

Release Document For Public Blockchain

RELEASE DOCUMENT ON RMIT PUBLIC BLOCKCHAIN

Date 20-September-2023

Table of contents

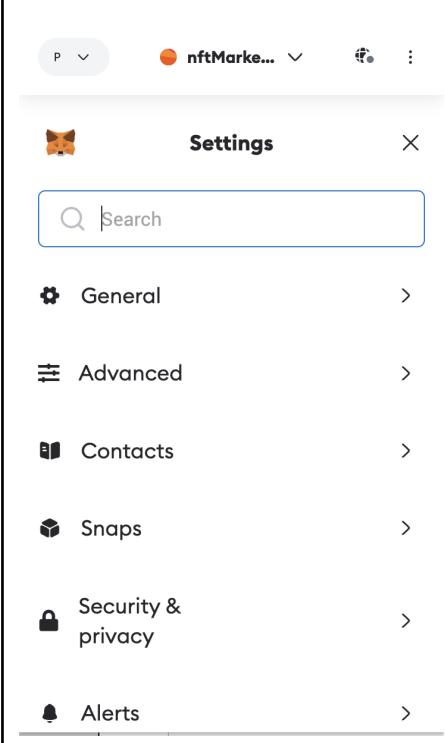
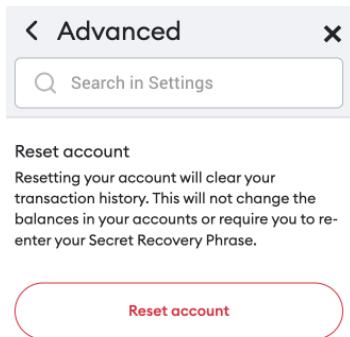
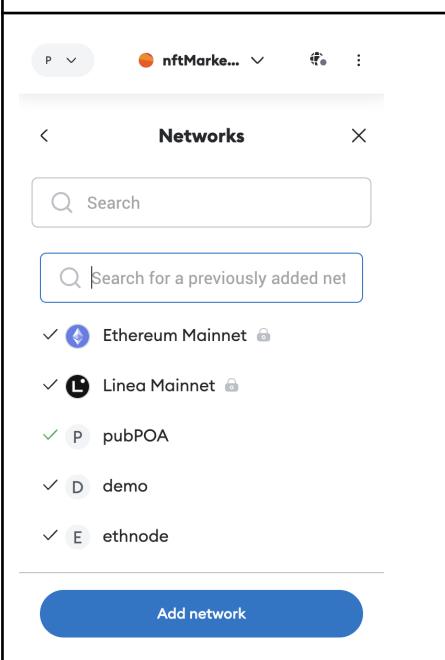
1 Cloud Infrastructure & Setup Guide.....	5
1.1 RMIT Public Blockchain.....	5
1.1.1 Blockchain nodes network diagram.....	5
1.1.2 Blockchain setup information.....	6
1.1.2.1 Genesis Block.....	6
1.1.2.2 Sealer nodes.....	6
1.1.2.3 Data nodes.....	7
1.2 Public blockchain network on Viettel Cloud.....	7
1.2.1 Deploy blockchain nodes.....	8
1.2.1.1 Deploy Sealer Nodes.....	8
1.2.1.2 Deploy Data Nodes.....	9
1.2.1.3 Deploy Load Balancer.....	9
1.2.2 Setup blockchain explorer.....	15
1.2.3 Backend for Applications.....	16
1.2.3.1 Backend for Faucet.....	16
1.2.3.2 File server.....	16
1.2.3.3 Database MongoDB.....	17
1.2.3.4 Log service.....	18
1.2.4 Setup Faucet Applications.....	19
1.2.4.1 Setup the faucet accounts.....	19
1.2.4.2 Setup Faucet App.....	20
1.2.5 Setup NFT marketplace.....	20
1.2.5.1 Deploy NFT marketplace smart contracts.....	20
1.2.5.2 Setup NFT marketplace Frontend.....	21
1.2.6 Setup Fractional NFT Application.....	22
1.2.6.1 Deploy Fractional NFT smartcontracts.....	22
1.2.6.2 Deploy fractional NFT Frontend app.....	23
1.2.7 Setup FT generate app.....	24
1.2.8 Setup ERC20 Votes App.....	25
1.2.9 Setup NFT ticket App.....	25
1.3 Guideline to control the blockchain network.....	26
1.3.1 Important notes.....	26
1.3.2 Bring the blockchain into private.....	27
1.3.3 Turn on and off blockchain node's load balancer.....	27
2 Application Usage Guide.....	29
2.1 Metamask.....	29
2.2 Faucet Application.....	30
2.2.1 Faucet Home page.....	31
2.2.2 Faucet Application Page.....	33
2.2.3 User Management.....	34

2.2.3.1 User details page.....	34
2.2.3.2 Create Account.....	36
2.2.4 Quota Request Management.....	37
2.2.4.1 Workflow.....	37
2.2.4.2 Display Panel.....	38
2.2.5 My Account.....	38
2.3 NFT Marketplace.....	39
2.3.1 Workflow.....	40
2.3.2 Marketplace.....	40
2.3.3 Wallet.....	41
2.3.4 Your NFTs.....	42
2.3.5 Create NFT.....	43
2.4 Fractional NFT (FNFT).....	44
2.4.1 Workflow.....	44
2.4.2 Fraction Vault.....	45
2.4.3 Wallet.....	46
2.4.4 Your NFTs.....	47
2.4.5 Create NFT.....	50
2.5 ERC20 FT generator.....	51
2.5.1 Home page.....	51
2.5.2 Create/Deploy new smart contract.....	52
2.5.3 Mint.....	53
2.5.4 Setup.....	54
2.5.5 Buy Fungible Tokens.....	55
2.6 ERC20 Votes Campaign.....	56
2.6.1 Home pages.....	56
2.6.2 Create new Vote Campaign.....	57
2.6.3 Transfer Vote.....	58
2.6.4 Vote.....	59
2.7 ERC721 NFT Ticket.....	60
2.7.1 NFT ticket setup page.....	60
2.7.2 Buy ticket page.....	61

2 Application Usage Guide

2.1 Metamask

To use Dapp on web page, users are required to install the Metamask wallet plugin on **chrome/brave**. Then configure this wallet to connect to your data node as below instructions. [Ref link](#)

	<p>Reset Account: to clear all previous cache states. This is a very important step, otherwise, you may not be able to submit any tx because of the stale state. Each time you change the network/reset/turn on the node you need to do that. On metamask go to Settings/Advanced then Click on Reset Account</p>  <p>Reset account Resetting your account will clear your transaction history. This will not change the balances in your accounts or require you to re-enter your Secret Recovery Phrase.</p> <p>Reset account</p> <p>Advanced gas controls Select this to show gas price and limit controls directly on the send and confirm screens.</p>
	<p>Make sure the POA node on your node is properly running (block number is increased)</p> <p>Setup network: On metamask go to Settings/Network. Click on Add Network.</p> <ul style="list-style-type: none">• RPC URL: https://mainnet.rmit.site• ChainID: 202306• Currency symbol: RMLC• Blockexplorer: https://explorer.rmit.site

Add a network

Network name
pubPOA

New RPC URL
https://mainnet.rmit.site

Chain ID ⓘ
202306

Currency symbol
RMLC

Ticker symbol verification data is currently unavailable, make sure that the symbol you have entered is correct. It will impact the conversion rates that you see for this network

Block explorer URL (Optional)
https://explorer.rmit.site

Cancel **Save**

Note: Metamask checks the Currency symbol on [Coingecko](#) or [Coinmarketcap](#) to fetch the fiat exchange rate. If you don't want to display the Fiat, change the currency symbol to the one which was not available on these 2 sites

2.2 Faucet Application

- The faucet application is at the link: <https://rmit.site/>
- Users are required to have a valid account to access the faucet. There are 3 types of users: Super Admin, Admin, Normal User, and Guest
 - Only the Super Admin and Admin user have permission to access management pages to manage the faucet
 - The Normal User can only access the faucet page to request the RMLC from the faucet. The number of RMLC and time between the requests are set up by Admin or Super Admin user
 - Guest may have permission to request the RMLC coin depending on the system setup by Admin or Super Admin

2.2.1 Faucet Home page

Faucet RMIT

Login

Earn Free RMLC

Are you looking for an easy-to-use faucet for free RMLC? What are you waiting for? Grab your free coins today!

Name *

Email *

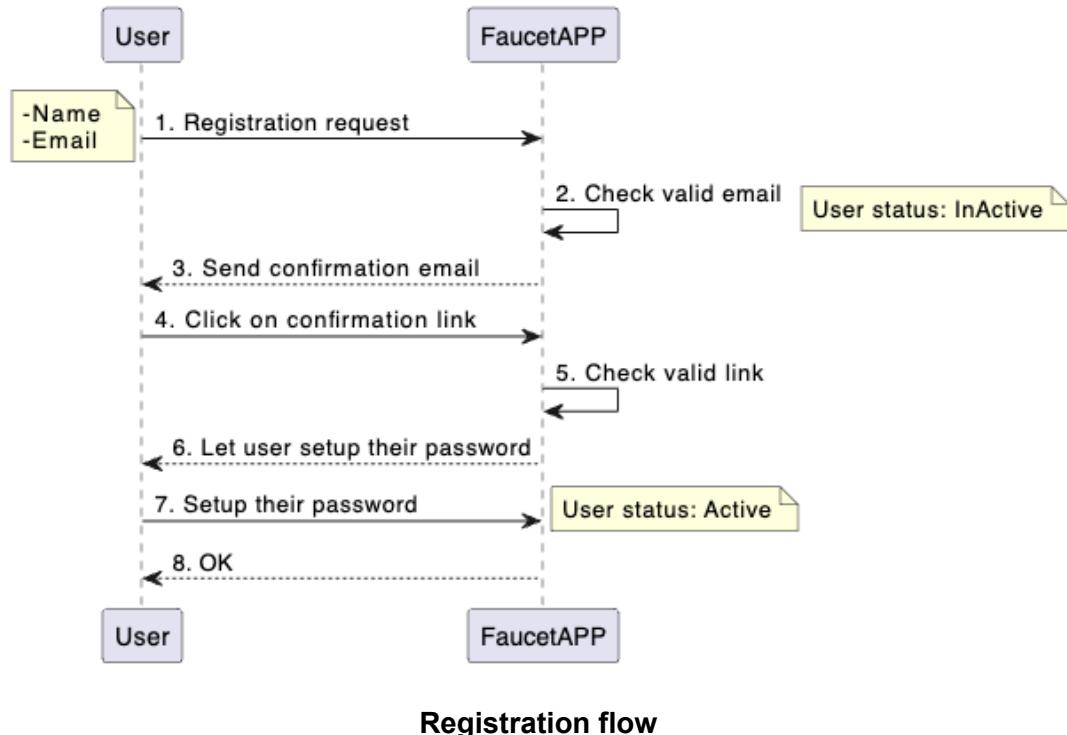
I agree that by using this service, my personal data of name and email address will be stored in a data server in Vietnam to help RMIT University Vietnam record the users of Faucet in the RMIT Public Blockchain.

