BLOCKCHAIN BASED STUDENT ACADEMIC RECORD SHARING SYSTEM FOR EXCHANGE STUDENTS

CMPE492 FINAL REPORT

January 7, 2019

Student Name: Barın Özmen
Student ID: 2012400045
Boğaziçi University
Department of Computer Engineering

Contents

INTRODUCTION AND MOTIVATION

First of all, i am supposed to build such a system that Universities that participate in exchange programs need to share academic information such as transcripts about the students. Traditionally, this is done by using paper transcripts sealed by university registers. In this project, a permissioned blockchain system is going to be deployed in order to facilitate academic information sharing. Quorum version of Ethereum Blockchain will be used which supports private transactions. Since the system may also be used by Erasmus students, the system will be designed with the new European GDPR regulation in mind.

To begin with, Blockchain Programming, one of the newest technologies and it is used in this project. As many people know that blockchain using etherium network mostly. But in this project, we will use Quorum network for building a registration system and show that what are the differences between these two.

At First, let us comprehend what is Blockchain Tecnology:

- *"Blockchain is a digital ledger in which transactions are recorded chronologically and publicly"*[2]
- *"A peer-to-peer infrastructure maintains a ledger (blockchain) of transactions among a distributed network of computers "*[2]
- *"Decentralized trustless protocol"*[2]

Both quorum network and etherium network using a pattern to match between two peers to be able to make transaction. The major difference is transaction style. Etherium based network using public key to transfer data. This public key can be seen from every user who are in that network.

On the other hand, quorum network uses private keys while making transaction. That means, transactions are aiming only those who the contract give permission to by knowing from their private keys. So in this project there will be 7 nodes(Universities) which send student transcripts towards each other via their private keys. Therefore, the transcripts of students will be know only by universities accepted those students.

This project aims to transfer data between universities easily and protected, with get away

from paper works and saving time from students and employees who has to do this process. Environmentally and socially it has a nice outcome. From the economic perspective, besides code work there are nothing to spend. After the contract released, the only thing is quick registration without error and with protection.

STATE OF THE ART

"Quorum is ideal for any application requiring high speed and high throughput processing of private transactions within a permissioned group of known participants. Quorum addresses specific challenges to blockchain technology adoption within the financial industry, and beyond."[3] This explanation is JP Morgan himself who built the quorum network.

Blockchain technology offers us a secure network. One of the biggest reward for using it security. Quorum offers one more thing that which is privacy. Besides knowing protected and secured well, quorum network also concerns about privacy. As we can see from this paper conclusion *"Analysis and large scale evaluation of the performance, scalability and security of consensus mechanisms. Suggestion of optimal protocols with respect to required network and security parameters"*[4] quorum focuses on performance with security and privacy issues.

Not only privacy, quorum seems to be has efficient delivery versus payment system. One of the article published in November 2017 says that *"Although we could not use the most efficient Zero-Knowledge protocols due to limited time and computing power, our architecture is easily upgradable and scalable. With our proposed improvements, such an architecture could be used for an efficient Delivery versus Payment systems such as Target2Securities in the Eurozone."*[5]

Quorum blockchain network is one of the newest technologies. In the most of the cases, Quorum is just used for finance sector. There are some software products that makes the transaction easy and private. Such as quorum software application who JP. Morgan build himself.

Technically, to be able to develop in quorum network, knowing only one programming language is not enough. Solidity language which uses smart contract is required for writing contracts. Besides that javascript language for addressing contract and deploying that contract is required. Lastly, shell scripting is required to setup whole project which is the hardest part of the project and for connecting network lively.

Highlight Shortcomings of SoA, in particular those that going to be improved in this project:

- *"Short lived applications or applications that are in any way limited in scope"*[6]
- *"Applications in which one way asynchronous communication is necessary"*[6]
- *"Applications that need GUI based functionality"*[6]

METHODS

To be able to develop this project Unix type of operating system is required. Linux based systems are compatible with the quorum blockchain network. First thing needs to be done is building the network. After building local network which can be moved to live network easily, deployment of smart contact is needed. For deployment specific location, addresses of nodes(in my case universities) are needed to make it private. After all done, from the terminal, using "privateFor" method is enough for to make transactions private. So, Linux based system, smart contract and javascript knowledge is required.

