Land Registry Application using Hyperledger Composer

Land Registration is the process of registering documents related to property or land that you own with legal entities. When you purchase a flat directly from the builder, registration of the land gives you the right to legally own, use and dispose of the property. When you have a legal ownership title over a land, the likelihood of fraud or misappropriation is faint.

Need for Land Registration

Land registration is required to maintain the ownership of land deeds; there are many reasons to get your land registered :

1. **Avoid conflicts** : Proper land registration will help individuals avoid conflicts arising from land disputes.

2. **Maintain ownership**: Land registration also helps to showcase the rightful owner of the property.

3. **Legal processes**: Many legal processes require individuals to showcase proper land deeds and documentation.

Problems / Challenges in Land Registration

- Land registration is a mere record of sales transaction.

- There could be multiple parties claiming the ownership of the land.

- Although the properties can be challenged in the court, the verification process is cumbersome and time consuming.

- Ownership could be tampered.

- Tampering of land deeds causes faulty people to own properties for their gain.

- This a major issue with developing countries and also creates a huge backlog for civil cases in courts.

- Benami Registrations: This is a transaction in which a land is transferred to one person for consideration paid by another person.

- This also leads to corruption and evasions of taxes.

Solution using Blockchain

Blockchain is an immutable distributed ledger shared with everyone involved in the network. Every participant interacts with the blockchain using a public-private cryptographic key combination. Moreover, immutable record storage is provided, which is very hard to tamper with. A solution like Hyperledger also provides the features to maintain users and roles, which will additionally help in securing and identifying owners.

The government can utilize the feature set of blockchain to reduce the difficulties faced in the traditional land registration process. A distributed ledger can be set up between the buyer, seller, bank, registration authority, and notary. Land details can be immutably stored and accessed from the blockchain.

Model File Land Registry Application

Participants: Buyer, Seller, Registrar

Assets: Land, Land Listing

Transactions: Created, Registered, Intent for Sale

Participants:

There are three types of clients: buyers, sellers, and registrars.

Model File Buyer:

```
Name of the buyer: The buyer provides his name, which is used to track the buyer details.
Datatype: String
Email of the buyer: The buyer provides his email ID, which can be used to track the buyer details.
Datatype: String
Identification number: This is a unique identification number for the buyer.
Datatype: Integer
Bank name: This is the name of the bank used by the buyer to buy the land.
Datatype: String
Bank address: This is the address of the bank used by the buyer to buy the land.
Datatype: String
Account number: This is the unique account number for the buyer that is registered with the bank.
Datatype: Integer
IFSC code: This is the IFSC code for the bank, which will be used for the transfer of funds.
Datatype: String
Balance (optional): This is the balance in the buyer’s bank account.
Datatype: Integer
```

Note: Identification of the buyer mentioned above can be any one of the following in the form of an alphanumeric string or an integer. For this case study, data type as an integer:

- Passport

- Identity proof
- Driving license
- SSN
- Any other identification proof

Seller:

```
1.  Name of the seller: The seller provides his name, which is used to track the seller
    details.
2.  Datatype: String
3.  Email of the seller: The seller provides his email, which can be used to track the
    seller details.
4.  Datatype: String
5.  Identification number: This is a unique identification number for the seller.
6.  Datatype: Integer
7.  Bank name: This is the name of the bank used by the seller to receive the funds against
    the Land.
8.  Datatype: String
9.  Bank address: This the address of the bank used by the seller to receive the funds against
    the land.
10. Datatype: String
11. Account number: This is the unique account number for the seller that is registered with
    the bank.
12. Datatype: Integer
13. IFSC code: This is the IFSC code for the bank, which will be used for the transfer of
    funds.
14. Datatype: String
15. Balance (optional): This is the balance in the seller’s bank account.
16. Datatype: Integer
17. SaleDeedDocs: This is the hex for the sales deed doc for the land owned by the seller.
18. Datatype: String
```

Note: Identification of the seller mentioned above can be any one of the following in the form of an alphanumeric string or an integer. For this case study, data type as an integer:

- Passport
- Identity proof
- Driving license
- SSN
- Any other identification proof

Registrar:

```
1.  Name of the registrar: The registrar provides his name, which is used to track the     det
    ails.
2.  Datatype: String
```

3. Email of the registrar: The registrar provides his email ID, which can be used to track the details.
4. Datatype: String

Assets:

Two types of assets in the case study: one to maintain the lands and another to maintain the land listings for sale.

