Software Requirements Specification

For

Digitalization of Land Records using Blockchain
30 March 2022

Prepared by

Specialization	SAP ID	Name
CSE BAO	500076372	Ishan Agarwal
CSE BAO	500075932	Sachin Kedia
CSE BAO	500075119	Sharique Ahmad Khan

Under
Mr. Deepak Kumar Sharma

Department of Informatics
School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand

Table of Contents

Revision History

Date	Change	Reason for Changes	Mentor Signature

1. INTRODUCTION

1.1 Purpose of the Project

Today's Land Record Management System is plagued with multiple loopholes which result in discrepancies in records, resulting in a burden of court cases on judicial system.
Some problems with current land record systems are:

1. Errors in public records affects ownership rights and cause financial strain
2. Illegal deeds by not documenting prior titles in the chainage may affect the ownership
3. Forged or fabricated documents affect the ownership
4. Undiscovered encumbrances (Ongoing Cases)
5. Unknown easements affects right to enjoy the property
6. Boundary Disputes

This project attempts to resolve these issues by implementing a blockchain network to store and manage land records. The project will store cryptographically secured history of land records as a blockchain ledger on all nodes of the system.

1.2 Target Beneficiary

All the members will easily be able to perform land transaction within system as well as to non-members. Land records will be easy to maintain and immutable. There will be complete transparency in the details of land record and any member within system can access the land record details. Admin can easily track the transaction and can verify to add it to the blockchain.

1.3 Project Scope

Blockchain based land record management system can be used nationwide which will ultimately help in reduce of land disputes and since anyone can access the information of land records, it will help to prevent frauds. Since the blockchain is immutable, no attacker would be able to change the ownership details. Thus, creating trust within the system. Government can easily keep track of land transactions. Even smart contracts can be used to automate the agreement between buyer and seller.

2. PROJECT DESCRIPTION

2.1 Reference Algorithm

Different Blockchain features implemented in the projects are: -

- **Distributed Ledger:** A distributed ledger is a consensus of replicated, shared, and synchronized digital data geographically spread across multiple nodes. All the available nodes (i.e., owners) will have a copy of blockchain stored in their system.
- **Peer to Peer Network:** As the network will be a peer-to-peer network, there will be a direct communication between buyer and seller.
- **Hashing:** A hash function is any function that can be used to map data of arbitrary size to fixed-size values. The values returned by a hash function are called hash values which will be used in chain formation as the PrevHash value will be stored in current block. Where hash of complete transaction details along with prevHash will be calculated.
- **Digital Signature:** A digital signature is basically used to ensure the transaction is being carried out between two genuine person and no other person could interfere between the transaction process. A combination of public-private key encryption will ensure the implementation of digital signature.
- **Consensus:** Process of verification and validation in a blockchain transaction is known as consensus. Here, we are using proof of authority as consensus mechanism, since here only admin has right to verify and add transaction detail to the blockchain.

2.2 SWOT Analysis

Strengths:

1. There will be no mediator in buy-sell transactions
2. All nodes will be in consensus with each other so any tampering will only be possible if data is changed in 51% of nodes.
3. Transactions will be secured via digital signature.

Weaknesses:

1. No provision of agreement among buyer and seller in the system.
2. This concept for now is designed only for small scale database.

Opportunities:

1. This concept can be used Nationwide which will ultimately help in reducing Land disputes.
2. Smart Contracts can be introduced in blockchain which can automate the task of Agreement between Seller and Buyer.

Threats:

1. To replicate current land records into the system will be very time taking

2.3 Project Features

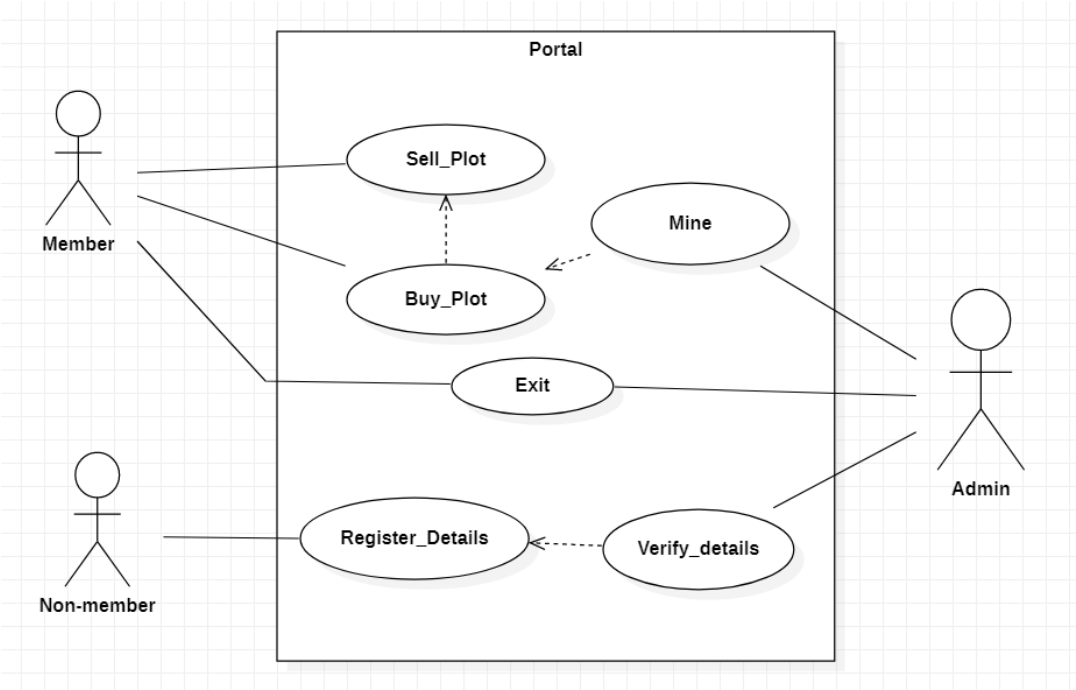


Figure 1. Use case diagrams for functionalities users can perform

User:

- Sell: In order to initiate transaction to sell plot which are being owned by the user
- Buy: Approve initiated transaction which are directed to the user's ID
- Exit: Logout from the portal.

Admin:

- VerifyUser: Verify new user details and add it to the society members list.
- Mine: Takeout the transaction from mempool and verify it to add it to the blockchain.
- Exit: Logout from the portal.

New User:

- Register: Enter all the details asked to get register in the society.

2.4 User Classes and Characteristics

The user of the software can easily maintain the land records information and can easily monitor the transaction without any risk of information getting attacked by outside hacker. All the past transaction and ownership history of the of the land can easily be maintained. And since the information are transparent to all, disputes related to ownership and boundaries will not arise.

2.5 Design and Implementation Constraints

Blockchain and transactions are maintained in a text file using File handling. Proper formats of information storage are defined in the file. All the information related to plots are stored in plot.txt. Information regarding users and ownership are maintained in users.txt. Blockchain is maintained in text file record.txt with proper format of Block number, transaction details, hash value and PrevHash value with blocks separated using block numbers. Each block in blockchain is maintained in separate text file Block<block_no>.txt. Transaction in mempool is maintained in mempool.txt with transactions separated by seller Id and stored in different lines.

2.6 Design diagrams

The below diagrams show functioning of the project at current stage of progress:

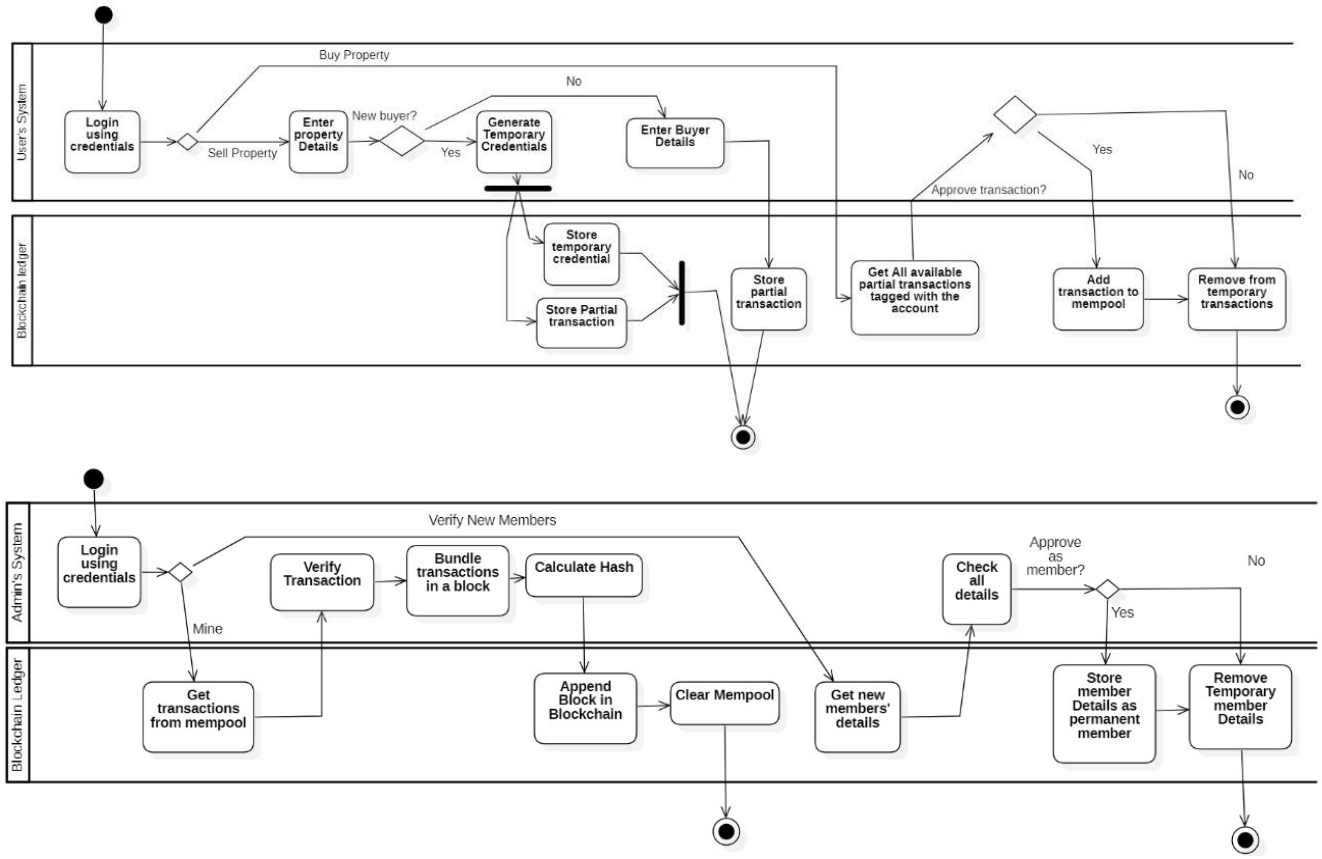


Figure 2. Activity diagram of the project

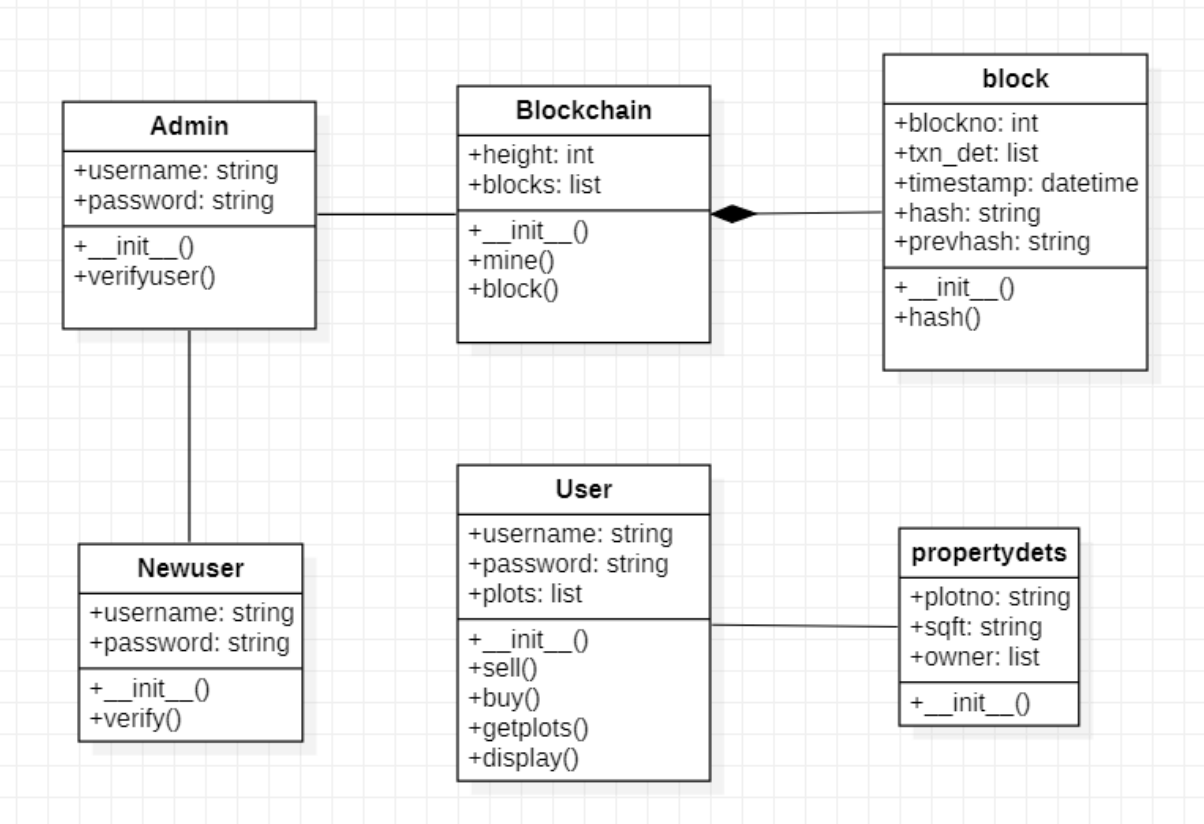


Figure 3. Class diagram of the project

2.7 Assumption and Dependencies

It is assumed that the land agreements and actual registry are being performed outside the blockchain and this blockchain is maintaining land records for the society which can be used for legal purposes as a proof in court and accredited and accepted by all the authorities.

3. SYSTEM REQUIREMENTS

3.1 User Interface

The console based application provides a menu allowing for easy control by a keyboard. The menu will allow user to access functionalities such as:

- Login Portal
- Buyer:
  - See details of owned plot
  - Approve transaction initiated by seller
- Seller:
  - See details of owned plot
  - Initiate transaction to sell plot
- Admin:
  - Verify non-society member information
  - Verify transaction from mempool and add to blockchain.

a. Software Interface

The program (until now) consists of 4 modules –

- **Blockchain –** This class maintain the complete blockchain and the functions to manage the blockchain. Functions members like loadBlockchain is used to load the last block information about the blockchain so that new blocks can be added on previous blocks. AddBlock is used to add new block in the blockchain.