Register

Faucet home page

This is the default page when non-login users visit the URL: <https://rmit.site/>. New users can input their name and email to register the account with the faucet application and then hit the **Register** button to make the user registration request. This server only accepts the email input from **rmit.edu.vn** and **rmit.edu.au**.

This action will register the **normal user account** with the faucet. Users need to check their mailboxes to complete the onboarding process. If users have not completed this process, their account status is InActive until they finish the onboarding.

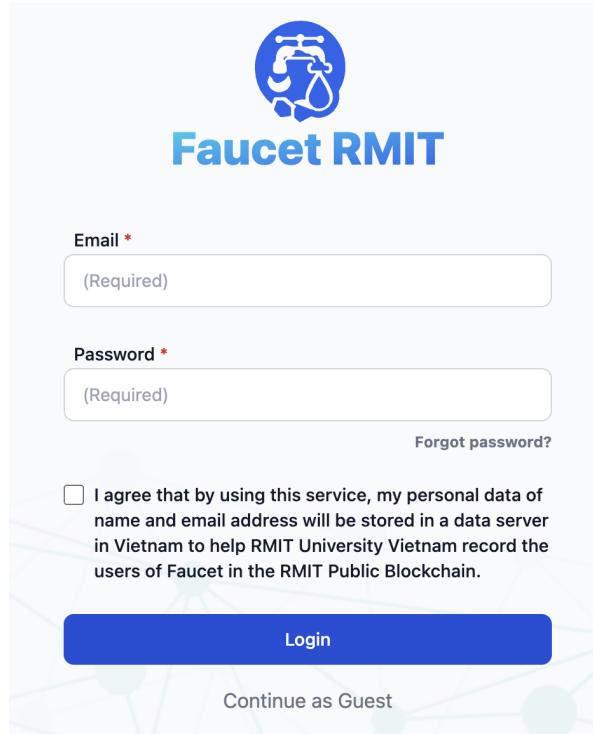


Note that, the OTP link on 3 is last for 5 min. If the time has expired, the user needs

to register for a new session or the admin user needs to manually send them the new OTP link through the mail. When the user is on the status of **active**, they can log in and request the faucet RMLC to use with Dapps. This credential is also used to upload the images to the server when creating the NFT token.

- **Login**

In order to log in, please hit the button **login** at the top right



The screenshot shows the 'Faucet RMIT' login interface. At the top is a logo of a faucet with water droplets. Below it is the text 'Faucet RMIT'. The form contains two input fields: 'Email *' and '(Required)' below it, and 'Password *' and '(Required)' below it. To the right of the password field is a 'Forgot password?' link. Below the fields is a checkbox with the text: 'I agree that by using this service, my personal data of name and email address will be stored in a data server in Vietnam to help RMIT University Vietnam record the users of Faucet in the RMIT Public Blockchain.' A large blue 'Login' button is centered at the bottom, and a 'Continue as Guest' link is located just below it.

Registration page

The user inputs their registered email and password to log in and then requests the RMLC from the faucet. The user also has an option to “**Continue as Guest**” to request the RMLC from the faucet, however, this one depends on the admin/super admin setup whether they want to open the faucet to the Guest users.

- **Continue as guest**

If the user chooses the option to continue as a guest, they also need to input their email (any email). The system will send them the One Time Login link through the mail so that they can use this link to access the faucet application

Login as Guest

Email *

I agree that by using this service, my personal data of name and email address will be stored in a data server in Vietnam to help RMIT University Vietnam record the users of Faucet in the RMIT Public Blockchain.

Verify

Continue as Guest

2.2.2 Faucet Application Page

After login the user can access this page to request the faucet RMLC

1 × **Faucet RMIT**

2 Super Admin

3

4

5

6

Request RMLC

Faucet application page

1. Faucet menu: The number of menus that the user can access depends on the role of the user
 - a. Admin/Super Admin can see the full menu as above in picture
 - b. Standard Users can only see and access **Faucet** and **My Account**
2. Login user account information
3. Connected Blockchain information: This is up to now blockchain status.
4. The faucet accounts information, this is the accounts that hold the faucet RMLC fund
5. The currently selected account that Metamask uses to connect this faucet page.
6. Users click on this button to request the RMLC into their selected account (5). The amount of RMLC/request is defined by the admin on the management dashboard.

2.2.3 User Management

- To access this, the user needs to be Admin or Super Admin
- Link: <https://rmit.site/app/admin/management>

The screenshot shows a table of users with the following columns: #, ID, NAME, USER NAME, ROLE, STATUS, REMAIN RMLC, RMLC/REQUEST, RATE LIMIT (MIN), and L REC. The table contains four rows of data.

#	ID	NAME	USER NAME	ROLE	STATUS	REMAIN RMLC	RMLC/REQUEST	RATE LIMIT (MIN)	L REC
1	e8ba15	Thinh Le	test@rmit.edu.vn	USER	INACTIVE	0.50	0.05	900000	N/A
2	e8b953	Anh Nguyen	anh.nguyen772@rmit.edu.vn	USER	ACTIVE	0.50	0.05	900000	N/A
3	7d7001	Viet	S3926719@rmit.edu.vn	USER	ACTIVE	0.50	0.05	900000	N/A
4	342f2a	Minh Nguyen	nguyen.minh244@rmit.edu.vn	USER	ACTIVE	0.50	0.05	900000	N/A

User management application page

1. Hamburger menu: Click here to see the full menus that this user can access
2. Create Accounts: This lets the super Admin/Admin user create a new user
3. Filter options: filter to find the users based on Role and Status categories.
4. Full-text search: quickly do the search for user information by text
5. The table displays the results of (3) and (4):
 - a. Admin users can click on each row to see the details of each user and manage them.

2.2.3.1 User details page

- When clicking on the row the user detail page will show:

The screenshot shows the user detail page for a user with ID e8ba15. The page includes fields for Full Name (2), Email (3), Phone (4), Roles (8), Status (9), and Two-factor authentication (10). It also shows RMLC requests (11), a quota history section (12), and a request history section (13). A large blue box at the bottom (15) displays a message: "Nothing to show".

User management user detail page

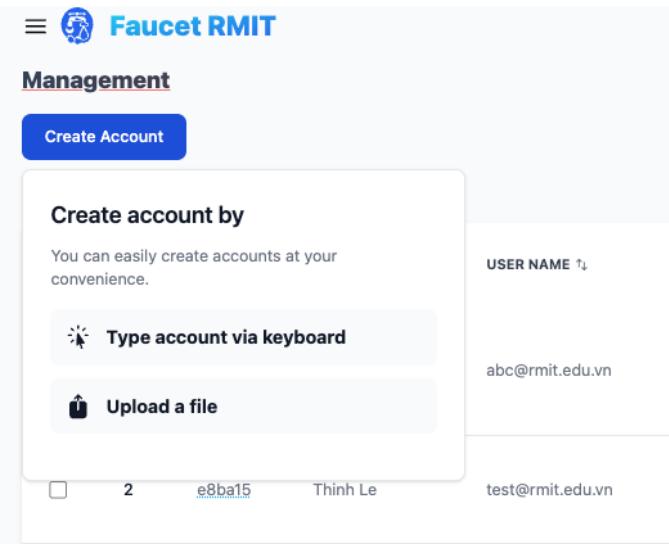
1. Account status: there are 4 status
 - a. INACTIVE: The Account was recently registered but have not been activated the account by the OTP link which has been sent to their mail yet
 - b. ACTIVE: Normal status, user can use this account as it is expected
 - c. BLOCKED: This account has been suspended by the admin. This can not be used to log in and request any RMLC anymore. In order to use this account, the admin need to ACTIVATE this account by (9)
 - d. DELETED: The account is marked as deleted. The user or Admin needs to recreate this account again to use
2. Account information:
 - a. Full name
 - b. Email
 - c. Phone
3. Total Request: the accumulated RMLC of all requests from this account.
4. Remain RMLC: (5) -(3)
5. Max RMLC request: The requests RMLC from this account will be accumulated (3) and then compared to this number to decide if this account can request more RMLC.
6. RMLC/request: This defines how many RMLC which faucet fund account will send to the requester each time of request
7. Raise limit: This defines how many minutes between 2 requests. After the first request, the user needs to wait for this time before making any more requests
8. Roles: This Is the User Role
 - a. USER: normal user/ standard user or end user, who will use this faucet to request the Rmit Lab Coin(RMLC) under configuration set by ADMIN
 - b. ADMIN: The admin user can do whatever a normal user can. Besides that, they also have permission to manage the system and change the configuration of other normal users and admin users (5)(6)(7)(8)
 - c. SUPER ADMIN: The admin user can do whatever an admin user can. Besides that, they also have permission to manage the system and change the configuration of other normal users, admin users, super admin users (5)(6)(7)(8)
9. Change status action: there are 2 actions
 - a. ACTIVE: Activate this user after clicking on the update account button
 - b. BLOCK: Block this user after clicking on the update account button. The blocked account can not login and request the RMLC from the faucet
10. 2FA setup: Enable two-factor authentication through the mail
11. Update status: Click on this button to submit all the configurations on (2)(3)(5)(6)(7)(8)(9)(10) to the server to change the account's configuration
12. Resend Activation Email: Click on this button to resend the activation mail. This button is only available on the account in status INACTIVE. If the account status is ACTIVE, this button is replaced by "Reset Password" to help a user reset their password. Note that, For security reasons, the activation link is only alive for 5 mins.

