Software Requirement Specification (SRS) for Art Gallery

1. Abstract

This is the requirements document for the case study of 3D Art Gallery.The system to be developed is for portraying and selling the pieces of art in an Art Gallery , based on the values depicted in the art and the overall quality work of the different Artists. Different conditions have to be satisfied while selling or buying through NFT's.This document follows the IEEE standard for a requirements specification document, with some variations.

2. Introduction

2.1 Purpose

This document is meant to delineate the features of the art gallery, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.This document explains the specifications and requirements of the "Art Gallery" a website. It describes the functional features and non-functional requirements of the site such as interface, design and other details related to Art Gallery. This specification document is intended for the entrepreneurs who are interested in opening their Art Gallery and students also can read this documentation for learning purposes.

Blockchain based art galleries would allow artists to flourish in their work. Autonomous works of art built upon a blockchain-based network and with **smart contract development** facilitated the sculpture.

2.2 Scope

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

In Scope:

a. Maintain records of art works made by artists as a portfolio.
b. Type of artwork artists want to display.
c. Necessary information about buyers like their transaction history, artworks owned.
d. Cryptocurrency is owned by the buyer and it should be sufficient to buy the desired artwork.

Out of Scope:

a. Internal working details of blockchain used.
b. Any market related prediction.

2.3 Definitions, Acronyms, and Abbreviations

- Web Application- A type of software that runs on some server. It is usually a piece of software that runs online.
- Apache Software- To Host The Web Locally. It Is Open Source Software.
- Mysqli- Database Management System for Managing Database. It Is Also Open Source Software.
- Xampp- Xampp Is Combination Of Apache, php And Mysqli Database.
- Server- Server Is A Powerful System That Hosts Some Web Applications.
- Open Source- A software that is freely available in the market to download its code for research or modifications.
- Web Browser -Software to surf the web.
- Admin- A user who manages the whole website at the backend. He is the most powerful user of the application. He is responsible for the whole database. He can insert, update and delete data from the database.
- Client- A user who uses the application from the front end. It can send the request to the server.
- Database- A database is the collection of related data that is stored in an efficient and compact manner. Data is stored in the form of tables in the database. Table is a collection of rows and columns.

References

IEEE STD 830-1998, IEEE recommended practice for software requirement specification.

Overview:

The rest of this SRS is organized as follows: Section 2 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed. Section 3 gives specific requirements which the software is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given.
Section 4 gives some possible future extensions of the system. Finally the appendices.

The remaining SRS contains

- The Overall Description
- Gallery Perspective
- Gallery Functions
- User Characteristics
- General Constraints
- Assumption and Dependencies

3. The Overall Description:

3.1 Website Perspective

As mentioned above, this website is totally independent and not a part of some existing system. This website needs a server i.e. web server or a local server, apache in our case. The client sends a request to the server, the application host on some server will communicate to the application and to the database management system if needed to facilitate the request. The database server will respond to the user according to demands of the user.

3.2 Online Art Gallery Functions

This web project is all about an Art gallery, which consists of two views.

1. Admin Panel
2. Client Panel

Where the admin is responsible for the whole database. Admin can perform all the designs, creates and update operations. Admin will be able to

- May add new items in the database using the admin panel
- May update the items in the database
- May delete the items in the database

A client who will use the application from the front end, will be able to visit the website. Clients will be able to select different types of paintings i.e. according to their interest. The client will be able to search different paintings with the name.

Art Gallery should embed the following features and function to its users, i.e. admin and client.

- Should make the admin able to login into the admin panel after authentication.
- Should make the admin able to view the entire available Gallery in the database.
- Should make the admin able to insert new paintings, new profile and new categories.
- Should make the admin able to view list all clients.
- Should make the admin able to make new admin, delete or update the existing user.
- Should make the client able to select more than one Gallery.
- Should provide the security for the system.

3.3 User Characteristics

There are two types of system users. The first is user, user can login and upload his information and also check another profile. The second user is the administrative who have the capability for maintain the record. Admin manage the whole database. Admin is the person who will control the application by entering new events, by updating or deleting new events. I.e. paintings, and categories. The user should know the details of a transaction.

3.4 General Constraints

Web application is based on internet which required both hardware and software, so we will facilitate the requirements about software which are given below. Web application are used the specific software that are include PHP 6.0 ,Apache and data Base scheme used Mysql and hardware requirements are dual core process with at least 1GB RAM for used this software. For better results better hardware will be consider a plus point.

3.5 Assumptions and Dependencies

There are some number of factors that are consider to be fulfill to make the project work according to the requirements described in document.

