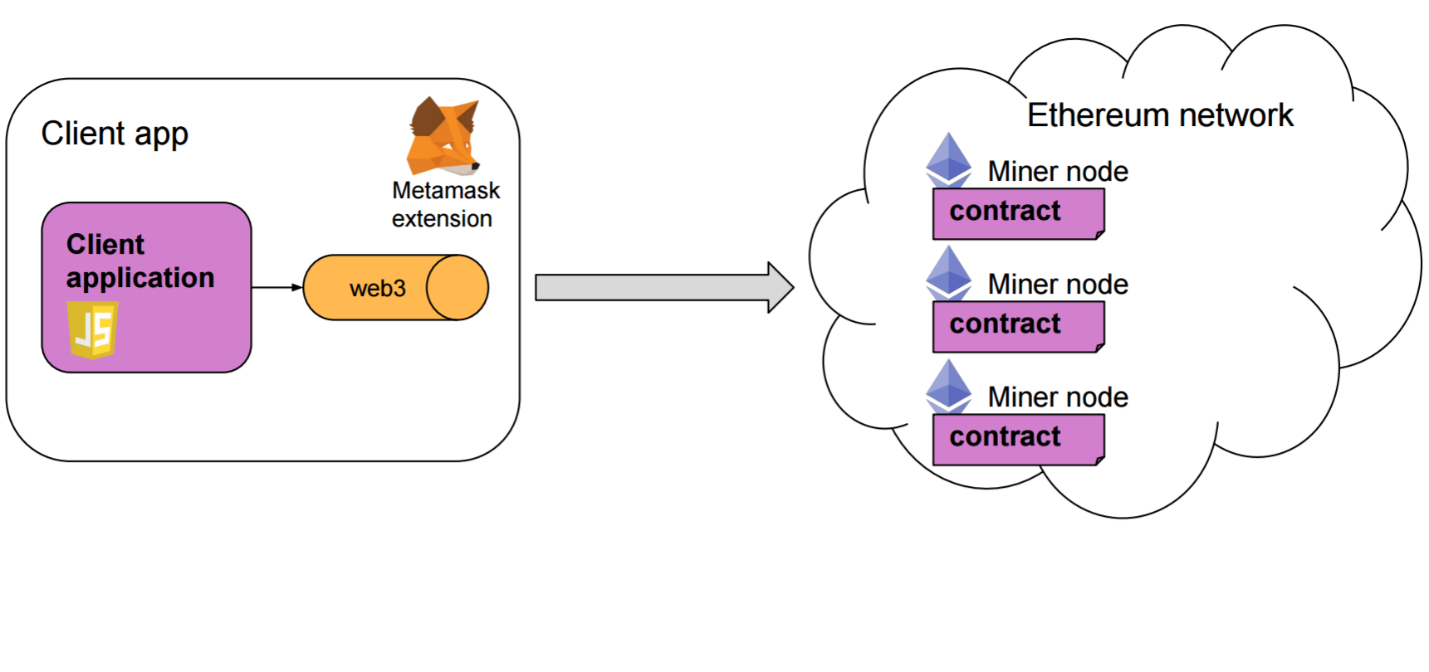
**SOFTWARE DESIGN**

**FunixPricingChain**

**1. Architecture**

*The diagram shows the achitecture of FunixPricingChain:*

**

Images

Hashes

Store Images

*FunixPricingChain*

*Smart contract*

IPFS

**2. Application Design**

|  |
| --- |
| **Contract Main** |
| struct IParticipant {  uint participantID;  string name;  string email;  uint deviation ;  uint deviationNew;  uint sessionNumber;  address addr;  }  address public admin;  mapping( address=>IParticipant) public participants;  IParticipant [10] public Iparticipants;  struct Session{  address addr;  uint totalPrice;  uint totalPercent;  uint proposedPrice;  uint lastPrice;  address [] participantsOfSes;  mapping(uint =>uint) priceOfAPar; // map from ID of participant to their latest price  mapping(address => uint []) parPrices;  uint [] givenPrices;  uint status; // State{ CREATED, STARTED, STOPPED, CLOSED }  }  mapping (address=>Session) detailSession;  address [] public sessions;  uint public count = 0; |
| *event Register(string name, string email);*  *event UpdateInfo(string name, string email);*  *event ProposedPrice(uint proposedPrice);*  *event CloseSession( uint lastPrice);* |
| *Functions:*   * constructor(){ admin = msg.sender; } * function addSession(address session) public {} * function getSessions() public view returns(address [] memory, uint){} * function getSessionNum() public view returns(uint){} * function getAdminInfo() public returns(address){} * function register( string memory \_name, string memory \_email) public{} * function update(string memory \_newName, string memory \_newEmail) public{} * function statusSession(address \_session,uint \_status) public {} * function price(address \_session, uint \_price) public{} * function getProposedPrice(address \_session)public returns(uint){} * function calculateLastPrice(address \_addr)public {} * function getLastPrice(address \_addr) external returns(uint){} * function getListOfPar(address \_session)public returns(address [] memory){} * function getGivenPrices(address \_session) public  returns(uint [] memory){} * function getProfileId(uint \_id) view public returns(uint ,address,string memory,string memory,uint,uint){} * function getProfileAdd(address \_addr) view public returns(uint ,address,string memory,string memory,uint,uint){} * function getTotalParNum() public view returns(uint){} * function getStatusSession(address \_session) public returns(uint){} |
| *modifier onlyAdmin(){ }* |