13. Delete Account: Click on this button to delete the account directly. Note that the system does not really delete this account, it just marks this account as DELETED. The end user can reuse the deleted account's email to register a new account
14. Block Account: Block the account directly.
15. Account history logs: To view all the request history of this account and all the account's quota RMLC changes are also logged here

2.2.3.2 Create Account

When user click on **Create Account** button on management page, there are 2 options that user can choose to create the account

- Type account via keyboard: Create the single accounts
- Upload a file : to mass create the account by upload the excel file



Create account menu

1. Type account via keyboard

Create single account

Users are required to input Full Name and Email. The email is used to activate the account. After clicking of Create account, the email from system will send to the mail of user to start the flow of onboarding.(see register flow)

2. Upload the file

The screenshot shows a web-based form titled "Management > Bulk Account Creation". The main section is titled "Bulk Account Creation". It contains a "CSV file * " label with a red asterisk, followed by a "Choose File" button and a text input field showing "No file chosen". Below this is a link "Click here to download CSV Template". At the bottom left is a "CSV configuration:" dropdown menu with "Include header" checked.

Create buck accounts

- User can download the CSV template from this form.
- Use excel to input the required information from this CSV
- Save this file as CSV
- Upload this file to server by clicking on "**Choose File**" button
- The system will auto create the account and start the onboarding flow for each user in the CSV sheet.

2.2.4 Quota Request Management

- Users are required to have role Admin/Super admin to access here
- To access this function **Menu/Quota Request Management**
- Link: <https://rmit.site/app/admin/management/quota-requests>

2.2.4.1 Workflow

This flow is used in case there are users who want to increase their **max total coin RMLC** can be requested. This is useful when users run out of RMLC quota. Here is expected flow

1. Users can visit their **Menu/My Account** then click on **Request more RMLC**.
2. The submitted requests will be available on the **Menu/Quota Request Management** panel. Only Admin/SuperAdmin can approve this request.
3. Admin/Super Admin visit **Menu/Quota Request Management** to approve/reject the request.
4. When the request is approved, the max total coin of requested users are **increased by 0.05 RMLC**
5. When the request is approved, it will disappear from this panel

2.2.4.2 Display Panel

The screenshot shows a web-based application interface for managing quota requests. At the top, there's a header with the logo 'Faucet RMIT' and a user name 'baoduong'. Below the header, the navigation path is 'Management > Quota requests'. The main title is 'Quota requests'. A status filter dropdown is set to 'All' (highlighted with a red box, labeled 1). To the right of the dropdown is a search bar with placeholder text 'Search ... (Optional)' and a 'Search' button (highlighted with a red box, labeled 2). The main content area displays a table with 5 rows of data. The columns are: #, REQUESTED BY, AMOUNT RMLC, STATUS, ACTIONS, UPDATED AT, and CREATED AT. Row 1: #1, baoduong, 0.05, PENDING, Approve (blue), Reject (red) (highlighted with a red box, labeled 3), 29/09/2023, 15:56:04, 29/09/2023, 15:56:04. Row 2: #2, baoduong, 0.05, APPROVED, by baoduong, 29/09/2023, 15:45:57, 29/09/2023, 15:45:27. Row 3: #3, Super Admin, 48, APPROVED, by Super Admin, 02/09/2023, 12:04:06, 02/09/2023, 12:04:06. Row 4: #4, Super Admin, 0.05, REJECTED, by Super Admin, 07/09/2023, 07:57:37, 27/08/2023, 16:25:47. Row 5: #5, Super Admin, 0.05, APPROVED, by Super Admin, 27/08/2023, 16:07:32, 27/08/2023, 16:07:16.

#	REQUESTED BY	AMOUNT RMLC	STATUS	ACTIONS	UPDATED AT	CREATED AT
1	baoduong	0.05	PENDING	Approve Reject	29/09/2023, 15:56:04	29/09/2023, 15:56:04
2	baoduong	0.05	APPROVED	by baoduong	29/09/2023, 15:45:57	29/09/2023, 15:45:27
3	Super Admin	48	APPROVED	by Super Admin	02/09/2023, 12:04:06	02/09/2023, 12:04:06
4	Super Admin	0.05	REJECTED	by Super Admin	07/09/2023, 07:57:37	27/08/2023, 16:25:47
5	Super Admin	0.05	APPROVED	by Super Admin	27/08/2023, 16:07:32	27/08/2023, 16:07:16

Quota request management

1. Request Status filter: To let user filter what he want to view
 - a. All : Display all requested items
 - b. APPROVED: Only approved ones are displayed
 - c. REJECTED: Only rejected ones are displayed
 - d. PENDING: Only pending ones are displayed
2. Search Action: Search requester by text
3. List view of request: List all items as the result of (1) and (2)
4. Request items: the request status and information. Admin user can approve or reject the pending request

2.2.5 My Account

- All User have their my account page to display the parameter they has been configured by admin or system
- To access this function **Menu/My Accounts**
- Link: <https://rmit.site/app/admin/my-account>

User Information:

- Name: [REDACTED]
- Email: [REDACTED]
- Phone: N/A
- Status: ACTIVE
- Roles: ADMIN (1)
- Two-factor authentication (2FA): Off

Quota Requests:

- Max coin/request: 1.00 RMLC
- Time between request: 60000 ms/request
- Max total coin: 50.05 RMLC
- Total Requested Coin: 7.25 RMLC
- Remain Coin: 42.80 RMLC (Request more RMLC (2))

History:

- Addresses (3)
- Quota History
- Request History

#	ADDRESS	ADDED AT
1	0xB09...870159	<input type="button" value="Copy"/> View on Block Explorer (Q)
		21/09/2023, 15:14:20

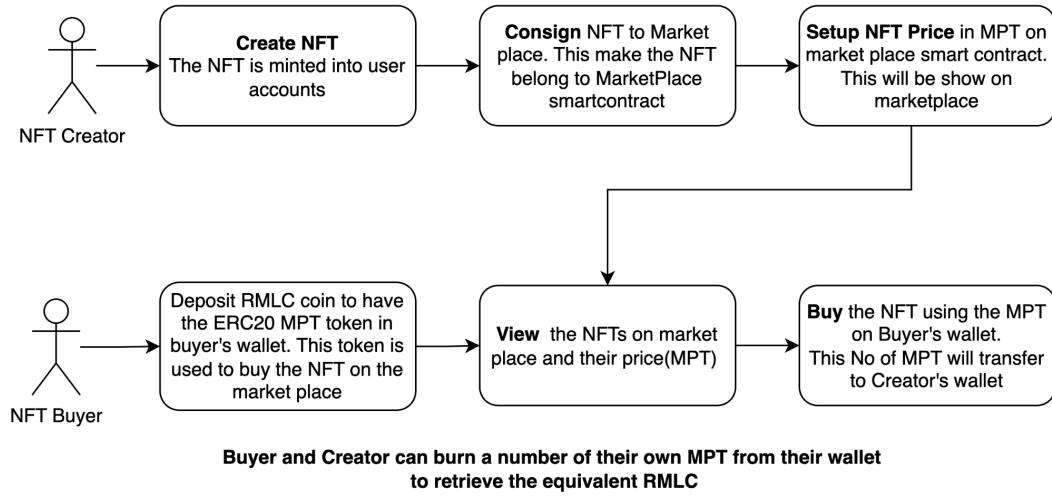
My Account

1. User information: User's configuration parameters.
2. Request more RMLC: Click on this button to if you want to increase your max total coin 0.05 RMLC. This request needs to be approved by the admin. In this case if this request is approve your max total coin is increased to 51 RMLC
3. List view of history logs: Display the events from and to this account.

2.3 NFT Marketplace

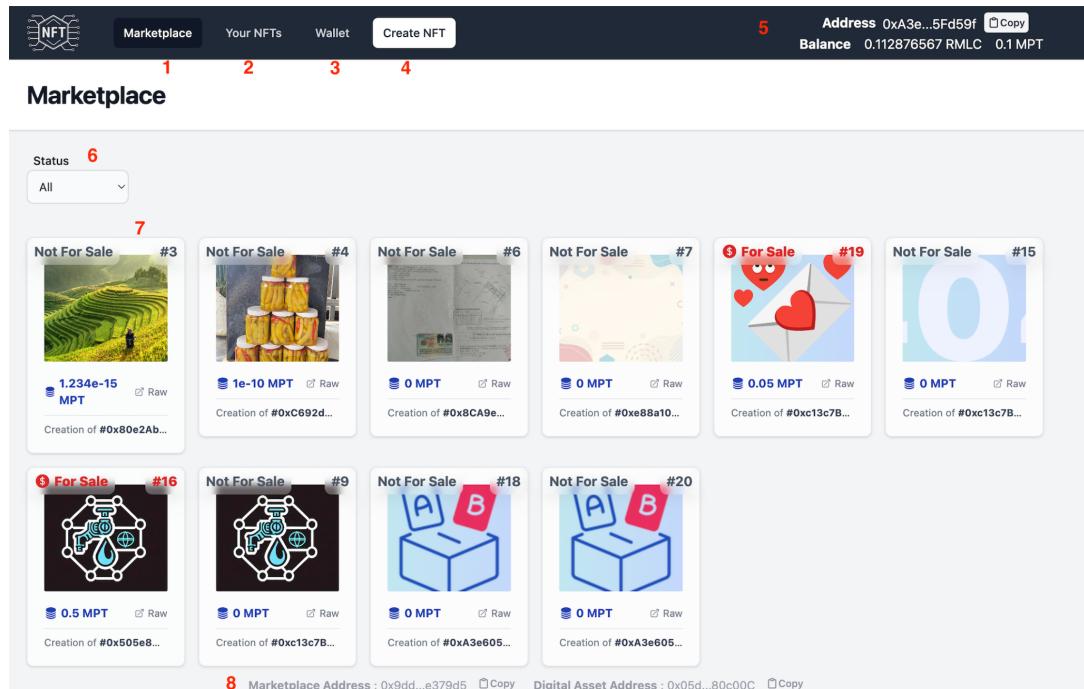
- The marketplace is at this link: <https://marketplace.rmit.site/marketplace>
- Users need to connect their Metamask and address to this site to be able to interact with the marketplace DAPP
- Users need to have enough RMLC to pay for transaction gas and to buy the MPT. The MPT is used to trade the NFT on the marketplace
- In order to create the NFT, the user is required to login to the faucet page, this page uses the credential of the faucet page to allow the user to upload their image asset into the server and use its URL as NFT's TokenURI

