**SOFTWARE REQUIREMENTS SPECIFICATION**

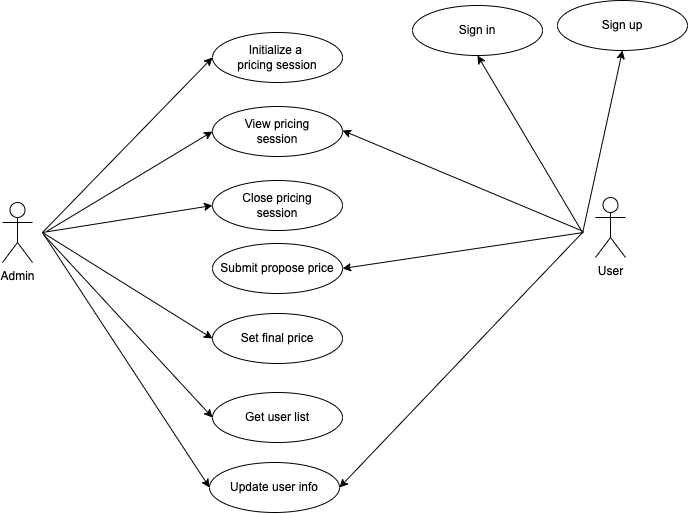
**FunixPricingChain**

**1. Product overview**

* Product name: FunixPricingChain – an online product pricing application.
* Product description:
  + Allow participant (admin) to post their products through sessions.
  + Other participants will join the session then submit the price that they think suitable for the product.
  + At the end of the session, admin can collect all the price then calculate the new price for the product base on algorithm, and participants can use the result price to compare with their experiences.
* Users – who will use the application:

|  |  |  |
| --- | --- | --- |
| *No.* | *User* | *Description* |
| *1.* | *Admin* | *The one who own the products, and want to get the price of these products* |
| *2.* | *Participants* | *The one who has experiences and knowledge about lot of products* |

* The benefit to users:
  + For admin: can get price of rare products
  + For participants: know the price of the products at the end, and can get reward if their contribute is almost correct.
* Functional Diagram:



**2. Functional Requirements**

2.1 . Initiate A Pricing Session

* Function name: Initiate A Pricing Session
* Function description: the function is to initiate a new session for pricing product
* User to use: Only admin will use the function
* Function Inputs:

|  |  |
| --- | --- |
| **Input** | **Requirement** |
| Name | String, required |
| Description | String, optional |
| Images | Array of String, optional |
| Timeout | Unsigned integer, optional |

* Function rule: only name is required, others fields are optional.
* Function result: a pricing session is initiated.

2.2 . View Pricing Session

* Function name: View Pricing Session
* Function description: the function is to let all user to get current active pricing session and the detail of the product of that session.
* User to use: Admin and participant.
* Function Inputs: no input.
* Function rule: no rule.
* Function result: detail of all session will be display

2.3 . Close Pricing Session

* Function name: Close Pricing Session
* Function description: the function is to close an existed pricing session.
* User to use: Only admin will use the function
* Function Inputs:

|  |  |
| --- | --- |
| **Input** | **Requirement** |
| Session index | Unsigned integer, required |

* Function rule: session index required.

Function result: a pricing session with correct index will be closed, and proposed price for the product will be calculated.

2.4 . Submit pricing

* Function name: Submit pricing
* Function description: the function is to submit pricing for a pricing session
* User to use: Only participant that is registered will use the function
* Function Inputs:

|  |  |
| --- | --- |
| **Input** | **Requirement** |
| Session index | Unsigned integer, required |
| Price | Unsigned integer, required |

* Function rule: session index and price is required. Price is > 0.
* Function result: a price will be submit and stored to specific session product.

2.5 . Set final price

* Function name: Set final price
* Function description: the function is to manage (CRUD) all product
* User to use: Only admin will use the function
* Function Inputs:

|  |  |
| --- | --- |
| Session index | Unsigned integer, required |
| Price | Unsigned integer, required |

* Function rule: session index and price is required. Price is > 0.
* Function result: final price will be stored in a session with index

2.6 . Get user list

* Function name: Get user list
* Function description: the function is to get list of user
* User to use: Only admin will use the function
* Function Inputs: no input
* Function rule: no rule.
* Function result: a list of user will be return and display.

2.7 . Update user info

* Function name: Update user info
* Function description: the function is to update user info
* User to use: Admin and participants.
* Function input:

|  |  |
| --- | --- |
| **Input** | **Requirement** |
| User address | Address, required |
| Email | String, required |
| Name | String, required |

* Function rule: all fields are required.
* Function result: user info (only email and name) will be updated.

2.8 . Sign in

* Function name: Sign in
* Function description: the function is to sign in
* User to use: only participant
* Function input:

|  |  |
| --- | --- |
| **Input** | **Requirement** |
| User address | Address, required |

* Function rule: user address is required
* Function result: if user is registered, user will be redirect to session page, else he is redirected to register page.

2.9 . Sign up

* Function name: Sign up
* Function description: the function is to sign up new new user
* User to use: participants
* Function input:

|  |  |
| --- | --- |
| **Input** | **Requirement** |
| Address | Address, required |
| Email | String, required |
| Name | String, required |

* Function rule: address, email and name are required
* Function result: new participant will be created if address is not existed.

**3. Non-Functional Requirements**

* *Web browser coverage: allow only for modern browser such as Chrome, Firefox, Safari, Microsoft Edge which can add Metamask extension*

1. **Product Upgrades**

- Rewards for participants: after a session is closed, some participants can get rewards if their predict price is almost correct to proposed price.

- Mobile app

- Responsive website

- Update algorithm to remove admin manually updating final price.