|  |
| --- |
| **Contract Session** |
| *Variables:*  address public mainContract;  Main MainContract;  address public admin;  struct Item{  uint itemID ;  string itemName;  string description;  }    Item public item;  uint public proposedPrice;  uint public lastPrice;  enum State{ CREATED, STARTED, STOPPED, CLOSED }  State public state = State.CREATED;  address [] public participants;  uint [] public givenPrices;  string [] public imageHashes;  string public imageHash; |
| *Event FinalPrice()*  *Event ProductionUpdate()* |
| *Functions:*   * *constructor(address \_mainContract, string memory \_itemName, string memory \_description) public {}* * *function setIdSession() public{}* * *function getProductInfo() view public returns(uint,string memory,string memory ){}* * *function setHash(string memory \_imageHash) public onlyAdmin{ }* * *function getImages() public view returns(string [] memory ){ }* * *function startSession() public inState(State.CREATED) onlyAdmin{}* * *function updateProductInfo(uint \_newItemId, string memory \_newName, string memory \_newDescription) public onlyAdmin{}* * *function getProposedPrice() public onlyAdmin inState(State.STARTED) returns(uint){}* * *function stopSession() public onlyAdmin inState(State.STARTED){}* * *function getPriceAndClose() public inState(State.STOPPED) returns(uint){}* * *function getListOfPar() public onlyAdmin returns(address[] memory){}* * *function getGivenPrices() public onlyAdmin {}* * *function getStatusSession()public view returns(State){}* |
| *modifier onlyAdmin(){ require msg.sender == admin}*  *modifier inState(State \_state){*  *require(state == \_state,'Invalid state');*  *\_;* |

**3. Function details**

* 1. **Main Contract**

3.1.1 constructor() public {}

* *Function name: constructor*
* *Function purpose : constructor function. Set admin= msg.sender*
* *List of parameters : none*
* *Return type:none*

3.1.2 register( string memory \_name, string memory \_email) public {}

* *Function name: register*
* *Function purpose : particpants register their information to create accounts*
* *List of parameters : string \_name, string \_email*
* *Return type: none*

3.1.3 update(string memory \_newName, string memory \_newEmail) public {}

* *Function name: update*
* *Function purpose : participants update their information of name ,email*
* *List of parameters :* string memory \_newName, string memory \_newEmail
* *Return type: none*

3.1.4 price(address \_session, uint \_price) public{}

* *Function name: price*
* *Function purpose : participants give their prices for products on each session*
* *List of parameters : address \_session, uint \_price*
* *Return type: none*

3.1.5 addSession (address session) public{}

* *Function name: addSession*
* *Function purpose : to store all session contract instances*
* *List of parameters : address session*
* *Return type: none*

3.1.6 getSessions public view returns(address [] memory, uint){}

* *Function name:* getSessions
* *Function purpose : get address of Sessions created*
* *List of parameters : none*
* *Return type: array of address and the number of sessions*

3.1.7 getSessionNum() public view returns(uint){}

* *Function name:* getSessionNum
* *Function purpose :return the number of sessions created*
* *List of parameters : none*
* *Return type: uint*

3.1.8 getAdminInfo() public returns(address){}

* *Function name:* getAdminInfo
* *Function purpose : return admin account*
* *List of parameters : address*
* *Return type: address*
  + 1. getProposedPrice(address \_session)public returns(uint){
* *Function name:* getProposedPrice
* *Function purpose : get proposed price of each session*
* *List of parameters : none*
* *Return type: uint*

3.1.10 calculateLastPrice(address \_session)public {}

* *Function name :* calculateLastPrice
* *Function purpose : calculate last price of session and update deviation, number of sessions of participants*
* *List of parameters :* address \_session
* *Return type: none*

3.1.11 getLastPrice(address \_session) public returns(uint){}

* *Function name:* getLastPrice
* *Function purpose : get last price of session*
* *List of parameters :* address \_session
* *Return type: address*