In this project, i used Mac operating system which is not compatible with quorum network. So i create a virtual machine to build quorum network and from the terminal access to that machine, i can read and write on that network nodes. After the necessities installed for building network[1](follow reference address), typically typing " > vagrant up" for building service and " > vagrant ssh" connects me to that virtual machine from ssh connection. After ssh connection is established, "./raft-init.sh" commend will be needed stated in reference address[1]. Before doing that, if we want our nodes(universities) can be connected from other networks(ex: live network), we must allow *rpccorsdomain* to that network. Simply we can do with ssh connection. Land on raft-start.sh file which is in linux machine, quorum-examples/7nodes/ directory. Modify that file such that append *' –rpccorsdomain "*" '* following lines of geth functions for each node. Then the following scripts can be written in order, *"./raft-init.sh", "./raft-start.sh"* to terminal for opening ports starting default from 22000 to 22006. Note that if you are running this network only in local network, adding rpccorsdomain process is not necessary. Just typing raft-init and raft-start is enough.

Second most important command is "truffle" commend which helps to deployment and manipulation of codes. From the terminal command, " > truffle compile" for compiling smart

contracts and " > truffle migrate " for deploying contracts. After all done, easily typing " > truffle console" to terminal make access to network for specified contract. Major difference between etherium based and quorum based network is making a transaction private. For ex:

"module.exports = function(deployer) // Pass 42 to the SimpleStrorage already deployed conract deployer.deploy(SimpleStorage, 42, privateFor: ["address_of_node"]) ;"

as we can notice that "privateFor:" parameter is making the transaction private and can be read and write only from networks specified in privateFor:[] even if all 7 nodes are in that network;

This project consists of 2 main part. First one is build the network and deploying smart contract on that network. Second part is for building a GUI for that project and integrating this contract to GUI using web and metamask. After second part is done, the registration process built in quorum network can be done from web easily.

In the first part there is not any GUI for that project and all is done via terminal. There are 2 ways to see the results when you integrate with contract which are, from typing "truffle console" as we mentioned before that gives result one by one. That means, for one command such as getNumber() will turn the value number. For another call, another command needed after finished first one. The other way is writing all the methods such as getter, setter into javascript codes which is actually same as console yet in this way we can write more than one commend and see the result of both from the terminal after we type " > truffle exec *.js" to the commend line. These are the only ways to collect and analyze the experimental data in this part. Like an old school, results must be wrote on simple paper or computer notepad to follow. Simple example javascript code for running js file:

```
1  var RegisterSystem = artifacts.require("./RegisterSystem.sol");
2  var addressBogazici = "QfeDAys9MPDs2XHExtc84jKGHxZg/aj52DTh0vtA3Xc=";
3  var student1 = "0x627306090abab3a6e1400e9345bc60c78a8bef57";
4  var student2 = "0xf17f52151ebef6c7334fad080c5704d77216b732";
5  module.exports = function(done) {
6     console.log("Getting deployed version of RegisterSystem...")
7     RegisterSystem.deployed().then(function(instance) {
8        console.log("Building the network...");
```

```
9         instance.registerStudent(student1, "Barın Özmen", privateFor: [addressBogazici]);
10          instance.registerStudent(student2, "Aras Uçar", privateFor: [addressBogazici]);
11          instance.registerCourse(student1, "CmpE483", privateFor: [addressBogazici]);
12          instance.registerCourse(student1, "CmpE492", privateFor: [addressBogazici]);
13      }).then(function(result) {
14          console.log("Transaction:", result.tx);
15          console.log("Finished!");
16          done();
17      }).catch(function(e) {
18          console.log(e);
19          done();
20      });
21  };
```

With running this simple code from terminal by typing *"truffle exec <registerSystem>.js"* will register these 2 students to Bogazici University node and add 2 courses to student1 on that network. Again it can be seen only by Bogazici University. To be able to see result, type following to commend line *"truffle console –network bogaziciUniversity"* to enter that network and in that network type following to see result: *"RegisterSystem.deployed().then(function(instance) return instance.getRegisteredStudents();) "*. The result can only be seen by Bogazici University node. Trying to access from other nodes is not permitted.

Second part is for integrating this network to web browser so that the transactions can be done easily. To be able to access this network from web, first metamask extension should be added to web browser. After that, as it mentioned above, *"–rpccorsdomain "*" "* should be added in geth console. Otherwise, connection will be refused. After all done, create a directory which has server and html files in it. Run html codes on that local server. In html codes ensure that web3 to connect nodes that we created for quorum network. Example to connect Bogazici Network(node2): *web3js = new Web3(new Web3.providers.HttpProvider("http://localhost:22001"));*.
After defining web3js to Bogazici node, it can simply called in script with following codes:
var contract = web3js.eth.contract(<abi>).at(<contractaddress>);
contract.getRegisteredStudents(function(error, result){ ... }
Get <abi> and <contractaddress> from deployed contract json file created when typed "truffle migrate" commend. With the result, it can be integrated into html body to show result.

