

Software Requirements Specification

for

Vehicle Registration and Ownership Management System

Prepared by Troyrangers

Faculty of Information Technology

University of Moratuwa

24/12/2022

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 2

2. Overall Description 3

2.1 Product Perspective 3

2.2 Product Functions 3

2.3 User Classes and Characteristic 3

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3. External Interface Requirements 5

3.1 User Interfaces 5

3.2 Hardware Interfaces 5

3.3 Software Interfaces 6

3.4 Communications Interfaces 6

4. System Features 7

4.1 SSI Issuing Process 7

4.2 SSI Verifying Process 8

4.3 NFT Minting Process 9

4.4 Mobile SSI Wallet 10

4.5 NFT Market Place 11

5. Other Nonfunctional Requirements 13

5.1 Performance Requirements 13

5.2 Safety Requirements 13

5.3 Security Requirements 13

5.4 Software Quality Attributes 13

5.5 Business Rules 14

6. Other Requirements 15

Appendix A: Glossary 16

Appendix B: Analysis Models 17

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The main purpose of this document is to describe and to demonstrate the functionality of the Vehicle Ownership Management system. It contains a detailed description of all the functional and non-functional requirements of the system. Also, this document shows the purpose and complete declaration of the system and explains system constraints and interactions with system users.

## Document Conventions

IEEE 830-1998 standard for writing SRS documents was used in preparing this SRS document.

## Intended Audience and Reading Suggestions

This document will be helpful to project managers, developers, users and testers of the system to get an idea about the functionalities of this system. We recommend project managers and users to go through the overall description of this document. Developers and testers are recommended to read the functional and non-functional requirements and the external interface requirements of this document.

## Product Scope

Vehicle Registration and Ownership Management System is an application which is used to digitize the registration and ownership management system of vehicles. We implement our system with security and transparency, as security is the most required factor in these kinds of systems. Therefore, it is not much better to use a centralized database, and therefore as the best way, we use a blockchain. The main task of our system is to convert a physical asset into a digital asset. As an example, in our system we convert vehicle asset into a NFT (Non-Fungible Token).

There is a special process that happened in registering the vehicle. When we are registering the vehicle, we can mint (create) NFT with all details of the vehicle. According to that, there can be photographs of the vehicle according to the standards, and also all the information related to the vehicle which are included in the scanned copy of vehicle registration book. Not only that, but also, we create a vehicle marketplace to provide the facility of selling and buying vehicles. In this process, transferring the ownership of the vehicle happens through the NFT. When the lifetime of the vehicle is over, we have to remove the vehicle of use. In that case we can burn-out related NFT. But the information related to that NFT will remain forever with that blockchain. In the current situation, we have to identify a person legally on the internet. But there is not a proper way to show a person’s digital identity. That means a person can’t be legally identified in internet. Because of that we have to issue a digital verifiable ID for them. As we can identify the vehicle owner through this digital verifiable ID, we can use this ID to login the digital marketplace.

## References

IEEE 830-1998 standard for writing SRS document.

# Overall Description

## Product Perspective

This system is newly introduced software. But it is a replacement for the current NIC issuing system and existing vehicle ownership registration system. The blockchain based vehicle registration and ownership management system is a web-based application which contains a mobile application that can be used to register a vehicle as a NFT and, record all the details of the vehicle and it also has a market environment that can buy and sell those registered vehicles and legally transfer ownership within the system. And this application contains the ability of register a person and verify this person by giving an SSI.

## Product Functions

1. User shall be able to request a SSI
2. User shall be able to request a NFT for his/ her vehicle.
3. User shall be able to store their SSI.
4. User shall be able to buy a vehicle through the NFT
5. User shall be able to sell a vehicle through the NFT

## User Classes and Characteristic

Users of this application are General user, SSI issuing officer, Seller, Buyer, Vehicle owner, and NFT minting officer.

