Software Requirements Specification

for

Soterra: An Individual Land Titling for Farmers using Blockchain Technology

Prepared by
Famitanico, Erhika Mae
Jacolbia, Patrick
Lector, Adrian Job
Robrigado, Mark Vincent
Saule, Elizabeth

BSCS 3 - 3

2021

Revision History

Name	Date	Reason For Changes	Version
First Revision	05/11/2021	Minor revisions	Version 1.1

Table of Contents

1. Introduction

1.1 Purpose

This document specifies requirements for a blockchain-based Land Titling system. The format of this document is based on IEEE standard 830-1998 and conforms to that standard where possible. The introduction of a blockchain-based Land Titling system will allow for a faster transaction of land registration, automation of the land registration, transparency of transactions with the help of a distributed ledger, avoid physical archives, prevent situations where records can be lost, misplaced, stolen, or manipulated, and provide greater security for the land titles of our farmers.

1.2 Scope

The project, Blockchain Individual Land Titling (BLIT), is a web based DApp (Decentralized Application) aimed to address the shortcomings of the current Land Titling system. The system requests the users to log in to use the system. After logging in, the user can add property, generate deeds, accept or reject deed requests, add users, view title details, and search titles.

The system restricts the users form accessing all the functionalities mentioned above. Access of features depends on the user's role in the system. Roles are not assigned by the system rather user with a specific role can add users to another role.

1.3 Definitions, Acronyms, and Abbreviations

1. ICLOA Individual Certificates of Land Ownership Award
2. LRA Land Registry Authority
3. DARPO Department of Agrarian Provincial Office
4. ARB Agrarian Reform Beneficiaries
5. DApp Decentralized Application

1.4 References

1. IEEE Standards Description: 830-1998
2. Blockchain Database System Concepts by Mastering Ethereum

1.5 Overview

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the system. General description and constraints and assumptions of the system is discussed in section 2 of this document. Section 3 gives the functional requirements, database requirements made while designing the system. It also gives the user viewpoint of product. Section 3 also gives the specific requirements of the product. Section 3 also discusses the external interface requirements and gives detailed description of functional requirements. Section 4 is for supporting information.

2. Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

The web based DApp runs in latest version of web browser such as Google Chrome, Firefox, Safari, Microsoft Edge on Microsoft Windows, MacOS, and Linux OS.

2.1.2 User Interfaces

Login Interface

The Login interface allows the user to connect to the system by clicking the “Connect to Blockchain” button wherein a MetaMask popup will appear and will ask them to sign the Login request.

Search Interface

The search interface will consist of search bar wherein the user can input the Property ID a submit button to run a query. The search result will be displayed at the bottom if the query string matched a record in the blockchain.

LRA Interface

This interface will consist of a series of tabs (could also be buttons) representing the few functions this interface controls. There should be a “dashboard” tab/button, a “deed request” tab/button, a “add DARPO” tab/button, a “search” tab/button, and a “logout” tab/button. The Dashboard tab/button should allow the user to view the height of the blockchain, the latest transaction, and the block history. The Add DARPO tab/button should allow the user to add another user to DARPO role by providing their Ethereum wallet address. The Deed Request tab/button should allow the user to view the generated deed request of the DARPO role user and register or reject the Deed Request. The Search tab/button will allow the user to access the Search interface. The Logout tab/button will allow the user to logout of the system.

DARPO Interface

This interface will consist of a series of tabs (could also be buttons) representing the few functions this interface controls. There should be a “available property” tab/button, a “add field validator” tab/button, a “search” tab/button, and a “logout” tab/button. The Available Property tab/button should allow the user to view all available properties added by Field Validator user role. The user may generate deed on the selected property by clicking the “generated deed” button and inputting the necessary information. The Add Field Validator tab/button should allow the user to add another user to Field Validator role by providing their Ethereum wallet address. The Search tab/button will allow the user to access the Search interface. The Logout tab/button will allow the user to logout of the system.

Field Validator Interface

This interface will consist of a series of tabs (could also be buttons) representing the few functions this interface controls. There should be a “properties” tab/button, a “add property” tab/button, a “search” tab/button, and a “logout” tab/button. The Properties tab/button should allow the user to view all available properties he/she added. The user can make the property available by clicking the “make available” button on the selected property. Add Property tab/button should allow the user to add another property by inputting all the necessary information and clicking the “add property” button. The Search tab/button will allow the user to access the Search interface. The Logout tab/button will allow the user to logout of the system.

2.1.3 Hardware and Software Interfaces

A monitor with Super VGA display standard with a resolution of 1024 x 768 is the minimum requirement to properly view the contents of the web page. Higher resolution monitor is highly recommended. The end user will need a pointing and a typing device to interact with the user interface for the proper functioning of the application.

The application server runs on HTTP server that handles server pages. The application uses blockchain database to keep track of the land titles, which connects automatically using the Ganache Truffle Suite. Windows, Linux, and Mac are all supported operating system.

2.1.4 Communications Interfaces

- Client on Internet will be using HTTP/HTTPS protocol
- Firewall security is required for securing the server
- TCP/IP protocol is basic need for client side
- Communication interfaces include HTTP server hosted on Ganache Truffle Suite.
- Ganache is communicating with MetaMask.
- MetaMask is communicating with Ethereum Blockchain.

2.1.5 Memory Constraints

The application is a web based DApp that requires a minimum of 2GB of RAM for proper execution.

2.1.6 Operational Context

- Minimum of 2GB RAM
- At least 1.9GHz x86- or x64-bit dual core processor with SSE2 instruction set
- Internet Connectivity
- Minimum web browser version is requirement to support Bootstrap 5.0:
  Google Chrome (version 60)
  Firefox (version 60)
  Safari (version 12)
  Microsoft Edge

2.2 Product Functions

Login

This function allows the user to enter the system. The user is required to have an Ethereum Wallet Account through MetaMask. After authentication user will have access to main menu. Availability of menu functions depends on user's role within the system.

Logout

This function can be done by all users. It terminates the user session.

Add User

This function allows user with specific role to add user to another specific role. LRA role can add user to DARPO role. DARPO role can add user to Field Validator role.

Add Property

This function allows users with Field Validator role to add property listing.

Generate Deed

This function allows users with DARPO role to generate individual deed from available properties added by the Field Validator role.

Register Deed

This function allows users with LRA role to register or reject induvial deed generated by users with DARPO role.

Search Deed

This function can be done by all users. The user is required to input the Property number and click the "Submit" button to query the system. If there is a match the results and some of the details regarding the deed will be displayed at the screen.

View Deed

This function can be done by all users. To view deed information the user should perform a search. If there is a match the result some of the details regarding the deed will be displayed at the screen. The user can click the "View Details" button to view all the information available regarding the deed.

2.3 User Characteristics

LRA

LRA user acts as the administrator of the system and should have the proper knowledge about the details required to accept and reject a deed request. Users with this role can see the height of the blockchain and all the transactions within the system. This user can also add users to DARPO role by providing their ETH address.

DARPO

DARPO users are responsible for generating deeds from available properties added by users with Field Validator role. They can add users to field validator role by providing their ETH address.

Field Validator

Field Validator user should have proper knowledge about the actual field validation on the ground to properly input the necessary details needed to add property in the Properties listing.

2.4 General Development Constraints

The application has been designed for public use and therefore required to keep the design, functional, and implementation constraints as few as possible however a few of them could be:

- The organization that is responsible to maintain the project or system should be familiar with blockchain and blockchain programming.
- The app will be developed using Solidity programming language, further updates and support for the language is a major constraint.
- Budget constraints.
- End user's knowledge about computers.

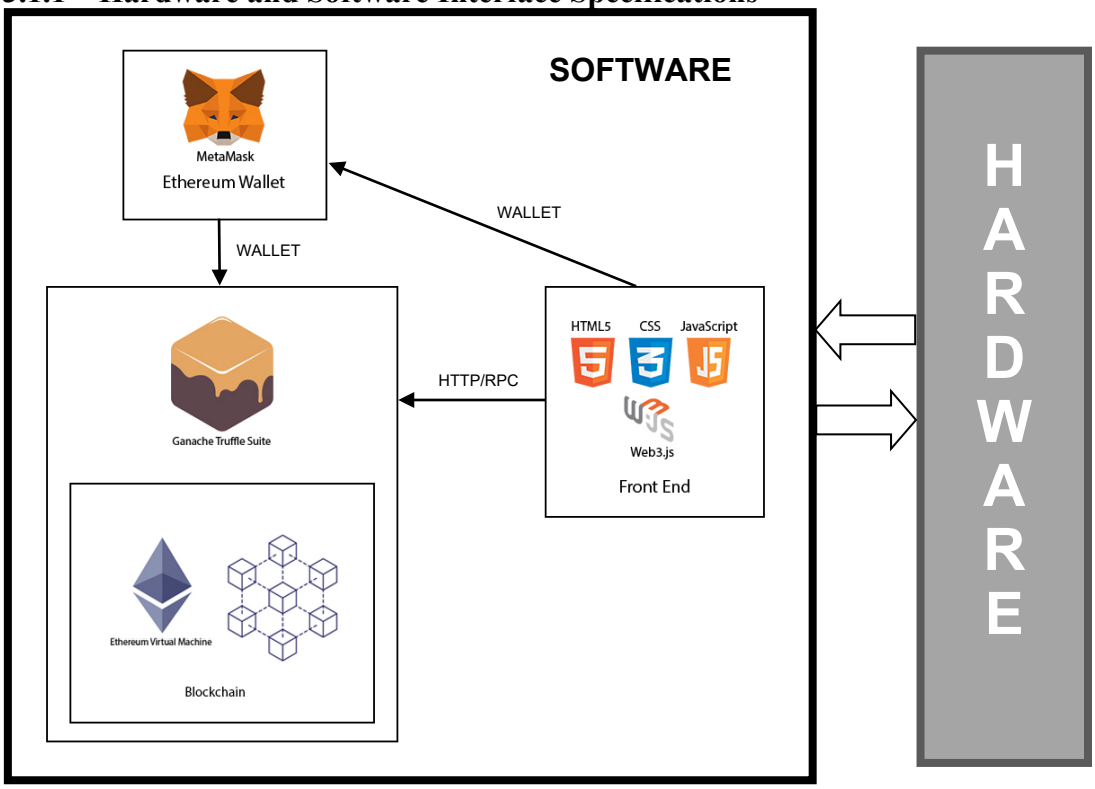
2.5 Assumptions and Dependencies

- The end user should have some basic knowledge about computers.
- The end user already has an Ethereum Wallet Account through MetaMask.
- Roles of each user class is predefined.
- Administrator account is already created in the system.

3. Specific Requirements

3.1 External Interfaces

3.1.1 Hardware and Software Interface Specifications



Hardware specification requirements is written in **2.1.3. Hardware and Software Interfaces** and **2.1.6. Operational Context**.