2.3.1 Workflow



Marketplace workflow

2.3.2 Marketplace

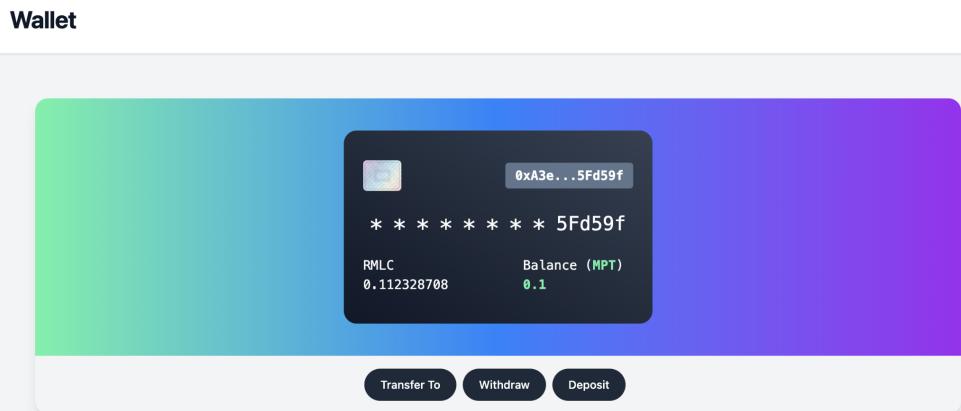


Market place

1. **Marketplace menu:** Click on this menu to visit the marketplace, where the app will list the NFTs which is consigned to the NFT marketplace smart contract

2. **Your NFTs menu** : Click on this menu to visit the page where the app will list all the NFTs which is belong to you (on the Digital Asset contract and on the Marketplace)
3. **Wallet menu**: Click on this menu to visit the page, where Users can see their wallet information including balance MPT, and balance RMLC. User can also interact with their own MPT on this page: deposit, transfer, withdraw
4. **Create NFT menu**: Click on this menu to visit the page, where user can create their NFT, then consign it to market place for sale
5. **Selected Account Information** : Users can see their wallet information including balance MPT, balance RMLC, and their selected account in Metamask
6. **Filter menu**: There are 3 options
 - a. **For Sale** : list all the NFTs that are consigned to market and ready to sell (public)
 - b. **Not For Sale** : list all the NFTs that are consigned to market and not ready to sell
 - c. **All**: display all
7. **NFTs list**: list the NFTs based on the filter (6). Each NFT is displayed as a card which includes the below information:
 - a. The thumbnail of NFT images
 - b. The sell status of NFT: Not for sell and for sell
 - c. The Price of this NFT in MPT
 - d. The raw link where the user can click on that to see the full image
 - e. When the user hovers the mouse point on the thumbnail the action button will display to let the user be able to buy this NFT (incase of for sale NFT)
 - f. NFT token ID: for example #3
8. **Contract address information**: This Dapp use 2 smart contracts the Digital Asset and Market place

2.3.3 Wallet



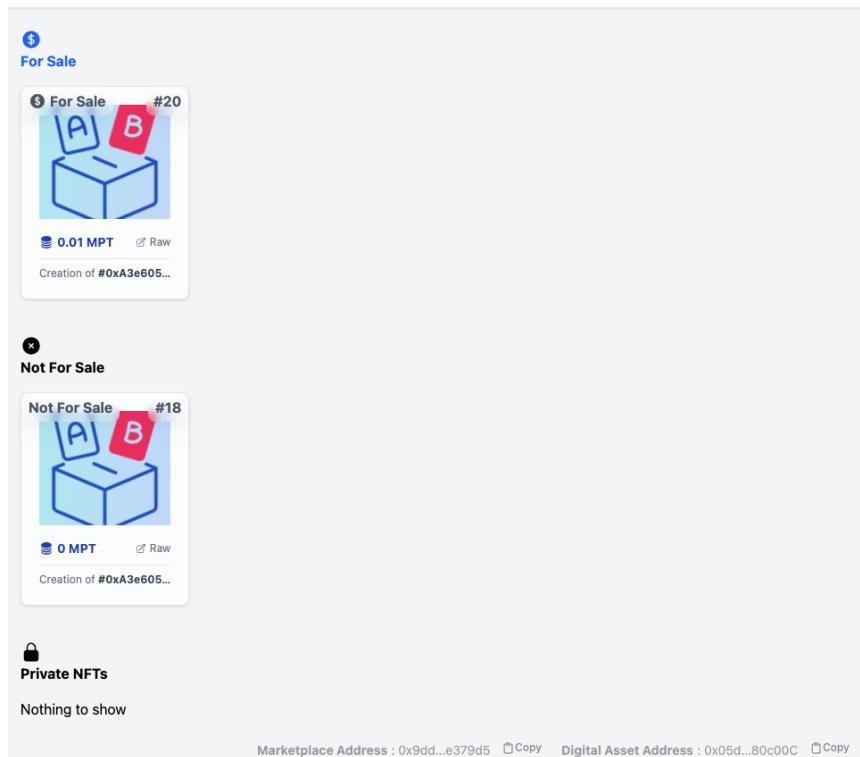
- ### Wallet
- Allow user use their native coin RMLC to buy the MPT token to use in this market place
 - Users can see their wallet information include : address information ,balance MPT, balance RMLC
 - This also support user to
 - **Deposit:** User deposits the number of his RMLC to retrieve the

- equivalent number of MPT. The exchange rate is 1. That means 0.1 RMLC will be equivalent to 0.1 MPT. In this case, users will actually pay more than 0.1 RMLC because they also need to pay for the gas fee.
- **Transfer To:** Let users transfer the number of their own MPT to the other address.
 - **Withdraw:** User can burn number of their own MPT to retrieve the equivalent native coin RMLC
 - Note that, in all case user need to pay for the gas fee using their native RMLC

2.3.4 Your NFTs

This page will list all of your belong NFT. Each NFT is displayed as a card.

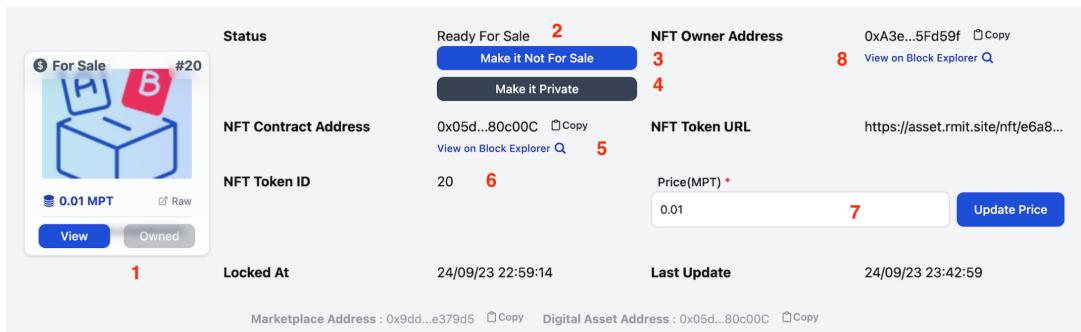
Your NFTs



Your NFTs pages

- For sale: Include the NFTs are setup to be ready for sale on the market place
- Not For sale: Include the NFTs are setup to not be ready for sale on the market place
- Private NFTs: Include the NFTs are not consigned to the marketplace.

When use hover the mouse on these card, they can click on View to see more details about the NFT



NFT details

1. The card to display the NFT information
2. Sale status of NFT : For sale and Not For Sale
3. The button to change the sale status of this NFT
4. Make it Private button to let users withdraw the NFT on the market place into their account. So they will self manage this NFT. The NFT marketplace doesn't control this NFT
5. Link to inspect the NFT Digital Asset on block explorer
6. The original NFT token ID on NFT Digital Asset smart contract
7. The price of this NFT in MPT, user can set any price then hit Update price
8. Link to inspect Transaction made by this address on blockchain explorer

2.3.5 Create NFT

Create NFT

1 Upload Asset
Upload your digital asset to the RMIT cloud storage

2 Create your NFT
Create a NFT on the RMIT blockchain

3 Market Place
Add your NFT to the RMIT market place

Great! Your NFT asset has been created

Token ID
21

URI
<https://asset.rmit.site/nft/902bcb157e4c8da00ac305678e9d8769-aa.jpeg>

TX Create NFT
0x357...f21fa2 [View on Block Explorer](#)

3 Add to the RMIT market place Keep it private Add NFT to MetaMask

Done
NFT Created!