- **Buyer –** Buyers are basically of two types Already a society member and non-society member. Already member can confirm the initiated transaction and the transaction get added to mempool. Non-Society member has to use the temporary credentials provided by seller to login and fill in the details, and once verified by the admin, he can verify the initiated transaction by the seller.
- **Seller –** Sellers can initiate the transactions to sell the plot which being owned by them to another society member or non-society member. In case of non-Society member, the system will generate the temporary credentials which the seller can send it to the buyer.
- **Admin –** Admin class has two functions verifyUser and verifyTransaction. In verifyUser, admin verify the details of the non-society member and approve them to become a society member. VerifyTransaction is used to verify the transaction from the mempool and once verified, transaction details are added to the blockchain and ownership of plot is transferred to the buyer.

4.NON-FUNCTIONAL REQUIREMENTS

4.1 Software Quality Attributes

The Program is a python-based program and it will work on any system which has python installed in it.

Appendix A: Glossary

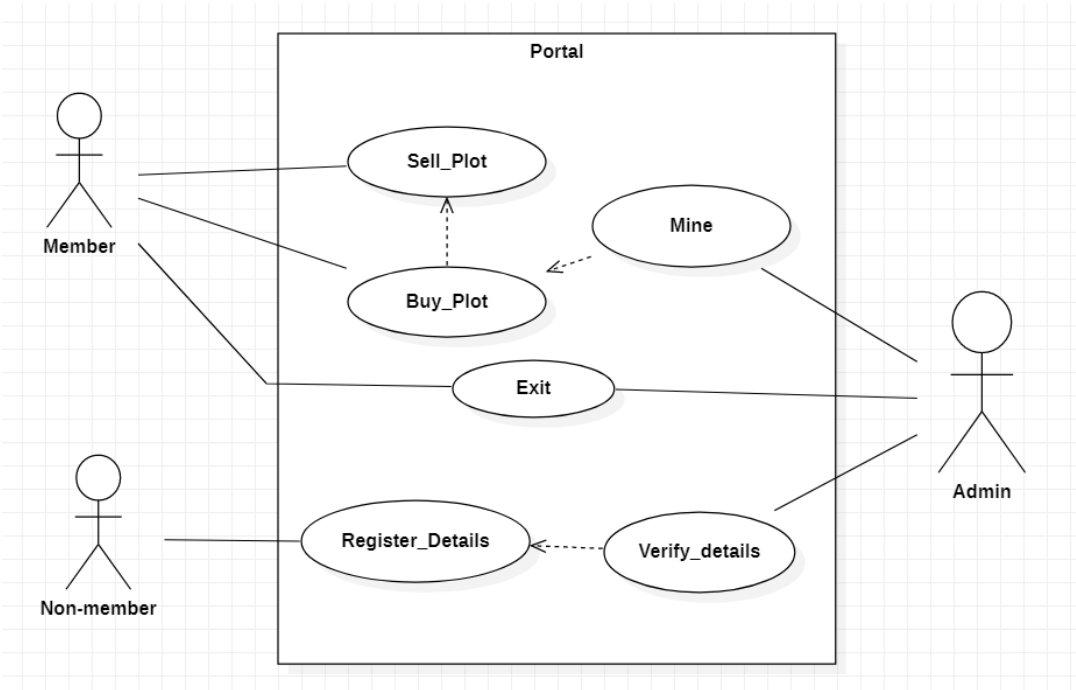
mempool: mempool are the temperory storage of the transaction details which are being initiated by the seller and approved by the buyer. Then admin verifies each transaction picking from the mempool.