3.1.2 Detailed Description of Inputs and Outputs

Case #	Use Case	Input
1	Login	none
2	Logout	none
3	Add DARPO	Text and numeric data (keyboard): Ethereum Address, Region, Province
4	Add Field Validator	Text and numeric data (keyboard): Ethereum Address, Region, Province, Municipality
5	Search Deed	Text and numeric data (keyboard): Property ID
6	View Deed	None
7	Add Property	Text and numeric data (keyboard): OCT/TCT number, Lot number, Survey Number, Total Area, Location, Location description
8	Generate Deed	Text and numeric data (keyboard): ARB, ARB spouse, Address, Cadastral number, LRC, number, RD branch, date/time
9	Register Deed	none

Case #	Use Case	Output
1	Login	Display homepage (depending on role) on screen
2	Logout	Display Login page on screen
3	Add DARPO	On screen message confirming DARPO added
4	Add Field Validator	On screen message confirming Field Validator added
5	Search Deed	Display search result on screen
6	View Deed	Display deed information on screen
7	Add Property	On screen message confirming Property added
8	Generate Deed	On screen message confirming Deed generated
9	Register Deed	On screen message confirming Deed registered

3.2 Functional Requirements

3.2.1 Use Case Diagram

3.2.2 Use Case Report

3.2.2.1 Login

Actor(s): All users.
Precondition(s):
1. Unauthenticated session.
2. User is logged into his/her Ethereum wallet via MetaMask.

Trigger: Clicking on “Connect to Blockchain” button in the home page.
Procedure:
1. The user clicks on the “Connect to Blockchain” button in the home page.
2. MetaMask popup dialog will appear asking the user to sign the login process.
3. The user clicks on “Sign” button on the popup menu.
    3.1. The system opens the authenticated session if user information was true.
    3.2. The system will go back to the home page if the user clicks on “Cancel” button.

Post condition(s):
1. The user is logged in to the system (Authenticated session).

Exception(s):
2a. User is not logged in to MetaMask Ethereum Wallet Account.

3.2.2.2. Logout

Actor(s): All users.
Precondition(s):
1. Authenticated session.

Trigger: The user clicks on “Logout” in any page.
Procedure:
1. The user clicks on the “Logout” button located on any page in the authenticated session.
2. The system terminates the session.
3. The system transfers to home page.

Post condition(s):
1. Unauthenticated session.

3.2.2.3. Add DARPO

Actor(s): Users with LRA role.
Precondition(s):
1. Authenticated session.
2. User to be added should already have an ETH address.

Trigger: The user clicks on “Add DARPO” in any page.
Procedure:
1. The user clicks on the “Add DARPO” button located on any page in the authenticated session.
2. The user input details such as Province, Region, and ETH address of the user to be added.
3. The user clicks “Add” button.
    3.1 The system adds DARPO role functionality to the added user.
    3.2 The system displays an error message if ETH address of user does not exist.

Post condition(s):

1. Added user can now access DARPO role functions.

Exception(s):

3a. ETH address not found

3a1. Systems displays error message and prompts user to try again.

3.2.2.4. Add Field Validator

Actor(s): Users with DARPO role.

Precondition(s):

1. Authenticated session.
2. User to be added should already have an ETH address.

Trigger: The user clicks on “Add Field Validator” in any page.

Procedure:

1. The user clicks on the “Add Field Validator” button located on any page in the authenticated session.
2. The user input details such as Province, Region, Municipality and ETH address of the user to be added.
3. The user clicks “Add” button.
   3.1 The system adds Field Validator role functionality to the added user.
   3.2 The system displays an error message if ETH address of user does not exist.

Post condition(s):

1. Added user can now access Field Validator role functions.

Exception(s):

3a. ETH address not found

3a1. Systems displays error message and prompts user to try again.

3.2.2.5. Search Deed

Actor(s): All users.

Precondition(s):

1. Authenticated session.

Trigger: The user clicks on “Search” in any page.

Procedure:

1. The user clicks on the “Search” button located on any page in the authenticated session.
2. The user types the query on the search bar.
3. The user clicks on the “Submit” button
4. The system looks for matches in the blockchain.

Post condition(s):

1. System displays search results.

Exception(s):

2a. User did not type anything in the search box.

2a1. System displays an error message and asks user to type a query for search.

4a. Record not found.

4a1. System displays “CLOA number not found” message.

3.2.2.6. View Deed

Actor(s): All users.

Precondition(s):

1. Authenticated session.

2. User search query returns a result.

Trigger: The user clicks on “View Details” in the search result.
Procedure:

1. The user clicks on the “View Details” button located beside a search result.
2. The system displays all the information available about the CLOA.

Post condition(s):

1. System displays all the information available about the CLOA.

3.2.2.7. Add Property

Actor(s): Users with Field Validator role.
Precondition(s):

1. Authenticated session.

Trigger: The user clicks on “Add Property” in any page.
Procedure:

1. The user clicks on the “Add Property” button located on any page in the authenticated session.
2. The user fill up all the fields and upload necessary documents.
3. The user clicks “Add Property” button.
    3.1 The system adds the property to the Properties page.
    3.2 The system displays an error message if not all fields have been filled up.

Post condition(s):

1. Property added to Properties page.

Exception(s):

3a. Missing Fields
    3a1. Systems displays error message and prompts user to fill all the necessary details.

3.2.2.8. Generate Deed

Actor(s): Users with DARPO role.
Precondition(s):

1. Authenticated session.
2. Property should be listed in the Available Properties Page.

Trigger: The user clicks on “Available Properties” in any page then select from available properties and click “Generate ICLOA” button.
Procedure:

1. The user clicks on the “Generate ICLOA” button.
2. The user fill up all the fields.
3. The user clicks “Generate” button.
    3.1 The system adds the generated ICLOA deed to the ICLOA Deed Request page of LRA user role.
    3.2 The system displays an error message if not all fields have been filled up.

Post condition(s):

1. ICLOA deed added to ICLOA Deed Request page of LRA user role.

Exception(s):

3a. Missing Fields
    3a1. Systems displays error message and prompts user to fill all the necessary details.

3.2.2.9. Register Deed

Actor(s): Users with LRA role.
Precondition(s):

1. Authenticated session.
2. ICLOA deed should be listed in the ICLOA Deed Request Page.

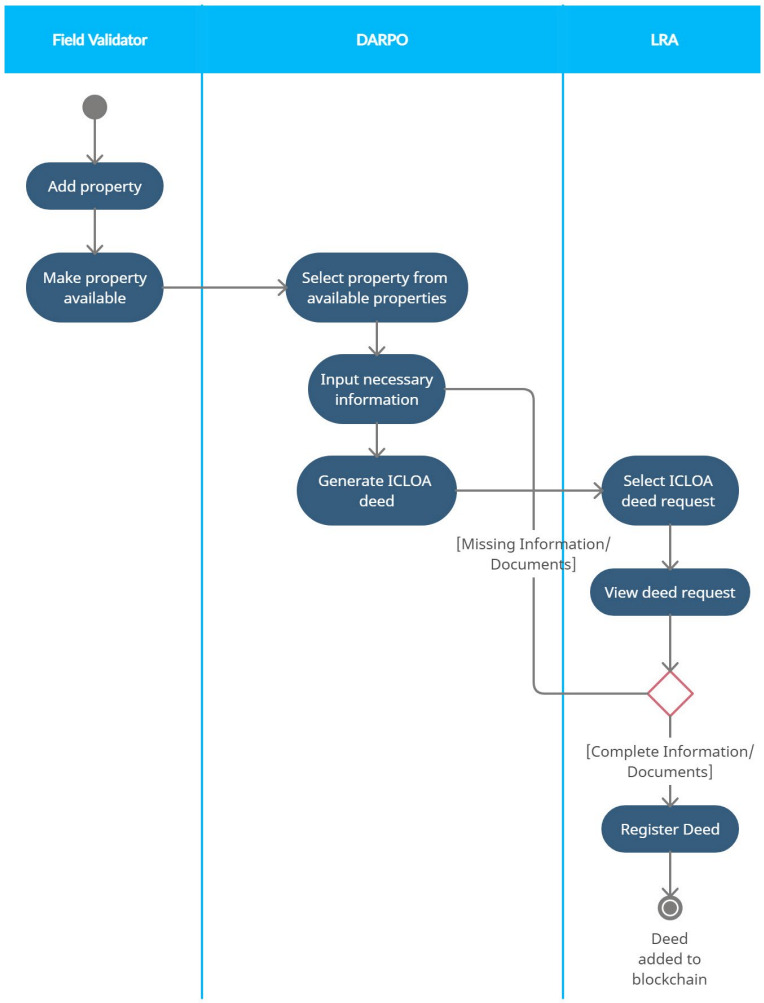
Trigger: The user clicks on “ICLOA Deed Request” in any page then select from available properties and click “Register” button.
Procedure:

1. The user selects deed request from the ICLOA Deed Request Page.
2. The user view all the details and check all the information about the deed.
3. The user clicks “Register” button.
    3.1 ICLOA deed added to the blockchain.
4. The user clicks “Reject” button.
    4.1 ICLOA deed is sent back to the DARPO to be reviewed.

Post condition(s):

3. ICLOA deed added to the blockchain or rejected.

3.2.3 Activity Diagram

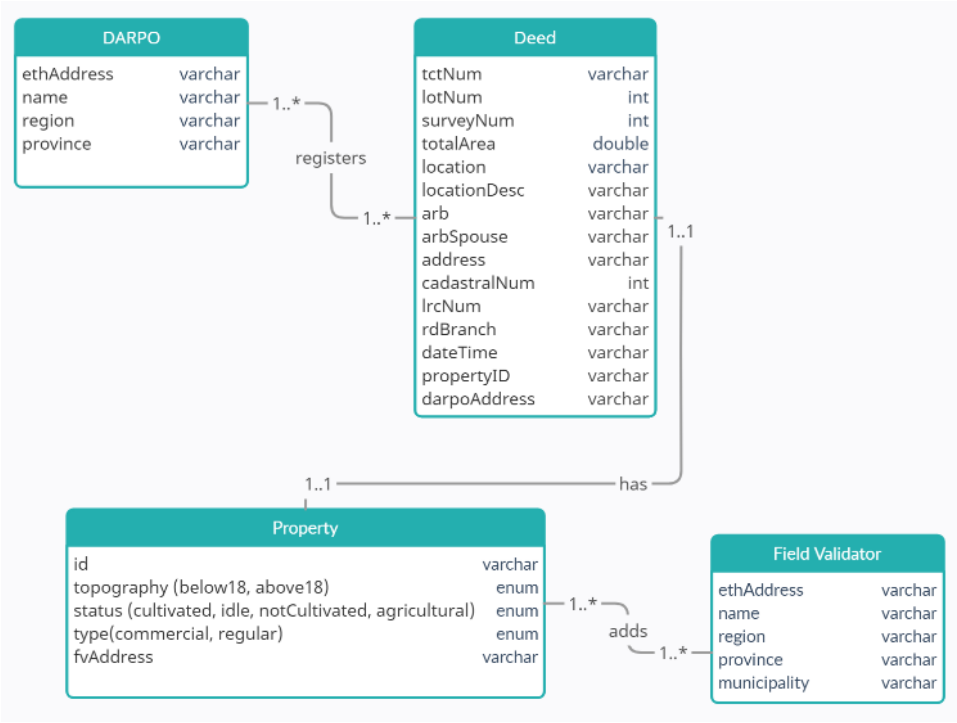


3.3 Performance Requirements

The system should have high performance rate when executing user's input and should be able to provide feedback or response within a short time span usually 50 seconds for highly complicated task and 20 to 25 seconds for less complicated task.