General users give a hard copy of SSI request form to SSI issuing officer. The SSI issuing officer logs into the system and enters user’s details into the system which is included in the SSI requesting form.

The vehicle owner can sell a vehicle through the marketplace and he or she wants to get a NFT which represents their vehicle. NFT minting officer enters vehicle’s details into the system which is included in the NFT request form. If user is a seller and he or she wants to open a request to sell his vehicle, then the NFT of this vehicle is marked as sale. If the user is a buyer, then he can see all NFTs which are in the selling state with the price.

## Operating Environment

Since our project is a web application it will follow the standard client server model. The UI of our application is compatible with Desktop, Laptop and Mobile Device screens. Data will be stored on a cloud-based system using Amazon Web Services as the cloud provider.

## Design and Implementation Constraints

• Front end of the application will be designed using ReactJs + NextJS, Html and CSS, React Native

• We will use Django to design the backend.

• Blockchain technology is used to save our data.

## User Documentation

We include online help in the application.

## Assumptions and Dependencies

The SSI issuing application and NFT minting application should be included in an internal network. Public users can't access them. All the users of the system should be connected to the internet for proper functioning of the system.

# External Interface Requirements

## User Interfaces

### 3.1.1 SSI Issuing Application

There is a one GUI for the login of the issuing officer. And also, there is another user interface for the SSI request form.

### 3.1.2 SSI Verifying Application

There is a third-party verification GUI and verification API.

### 3.1.3 Mobile SSI Wallet

There is a mobile application UI. This stores SSI, present SSI. And also, scans the QR, and manages the SSI.

### 3.1.4 NFT minter

There is a one GUI for the login of NFT minting officer. And there is another user interface for the NFT request form.

### 3.1.5 Marketplace

There are GUIs for the login of seller/ buyer, marketplace operations such as listing NFTs, get NFT full details, updating NFT description, buy, sell.

## Hardware Interfaces

In this system we expect to implement as a web-based application and a mobile application system. So, any device can work with any kind of web browser, this system can also access without any interruption.

This system does not need specific internet speed or extra CPU or GPU power, there normal data processing exists. So that a device depends on some particular hardware specification for normal browsing, it will be the minimum hardware requirement for running this application.

## Software Interfaces

### 3..3.1 Indy

Our system communicates with the Indy blockchain using Aries cloud agent python software interface.

### 3.3.2 Polygon

Hardhat, Web3model, ethers are three JavaScript libraries. Communication between the system and the polygon blockchain happens through these libraries.

### 3.3.3 IPFS

“ipfs- http-client” is a java script library. All the communications of the system are happened through this library.

## Communications Interfaces

* In this system, the API is needed to communicate between Django and Aries Cloud Agent python (ACA-Py).
* We implement an Email service in the Communication interface.
* Application and backend communication will happen on HTTPS web requests.
* There are two web servers for SSI issuer and SSI verifier.
* We make an SSL certificate chain, and there is a self-sign certificate`` in between the server and cloud flare.

# System Features

This illustrates organizing the functional requirements for the product by system features, the major services provided by the product. We can divide the functional requirements into the following categories.

## SSI Issuing Process

### 4.1.1 Description and Priority

For a person he or she must have a unique identity which is digitally verifiable to log in to the blockchain based vehicle registration management system. Therefore, each person has to generate his own SSI. That part is done by using this SSI Issuer system. This part has the highest priority. Without the SSI for a particular person, he or she can’t log in to the system. And the penalty of not creating this SSI Issuer module is that we can’t mint a NFT without a SSI and this SSI generating part will be done by using this SSI Issuer module. Cost can be represented according to three ways as time, money, and labor. To develop this module, one developer has to work for nearly 5 months. As the monetary cost of this module, we have to spend for servers to host the private blockchain. The risk to develop this SSI Issuer part is that this uses blockchain technology and it is quite new and an evolving technology. And therefore, it is hard to find resources to develop this particular module.