Marketplace Address : 0x9dd...e379d5 [Copy](#) Digital Asset Address : 0x05d...80c00C [Copy](#)

Create NFT pages

1. **Upload Asset:** This is just a normal task to upload the NFT images into the server. This step may require the login credential on the faucet page because

the backend uses this credential to verify who the user is before letting him upload the asset into the server. After the upload has been done, these steps will return the URL of the asset, this URL will be used as the TokenURI of NFT

2. **Create your NFT:** Use the Token URI in step 1 to mint the new NFT. Note that the user is required to have enough RMLC to pay for the gas fee of minting a new NFT
3. **Market place:** This step lets the user choose:
 - a. Add to RMIT marketplace: Consign this newly created NFT to marketplace smart contract
 - b. Keep it private: The NFT is self-managed by user accounts
 - c. Add to metamask: Add this newly created NFT to the metamask wallet

Note that: Even if the user consigned the NFT to the marketplace, this NFT is still not ready for sale. The default setup of NFT on the marketplace is:

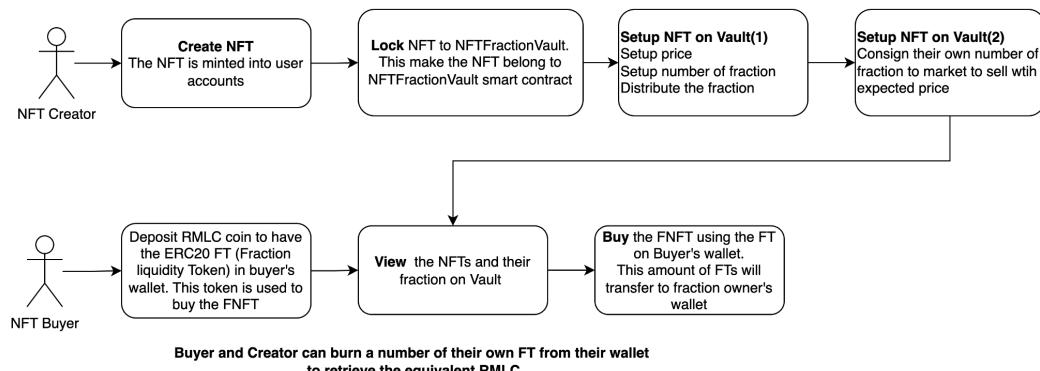
- Sale status: Not For Sale
- Price: 0 MPT

So, the user needs to set up this NFT once again on the marketplace by clicking on the shortcut Setup your NFT (after finishing step 3) or visit yourNFT pages to find it then set it up.

2.4 Fractional NFT (FNFT)

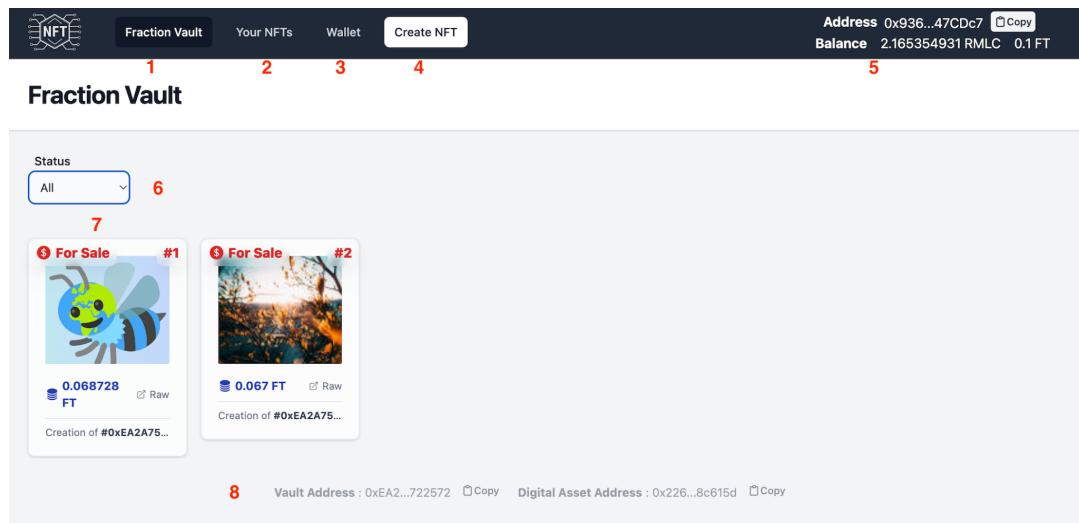
- The marketplace is at this link: <https://fract.rmit.site/>
- Users need to connect their Metamask and address to this site to be able to interact with the fraction NFT DAPP
- Users need to have enough RMLC to pay for transaction gas and to buy the MPT. The MPT is used to trade the FNFT on the marketplace
- In order to create the NFT, user is required to login to the faucet page, this page uses the credential of the faucet page to allow the user to upload their image asset into the server and use its url as NFT's TokenURI

2.4.1 Workflow



Fractional NFT(FNFT) workflow

2.4.2 Fraction Vault

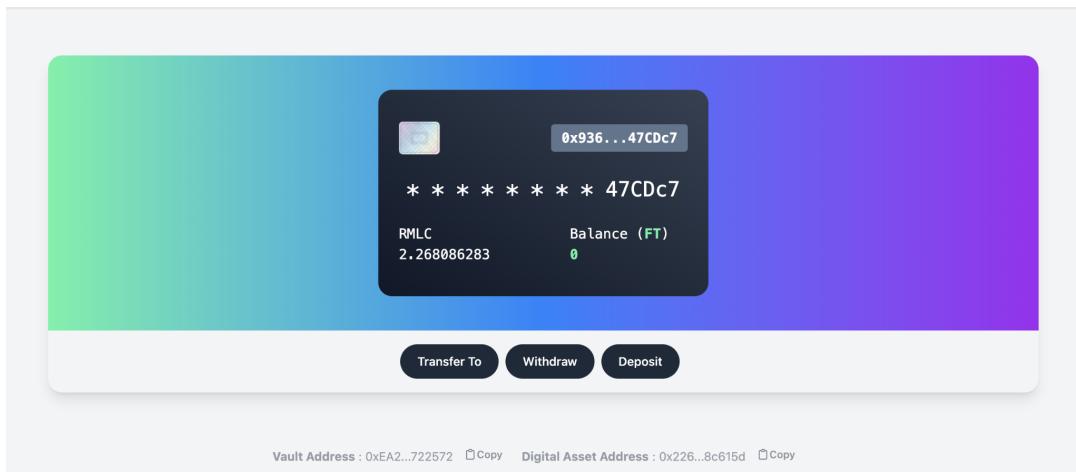


Fractional Vault

1. **FractionVault menu:** Click on this menu to visit the Vault marketplace, where the app will list the NFTs which is consigned to the Vault smart contract
2. **YourNFTs menu :** Click on this menu to visit the page where the app will list all the NFTs that belong to you (on the Digital Asset contract and on the Vault smart contract)
3. **Wallet menu:** Click on this menu to visit the page, where Users can see their wallet information including balance FT, and balance RMLC. User can also interact with their own FT on this page: deposit, transfer, withdraw
4. **Create NFT menu:** Click on this menu to visit the page, where user can create their NFT, then consign to Vault smart contract
5. **Selected Account Information :** Users can see their wallet information including balance MPT, balance RMLC, and their selected account in Metamask
6. **Filter menu:** There are 3 options
 - a. **For Sale :** list all the NFTs that are consigned to market and ready to sell (public)
 - b. **Not For Sale :** list all the NFTs that are consigned to market and not ready to sell
 - c. **All:** display all
7. **NFTs list:** list the NFTs based on the filter (6). Each NFT is displayed as a card which includes the below information:
 - a. The thumbnail of NFT images
 - b. The sell status of NFT: Not for sell and for sell
 - c. NFT token ID: for example #3
 - d. The Price of this NFT in FT
 - e. The raw link where the user can click on that to see the full image
 - f. When the user hovers the mouse point on the thumbnail the action button will display to let the user be able to **view** this NFT
8. **Contract address information:** This DAPP use 2 smart contracts the Digital Asset and Vault

2.4.3 Wallet

Wallet



Wallet

- Allow user to use their native coin RMLC to buy the FT token to use in this marketplace
- Users can see their wallet information include: address information, balance MPT, balance RMLC
- This also supports user to
 - **Deposit:** User deposits the number of his RMLC to retrieve the equivalent number of FT. The exchange rate is 1. That means 0.1 RMLC will be equivalent to 0.1 FT. In this case, the user will actually pay more than 0.1 RMLC because they also need to pay for the gas fee.
 - **Transfer To:** Let the user transfer the number of their own FT to the other address.
 - **Withdraw:** User can burn a number of their own FT to retrieve the equivalent native coin RMLC
- Note that, in all case, the user need to pay for the gas fee using their native RMLC

2.4.4 Your NFTs

This page will list all of your belongings NFT. Each NFT is displayed as a card.

Your NFTs

The screenshot shows a dashboard titled 'Your NFTs'. It has three main sections:

- For Sale:** Shows a single item: 'Not For Sale #2' with a thumbnail of a landscape, status '0 FT', and a 'Raw' link.
- Not For Sale:** Shows a single item: 'Not For Sale #2' with a thumbnail of a landscape, status '0 FT', and a 'Raw' link.
- Private NFTs:** Shows a message 'Nothing to show'.

At the bottom, there are vault and digital asset addresses with copy buttons.

Your NFTs pages

- **For Sale:** Include the NFTs that are setup to be ready for sale on the Vault
- **Not For Sale:** Include the NFTs that are setup to not be ready for sale on the Vault
- **Private NFTs:** Include the NFTs that are not consigned to Vault.

When users hover the mouse on these cards, they can click on View to see more details about the NFT

The screenshot shows the details of an NFT (#2) before it was fractionalized. The top bar includes tabs for Fraction Vault, Your NFTs, Wallet, and Create NFT, along with address and balance information.