3.4 Database Requirements

3.4.1 ERD



3.4.2 Data Dictionary

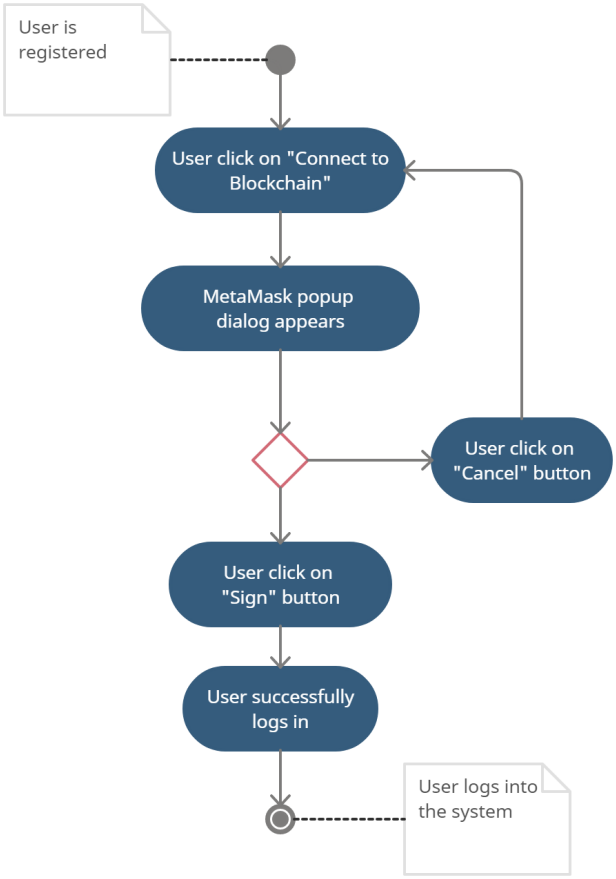
Field Name	Data Type	Data Format	Field Size	Description
ethAddress	varchar		100	Ethereum Address
name	varchar		50	Name of user
region	varchar		50	Region of user
province	varchar		50	Province of user
municipality	varchar		50	Municipality of user
tctNum	varchar		50	Transfer of Title Number
lotNum	int		10	Lot Number of property
surveyNum	int		10	Survey Number of property
totalArea	double		5,2	Total Area of property
location	varchar		100	Location of property
locationDesc	text			Georeference description of lot
arb	varchar		50	Agrarian Reform Beneficiary
arbSpouse	varchar		50	Spouse of ARB
address	varchar		100	Address of ARB
cadastralNum	int		10	Cadastral Number of property

lrcNum	varchar		10	Land Registration Commission Number
rdBranch	varchar		50	Registry of deed branch office
dateTime	date/time	DD/MM/YYYY	10	Date/Time of registration
propertyID	varchar		50	Unique ID of property
topography	enum	(above18, below18)	20	Topography of property
status	enum	(cultivated, idle, agicultural, notCultivated)	20	Current status of property
type	enum	(commercial, regular)	20	Type of property