Land:

1. Land ID: This is a unique ID for each land stored in the business network. It also acts as a primary key and is used to track the land details.
2. Datatype: String
3. Owner: This is the owner of the land registered over the network.
4. Datatype: String
5. Registration date: This is the date on which the land is registered with the Composer application.
6. Datatype: String
7. Land type: This is the information detailing the type of the land, such as “2 BHK, 3 BHK or villas.” You can define your own types too.
8. Datatype: String
9. Location: This defines the address of the land.
10. Datatype: String
11. Public: This is a flag used to define whether the land is a public property or not.
12. Datatype: Boolean
13. Private: This is a flag used to define whether the land is a private property or not.
14. Datatype: Boolean
15. Market price: This is the market price for the lands that are registered over the network.
16. Datatype: Integer
17. Intent for sale: This is a flag set by the owner of the land to showcase the intent for sale.
18. Datatype: Boolean
19. Status: This defines the status of the Land: "Intent of Sale" or "Registered."
20. Datatype: String
21. Default: Registered

Land listing:

1. Land listing ID: This is a unique ID for each land listed in the business network. It also acts as a primary key and is used to track the land listing details.
2. Datatype: String
3. Owner: This is the owner of the land listed in the network.
4. Datatype: String
5. Registration date: This is the date on which the land is registered with the Composer application.
6. Datatype: String
7. Land type: This is the information detailing the type of land, such as "2 BHK, 3 BHK or villas." You can define your own types too.
8. Datatype: String
9. Location: This defines the address of the land.
10. Datatype: String
11. Market price: This is the market price of the lands that are listed for sale over the network.
12. Datatype: Integer
13. Status: This defines the status of the lands: "Created", "Intent of Sale", or "Registered."

```
14. Datatype: String
15. Default: Intent of Sale
```

Transactions:

The business network will contain processes to handle the properties. The class project will involve transactions to add the property over the network, register the property under a new owner's name after sale, and add the property for sale over the network. The transactions will be defined

```
1.  LID: This is a unique ID to represent land.
2.  Datatype: String
3.  Land: This is a reference to the asset land, where a new object of an asset is passed  on
    to the transaction.
4.  Datatype: Land
```

Registered:

```
1.  LID: This is a unique ID to represent the land listing that is transacted to be registered
    under the new owner's name.
2.  Datatype: String
3.  Land listing: This is a reference to the asset land listing, where a new object of an asse
    t is passed on to the transaction for registering a new owner.
4.  Datatype: Land listing
5.  Buyer: This is a reference to the participant buyer who is purchasing the land.
6.  Datatype: Buyer
```

Intent for Sale:

```
1.  Land: This is a reference to the asset land, where a new object of asset is passed on  to
    the transaction.
2.  Datatype: Land
3.  Land listing: This is a reference to the asset land listing, where a new object of     ass
    et is passed on to the transaction for setting up the Intent for Sale.
4.  Datatype: Property listing
5.  Seller: This is a reference to the participant seller who is selling the land.
6.  Datatype: Seller
```

Business Logic Definitions for Transactions:

If the user decides to register the land data with the network, they can call the create function to register the same.

Created transaction business logic:

- This function is used by the participants to add new land to the business network archive.

- Parameters: The transaction definition from the model file will be passed as the parameter. Also, the mapping for the transaction will be added to the script file defined to provide the business logic for the application.

- Return: The ID as a string is returned for the new land added to the business network.

Business Logic Definitions Intent for Sale Transactions:

If the user decides to list the land for sale in the network, they can call the Intent for Sale function to perform the same.

Intent for Sale business logic:

- This function is used by the participants to list lands for sale in the business network archive.

- Parameters: The transaction definition for Intent for Sale from the model file will be passed as the parameter. Also, the mapping for the transaction will be added to the script file defined to provide the business logic for the application.

- Return: The ID as a string for the new land listed is returned to the business network.

Business Logic Definitions for Registered Transactions:

When the users finalize the sale of the land then they can call the registered transaction to register the land back to the land registry with the new owner:

Registered Business logic:

- This function is used by the participants to confirm the sale of the land in the business network archive.

- Parameters: The transaction definition for registered transaction from the model file will be passed as the parameter. Also, the mapping for the transaction will be added to the script file defined to provide the business logic for the application.

- Return: The ID as a string for the new land object is returned to the business network.