The mentioned system will use the latest tool available in the market, so it is assumed the user will use the latest web browser for working that is load the UI properly.
There is huge involvement of JavaScript code so it is assumed that the user will not disable the JavaScript in the browser to run the application smoothly.
System date corresponds to Actual date and Time
Users of the system are technically skilled persons and they understand that how to handle this system.

4. Specific Requirements:

Functional Requirements:

We describe the functional requirements by giving various use cases.

4.1 Interface Requirements

4.1.1 User interface

There are two types of user interface.

(1) Client User Interface
(2) Administrator User Interface

4.1.2 Client interface

The minimal requirements are that the user would be able to communicate to the system using GUI. GUI stand for graphical User Interface.GUI is bases on HTML5, CSS3, JavaScript and different frameworks and libraries. a JS library and Bootstrap, a CSS framework. Different GUI based component are used for the following actions.

4.1.3 Administrator Interface

The minimal requirements required for the administrator interface that the admin will access the control panel through a GUI using some web browser. The GUI again bases on unity. No command line utility will be given to the user to access the system.

4.1.3 Hardware Interfaces

Art Gallery is a app that is why all hardware interface of this, will be those of server On top of which it will be running. Hence the app will incorporate the server for all the hardware instances such as CPU, memory and communication.

4.1.4 Software Interfaces

As our mentioned system runs on a polygon based blockchain, so a browser that supports blockchain wallet extensions and transaction of NFTs is required.

4.2 Functional Requirements

This section narrates the features and process performed by the web site. Firstly, the following features of the system and their interaction with administrator are described check paintings, Delete artist Update profiles, Update User, and Delete User. Then the following features related to User are described: Search Paintings, Registration, and feedback, they are followed by a detailed specification of the functionality of the art gallery. Data definition will also be discus in this document .Data flow diagram will be provided in these documents.

4.2.1 Add Artist profile

4.2.1.1 Introduction This functional feature deals with the administrator. The admin can add new painting in the database by using the interface of the Art Gallery.

4.2.1.2 Inputs The admin will enter the Artist name, Artist address, time, date and other details about profile using different UI elements.

4.2.1.3 Processing The server with communicate the polygon marketplace to insert the new item in the database.

4.2.1.4 Outputs A success message will be shown to the user and database will be updated.

4.2.2.5 Error Handling An error message will be shown upon some error while updating the item in the database.

4.2.3 Delete Artist profile

4.2.3.1 Introduction This functional feature also deals with the administrator. The admin can delete Artist in the database by using the interface of the art gallery.

4.2.3.2 Inputs The admin will delete the item in the database with the help of ID of the artist. He would select the id of the artist from a dropdown menu.

4.2.3.3 Processing The request will communicate the distributed nodes to delete the item in the database.

4.2.3.4 Outputs A success message will be shown to the user and database will be updated.

4.2.3.5 Error Handling An error message will be shown upon some error while deleting the item in the blockchain.

4.2.4 Make A New User

4.2.4.1 Introduction This functional feature too deals with the administrator. The admin can make new admin in the blockchain by using the interface of the art gallery.

4.2.4.2 Inputs The admin will enter the name, username, email id and password of the new admin as input.

4.2.4.3 Processing A request will be sent to the polygon marketplace to add new user's details.

4.2.4.4 Outputs A success message will be shown.

4.2.4.5 Error Handling An error message will be shown upon some error while inserting the new user in the database

4.2.5 Update Existing User

4.2.5.1 Introduction This functional feature also deals with the administrator. The admin can update an existing user in the database by using the interface of the Art Gallery.
4.2.5.2 Inputs The admin would edit the user name, user address and other details about the user using different UI.
4.2.5.3 Processing A request will be sent to the polygon marketplace to update new user's details.
4.2.5.4 Outputs A success message will be shown to the user and database will be updated.
4.2.5.5 Error Handling An error message will be shown upon some error while updating the admin in the database.

4.2.6 New User Registration

4.2.6.1 Introduction: This functional feature deals with the user who is using the application's front-end. He would have to register him before any detail.
4.2.6.2 Inputs: The user will enter his complete name, address, his phone number, email address to register him.
4.2.6.3 Processing: A request will be sent to the polygon marketplace to add new user's details.
4.2.6.4 Outputs Database will be updated with a new user and a success message will.
4.2.6.5 Error: Handling An error message will be shown upon some error while searching a specific item the database.