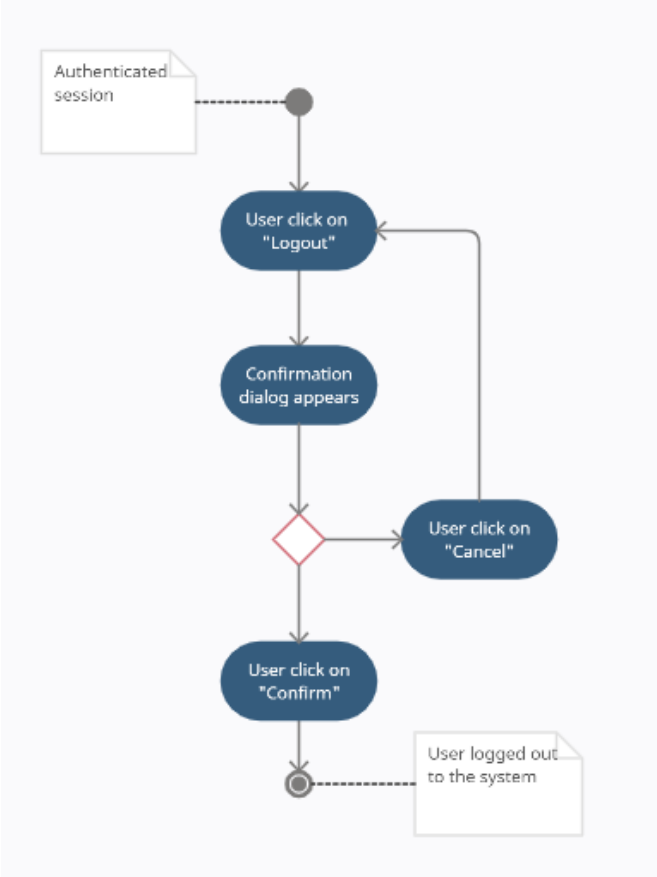
3.5 Object Model

3.5.1 Per Use Case Activity Diagram

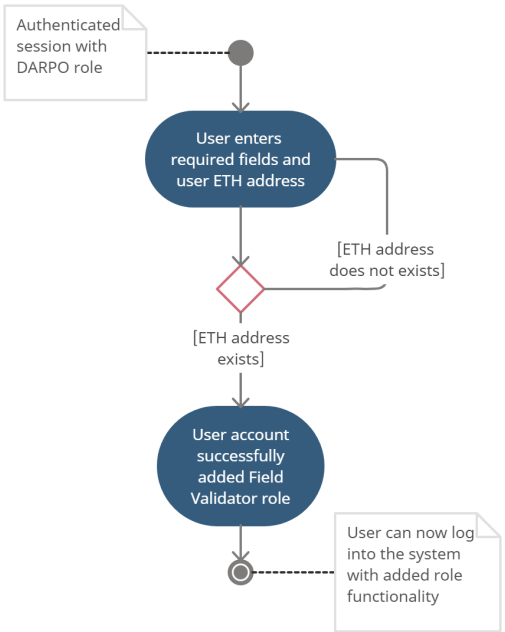
3.5.1.1 Login



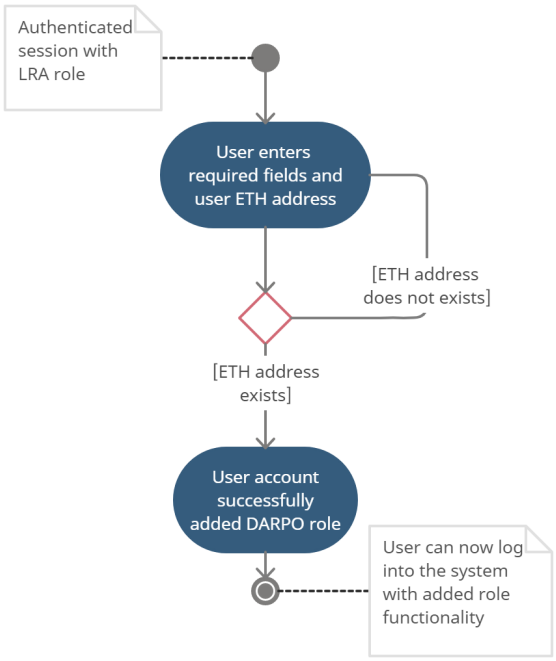
3.5.1.2 Logout



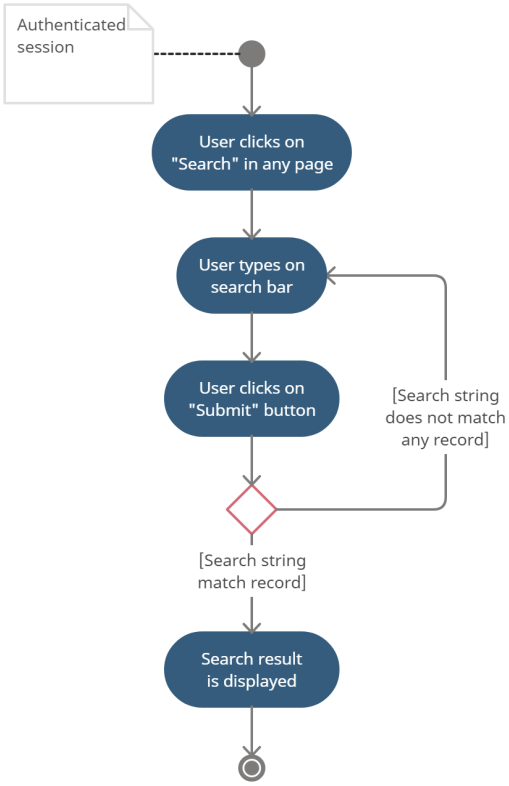
3.5.1.3 Add DARPO



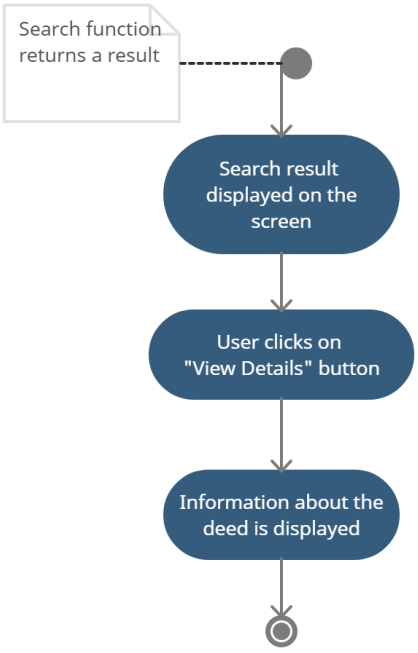
3.5.1.4 Add Field Validator



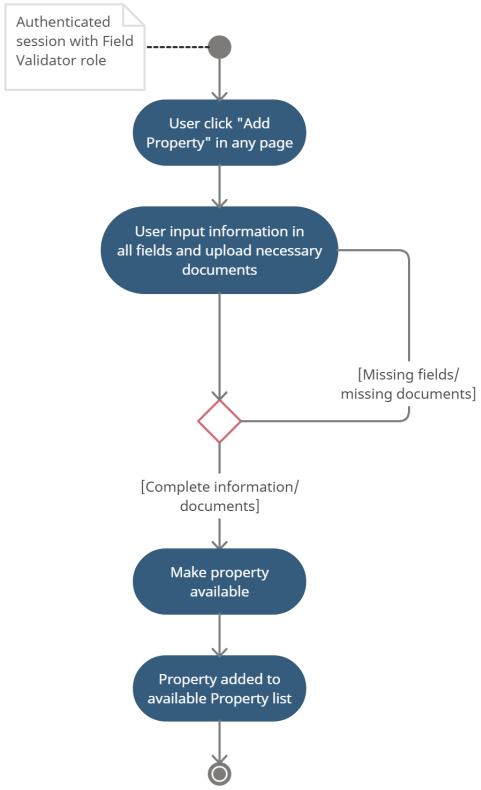
3.5.1.5 Search Deed



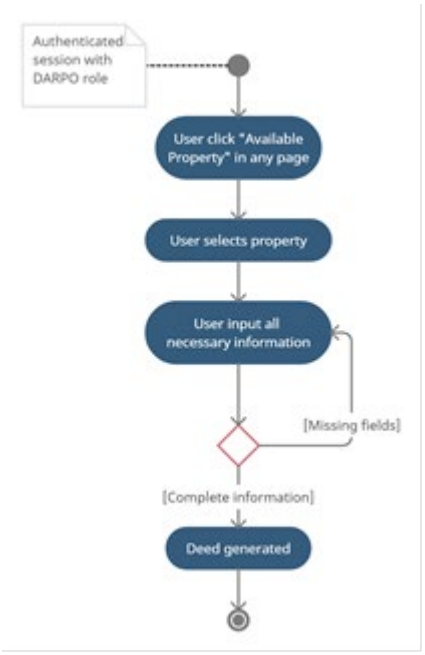
3.5.1.6 View Deed



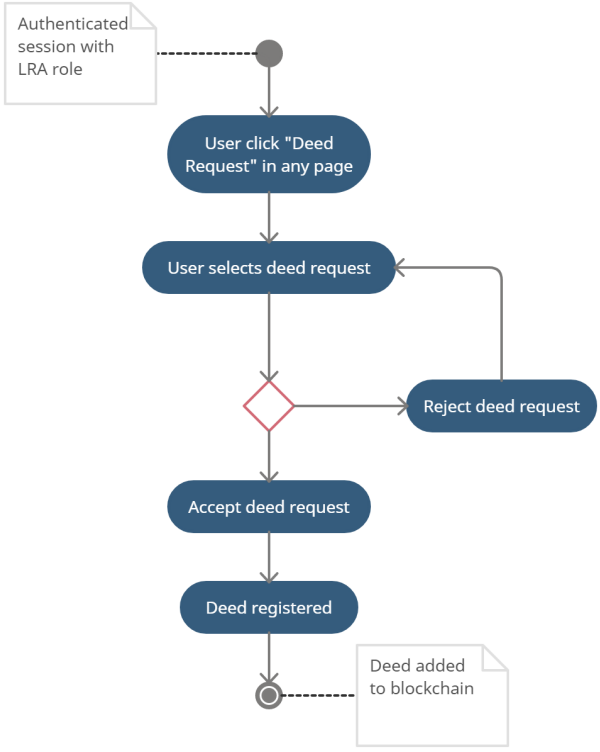
3.5.1.7 Add Property



3.5.1.8 Generate Deed

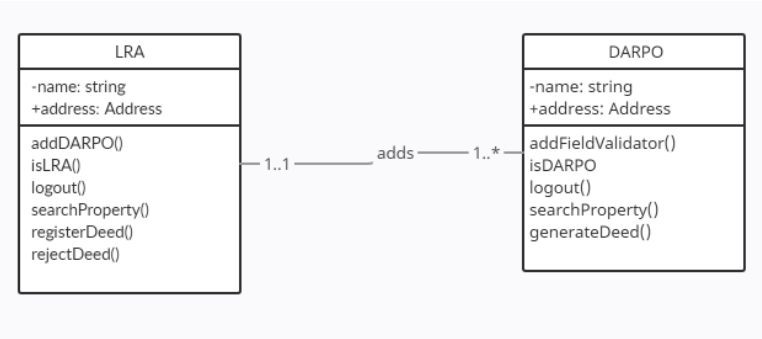


3.5.1.9 Register Deed

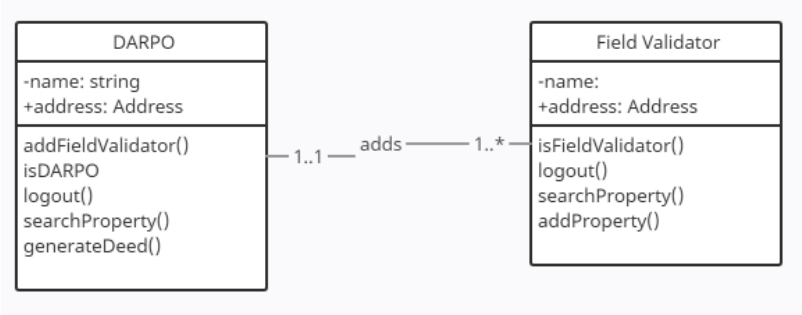