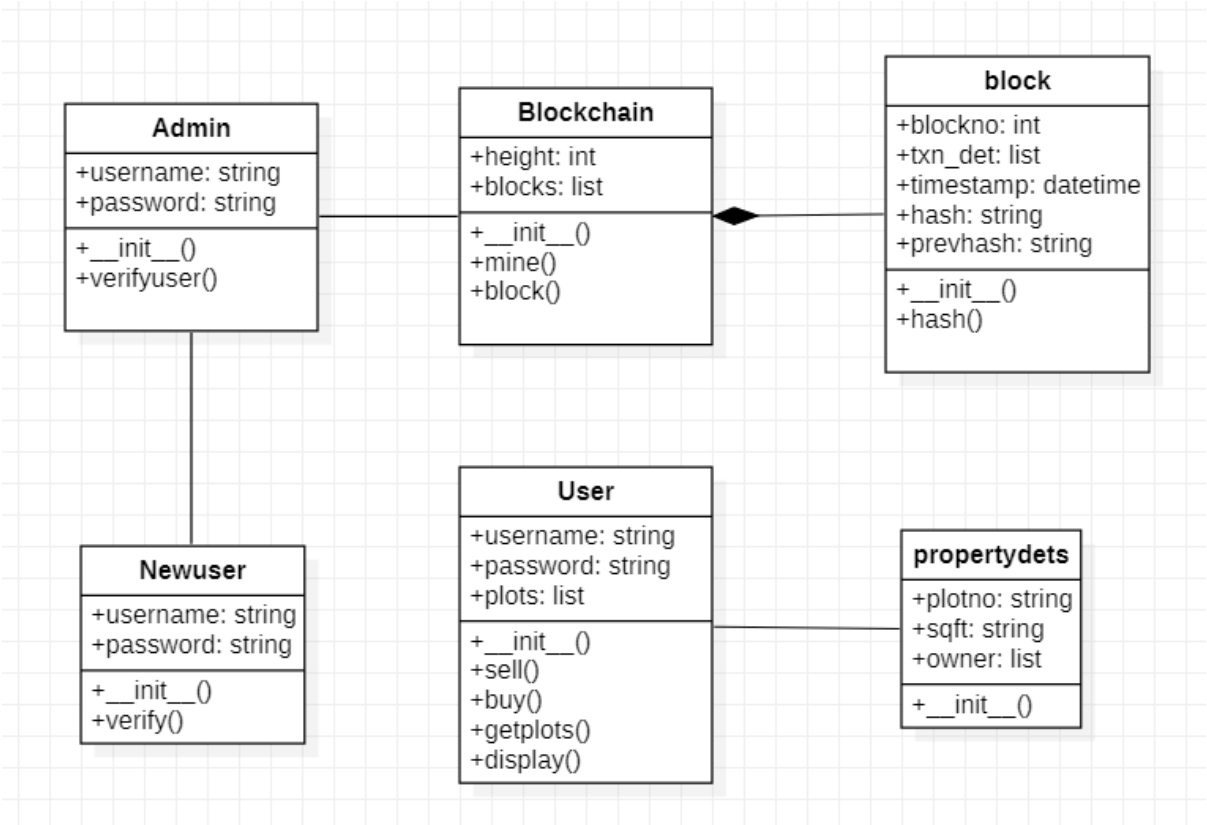
Ledger: Ledger are the records of transactions which are being performed between seller and buyer. Blocks in the blockchain stores the ledger in it along with other information like, hash value, time stamp, block number, etc.