Status: Not For Sale (2)
WithDraw From NFTVault (3)

NFT Contract Address: 0x226...8c615d (4)
View on Block Explorer (5)

NFT Token ID: 2 (6)

No. fraction *: (Required) 7
Price each fraction (FT) *: (Required) 7
Total price (FT): 0 (7)

Fractionalize NFT button (8)

Locked At: 25/09/23 01:36:59
Last Update: 25/09/23 01:36:59

At the bottom, there are vault and digital asset addresses with copy buttons.

NFT details page before fractionalized

1. The card to display the NFT information
2. Sale status of NFT: For Sale and Not For Sale
3. Withdraw From NFTVault: this button lets the user be able to withdraw NFT to their account and then self-manage this NFT. The NFT marketplace doesn't control this NFT
4. Link to inspect Transactions made by this address on blockchain Explorer

5. Link to inspect the NFT Digital Asset on Block Explorer
6. The original NFT token ID on NFT Digital Asset smart contract
7. Fraction setup control:
 - a. No. fraction: The number of fractions to which the NFT is fractionized to
 - b. Price each fraction(FT): the price in FT per each fraction
 - c. Total price (FT) = (a)*(b)
 - d. Click the Fractionalize NFT button to submit these above parameters to fractionalize this NFT.

After the NFT is fractionalized, all fractions are minted to the first owner. The first owner has all permission to distribute these fractions or trade on the Vault

Status		NFT Owner Address		Address 0xA3e...5Fd59f Copy	
NFT Contract Address	0x226...8c615d Copy View on Block Explorer	NFT Token URL	https://asset.rmit.site/nft/902...		
NFT Token ID	2	Total Price	2	0.067 FT	
Total Fraction	100 1	ERC20 Contract	3	0x058...830fcD Copy View on Block Explorer	4
Fraction Details 5					
ADDRESS 0x936...47CDc7 Copy View on Block Explorer		NO. FRACTION	N/A		
0xA3e...5Fd59f Copy View on Block Explorer		20	6	Transfer	Consign To Market
Total Locked Fractions		Total Locked NFTs Price 0.0201 FT 7			
List Locked Owner Addresses: 0x936...47CDc7 Copy View on Block Explorer		30	Buy 7		
Locked At	25/09/23 01:36:59	Last Update	25/09/23 02:15:59		
Vault Address : 0xEA2...722572 Copy Digital Asset Address : 0x226...8c615d Copy					

NFT details page after fractionalized

1. Total Fraction: Number of fractions which this NFT is fractionalized to
2. Total price(FT) = (1)* PricePerFraction \Rightarrow PricePerFraction = (2)/(1)
 - a. The price is computed as the weighted average value
3. ERC20 contract: when fractionalized the NFT into multiple fractions, the Vault utilizes the ERC20 contract to generate the fractions as ERC20 token and bind to this NFT.
4. Link to inspect ERC20 fraction on block explorer
5. Fraction details: Where list all the fraction owner's address and their fraction balance. User can:
 - a. For their own fraction user can
 - i. **Transfer** fraction to the other by click on Transfer button
 - ii. **Consign** to Vault market by clicking on the Consign to market. This action also requires the user to input the number of fractions and the expected price/fraction. The Vault will compute the weighted average value of FT based on the last price and this price. new price = (total fraction - consign fraction)*oldprice + consign fraction* consign fraction
 - b. Users can also buy the locked/consigned fractions with the price

computed above. When a user clicks on the **buy** button, the user is required to pay the number of FT to own the all locked fractions. These FTs will be automatically distributed into the fraction's lockers after the buy action is successful.

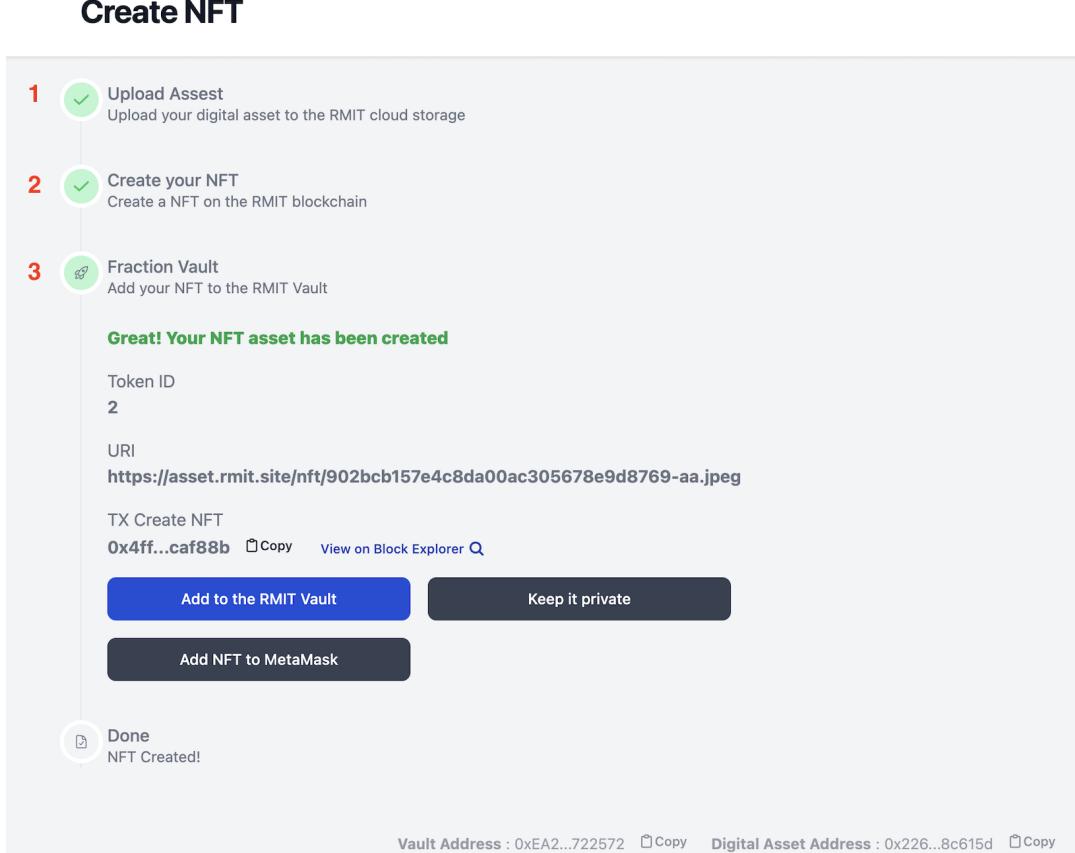
If one user collects all fractions, this user has a chance to reunion the token by clicking on the ReUnion button. This action will unbind the NFT to the ERC20 token, this means the user is own full NFT and doesn't want to share the NFT to the others. Note that, if one user wants to withdraw the NFT to self manage, they must reunion the token first to fully own this token on the vault, then withdraw.

Status	Not For Sale	NFT Owner Address	0xEA2...722572 Copy View on Block Explorer
NFT Contract Address	0x226...8c615d Copy View on Block Explorer	NFT Token URL	https://asset.rmit.site/nft/902...
NFT Token ID	2	Total Price	0.067 FT
Total Fraction	100 Re-union NFT	ERC20 Contract	0x058...830fcD Copy View on Block Explorer
Fraction Details			
ADDRESS		NO. FRACTION	ACTIONS
0x936...47CDC7 Copy View on Block Explorer		100	Transfer Consign To Market
Locked At	25/09/23 01:36:59	Last Update	25/09/23 02:15:59
Vault Address : 0xEA2...722572 Copy Digital Asset Address : 0x226...8c615d Copy			

Reunion and Make it Private

If one user collects all fractions, this user has a chance to make the NFT private to market by clicking on re-union NFT button, then clicking on withdraw from NFT Vault button.

2.4.5 Create NFT



Create NFT pages

1. **Upload Asset**: This is just a normal task to upload the NFT images into the server. This step may require the login credential on the faucet page because the backend uses this credential to verify who the user is before letting him upload the asset into the server. After the upload has been done, these steps will return the URL of the asset, this URL will be used as the TokenURI of NFT
2. **Create your NFT**: Use the Token URI in step 1 to mint the new NFT. Note that the user is required to have enough RMLC to pay for the gas fee of minting a new NFT
3. **Fraction Vault**: This step lets the user choose:
 - d. Add to the RMIT Vault: Consign this newly created NFT to Vault smart contract
 - e. Keep it private: The NFT is self-managed by user accounts
 - f. Add to metamask: Add this newly created NFT to the metamask wallet

Note that: Even if the user consigned the NFT to Vault, this NFT is still not ready for sale. The default setup of NFT on Vault is:

- Sale status: Not For Sale
- Number of Fraction : 0 (Not fractionalized yet)
- Total price: 0 MPT

So, the user needs to set this NFT once again on Vault by clicking on the shortcut Setup your NFT (after finishing step 3) or visit yourNFT pages to find it then set it up.

2.5 ERC20 FT generator

- The app is on this link: <https://ftgen.rmit.site/>

2.5.1 Home page

The screenshot shows the main interface of the ERC20 FT Generator. At the top, there is a header "ERC20 FT Generator" and a button "Create ERC20 Token". Below the header, there is a section titled "ERC20 Token" with a sub-section "At Address" containing the address "0x60BB754095013E34EDE0f043b63De1E16cbc3cdC". A red number "3" is placed next to this input field. Below this is a section titled "Fungible Token Information" with five buttons: "Mint" (red), "Withdraw" (blue), "Setup" (blue), "Buy" (blue), and "Transfer" (blue). Red numbers 4 through 8 are placed above these buttons. To the right of the buttons is a list of contract details, starting with "Contract ERC20TK(EFT): Add To Metamask" (red) and ending with "Current Account's Token Balance(EFT): 2". A red number "9" is placed next to the "Add To Metamask" link. Below this is a table showing account balances:

Idx	Address	Balance
1	0xA3e6053C9CE82480FC9431b411CA81Cf0c5Fd59f	2
0	0x936c0523cdE7800Da5f7a1A2db8C961f0C47CDc7	5

At the bottom, it says "Showing 1 to 2 of 2 entries" and has navigation buttons for "Previous", "1", and "Next".