3.5.2 Per Use Case Class Diagram

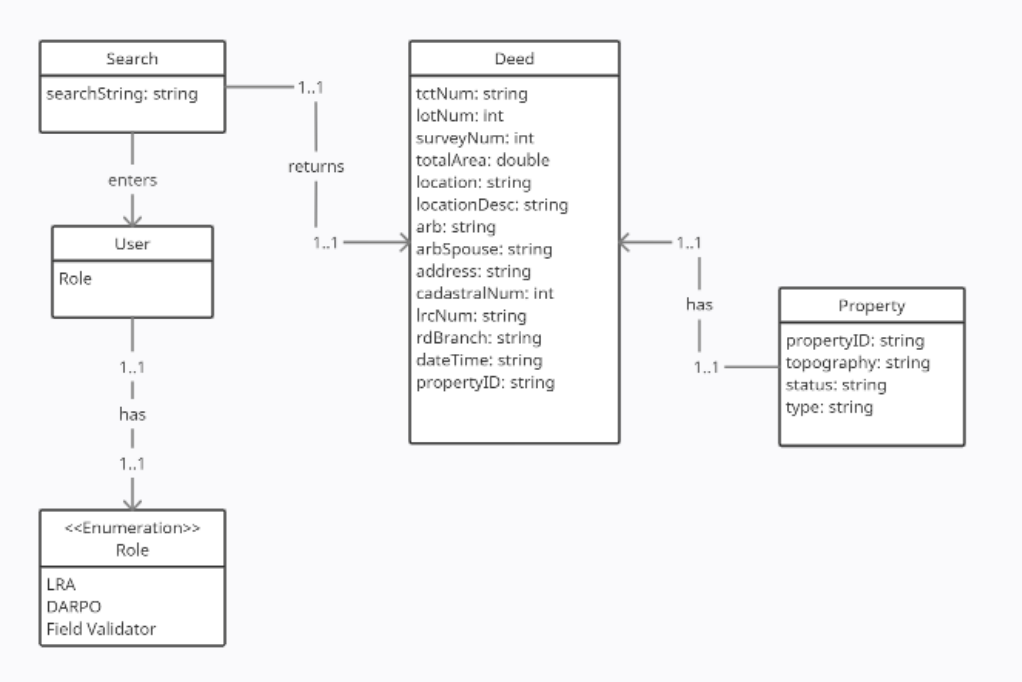
3.5.2.1 Add DARPO



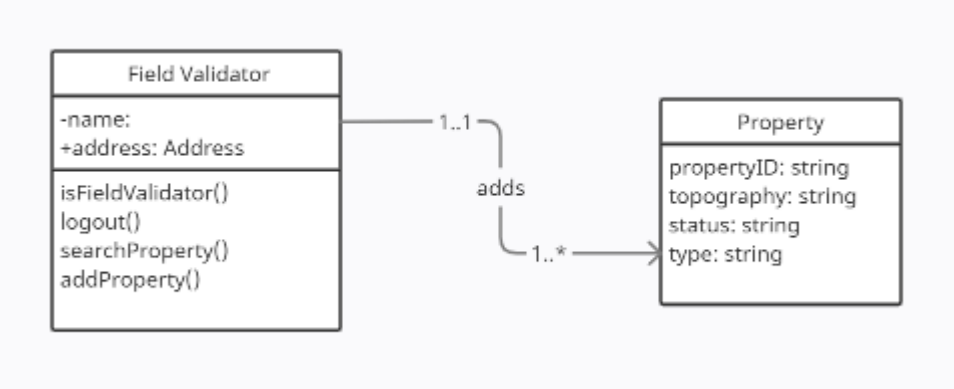
3.5.2.2 Add Field Validator



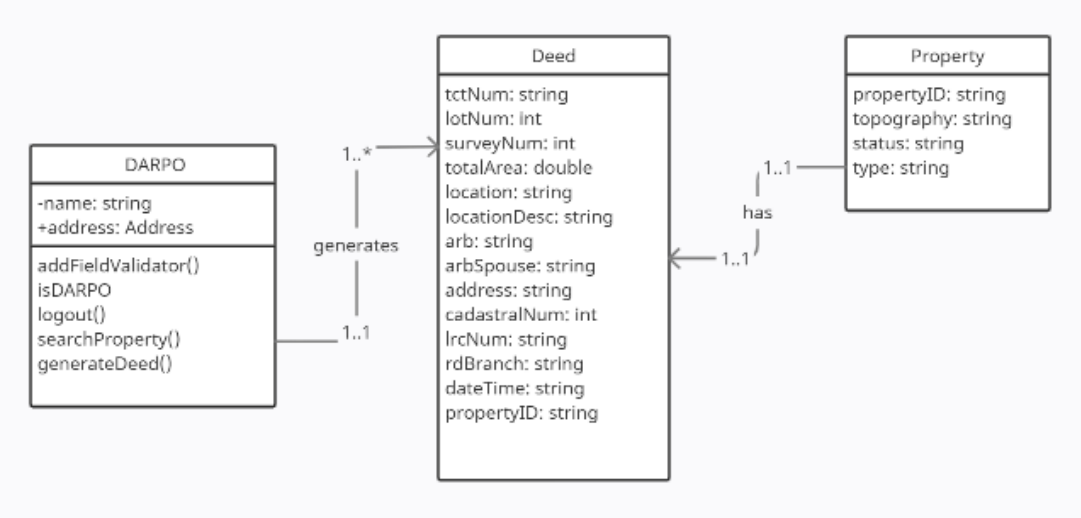
3.5.2.3 Search Deed



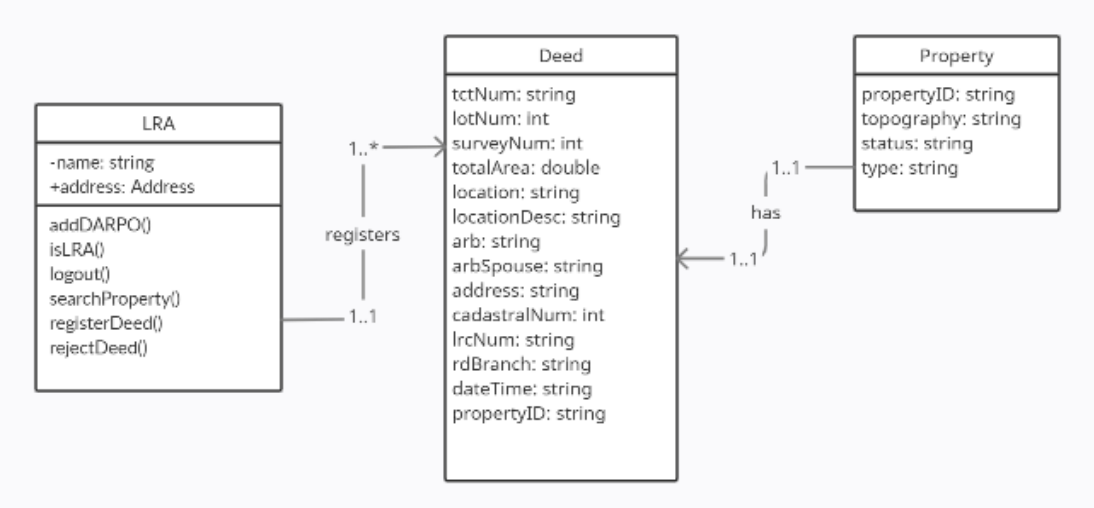
3.5.2.4 Add Property



3.5.2.5 Generate Deed

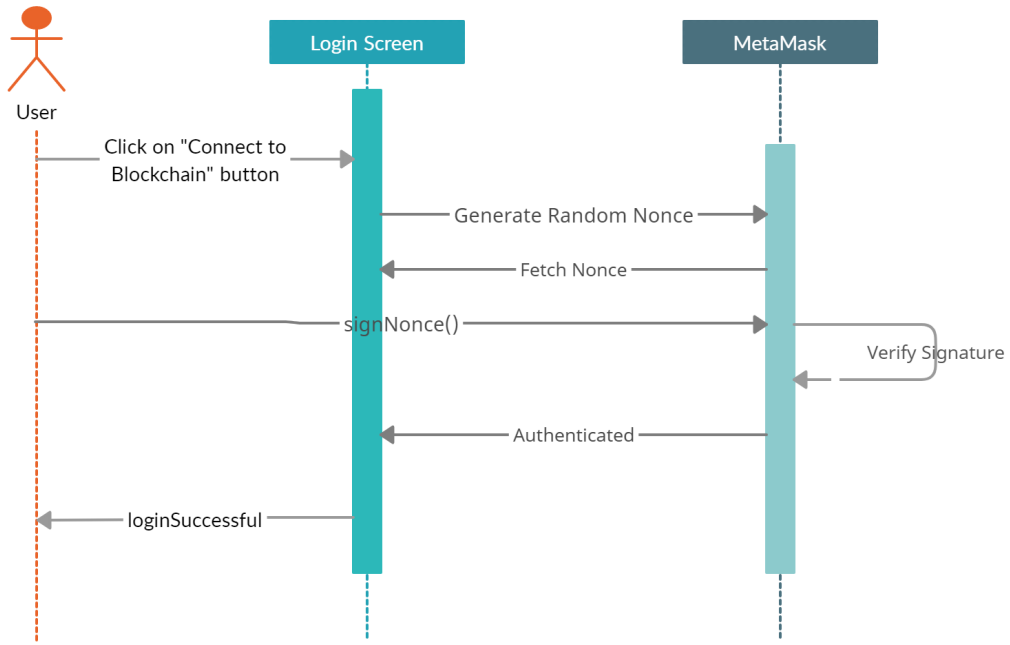


3.5.2.6 Register Deed

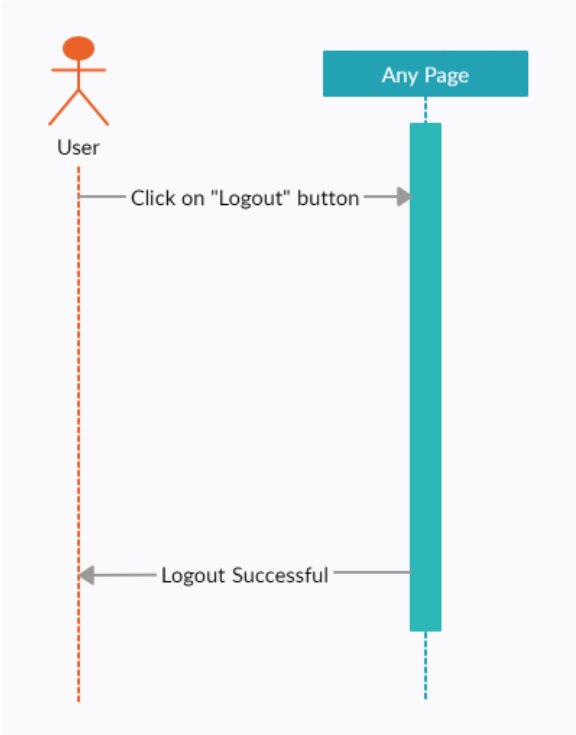


3.5.3 Per Use Case Sequence Diagram

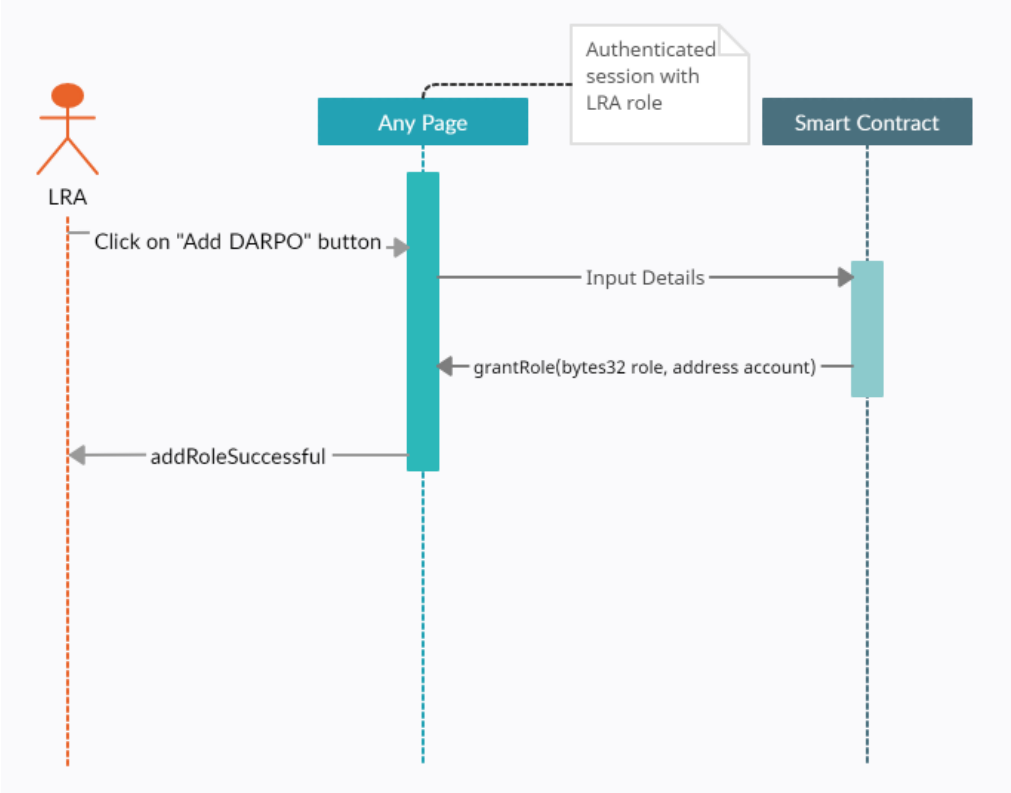
3.5.3.1 Login



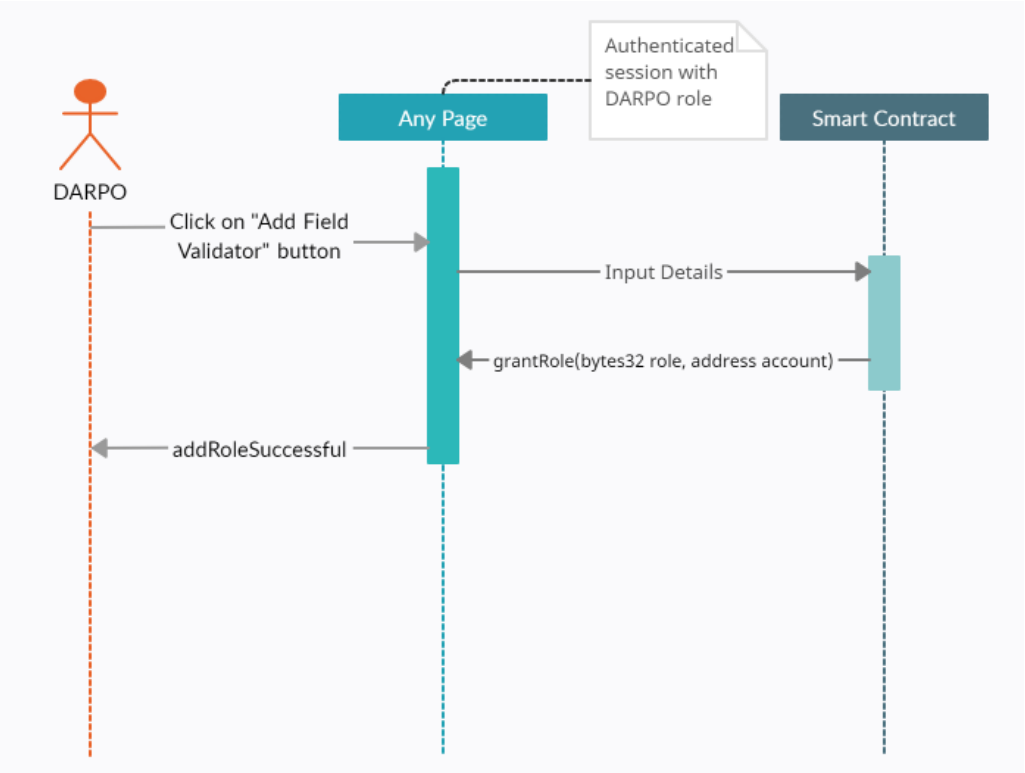
3.5.3.2 Logout



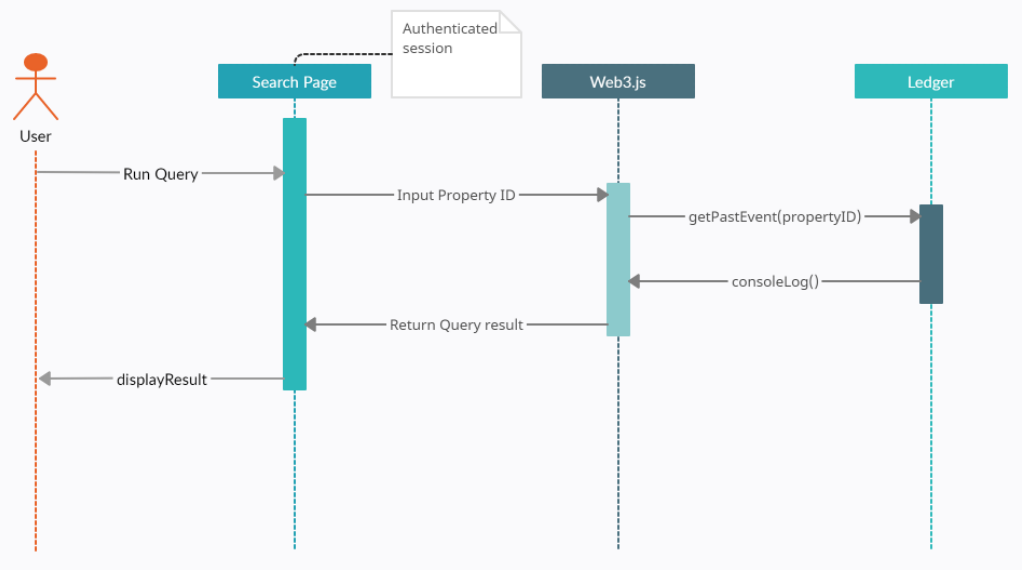
