**Quad chart diagram**

|  |  |
| --- | --- |
| **Use Case: RxBlock**  Problem statement: -  A decentralized blockchain based solution for medical prescriptions. | **Issues with existing centralized model:**   1. No robust mechanism to validate the authenticity of the medical prescriptions. Fake prescriptions cause rampant issuance of medical drugs. 2. No reliable data of overprescribed individuals. 3. Absence of age-wise data of people who have been victims of medical drug abuse. 4. Lack of a trustworthy system to safely store and retrieve health records (prescriptions) of a patient. 5. Absence of a method to record payment transactions. |
| **Proposed blockchain based solution:**   1. The blockchain method validates the prescription by validating the doctor’s and patient’s details as mentioned on the prescription and as stored in the database using smart contract functions. 2. Prescriptions of the patients are securely stored. 3. A reliable payment mechanism is enabled between the patient and the pharmacy. 4. The transactions are recorded on the blockchain for enforcing trust and for later use. 5. The data gathered from this decentralized methodology can help the agencies for analysis and to combat medical drug abuse thusly. | **Benefits:**   1. Provides a trustworthy, decentralized, and secure method for patients to access their prescriptions. 2. Saves time and effort to verify and validate the prescriptions. 3. A smooth payment service between the patient and the pharmacist. 4. This methodology has the potential to develop a token-based patient to pharmacist business model. 5. The data gathered of fake prescriptions using this solution, will be very beneficial to tackle medical drug abuse. 6. Creative solution-oriented model for the pharmaceutical industry. 7. This solution offers better experience to doctors, patients, and pharmacy. |

**Use case diagram**

A picture containing text, metalware, coil spring

Description automatically generated

**Sequence diagram**

Graphical user interface, application

Description automatically generated

**Contract diagram**

|  |
| --- |
| **RxChain** |
| address Doc;  address payable Pharm;  address payable Pat;  bool isvalid=false;  counter private PresID;  struct PresDetails{  string PresDate;  address Pat;  address Doc;  string[] medicines;}  struct Patient { uint256 age;  string name;  string email;  address payable Pat;  }  struct Doctor{  uint256 age;  string name;  string email;  //clinic or hospital name  string clinic\_name;  address Doc;}  struct Pharmacy{  uint256 age;  string name;  string email;  string store\_address;  address payable Pharm;}  mapping(uint256=>PresDetails) public prescription;  mapping (address=>Patient) public patient\_membership;  mapping (address=>Doctor) public doctor\_membership;  mapping (address=>Pharmacist) public pharmacy\_membership;  mapping(address => bool) public genuine\_doctors;  Doctor[] public doctors\_list;  Patient[] public patients\_list;  Pharmacy[] public pharmacy\_list;  PresDetails[] public prescription\_list;  mapping (address=>uint256) private deposits; |
| modifier onlyDoctor  modifier onlyPatient  modifier onlyPharmacy  modifier DoctorListed(address \_doc)  modifier DoctorAlreadyListed(address \_doc) |
| **//events** |
| function Doctor\_Register(address doc\_new,string memory \_name, uint256 age,string memory \_email,string memory \_clinic\_name) public DoctorListed(doc\_new) returns(bool)  function Doctor\_Unregister(address doc\_new) onlyDoctor public DoctorAlreadyListed(doc\_new) returns(bool)    function Patient\_Register(address payable pat\_new,string memory \_name, uint256 age,string memory \_email) public returns(bool)  function Patient\_Unregister(address payable pat\_new) public returns(bool)  function Pharmacy\_Register(address payable pharm\_new,string memory \_name, uint256 age,string memory \_email,string memory \_store\_address) public onlyPharmacy returns(bool)  function Pharmacy\_Unregister(address payable pharm\_new) public onlyPharmacy returns(bool)  function Prescription\_Register(address payable \_Pat, address \_Doc, string memory \_PresDate,string[] memory medicines) public onlyDoctor returns(bool)    function validate\_prescription(address payable \_pat, address \_doc, uint pres\_id) public onlyPharmacy returns (bool) {    //payment functions |