### 4.1.2 Stimulus/Response Sequences

First the SSI issuing officer submits the already provided username and the password then he or she can log in to the system. After login he or she directs into a SSI generating details form. Then that form will be filled out by the SSI issuing officer and provided as input for the system. As a result of that form submission, the SSI Issuer module will generate a SSI. Then a verifiable email will be sent to the relevant email address which is provided in the SSI requesting form. In that email, there is a link to verify the email provided. After verifying the email, a QR code and a link will be displayed. After scanning the QR code using the SSI Mobile Wallet or clicking the link the general user can make a connection through the system with the SSI Mobile Wallet. According to the generated connection with the SSI Mobile Wallet, this SSI Issuer module will send the SSI which is generated related to the submitted data of that general user.

### 4.1.3 Functional Requirements

REQ-1: The system should display a login page for the SSI issuing officer

REQ-2: The system should allow entering a SSI requesting form.

REQ-3: The system should generate a SSI related to the submitted data.

REQ-4: The system should allow making a connection with the SSI Mobile wallet.

REQ-5: The system should send a verifiable email to the general user.

## SSI Verifying Process

### Description and Priority

The SSI Verifier offers two unique approaches for verifying your identity. The first and the most important verifying process related to our project is that this SSI verifier module will be used in the login to the NFT Market Place. The other verifying approach is that the third-party person (such as police officers) can use this SSI Verifier to check whether the real identity of a general person. This module has a higher priority. Because without this module we can’t filter the ones who are eligible to enter the NFT Market Place. And other than that, this module is beneficial to find the real identity of a person in day-to-day life normal activities. If we do not develop this module the penalty is that we won’t be able to filter the users for the NFT Market Place. Cost can be represented according to three ways as time, money, and labor. To develop this module, one developer has to work for nearly 5 months. When taking the monetary cost for this module, the SSI Verifier module will be built on top of Hyperledger Indy, and for this, we have to use a private blockchain, therefore the monetary cost is only for the servers.

### 4.2.2 Stimulus/Response Sequences

There are two distinct stimulus interactions that happen with SSI Verifier module. When taking the first interaction process, that SSI verifier has a QR code and that is read using the SSI Mobile Wallet. Then the SSI will be sent to the SSI Verifier. And as a result of that, if the SSI is valid then the user will be directed to the NFT Market Place Home page and if it is not then the user will be provided with an error message.

### 4.2.3 Functional Requirements

REQ-6: The system must verify the Self-Sovereign Identity (SSI).

## NFT Minting Process

### 4.3.1 Description and Priority

When a vehicle owner wants to sell a vehicle through the NFT marketplace, he or she wants to generate an NFT which represents their vehicle. The NFT is minted in the polygon blockchain with the help of the NFT minting officer (because this person is the one who is responsible to submit the data related to the vehicle). And then the ownership of the NFT is transferred to the owner of the vehicle. Without an NFT, the vehicle owner can’t sell a vehicle within the system.

The benefit of this NFT Minting module is that we can digitize and give a digital identity to a real-world existing vehicle. And that will help to maintain an easily manageable (because we do not have a hard legal background to change the ownership of a vehicle and we can just simply record the transaction on the Polygon blockchain) and secure NFT Vehicle Marketplace. For this NFT Minter, we use Polygon Blockchain which is a public blockchain. Therefore, we have to bear a cost for each transaction in a public blockchain.

### 4.3.2 Stimulus/Response Sequences

NFT Minting Officer submits the data relevant to a vehicle which is provided by a vehicle owner to the NFT Minter module. As a result of that the NFT minter module will generate a NFT related to that provided data. After the creation of the NFT that NFT can be seen in the NFT Marketplace.

### 4.3.3 Functional Requirements

REQ-7: The system must provide a facility for the NFT minting officer to log in to the system.

REQ-8: The system must display a form to enter the details of vehicle.

REQ-9: The system must create the NFT based on the provided data.

