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

[**Table of Contents**](#_ldjob8k7zpz5) **2**

[**Change Log**](#_3mmssxfzzqjn) **3**

[**1. Introduction**](#_l3390y66rn29) **3**

[1.1 Purpose](#_raqcr08ft5jf) 4

[1.2 Scope](#_ekdlndfuat4s) 4

[1.3 Product Overview](#_794u8193ei48) 4

[1.3.1 Product Perspective](#_u0b9202i4m2v) 4

[1.3.1.1 System Interfaces](#_dpfjssi26jrz) 4

[1.3.1.2 User Interfaces](#_q8bkg2cfnp1v) 4

[1.3.1.3 Hardware Interfaces](#_1f12c3vel47) 4

[1.3.1.4 Software Interfaces](#_lp0p6wvupnl4) 4

[1.3.1.5 Communication Interfaces](#_gs8183eeqg25) 5

[1.3.1.6 Memory Constraints](#_r0gw2cyx6t3s) 5

[1.3.1.7 Site Adaptation](#_wgv9kza1ogix) 5

[1.3.1.8 Interfaces with Services](#_f6zdob3ylj8g) 5

[1.3.2 Product Functions](#_4loi2xm9utq5) 5

[1.3.3 User characteristics](#_uxist8eut88c) 5

[1.3.4 Limitations](#_z4bc3fijjrhe) 5

[1.4 Definitions](#_5zf4bh61neff) 5

[**2. References**](#_pmg361g9dik2) **6**

[**3. Requirements**](#_k6i0q6aybig9) **6**

[3.1 Functions](#_5qim5rvwe1sh) 7

[3.2 Performance Requirements](#_4msnyoax8kaz) 7

[3.3 Usability Requirements](#_6x7gn3nm5n3w) 7

[3.4 Interface Requirements](#_40d9vtthio42) 8

[3.5 Logical Database Requirements](#_x8ijk12x423z) 8

[3.6 Design Constraints](#_a9808y4mgnt7) 8

[3.7 Software System Attributes](#_rdexucnqatfi) 8

[3.8 Supporting Information](#_tr53z2k70ttg) 9

[4. Verification](#_pmrj9kiuhk16) **9**

[4.1 Functions](#_q4z6zatp14lf) 9

[4.2 Performance Requirements](#_6u0iq5vieij6) 9

[4.3 Usability Requirements](#_k2xow3cnec0g) 9

[4.4 Interface Requirements](#_8vh19cd6nzxs) 9

[4.5 Logical Database Requirements](#_nge1i15i42vf) 9

[4.6 Design Constraints](#_b7z767u7pwm1) 9

[4.7 Software System Attributes](#_sikkj0y2pk7s) 10

[4.8 Supporting Information](#_73sk9wz65fyf) 10

[5. Appendix A – Tailoring Policies](#_6kbclnxz4hh1) **10**

[5.1 Assumptions and dependencies](#_xfu31x3m7pw2) 10

[5.2 Acronyms and Abbreviations](#_pqknufusr9jb) 10

[5.3 Tailoring Policies](#_z0xischrcbp) 11

[6. Appendix B –Copyright](#_erciyessst7q) **11**

[6.1 Author Names](#_64of79n4k05i) 11

[6.2 Creative Commons License](#_r5hin4zawkzb) 11

# Change Log

|  |  |  |
| --- | --- | --- |
| Date | Reason For Changes | Version |
| Sep 18th, 2022 | First SRS initialization | V1.0 |
| Sep 20th, 20222 | * Updated Appendix A * Replace Appendix B Copyright to Appendix C * Added Appendix B Analysis Models | V1.1 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 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.]*

1. **Primary reason**: “Support Who You Love” (referred as **SWYL**) is a web-based music donation/NFT marketplace platform
2. **Primary audience of the SRS**:

* Professor Bastian Tenbergen - Scrum Master **| TBD**
* Nam (Logan) Nguyen –Stakeholder | Project Owner **| TBD**

1. **Primary audiences of the system:**

* Musicians/artists who create music products and sell music NFTs (referred as **CREATORS**)
* 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**)

## **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:
  + transferring the NFTs’ ownerships from one party to another
  + creating and posting music NFTs
  + sending and receiving donations
  + buying and selling music NFTs.
  + hosting a digital 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**

### 1.3.1 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 records the original **authors/creators** of the NFTs
* A feature that allows users to configure their own accounts
* A feature to transfer music NFTs from one crypto wallet to another
* A feature allows creators receive donations and royalty rewards instantly and automatically after every transaction

- **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** mint music NFTs and store them in a blockchain network
* A feature that allows **creators** add royalty fees to their NFTs
* A feature that allows **creators** to post their music NFTs
* A feature that allows **creators** to create digital community
* A feature that allows **creators** to create and manage posts in the community
* A feature that allows **creators** to push their products with restriction
* A feature that allows **creators** to decide if they want to sell the NFTs

- **BUYERS** accessible features

* A feature that allows **buyers** to buy the music NFTs
* A feature that allows **buyers** to re-list the NFTs

- **SUPPORTERS** accessible features

* A feature that allows **supporters** to one-time donate the product
* A feature that allows **supporters** to share the products on social media
* A feature that allows **supporters** interact in the community
* A feature that allows **supporters** 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**: Musicians/artists who create music products and sell music NFTs
* **BUYERS**: Users who want to buy and sell music NFTs
* **SUPPORTERS**: 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 |
| **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:
  + On-chain data will be stored on a blockchain network
  + Off-chain data will be stored on a NoSQL database
* 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

## 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
* **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.]*

## 4.2 Performance Requirements

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

## 4.3 Usability Requirements

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

## 4.4 Interface Requirements

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

## 4.5 Logical Database Requirements

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

## 4.6 Design Constraints

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

## 4.7 Software System Attributes

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

## 4.8 Supporting Information

*[See sections 4.0, 3.8 for specific directions about what outputs should be included here.]*

# 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 | simply 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 minting page | A page available to creators allowing creators to fill in NFT metadata then mint it to the 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 |  |
| 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

# 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](mailto:bastian@tenbergen.org)

**Mikayla Conner-Spagnola**, MA

Independent Consultant

[mconner@oswego.edu](mailto:mconner@oswego.edu)

Department of Computer Science

State University of New York at Oswego

Oswego, NY 13126, United States

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. For more information, please see <http://creativecommons.org/licenses/by-sa/4.0/>