RESULTS

In this project, aim is to register a student into a university and then apply this student as master student onto another university with the private transcript of student's. That means no other universities(nodes) can see the transcript and any other informations of this student. In that case, i first deploy a contract can be seen all 7 nodes:
privateFor: [addressBogazici, addressHacettepe, addressItu, addressOdtu, addressKoc, addressOsmangazi]

Again in the same deployed contract which holds the adresses of node 0,1..,7 are deployed and see if the transaction between the node 0 and node 4 can be seen from node 7 or not. To test it, first i made a transaction which increment the stored value in that contact by one and call it only private for node 1, not other nodes included:
SimpleStorage.deployed().then(function(instance){ return instance.set(2, {privateFor: [addressBogazici]});}
and see that the value stored in node0 and node1 incremented by 2. Values in the other nodes are still equal to 1. Than i change the address to node7 and increment the value by 5. And see that the value is in the node 0 is incremented to 8(1 + 2 + 5), node1 stays same value 3(1 + 2) and at last, node 7's value incremented to 6(1 + 5). That means We can build a contract privately and furthermore, we can make transaction much more private than contract itself.

The following images is for showing the real registration example of this project is used both in terminal and web:

Starting with login page to enter network we wish. There are 6 universities, each assigned to string which are "bogazici, odtu, itu, koc, osmangazi, hacettepe" as username:

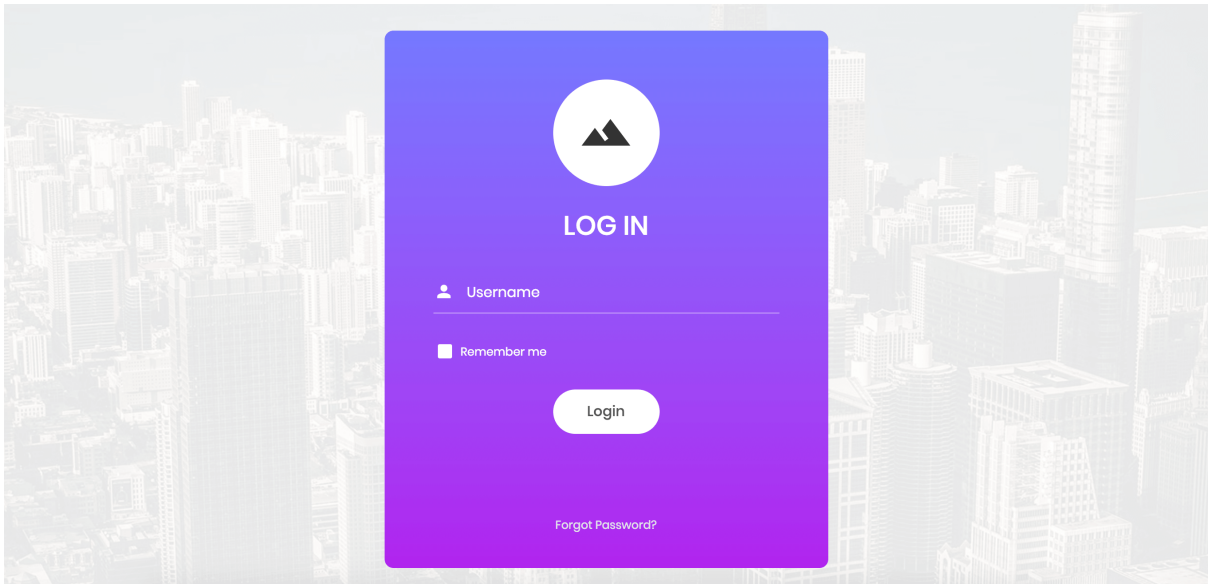

Figure 1: Login Page

On back hand side, with truffle commend, calling register functions.