4.2.7 User Login:

4.2.7.1 Introduction: The user may login to artist detail by using the credentials assigned to him upon registration.
4.2.7.2 Inputs: The user will enter his email address and password to login.
4.2.7.3 Processing: The application will communicate to the blockchain database to check if the user has registered already.
4.2.7.4 Outputs: User will be logged in if the entered credentials meets the one in the database and a success message will be shown otherwise he will be asked to enter correct data.
4.2.7.5 Error: Handling An error message will be shown upon some error while logging into the database

4.4 Non-Functional Requirements

4.4.1 Performance

1. Average load time of the app should be less than 5 seconds.
2. Average response time of the system should be less than 10 seconds.
3. Our system should easily accessible on the system having minimum internet speed of 1MB/s

4.4.2 Reliability There will be a maximum of 1 bug/KLOC.

4.4.3 Supportability: All code will be written as specified by the Hungarian Naming Convention.

4.4.4 Availability Seminar Generator will be available to client 7 days a week and 24 hours a day. In case of any system crash a backup will be available to make the system live.

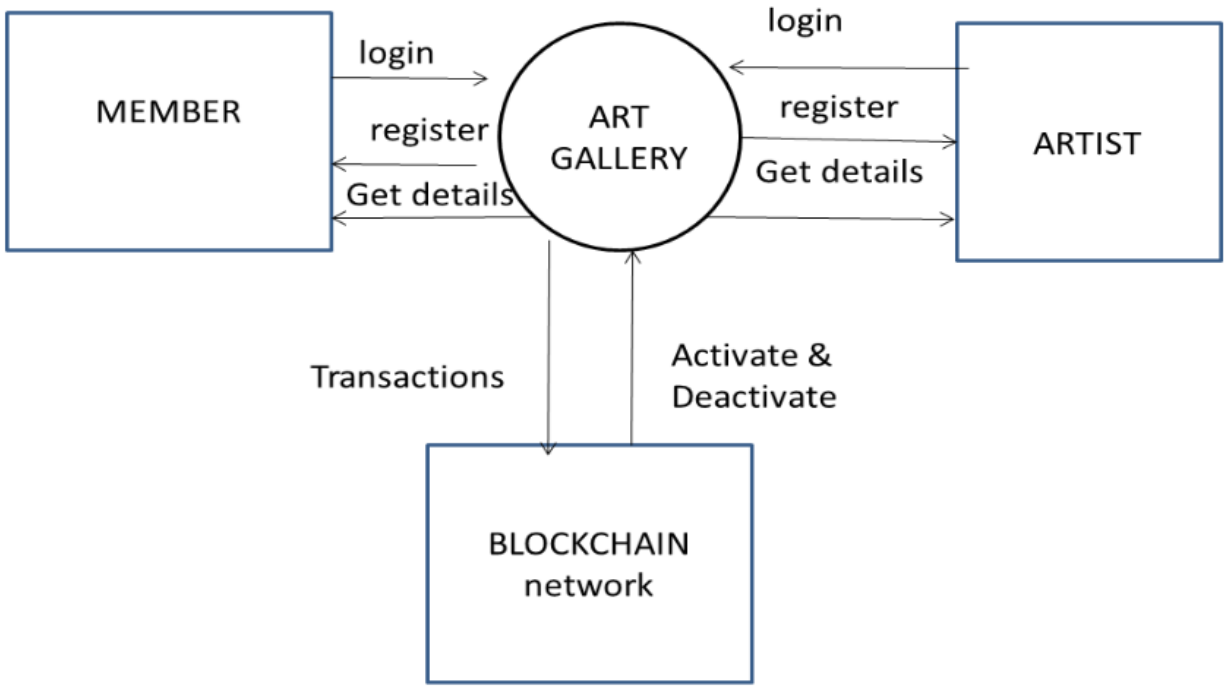
4.4.5 Security Users' information will be secure and he will be able to access only their own personal information

Design Constraints:

4.6.1 Parent component: There is no parent of "Gallery". It works its own.

4.6.2 Application Language Language used for this application is mainly "Solidity" for server side logics and HTML.CSS and Java script for front-end side.

Dataflow diagram:



5. Appendix

Appendix A

- Glossary Actor It represents a role, external entity that interacts with our system
- Use case Graphical Represent of interaction among system and actor.
- Scenario The time of actual input and expected output.
- Use case diagram Represent use the services and functionality by the system by the actor.
- Use case Realization How the use case realizes in the design table model.
- Activity diagram Graphical represent of the process flow of use case etc.
- Sequence diagram Represents the interaction of an object to perform a job.
- Collaboration diagram Alternative representation of sequence diagram.
- Data model Represent the persistence object model