3.5.3.3 Add DARPO



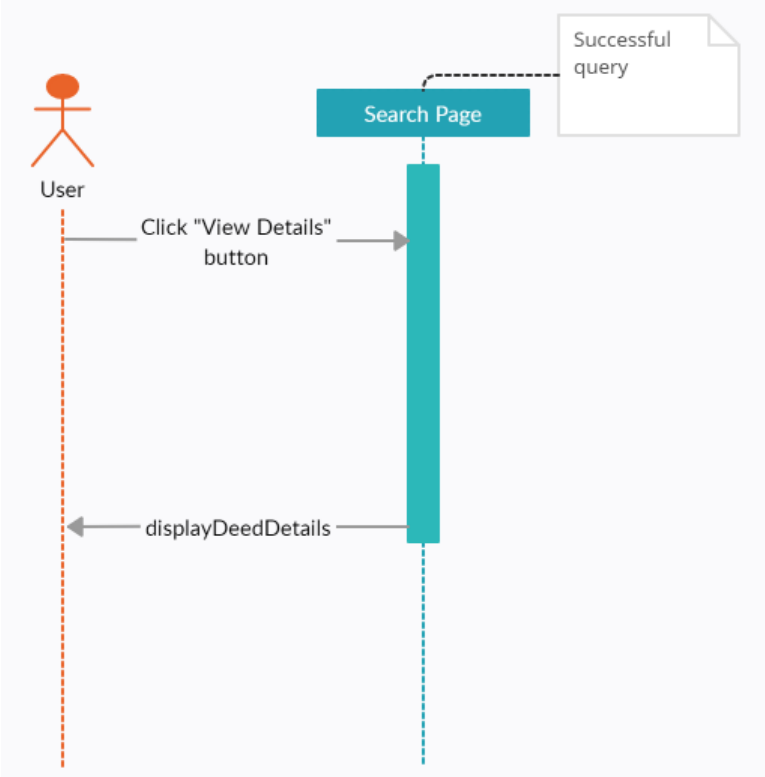
3.5.3.4 Add Field Validator



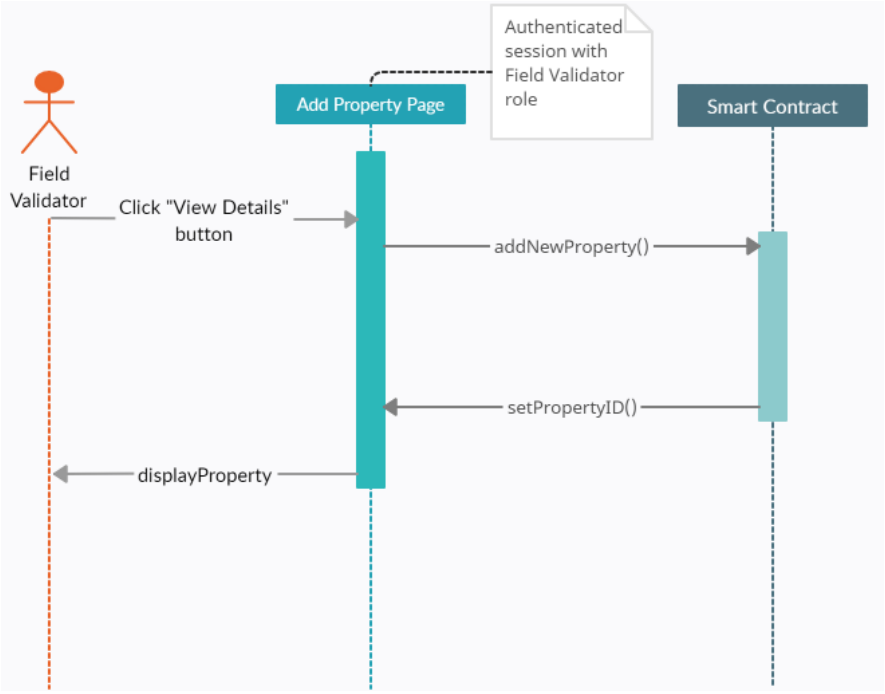
3.5.3.5 Search Deed



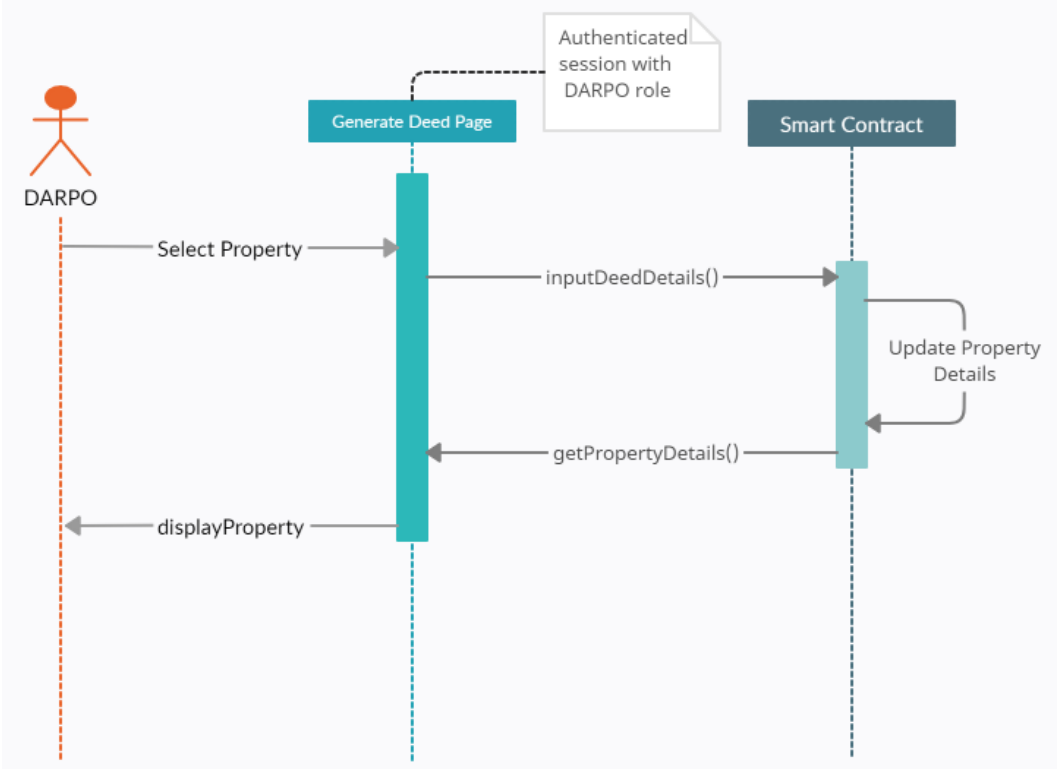
3.5.3.6 View Deed



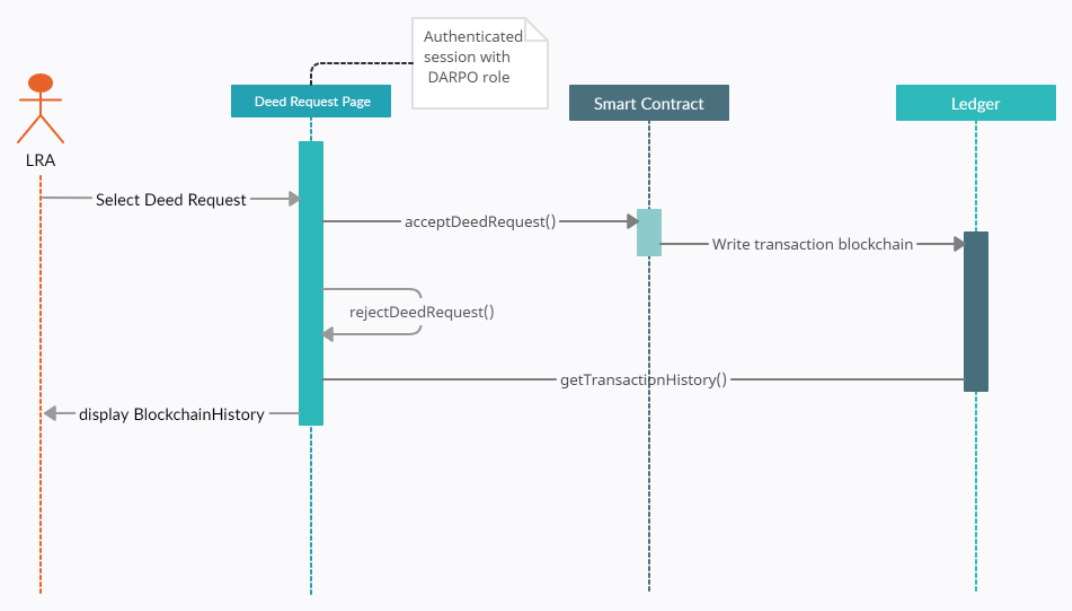
3.5.3.7 Add Property



3.5.3.8 Generate Deed



3.5.3.9 Register Deed



3.6 Software Systems Attributes

3.5.1 Reliability

The system should be highly reliable because full copies of the blockchain ledger are maintained by all active nodes. If one node goes offline, the ledger is still readily available to all other participants in the network therefore lacks a single point of failure.