ERC20 FT Generator main page

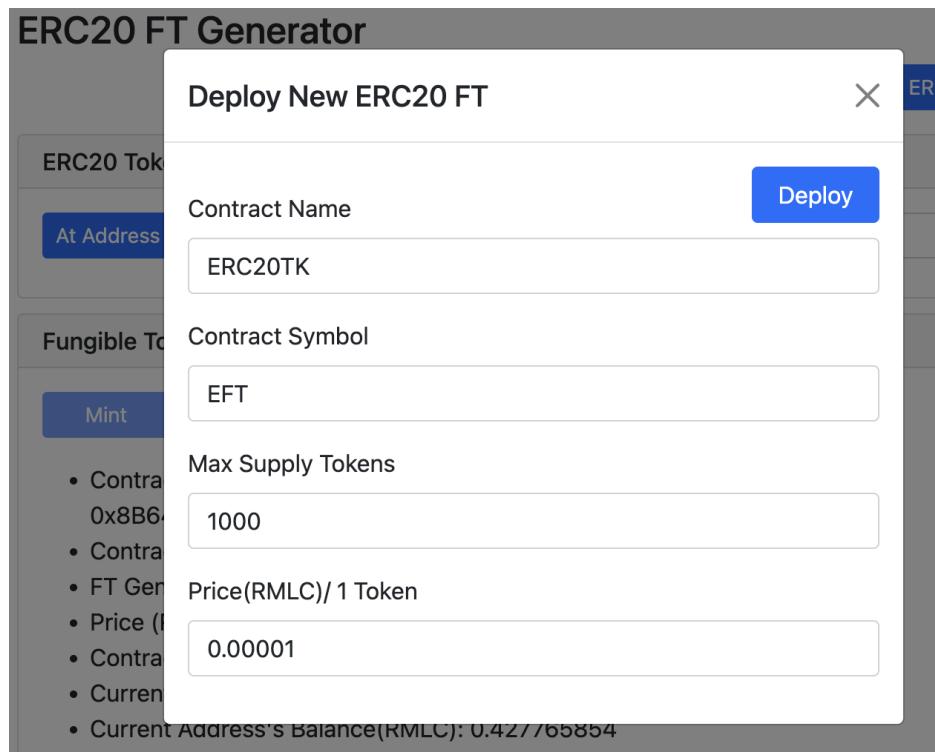
- Create ERC20 Token** button: this button is used to deploy the new ERC20 contract. The contract bytecode and ABI were prebuilt by truffle and loaded into this webpage, whenever the user clicks on this button, the contract bytecode will be deployed using the network and the account given by meta mask plugin
- At Address** button: when you need to connect this web app with the available ERC20 contract (had been deployed before). This will bind the contract ABI to the contract address given by (3)
- Contract Address** text input: This is the address of the ERC20 FT contract which had been deployed before.
- Mint** button: This button is used to mint a number of FT to a given address address. Only the contract owner can use this button.
- Withdraw** button: This button is to withdraw the native token owned by this smart contract to the sender. Only the contract owner can use this button.
- Setup** Button: This button is used to adjust the parameters of the binding contract: The maximum number of tokens that can be issued, the price per

token, and the feature of the paid-to-mint token. Only the contract owner can use this button.

7. **Buy** button: When the feature of the paid-to-mint token is enabled, any user can pay the amount of the native token(RMLC) to mint a number of tokens. The price and limit of tokens are set by the setup parameter in (6)
8. **Transfer** button: If your FT balance is positive, you can click on this button to transfer your own FT token to the other.
9. **Contract** overview: Display the binding smart contract information. User also can click to **Add To Metamask** to add this contract to metamask wallet
10. **Token holders table:** This table will display all the addresses of token holders and their token balance of the binding smart contract given by (2). Anyone can view this table

2.5.2 Create/Deploy new smart contract

When the user clicks on the button “**Create ERC20 Token**” to deploy the new ERC20 token, then configure the contract parameters needed for deployment as the picture below



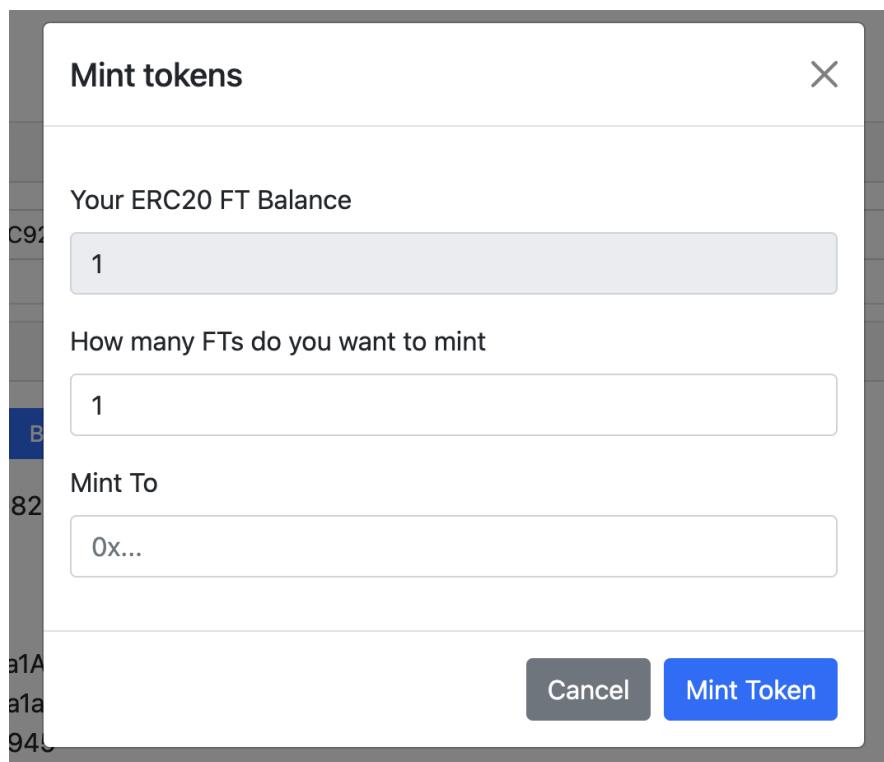
Configure parameters to deploy the ERC20 token

- Contract Name: the smart contract will be deployed under this name

- Contract Symbol: the smart contract will be deployed and use this as symbol for the token
- Max supply Tokens: This defines the maximum number of tokens can be generated/minted by the contract.
- Price(RMLC)/ 1 Tokens: This defines the price/1 token in RMLC. This will be used to let any user be able to pay to this smart contract amount of RMLC to mint a number of tokens. This feature only works if the owner enables the function to let the user do that (by default enable when the contract is first deployed)

2.5.3 Mint

Contract owner clicks on Mint button to airdrop number of FT to given address



Mint parameters

- The first parameter is read-only. This is the current number of FT that user owned.
- The 2nd parameter is the integer number which defines how many FT are air dropped to the address given by last parameter
- The 3rd parameter is the destination address which will receive the FTs

2.5.4 Setup

The contract owner clicks on **Setup** button on the main page to adjust the contract parameters as below

Setup Fungible Token X

Setup Total Supply

Price(RMLC) per FT

Is this buyable?

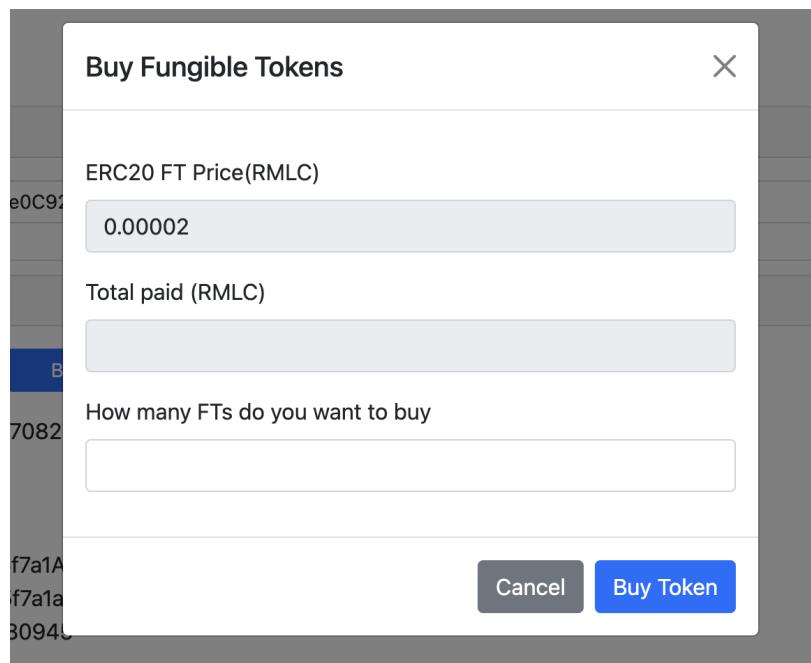
Cancel Submit

Setup parameters

- The 1st parameter is the default current maximum number of tokens that smart contract can mint. The contract owner can adjust this number as long as this is greater than the number of issued tokens
- The 2nd parameter is the float number(default number is current value) which defines the price of 1 FT
- The 3rd parameter is to enable/disable the feature to let anyone be able to buy the FT as long as they pay enough RMLC for that. The default value is the current status of the smart contract.

2.5.5 Buy Fungible Tokens

When this feature is enabled by the contract owner, Anyone can click on **buy** button to pay for the amount of RMLC to mint FT



Buy parameters

- The first parameter is read-only (the price of each FT in RMLC)
- The second parameter is read-only also, this is the total RMLC which users need to pay to the mint number of FT given by last parameters.
- The third parameter is the number of tokens the user wants to mint.

