Introduction

The purpose of creating the Land/Property registry Smart Contract is to ensure that there is transparency and efficiency in acquiring, leasing, succession, collateralizing and successful arbitration of land and land cases. The smart contract enables a clear view of land history and owner history and provides a land/property trustee board as validators of all transactions. A deed is issued to a validated account.

Features

The Following functionalities are needed in the smart contract:

1. Ability to define the owners of the smart contract.
2. Ability to transfer the ownership of the smart contract(sale/succession)
3. Ability to renounce the ownership of the smart contract after a specified period of time
4. Ability to create a land trustee board(validators)
5. Ability to have multiple owners
6. Ability to make payments on land rates and taxes
7. Ability to create non fungible token as title deed
8. Ability to query smart contract for land/property data
9. Ability to suspend transfer into a child contract pending resolution.

Requirements

The contract should be able to:

1. Have a way to loop through the properties/land info
2. Have a profile for properties/land
3. Have the ability to map owners by unique integer id to the properties they own
4. Have a way to map properties to the owners
5. Have the ability to pay any fees required
6. Have the ability to obtain a signed deed
7. Ability to add validators and remove them.

Specifications

The following specifications are desired in the smart contract

Parent Contract (Inheritance)

The contract requires some functions to be accessible only by the owner of the smart contract and thus it’s a requirement to inherit the **Ownable.sol** library of OpenZeppelin.

<https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/ownership/Ownable.sol>

**Note:** Ensure that the inheritance of openZepplin contract is also accounted for

Library Functions

The contract uses additions and subtractions in various functions, thus, it’s also a requirement to import and use the **SafeMath** library of OpenZeppelin.

<https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/math/SafeMath.sol>

**Note:** The SafeMath library provided only deals with uint256 functions. You are also required to create a SafeMath library for uint16 functions as the contract requires addition and subtraction functionality for that variable as well.

Recommended to create a **SafeMath16** library out of the provided openZeppelin link and import that in the contract as well

Structs

The following structs are required:

|  |  |
| --- | --- |
| **Name** | **Structure** |
| Land | Land Struct handles the land mapping of address as well as the id to which they are assigned to.   |  |  | | --- | --- | | **Name** | Type | | value | *uint* | | id | *uint* | | location | *string* | | holdername | *string* | |
|  | |
| Holder | Holder struct gives information of the land holder   |  |  | | --- | --- | | **Name** | Type | | Name | *string* | | ID | uint | | phone | uint | | tax\_pin | string | |
|  | |
|  |  |

Variables

The following variables are recommended to be used:

|  |  |
| --- | --- |
| **Name** | **Type** |
| stampduty | *integer* |

Arrays

The following arrays are recommended to be used:

|  |  |
| --- | --- |
| **Name** | **Type** |
| properties | *array* |
| validators | *array* |
| holders | *array* |

Events

The following events are recommended to be included in the smart contract:

|  |  |
| --- | --- |
| **Events** | **Purpose** |
| holderAdded | This is the owner of the land to be done when a deed is issued . |
| transferApproved | During a sale or lease this is used to emit after a successful transfer is approved by validators. |
| deedIssued | This is emitted if land is given a new holder |

Modifiers

The following modifiers is recommended to ensure proper restrictions on the smart contract functions:

|  |  |
| --- | --- |
| **Modifiers** | **Purpose** |
| onlyOwner() | To check if the caller of the smart contract is the holder. |

Functions

The following functions should be present in the smart contract,

For Reference: <https://solidity.readthedocs.io/en/v0.4.24/contracts.html>

#### Admin Related Functions

|  |  |
| --- | --- |
| **Functions** | **Purpose** |
| regValidators | * To register the validators |
| regLand | * To register the land |
| landProfile | * View function that returns registered land |
| addHolder | * After a deed is issued a holder of the deed details are added |
| removeHolder | * Removes holder on the deed |
| valuateProperty | * A property is valued by validators |
| leaseProperty | * Renting of property for a period of time |

#### 

#### Overriding Ownable Functions

|  |  |
| --- | --- |
| **Functions** | **Purpose** |
| transferOwnership | * Transfer of property holder from old holder to new holder |
| renounceOwnership | * Holder renouncing ownership |