3.5.2 Availability

The entire system should be available all year round, except for maintenance. The maintenance period should be scheduled and short. The users should be reminded of the unavailability period, well in advance.

3.5.3 Security

The system, at any time, should be accessed only by the authenticated users. Every transaction should be hashed and recorded into an immutable and distributed ledger.

3.5.4 Maintainability

The document should be easy for the users who execute the system day to day. Code must be readable, understandable, well documented and must be able to hand off code to future developers.

3.5.5 Portability

The system should support new versions of the related browsers. All tools and frameworks used should be standard and supported by most platforms.

3.5.6 Usability

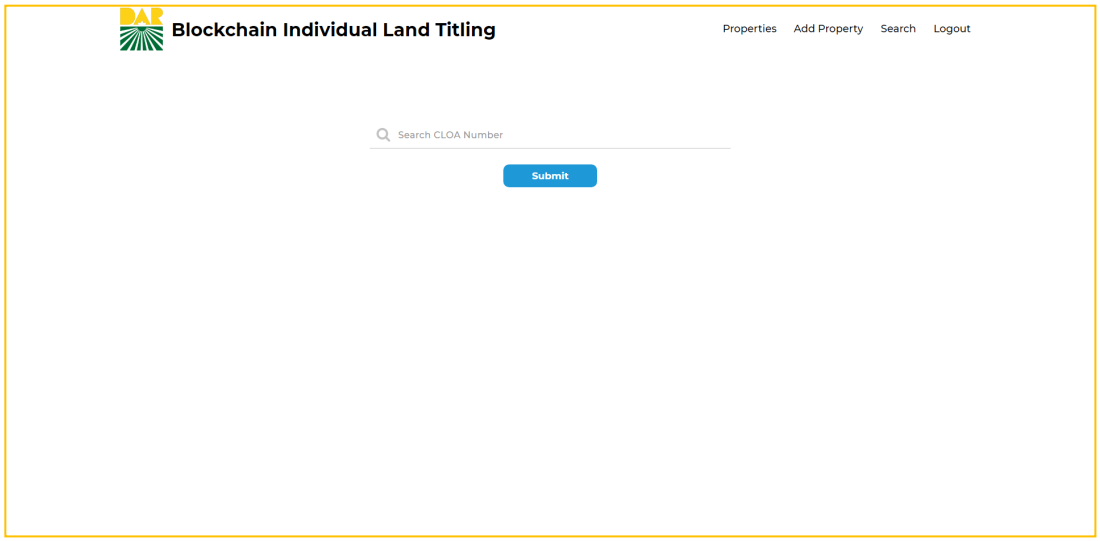
The GUI should be easy to learn and use by users of any technical background. An easy to understand documentation should be provided with the system.

