Writing Transaction File

Assuming that you have completed the Model file for the Class Project before coming to this section.

Business Logic Definitions for Transactions:

If the user decides to register the event data with the network, they can call the create event function to register the same. Following is the interface for the Create event function:

Create Event Transaction:

Created Event transaction business logic:

- This function is used by the participants to add new events to the business network archive.
    - Parameters: The transaction definition from the model file will be passed as the parameter. Also, the mapping for the transaction will be added to the script file defined to provide the business logic for the application.
    - Return: The ID as a string is returned for the new event added to the business network.

Created transaction flow:

- Take the input from the user for the new Event.
- Create an array of ticket objects and add the tickets as per the number of tickets provided under the event.
- Keep the status of the tickets as “UNSOLD”
- Add the event to the event asset registry.
- Add the tickets to the ticket asset registry
- Save the event asset registry and ticket asset registry over the Blockchain State Storage.
- Return the saved Event ID to the user.

If the user decides to sell the ticket in the network, they can call the sell Ticket function to perform the same. Following is the interface for the sell Ticket function:

Sell Ticket Transaction:

Sell Ticket business logic:

- This function is used by the participants to sell the tickets in the business network archive.
    - Parameters: The transaction definition for sell Ticket from the model file will be passed as the parameter. Also, the mapping for the transaction will be added to the script file defined to provide the business logic for the application.
    - Return: The ID as a string for the sold ticket is returned to the business network.

Sell Ticket transaction flow:

- Take the input from the user on the ticket to sell.
- Check if the ticket is “UNSOLD”, if not return an error to the user.
- Change the state to “SALE” and owner to the buyer for ticket asset.
- Add the ticket to the ticket listing asset
- Save the ticket asset registry and ticket listing registry over the Blockchain State Storage.
- Return the saved ticket id to the user.

When the users who have purchased the tickets want to sell the same with the new owner:

Resell Ticket Transaction:

Resell Ticket Business logic:

- This function is used by the participants to resell the tickets in the business network archive.
    - Parameters: The transaction definition for resell ticket transaction from the model file will be passed as the parameter. Also, the mapping for the transaction will be added to the script file defined to provide the business logic for the application.
    - Return: The ID as a string for the ticket is returned to the business network.

Resell Ticket transaction flow:

- Take the input from the user for reselling the ticket to the new owner.
- Check if the ticket already has a flag of “RESELL” or “UNSOLD”, throw an error to the user regarding the same.
- Get the ticket from the ticket listing registry.
- Update the ticket to the new owner
- Save the ticket asset registry and ticket listing asset registry over the Blockchain State Storage.
- Return the saved ticket ID to the user.

Use Ticket Transaction:

Use Ticket Business logic:

- This function is used by the participants to use the tickets in the business network archive.
    - Parameters: The transaction definition for use ticket transaction from the model file will be passed as the parameter. Also, the mapping for the transaction will be added to the script file defined to provide the business logic for the application.
    - Return: The ID as a string for the ticket is returned to the business network.

Use Ticket transaction flow:

- Take the input from the user for using the ticket.

- Check if the ticket already has a flag of “UNSOLD”, throw an error to the user regarding the same.
- Get the ticket from the ticket listing registry.
- Update the ticket to the state “USED”
- Save the ticket asset registry and ticket listing asset registry over the Blockchain State Storage.
- Return the saved ticket ID to the user.

Following the definitions defined above, you would be able to code the Transaction Processor File for "Concert Ticket Application." Do add the assumptions as comments with the transaction processor file.

The next part is to define the access control file; the guidelines for the same will be provided, make sure you learn about the access control before coding the next part. You can keep on adding the code to a gradual approach.