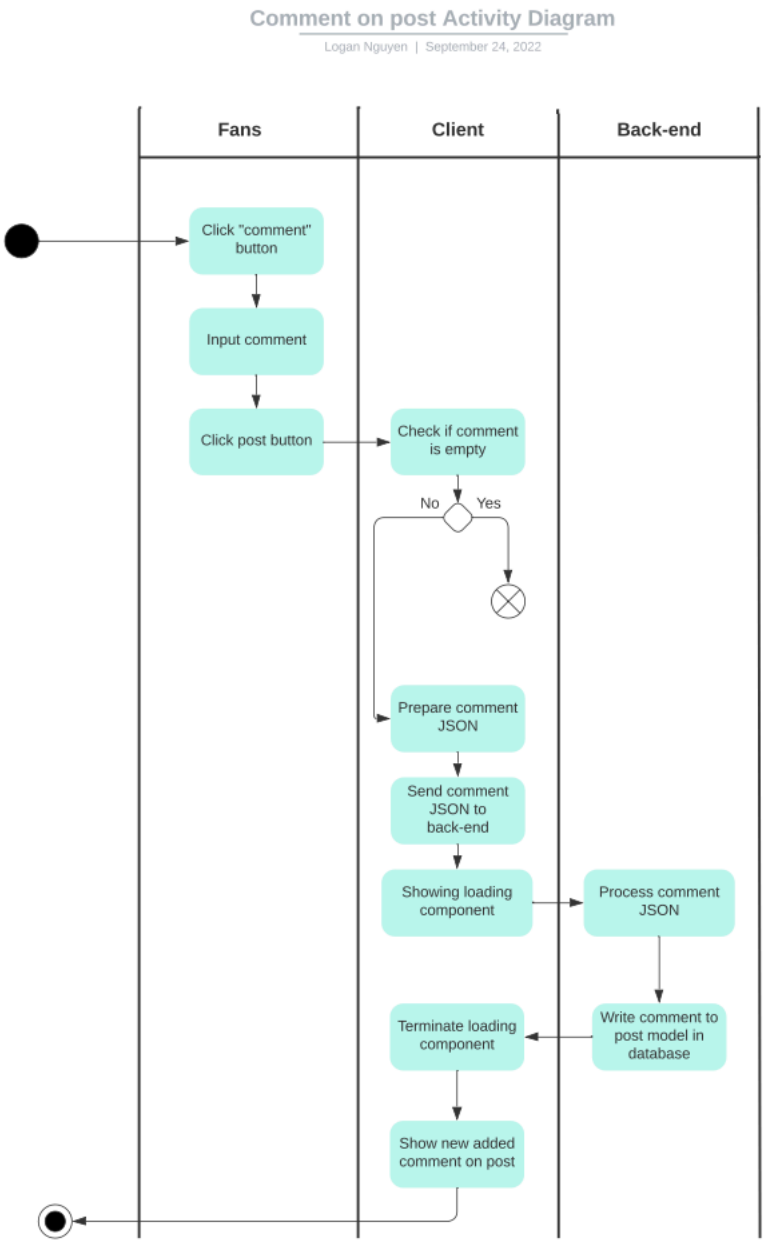

Figure 8.4: Comment on post activity diagram

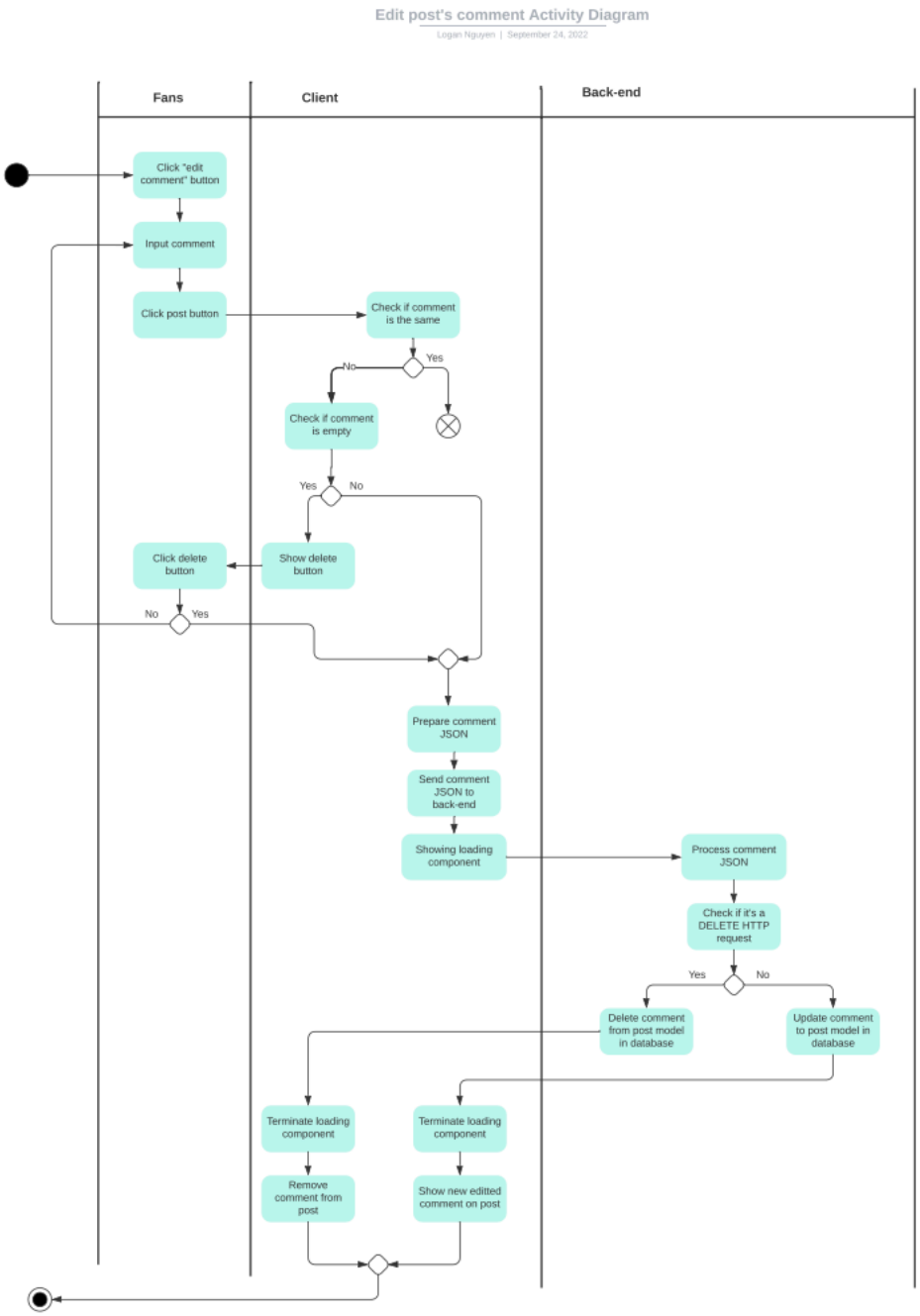

Figure 8.5: Edit post's comment activity diagram

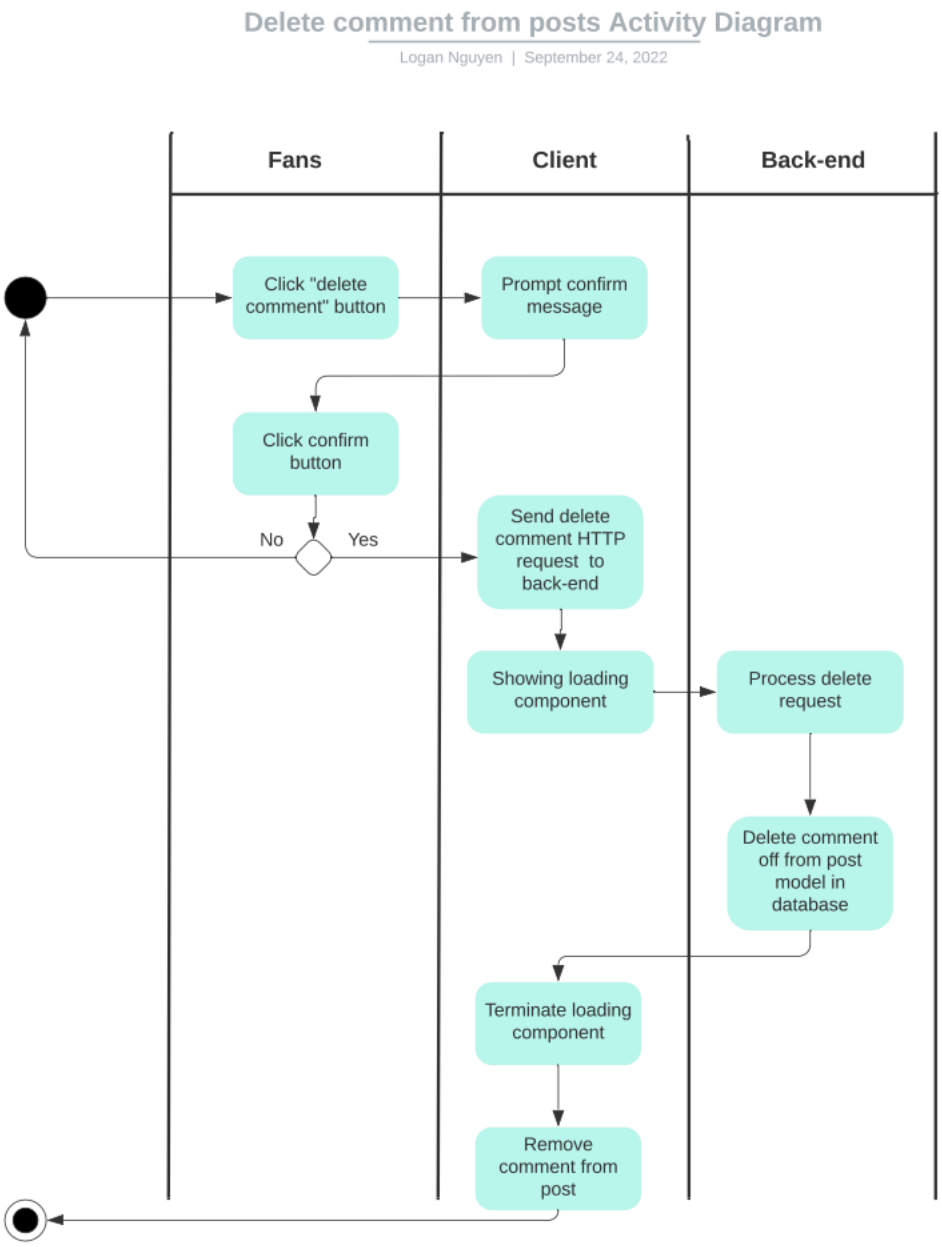

Figure 8.6: Delete comment activity diagram