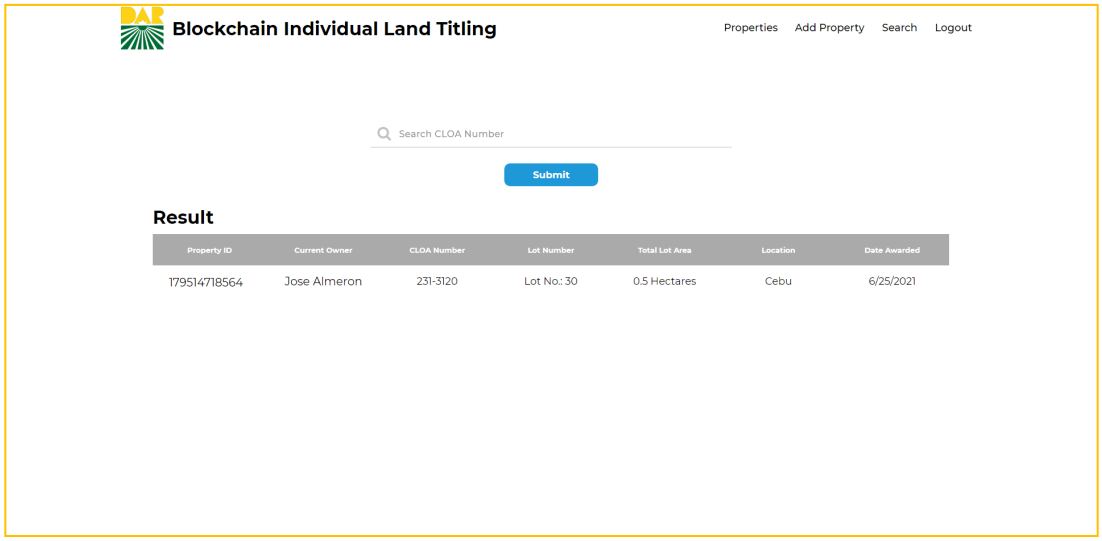
4. Supporting Information

4.1 Appendix: Proposed User Interface Design

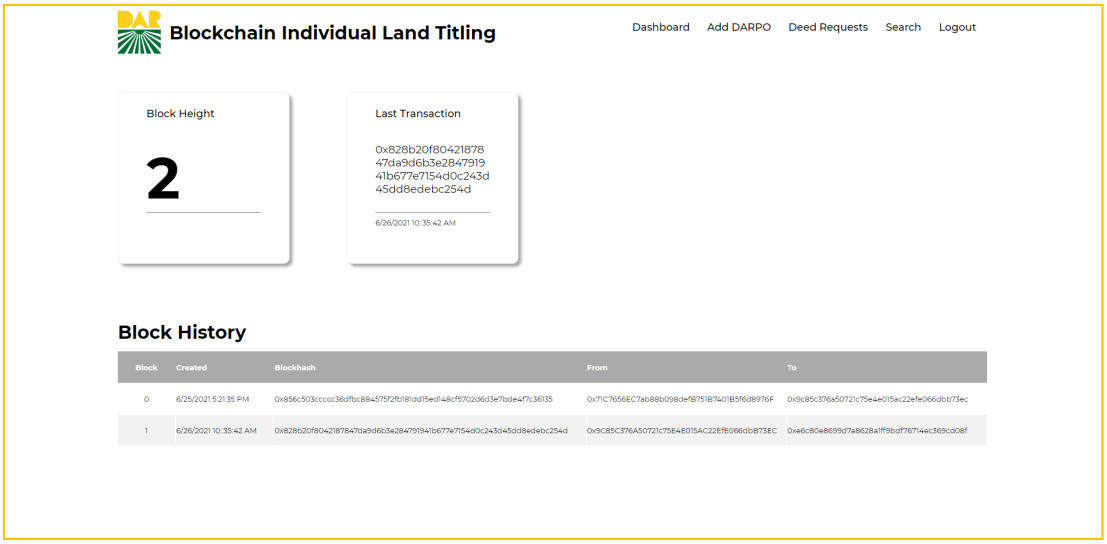
Login Interface

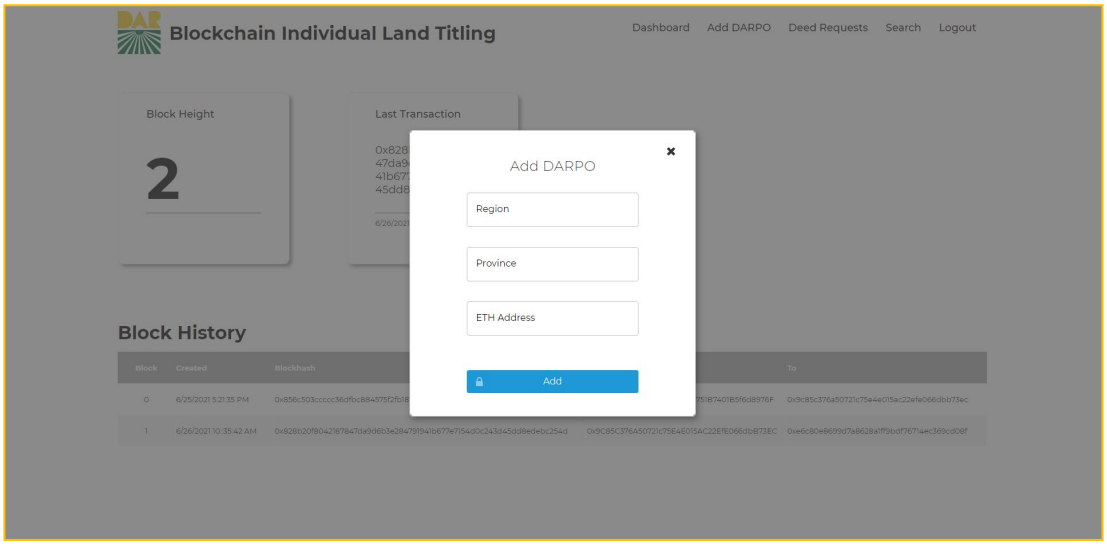
Search Interface

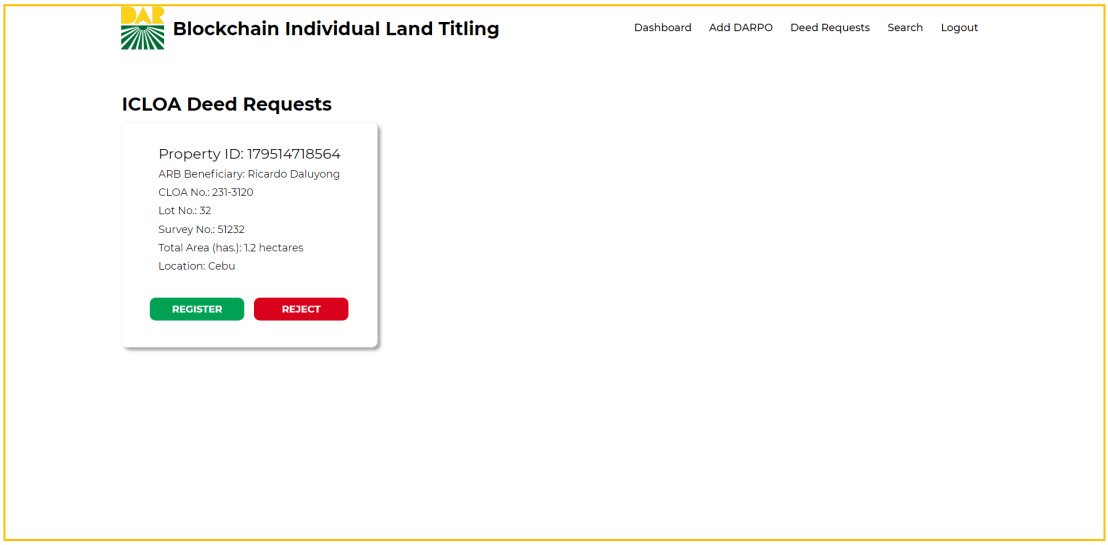




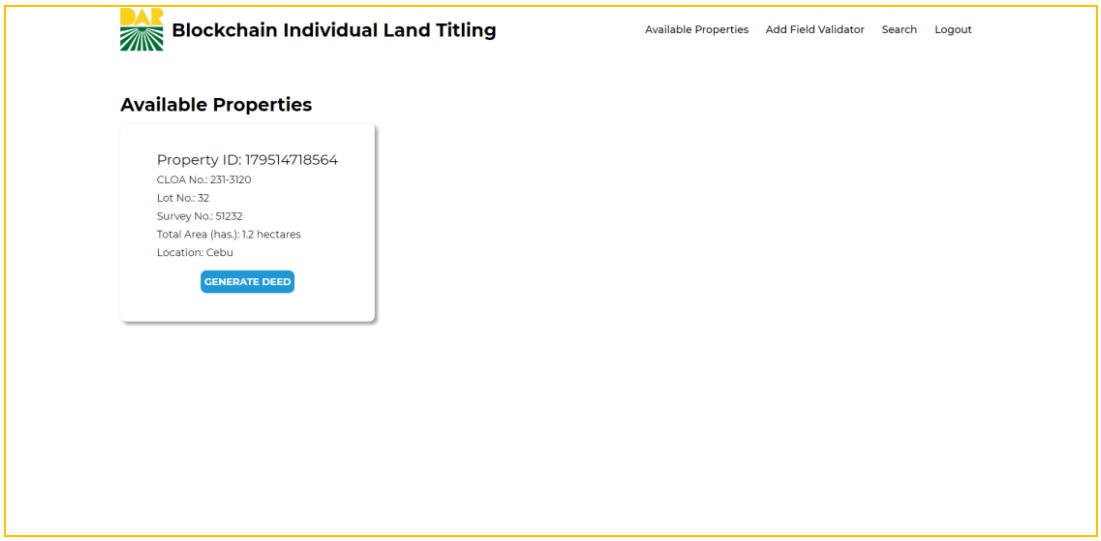
LRA Interface

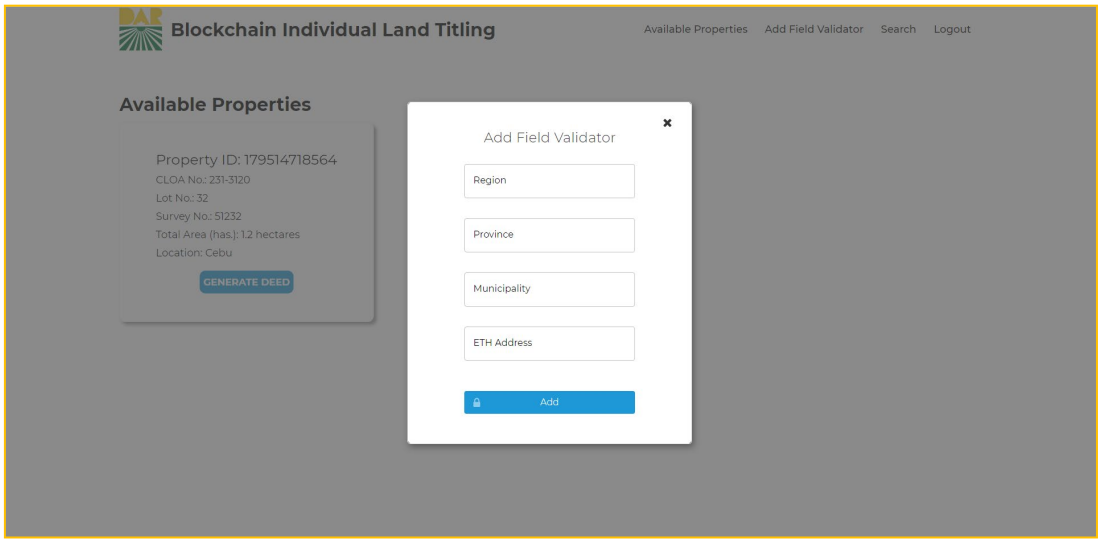


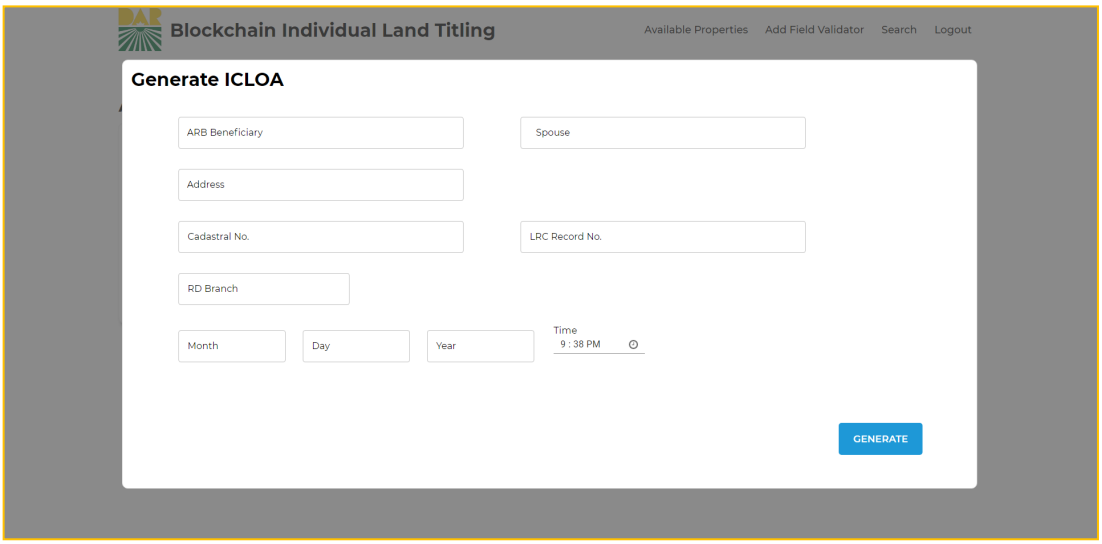




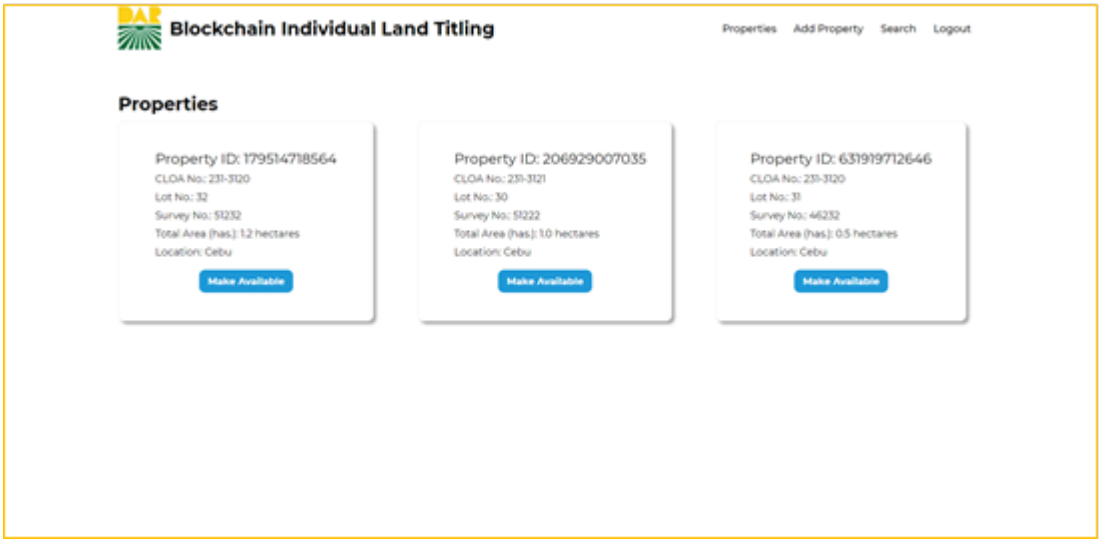
DARPO Interface

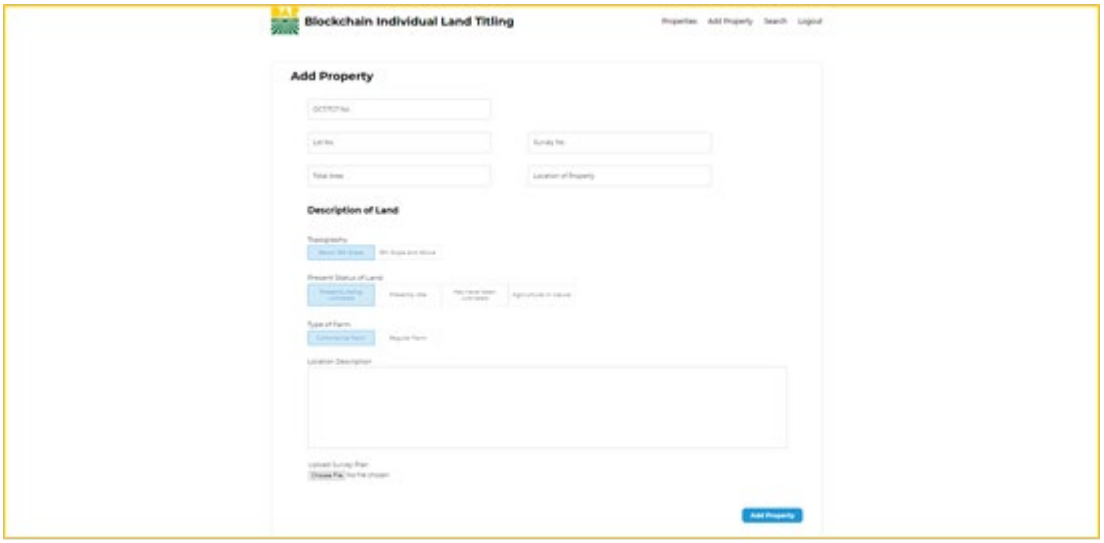






Field Validator Interface





4.2 Index