2.6 ERC20 Votes Campaign

- The app is on this link: <https://erc20vote.rmit.site/>

2.6.1 Home pages

ERC20 VOTE DEMO APP

1 Create New Vote Campaign

2

3

4 Transfer

5

6

7

Idx	Vote Options	Votes	Voters
2	option 3	0	
1	option 2	0	
0	option 1	1	0x936c0523cdE7800Da5f7a1A2db8C961f0C47CDC7

Show 10 entries

Search:

Showing 1 to 3 of 3 entries

Previous 1 Next

Main page of Vote Application

- Create New Vote Campaign** button: this button is used to deploy the new ERC20 contract. The contract bytecode and ABI were prebuilt by truffle and loaded into this webpage, whenever the user clicks on this button, the contract bytecode will be deployed using the network and the account given by meta mask plugin
- At Address** button: when you need to connect this web app with the available ERC20 contract (had been deployed before). This will bind the contract ABI to the contract address given by (3)
- Contract Address** text input: This is the address of the ERC20Vote contract which had been deployed before.
- Transfer** button: This button is used to transfer a number of FT votes to a given address address. Only the contract owner can use this button.
- Vote** button: This button is to let user burn the number of their FT as votes to vote for option
- Contract** overview: Display the binding smart contract information

7. **Vote result table:** This table displays who votes for which option

2.6.2 Create new Vote Campaign

When the user clicks on the button “**Create New Vote Campaign**” to deploy the new ERC20 Vote, then configure the contract parameters needed for deployment as in the picture below:

Deploy New Vote Campaign X

Contract Name	Deploy
ERC20VoteCampaign	
Contract Symbol	
EVC	
Total Number of Votes	
3	
End Time	
09/21/2023, 03:36	
option 3	Add
option 1	Remove
option 2	Remove

Configure parameters to deploy the ERC20 Vote

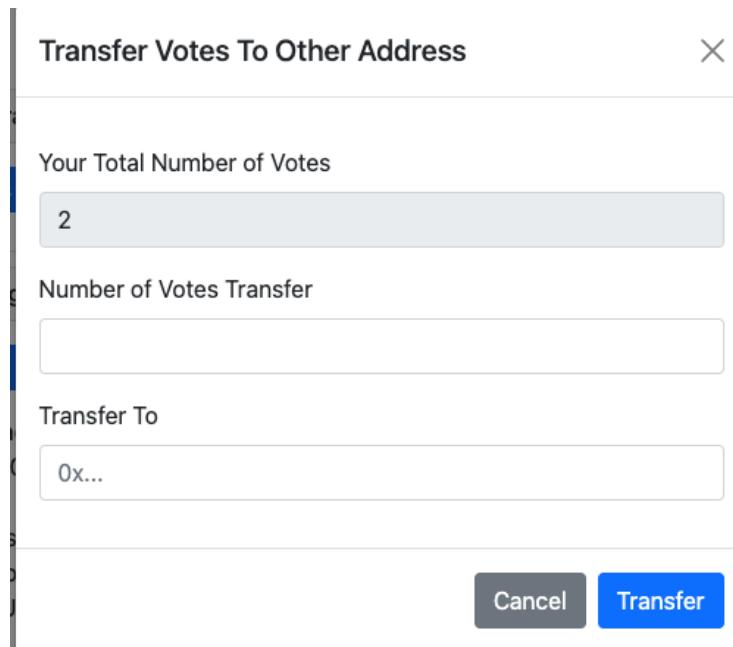
1. **Contract Name:** the smart contract will be deployed under this name
2. **Contract Symbol:** the smart contract will be deployed and used as the symbol for the token
3. **Total Number of Votes:** Define how many FTs as votes this campaign will issue. When first deployed, all the votes will be minted to the contract owner. Note that this number can be equal to the number of options, can be greater, or can be equal to the number of voters; depending on how you want to do the vote.
4. **End time:** Define when the vote campaign will be finished, after this time no one can vote even if they still have the votes.
5. **Options:** Text input to describe the option used during the vote. After the user inputs the text, the user can click the **Add** button to prepare the

vote options array before submitting to deploy the smart contract. Note that, the user can remove the added option by clicking on the **Remove** button. Users must create more than one option before creating a new campaign.

2.6.3 Transfer Vote

- When first deployed, all the votes are minted to the contract owner, so he needs to distribute these votes to the other voters to let them do the vote.
- When one user wants to delegate their votes to the other, they use this function to transfer their vote.

User need to click on **Transfer** Votes button:



The dialog box is titled "Transfer Votes To Other Address". It contains three input fields: "Your Total Number of Votes" (value: 2), "Number of Votes Transfer" (empty), and "Transfer To" (value: 0x...). At the bottom are "Cancel" and "Transfer" buttons.

Transfer Votes to the other users

- The first parameter is read-only which describes the number of current user's own votes. User can use this vote to vote for the option or transfer to the other
- The second parameter is the number of vote that the user want to transfer to the other
- The 3rd parameter is the destination address which will receive the Votes

2.6.4 Vote

Any user can click on the button Vote in the main page to do the vote as long as they had shared the votes before (the number of their own votes >0).

The form consists of the following fields:

- Your Total Number of Votes: 2
- Number of Votes Use: 1
- Select an vote option:
 - option 1
 - option 2
 - option 3

Buttons at the bottom:

- Cancel
- Submit

Votes' owner vote for options

- The first parameter is read-only which describes the number of current user's own votes. The user can use this vote to vote for the option or transfer to the other
- The second parameter is the number of votes that the user wants to vote for the option in the last parameter.
- The 3rd parameter is the selected option, the user needs to choose one option to vote.

2.7 ERC721 NFT Ticket

- The app is on this link : <https://nftticket.rmit.site/>

2.7.1 NFT ticket setup page

NFT TICKET DEMO APP

1 Create New Ticket Campaign

Ticket Contract

At Address Load contract from Address 0x... 3

2

Status: Connected as Owner

4 Gift For 5 Withdraw 6 Configure

• Campaign Name: Ticket1(TK1) at Address 0xffc798F7dd9437fD60B19e975084B6f3932aaa42
• Link for Buyer: <https://nftticket.rmit.site/buyticket.html?contract=0ffc798F7dd9437fD60B19e975084B6f3932aaa42>
• Contract Owner: 0x936c0523cdE7800Da5f7a1A2db8C961f0C47CDc7
• Contract Balance(RMLC): 0
• Issued Ticket: 1/50 7
• Ticket Price(RMLC): 0.001
• Created Date: 9/25/2023, 3:33:14 AM
• Ticket General Description:
THis is new TK

Show 10 entries 8 Search:

TID	Price	IssueDate	Owner	TicketInfo
1	0	9/25/2023, 3:37:44 AM	0x936c0523cdE7800Da5f7a1A2db8C961f0C47CDc7	Not For Sale THis is new TK

Showing 1 to 1 of 1 entries Previous 1 Next

TID	Price	IssueDate	Owner	TicketInfo
1	0	9/25/2023, 3:37:44 AM	0x936c0523cdE7800Da5f7a1A2db8C961f0C47CDc7	Not For Sale THis is new TK

NFT ticket seller page

- Create New Ticket Campaign** button: this button is used to deploy the new ERC721 contract. The contract bytecode and ABI were prebuilt by truffle and loaded into this webpage, whenever the user clicks on this button, the contract bytecode will be deployed using the network and the account given by meta mask plugin
- At Address** button: when you need to connect this web app with the available ERC721 contract (which had been deployed before). This will bind the contract ABI to the contract address given by (3)
- Contract Address** text input: This is the address of the ERC721 NFT ticket contract which had been deployed before.
- Gift For** button: Only the contract owner can use this button to mint the number of NFT tickets as gifts to the other
- Withdraw** button: Only the contract owner can use this button to withdraw all RMLC owned by the smart contract. The RMLC comes from others when they buy the NFT ticket.
- Configure** button: Only the contract owner can use this button to reconfigure the NFT ticket smart contract:
 - The ticket price (RMLC)
 - The maximum number of tickets can be issued by smart contract
- Contract** overview: Display the binding smart contract information
 - Users can share the **link to buyer** URL to another user to let them buy your NFT ticket.
- Contract detail table:** Display all the issued NFT ticket information

2.7.2 Buy ticket page

Users can use their RMLC to buy the amount of tickets.

The screenshot shows the 'NFT TICKET DEMO APP' interface. At the top, there's a 'Ticket Contract' section with a blue button labeled 'At Address' containing the value '0x28d55bFD5Be199Cd1c7238eeB58Faa86BE0ADD80'. A red number '1' is placed below the button. To its right is a red number '2'. Below this is a 'Campaign Status' section with a blue 'Buy Ticket' button. A red number '3' is placed below the button. To its right is a red number '4'. Below the status is a table titled 'Action' with columns: Action, TID, Price, IssueDate, and TicketInfo. The table contains four rows, each with a 'Transfer' button and a 'To Metamask' button. The rows are numbered 1 through 4 from bottom to top. A red number '5' is placed to the left of the table header. At the bottom of the table, it says 'Showing 1 to 4 of 4 entries'. On the right side of the table, there are 'Previous' and '1' buttons.

NFT Ticket Buyer page

1. **At Address** button: when you need to connect this web app with the available ERC721 contract (which had been deployed before). This will bind the contract ABI to the contract address given by (2)
2. **Contract Address** text input: This is the address of the ERC721 NFT ticket contract which had been deployed before.
3. **Buy ticket** button: The user can use this button to buy the number of NFT tickets. This button will cause a popup modal to require user input
 - a. How many tickets do you want to buy?
 - b. Note that you must pay **(a)* Ticket price(RMLC)** to own the number of tickets in (a)
4. **Contract** Overview: Display the binding smart contract information
 - a. User can share **the link to the seller** URL to back to NFT setup page
5. **Tables ticket details:**
 - a. Display all the issued tickets
 - b. The ticket owner has the chance to click on the button **Transfer** to transfer this ticket to the other user
 - c. **To Metamask:** add this NFT token to your metamask wallet.