REQ-10: The system must change the ownership of the NFT when selling and buying vehicles.

## Mobile SSI Wallet

**4.4.1** **Description and Priority**

The general user can buy or sell a vehicle within the NFT Marketplace and to involve that process that particular person must have a SSI to log in to that Marketplace. For that, the user should scan a QR code from his or her mobile SSI wallet and after scanning that QR code, the mobile SSI wallet establishes a connection between the SSI Verifier and that SSI Mobile Wallet. After that connection was established that SSI will be sent to the SSI verifier module. This is the benefit of this module. If this module won’t developed, then there is not a way to store the SSI which is used to identify a general user. For this Mobile SSI Wallet, we use Hyperledger Indy which is a private blockchain. Therefore, we have to bear a cost only for servers.

### 4.4.2 Stimulus/Response Sequences

After entering the password then the general user can login to the Mobile SSI Wallet. Then the home page of the wallet will be displayed. To make a connection with a SSI issuer first have to scan the QR Code issued by the SSI Issuer. After the connection is established SSI issuer will send the relevant SSI to this mobile wallet and that sent credential will be saved.

### 4.4.3 Functional Requirements

REQ-11: The system must provide a facility to scan the QR code.

REQ-12: The system must store and display the credentials.

REQ-13: The system must store and display the connections.

## NFT Market Place

### 4.5.1 Description and Priority

In the NFT marketplace, the user can sell or buy vehicles. If the user is a seller, he or she wants to open a request to sell his vehicle, then the NFT of this vehicle is marked as sale. If the user is a buyer, he can see all NFTs which are in the selling state with the price. The benefit of the NFT Market Place is that general users can sell and buy vehicles in a secure manner. If we do not create this part of the project the penalty is that we won’t be able to sell or buy vehicles and there is no use of what we did up this by generating NFTs of vehicles and SSIs related to general users.

### 4.5.2 Stimulus/Response Sequences

Upon logging in with their SSI credentials, the general user is able to view a selection of vehicles available for sale on the home page of NFT Vehicle Marketplace. In the home page there can be seen each vehicle as a card with an image of that vehicle and few details. After clicking on any card, the general user is directed to a page which can be seen all the details (past owners, price, vehicle technical details, vehicle number, etc.) related to that vehicle. There is a separate tab to view the NFTs which belong to that user. There is a separate button to move to the selling pages. And the seller can fill in a few data fields and make a particular NFT available for selling. And if any buyer needs to buy a particular vehicle, that person can click on the vehicle card that he wants to buy. And after clicking there is an option to buy that vehicle and after using that option the buyer can do the transaction using cryptocurrency. And then the transaction happens with the aid of the meta mask wallet and the ownership of the NFT is transferred to the new owner.

### 4.5.3 Functional Requirements

REQ-14: The system must provide the chance to log in to the marketplace.

REQ-15: The system must provide a facility for all the system users to view a selection of vehicles that are available for sale.

REQ-16: The system must provide a facility to view all the details related to the vehicle which are up to sale.

REQ-17: The system must provide facilities to do the transaction of vehicle using cryptocurrency.

REQ-18: The system must provide the chance to change the ownership of a NFT from a seller to buyer.

REQ-19: The system must provide the chance to view the NFTs belonging to them self through the system.

# Other Nonfunctional Requirements

## Performance Requirements

The response time would be an important feature when measuring performance. The response time should be low enough so that the user won’t be inconvenienced when using the application. Another important feature would be the user-friendliness of the application. The user should not find it difficult or unpleasant to navigate the application.

## Safety Requirements

It’s important that any data that we store, especially that of the users, is kept safe from data loss for any reason. The data that is stored in the database should be backed up regularly to ensure that there isn’t any loss of data. The backups should be able to restore the data in case of any loss of data. It’s also important to maintain the server so that it’s kept running and to ensure that it can handle increased data capacity as the number of users increases.