3.1.12 getListOfPar(address \_session) public returns(address [] memory){}

* *Function name:* getListOfPar
* *Function purpose : return list of participants’s address*
* *List of parameters :* address \_session
* *Return type: array of address*

3.1.13 getGivenPrices(address \_session) public returns(uint [] memory){}

* *Function name:* getGivenPrices
* *Function purpose : return all prices that wer given for a session*
* *List of parameters :* address \_session
* *Return type: array of uint*

3.1.14 getProfileId(uint \_id) view public returns(uint ,address,string memory,string memory,uint,uint){ }

* *Function name:* getProfileId
* *Function purpose : get profile of a participant by his id*
* *List of parameters :* uint \_id
* *Return type:* uint ,address,string memory,string memory,uint,uint

3.1.15 getProfileAdd(address \_addr) view public returns(uint ,address,string memory,string memory,uint,uint){ }

* *Function name:* getProfileId
* *Function purpose : get profile of a participant by his address*
* *List of parameters :* address \_addr
* *Return type:* uint ,address,string memory,string memory,uint,uint

3.1.16 getTotalParNum() public view returns(uint){

* *Function name:* getTotalParNum
* *Function purpose : get the total number of participant that registered*
* *List of parameters :* none
* *Return type: uint*

3.1.17 getStatusSession(address \_session) public onlyAdmin returns(uint){

* *Function name* getStatusSession
* *Function purpose : return the status of a session*
* *List of parameters :* address \_session
* *Return type: uint*

**3.2 Session Contract**

3.2.1 constructor(address \_mainContract,address \_admin, string memory \_itemName, string memory \_description) public {

* *Function name:* constructor
* *Function purpose : create a new session, set admin address for the contract and push the session contract address to an array to store in Main contract*
* *List of parameters :* address \_mainContract,address \_admin, string memory \_itemName, string memory \_description
* *Return type: none*

3.2.2 startSession() public inState(State.CREATED) onlyAdmin{

* *Function name:* startSession
* *Function purpose : for admin to start a pricing session*
* *List of parameters : none*
* *Return type: none*

3.2.3 updateProductInfo(uint \_newItemId, string memory \_newName, string memory \_newDescription) public onlyAdmin{}

* *Function name: updateProductInfo*
* *Function purpose : for admin to update information of products*
* *List of parameters :* uint \_newItemId, string memory \_newName, string memory \_newDescription
* *Return type:none*

3.2.4 getProductInfo() view public returns(uint,string memory,string memory ){}

* *Function name:* getProductInfo
* *Function purpose : get information of a session*
* *List of parameters : none*
* *Return type:* uint,string,string

3.2.5 setIdSession() public{}

* *Function name:* setIdSession
* *Function purpose : to set ID for the session automatically*
* *List of parameters : none*
* *Return type: none*

3.2.6 Start() public onlyAdmin {}

* *Function name: start*
* *Function purpose : for admin to start a pricing session*
* *List of parameters : none*
* *Return type*

3.2.7 setHash(string memory \_imageHash) public onlyAdmin{}

* *Function name:* setHash
* *Function purpose : get and push hashes of images to an array*
* *List of parameters :* string memory \_imageHash
* *Return type: none*

3.2.8 getImages() public view returns(string [] memory ){}

* *Function name:* getImages
* *Function purpose : to get hashes of images*
* *List of parameters : none*
* *Return type:array of strings*

3.2.9 getProposedPrice() public inState(State.STARTED) onlyAdmin returns(uint){}

* *Function name:* getProposedPrice
* *Function purpose : to get proposed price of a session*
* *List of parameters : none*
* *Return type: uint*

3.2.10 stopSession() public onlyAdmin inState(State.STARTED){}

* *Function name:* stopSession
* *Function purpose : to stop session , calculate last price and update deviation, number of session of participant*
* *List of parameters : none*
* *Return type: none*

3.2.11 getPriceAndClose() public inState(State.STOPPED) onlyAdmin returns(uint){

* *Function name:* getPriceAndClose
* *Function purpose : to get last price result and close the session*
* *List of parameters : none*
* *Return type: uint*

3.2.12 getListOfPar()public onlyAdmin returns(address[] memory){

* *Function name:* getListOfPar
* *Function purpose : to get the list of participants that joined the session*
* *List of parameters : none*
* *Return type: array of address*

3.2.13 getGivenPrices()public onlyAdmin{

* *Function name:* getGivenPrices
* *Function purpose : to get all prices that were given for a session*
* *List of parameters : none*
* *Return type: array of uint*

3.2.214 getStatusSession()public view returns(State){

* *Function name:* getStatusSession
* *Function purpose to get status of a session*
* *List of parameters : none*
* *Return type: State*