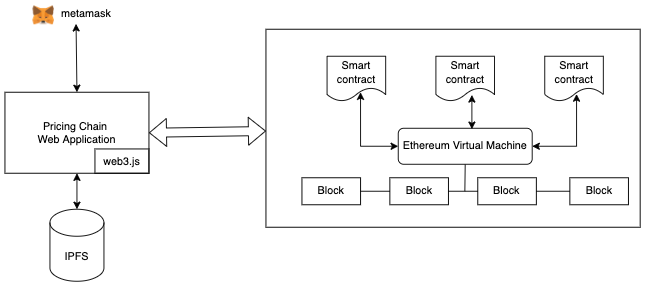
**SOFTWARE DESIGN**

**FunixPricingChain**

**1. Architecture**

**

**2. Application Design**

|  |
| --- |
| *Main Contract* |
| *+ sessionList*  *+ participantList* |
| *initSession*  *closeSession*  *setFinalPrice*  *getSessions*  *submitPrice*  *getParticipantList*  *register*  *signIn*  *getParticipantDetail*  *updateParticipantInfo* |

|  |
| --- |
| *Session* |
| *+ parentContract*  *+ name + description*  *+ images*  *+ initTime*  *+ timeout*  *+ reward*  *+ finalPrice*  *+ proposeList* |
| *constructor*  *close*  *update*  *setFinalPrice*  *submitPrice*  *calculateProposePrice* |

**3. Functional Design**

3.1 Main contract

3.1.1 Init session

* *Function name: initSession*
* *Function purpose : init new session*
* *List of parameters (name: string, description: string, images: string[], timeout: uint256)*
* *Return type: void*

3.1.2 Close session

* *Function name: closeSession*
* *Function purpose : close a ongoing session*
* *List of parameters (sessionIndex: uint256)*
* *Return type: void*

3.1.3 Set final price

* *Function name: setFinalPrice*
* *Function purpose : set final price for a session*
* *List of parameters (sessionIndex: uint256, finalPrice: uint)*
* *Return type: void*

3.1.4 Get Sessions

* *Function name: getSessions*
* *Function purpose : get list of session*
* *List of parameters: none*

*Return type: Session[]*

3.1.5 Submit price

* *Function name: submitPrice*
* *Function purpose : submit propose price*
* *List of parameters: (sessionIndex: uint256, finalPrice: uint)*
* *Return type: void*

3.1.6 Get participant list

* *Function name: getParticipantList*
* *Function purpose : get list of participant for admin*
* *List of parameters: none*

*- Return type: Participant[]*

3.1.7 Sign in

* *Function name: signIn*
* *Function purpose : check user is registered of not*
* *List of parameters: none*
* *Return type: boolean*

3.1.8 Register

* *Function name: register*
* *Function purpose : register new participant*
* *List of parameters: emai: string, name: string*
* *Return type: void*

3.1.9 Get participant detail

* *Function name: getParticipantDetail*
* *Function purpose : get detail of one participant*
* *List of parameters: participantAddr: address*
* *Return type: Participant*

3.1.10 Update participant info

* *Function name: updateParticipantInfo*
* *Function purpose : update email and name of participant*
* *List of parameters: participantAddr: address, email: string, name: string*

3.2 Session contract

3.2.1 Constructor

* *Function name: constructor*
* *Function purpose : iinit new session with data*
* *List of parameters (parentContract: address, name: string,description: string, images: string[], timeout: uint256, reward: uint256)*
* *Return type: void*

3.2.2 Close

* *Function name: close*
* *Function purpose : close current session*
* *List of parameters: none*
* *Return type: void*

3.2.3 Set final price

* *Function name: setFinalPrice*
* *Function purpose : set final price for current session*
* *List of parameters: finalPrice: uint256*

*- Return type: void*

3.2.4 Submit propose price

* *Function name: submitPrice*
* *Function purpose : store propose price for current session*
* *List of parameters: participantAddr: address, price: uint256*
* *Return type: void*

3.2.5 Calculate propose price

* *Function name: calculateProposePrice*
* *Function purpose : calculate and update propose price for current session*
* *List of parameters: none*

*- Return type: void*