

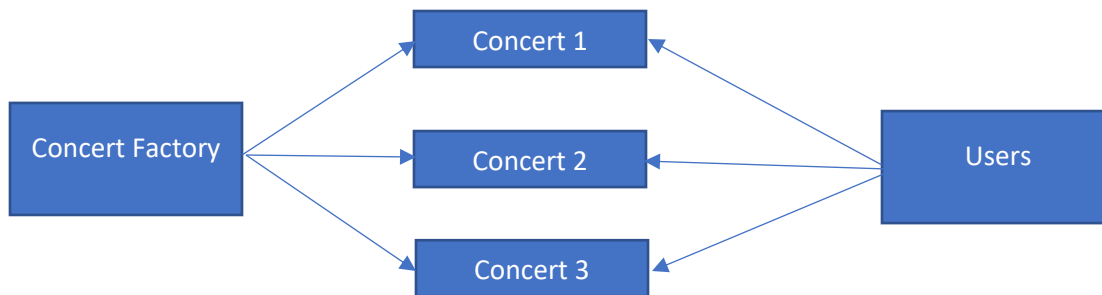
About MusicDAPP

MusicDapp lets concert organizers to create concerts and deploy the concerts details over Ethereum Blockchain. In this way organizers can prevent counterfeit tickets and ticket reselling.

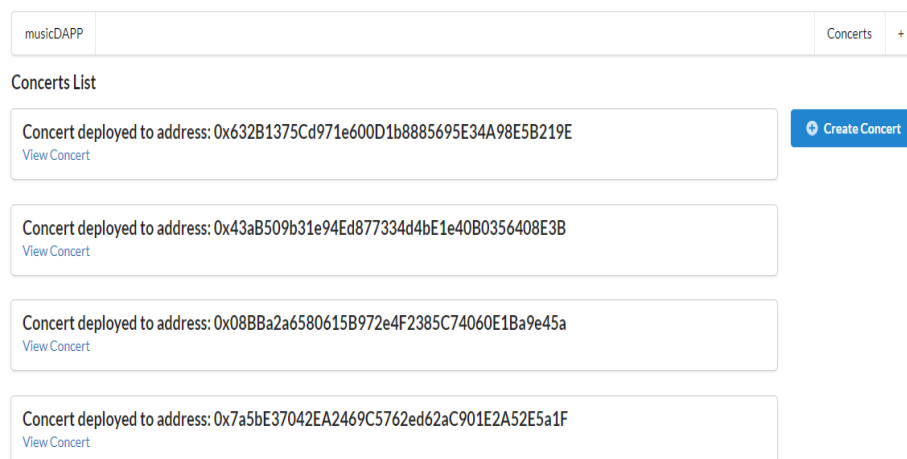
musicContract.sol: This solidity file has two contracts ConcertFactory and MusicDapp.

Concert Factory lets organizers to create contract by entering the required field of MusicDapp concert. Concert Factory deploys a specific instance of MusicDapp.

MusicDapp: stores the concert details and provides Buy Ticket and Check Ticket functions.



1. Home Page: This is the first page of MusicDAPP, it has list of deployed concerts



2. Create new Concert: Organizers can create new contracts by clicking Create Concert Button or + symbol in the menu bar. They will be directed to following page

musicDAPP

Concerts

+

Create a New Concert

Upload Concert Image (this prefills IPFS Hash)

Choose File

No file chosen

Upload

Name of Concert

Description of Concert

Specify Ticket Cost

wei

Specify Number of Seats

Date Time

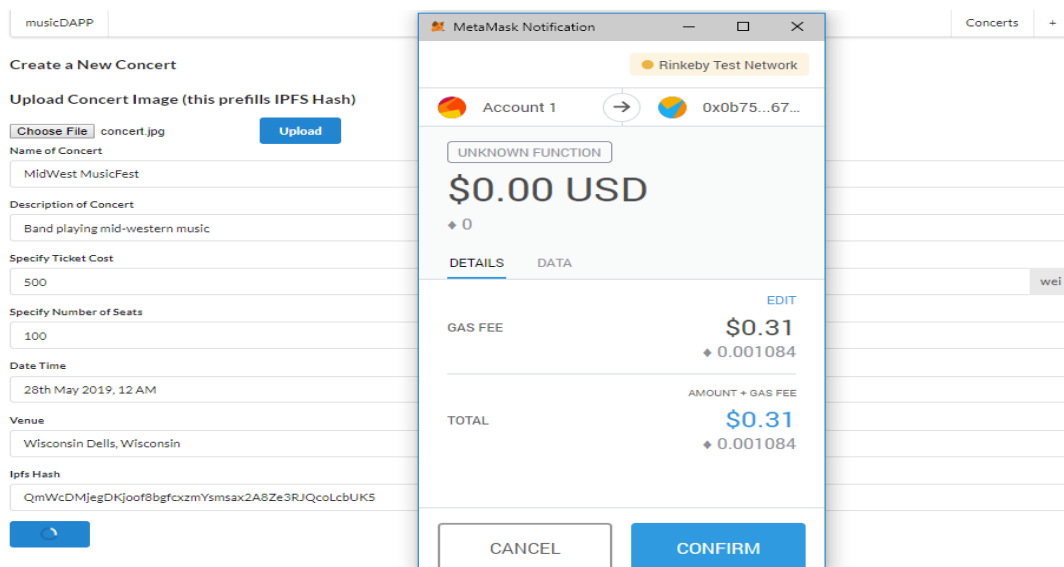
Venue

Ipfs Hash

Create

All the fields in this page is a required field:

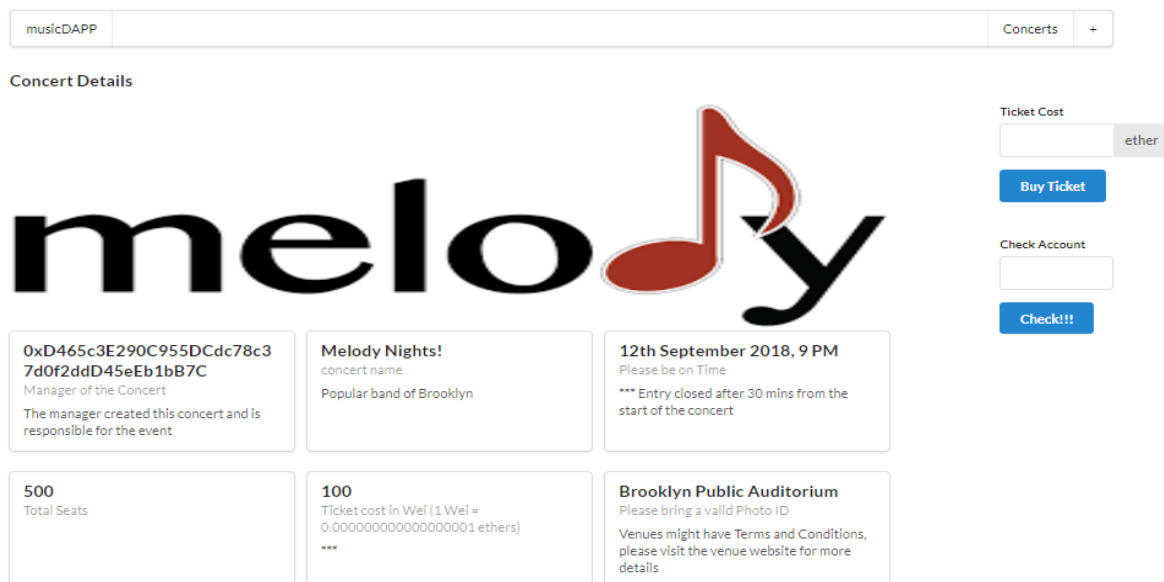
- By clicking Upload the Organizer will upload the image to IPFS, and the Ipfs Hash Field is prefilled
- The organizer must fill up all the fields and click create button. The information entered by the organizer will be sent to ConcertFactory contract.
- By clicking the create button the organizer is trying to deploy an instance of MusicDapp Concert through concert factory. After clicking create MetaMask pops up to authorize the transaction. Average gas price is 0.001 Ethers!!



- After successful transaction is organizer is redirected to Home Page where he can check/see the details of his concert.

3. View Concert:

By Clicking the View Concert link in the contract address card of homepage the user can see the contract details. The IPFS hash stored in the contract is used render the image. (By this way we reduced Gas Price exponentially).



4. By entering the ticket amount the user can buy the concert ticket:

By Buying the ticket the user's Ethereum wallet address is added to customerList in smart contract.

If the transaction goes through the successfully a success message will be show or if transaction failed the error message is shown

Ticket Cost

ether

Message from Contract
Returned error:
Error:
MetaMask Tx
Signature:
User denied transaction signature.

Buy Ticket

Ticket Cost

ether

Message from Contract
Transaction Successful

Buy Ticket

5. The User/Organizer can check whether an address is added to the customer list in the MusicDapp smart contract.

The image displays two identical 'Check Account' forms side-by-side. Each form consists of a title 'Check Account', an empty input field, a red message box, and a blue 'Check!!!' button.

Left Form:

- Title: Check Account
- Input field: (empty)
- Message from Contract: Valid Ticket!!!
- Button: Check!!!

Right Form:

- Title: Check Account
- Input field: (empty)
- Message from Contract: Invalid Ticket!!!
- Button: Check!!!

Note:

1. A physical ticket is not created for this project intentionally, because there is no feature for User who buys ticket to Login with his credentials.
2. An authentication feature is not implemented because, for a DApp to be fully decentralized, the user credentials should be stored in a Cloud service. For Ex: Amazon has a blockchain template for storing the credentials