## Security Requirements

The system should have secure authentication and authorization for all users. Each user should log in to the system using usernames & passwords provided by the Organization. In addition, it should have secure communication channels for all data transmission and should have data encryption for all sensitive information.

## Software Quality Attributes

There are several software qualities that would be vital in order to put out quality software. Some of these attributes are,

• Correctness

• Reliability

• Adequacy

• Learnability

• Robustness

• Maintainability

• Readability

• Extensibility

• Testability

• Efficiency

If these qualities are present in software, it can be considered to be of high quality.

## Business Rules

The product can be used by both social influencers and businesses. The user roles for both these users would be almost the same on all occasions. The businesses would have the additional capability of adding projects and events. The administrator is in charge of adding users and ensuring that all reported comments and users are looked into.

# Other Requirements

When we figured out both international and legal requirements we came across with these requirements. According to that, the SSI which we create for users must have the ability to be used in any situation which we need to verify the identity of that user. And also, the NFT which we mint for a particular vehicle must be accepted in any NFT marketplace. And other than that, there is an international requirement which we need to fulfill. That is our SSI, NFT, and the process going on in the NFT Market Place have to be accepted by any country. And there is an initialization requirement also. That is, the government has to allow permission to deploy our system.

Appendix A: Glossary

**List of Abbreviations**

|  |  |
| --- | --- |
| **Abbreviation** | **Meaning** |
| DeFi | Decentralized Financing |
| EIP | Ethereum Improvement Proposals |
| ERC | Ethereum Request for Comment |
| ETH | Ethereum |
| IPFS | Inter Planetary File System |
| NFT | Non-Fungible Token |
| RTA | Road Transport Authority |
| MVT | Model View Template |
| DLT | Distributed Ledger Technology |
| ACA-Py | Aries Cloud Agent Python |
| VOMS | Vehicle Registration and Ownership Management System |
| SRS | Software Requirement Specification |
| QR | Quick Response |
| HTTP | Hypertext Transfer Protocol |
| POC | Proof of Concept |

Appendix B: Analysis Models

Diagram

Description automatically generated

Figure ‑ Input Output Diagram

Class Diagram



Figure ‑ Class Diagram

ER Diagram

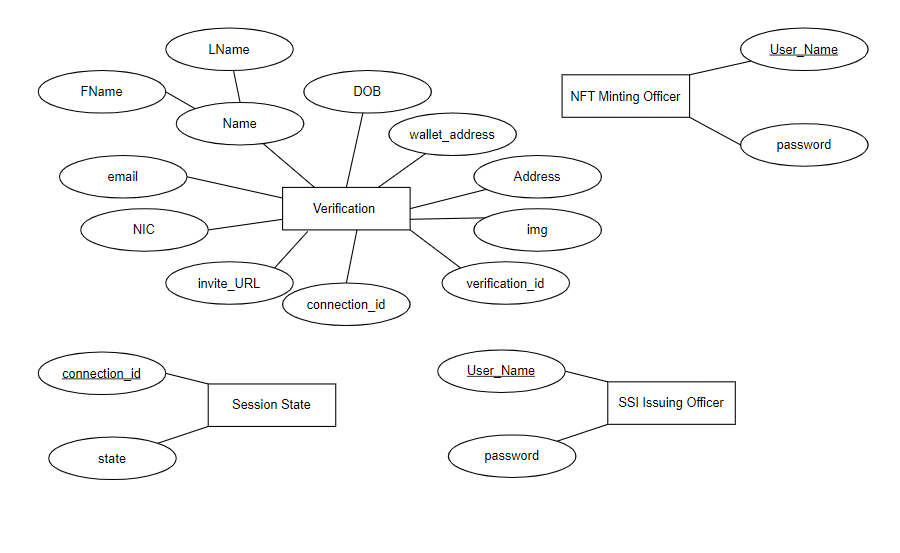


Figure ‑ ER Diagram