Mnemonics and ropsten endpoint information have been removed from app.js and truffleconfig files.

Contract address is at 0x2250Bcd37C365C258D6DdEc3CD146FB3ac4C6DbC

```
thoth-contract — -bash — 160×55
Starting migrations...
> Network name: 'ropsten'
> Network id:
                       3
> Block gas limit: 8000000 (0x7a1200)
1_initial_migration.js
_____
    Deploying 'Migrations'
   > transaction hash: 0x86c1e32ed38691c9e30c3aa1d8dd363ee8b52a2188dacaa326b27597e8a91de1
> Blocks: 1 Seconds: 44
> contract address: 0xCF8de166B63CB7c9A997698316db5Fff4b34cd52
   > block number: 9203056
> block timestamp: 1607197062
> account: 0x242f37cd22F82D0A120979161Ac2019E683510a7
> halance: 15_583926231856686E01
   0x242f37cd22F82D0A1209

15.583926231854686591

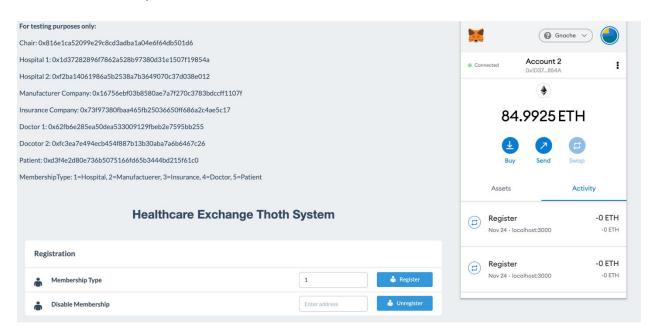
> gas price: 20 gwei

> value sent: 0 ETH

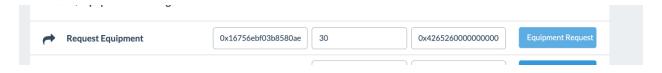
> total cost: 0.003802
    > Saving migration to chain.
    > Saving artifacts
    > Total cost:
                             0.00380022 ETH
2_deploy_contracts.js
_____
    Deploying 'Thoth'
   > transaction hash: 0xe88b0arcs
Seconds: 48
                              0xe88b0afc51fa05a02d522462815e845192ac201ba4d48d8301d0bc145c28f58c
    > contract address: 0x2250Bcd37C365C258D6DdEc3CD146FB3ac4C6DbC
    > block number: 9203058
> block timestamp: 1607197122
> account: 0x242f37cd22F82D0A120979161Ac2019E683510a7
   15.552186151854686591
    > Saving migration to chain.
    > Saving artifacts
    > Total cost:
                               0.03089332 ETH
```

The following demo is explained through multiple accounts from Ganache but the app has been deployed on Infura.

In the following example the second account will act as a Hospital so it will register by typing 1 into the membership field.



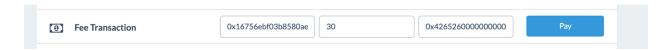
When requesting an item, the Hospital inputs the manufacturer's address, their payment offer in ethers, and a password



The manufacturer will respond to the requesting address by accepting (1) or denying (0)



The requester/Hospital account will pay the fee to the manufacturer's address, input the payment offer, and the password. The Smart Contract will check the original request hash to validate it.





The Manufacturer or whoever is moving the equipment can update the current address or location of where the requested equipment is in the Report Location. If it has arrived at the Hospital it inputs the Hospital address id into the Report Location.

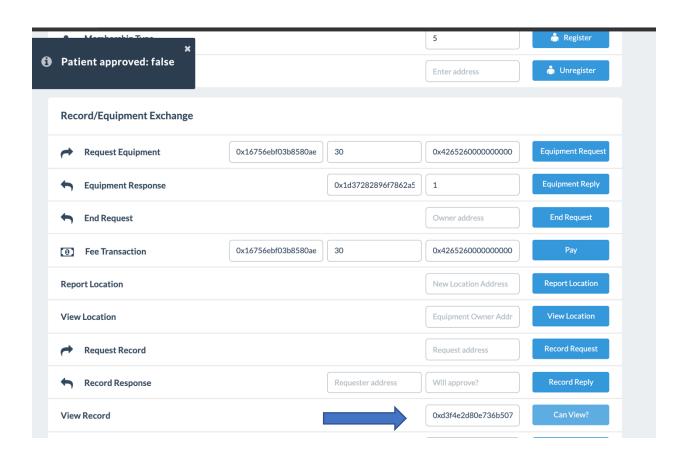
The Hospital can view where the current location of their equipment by inputting the address of the equipment owner, in this case the Manufacturer, into View Location.

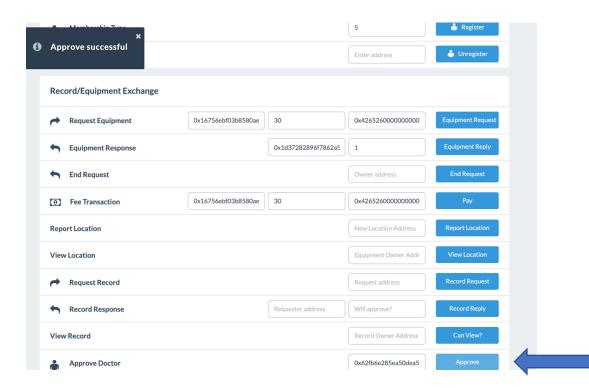


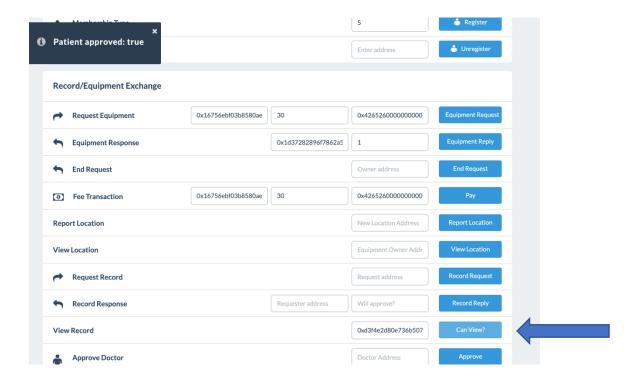
The Record request and response works similar to the equipment request/response. The user inputs the address of the record owner, either the patient or Insurance company representative. The recipient responds to the request by accepting (1) or denying (0) in the Will Approve box with the requester address.

For Doctors they enter the patients address in the view Record and the app will tell them if they are approved or not. The patient account would have to enter the Doctor's address in the approve field. Afterwards the system would approve the Doctor to view a patient's file.

Please note that I am switching between the Doctor and patient account in Ganache when roleplaying/testing the app.







The patient can undo/deny Doctor Access by inputting the Doctor's address in the Remove Doctor.