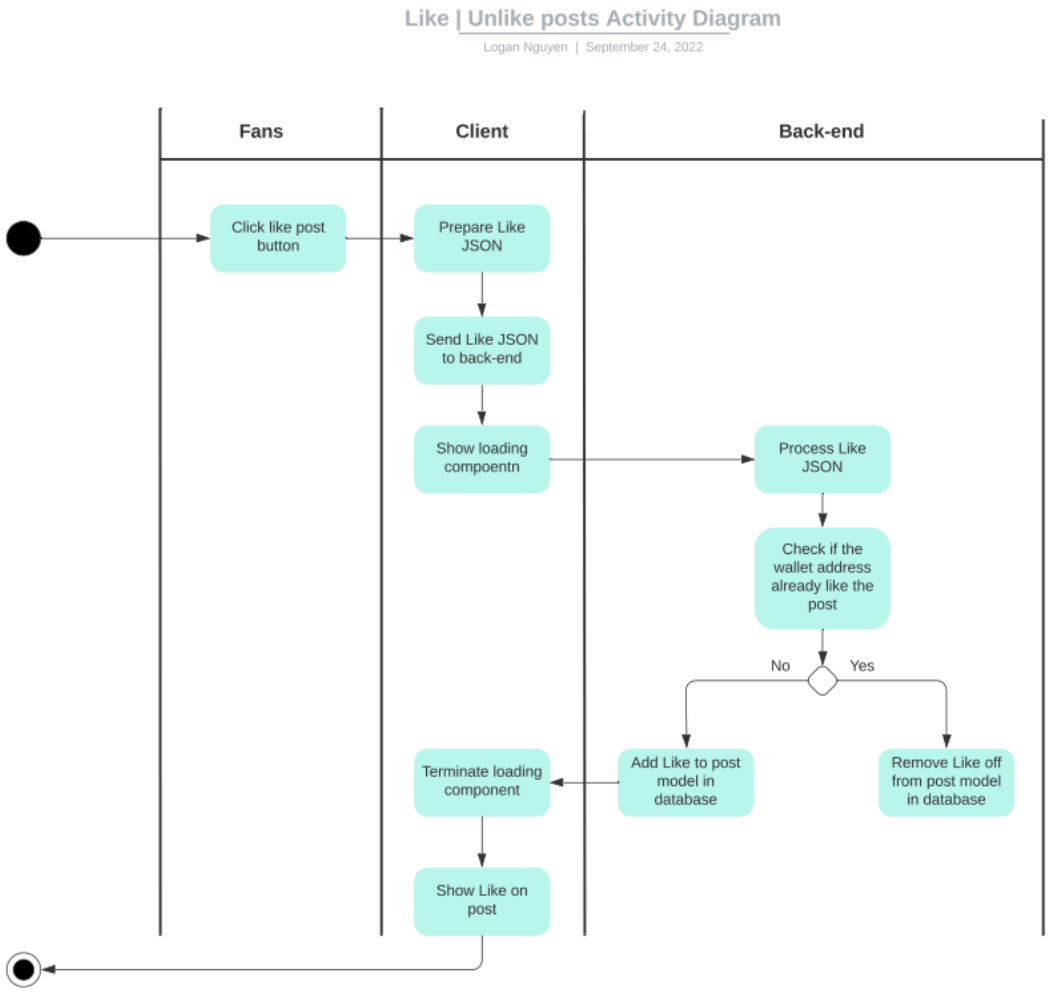

Figure 8.7: Like | Unlike post activity diagram

6.3 System Architecture (SA)

7. Appendix C – Copyright

This document is based on a template meeting the ISO/IEC/IEEE 29148-2018 standard, available at https://www.iso.org/standard/72089.html. Template authors are:

Dr. rer. nat. Bastian Tenbergen,
Associate Professor of Software Engineering
bastian@tenbergen.org

Mikayla Conner-Spagnola, MA

Independent Consultant
mconner@oswego.edu

Department of Computer Science
State University of New York at Oswego
Oswego, NY 13126, United States