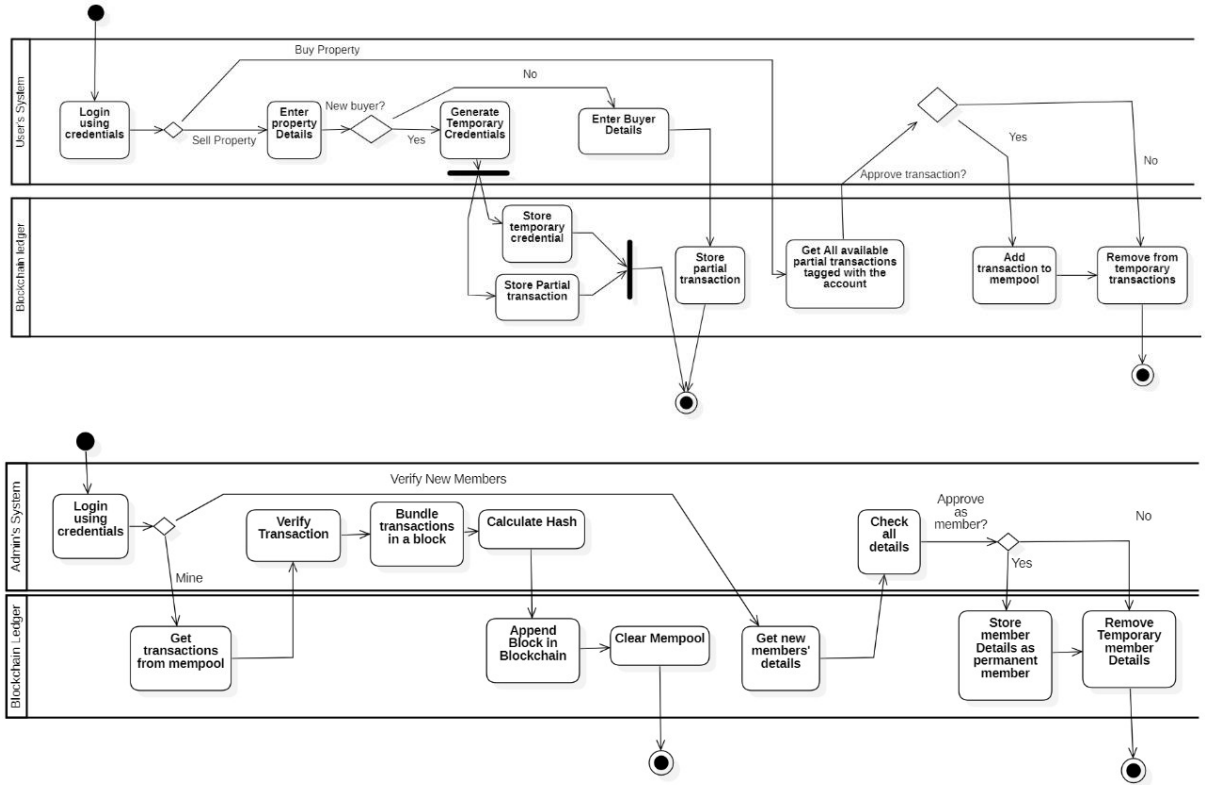
Consensus: Process of verification and validation in a blockchain transaction is known as consensus. Here, we are using proof of authority as consensus mechanism, since here only admin has right to verify and add transaction detail to the blockchain.

Immutable – Blockchain are immutable which means the information that once being stored in blockchain cannot be changed or altered. If changed, then the complete chain would become invalid because of change in hash value of the block and previous Hash value. And since, it is distributed and not centralized, fault can easily be found and handled.

Appendix B: Analysis Model






User's System
Buy Property
No
Login using credentials
Sell Property
Enter property Details
New buyer?
Yes
Generate Temporary Credentials
Enter Buyer Details
Approve transaction?
Yes
No
Blockchain ledger
Store temporary credential
Store Partial transaction
Store partial transaction
Get All available partial transactions tagged with the account
Add transaction to mempool
Remove from temporary transactions

Admin's System
Verify New Members
Login using credentials
Mine
Verify Transaction
Bundle transactions in a block
Calculate Hash
Check all details
Approve as member?
No
Yes
Blockchain Ledger
Get transactions from mempool
Append Block in Blockchain
Clear Mempool
Get new members' details
Store member Details as permanent member
Remove Temporary member Details