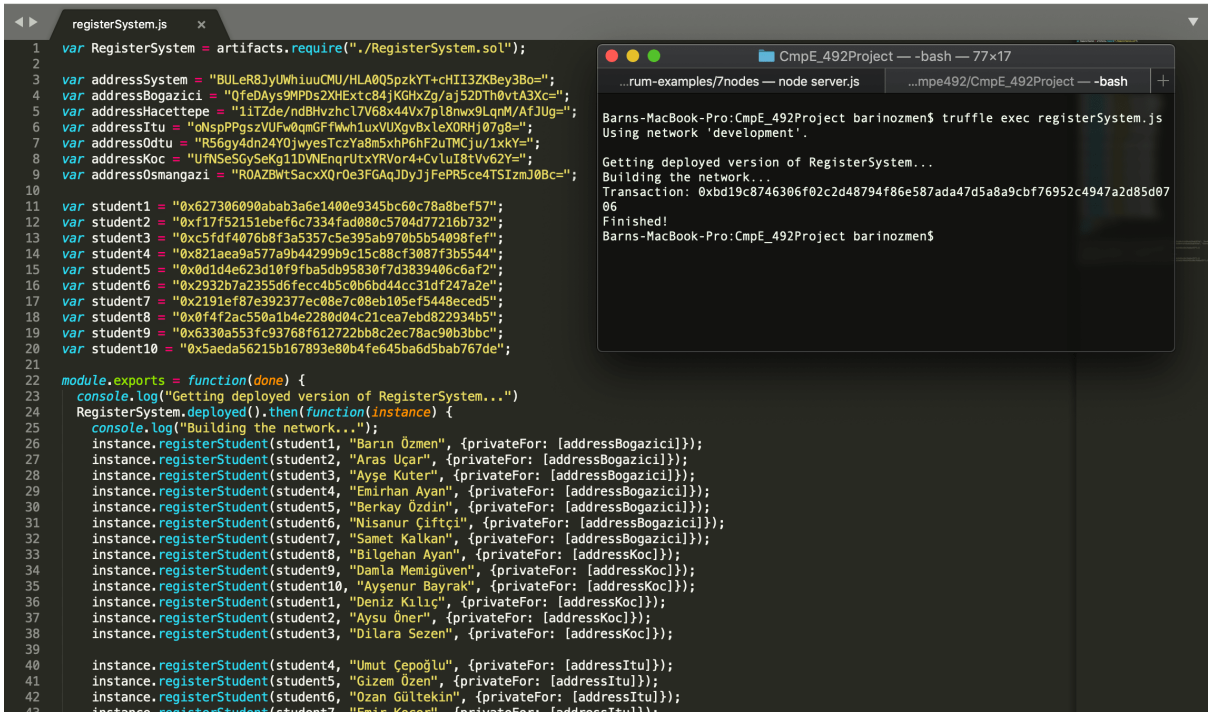

Figure 2: Truffle Execution

Login Bogazici Network and click "Get Registered Students" button. See that only first 7 students are seen by Bogazici University:

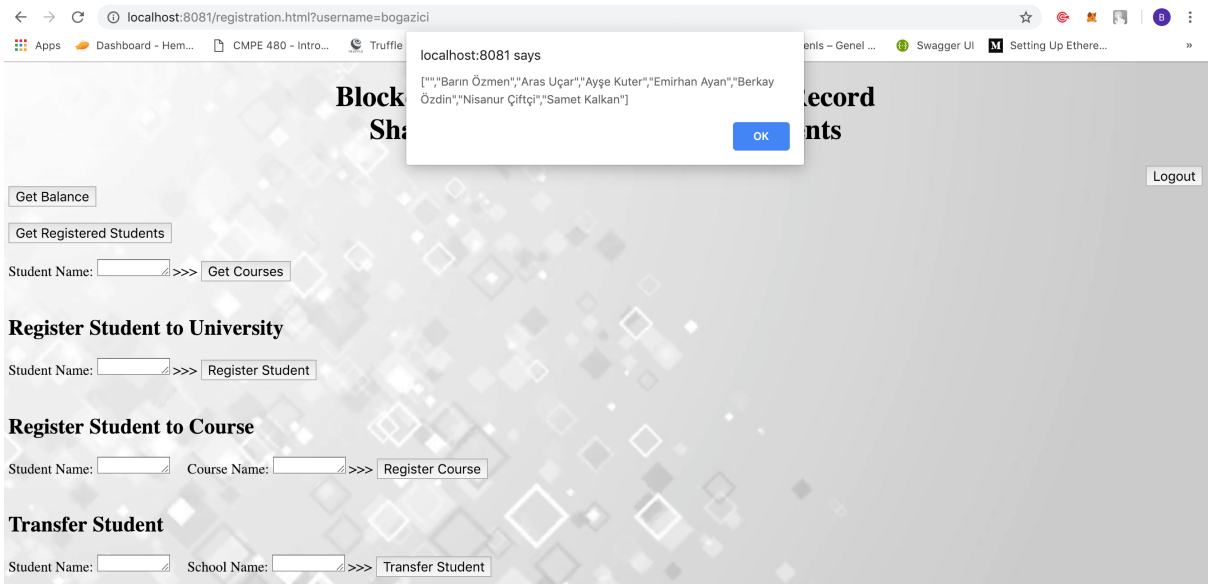

Figure 3: Get Registered Students Bogazici

Login Odtu Network and again click "Get Registered Students" button to see the result.

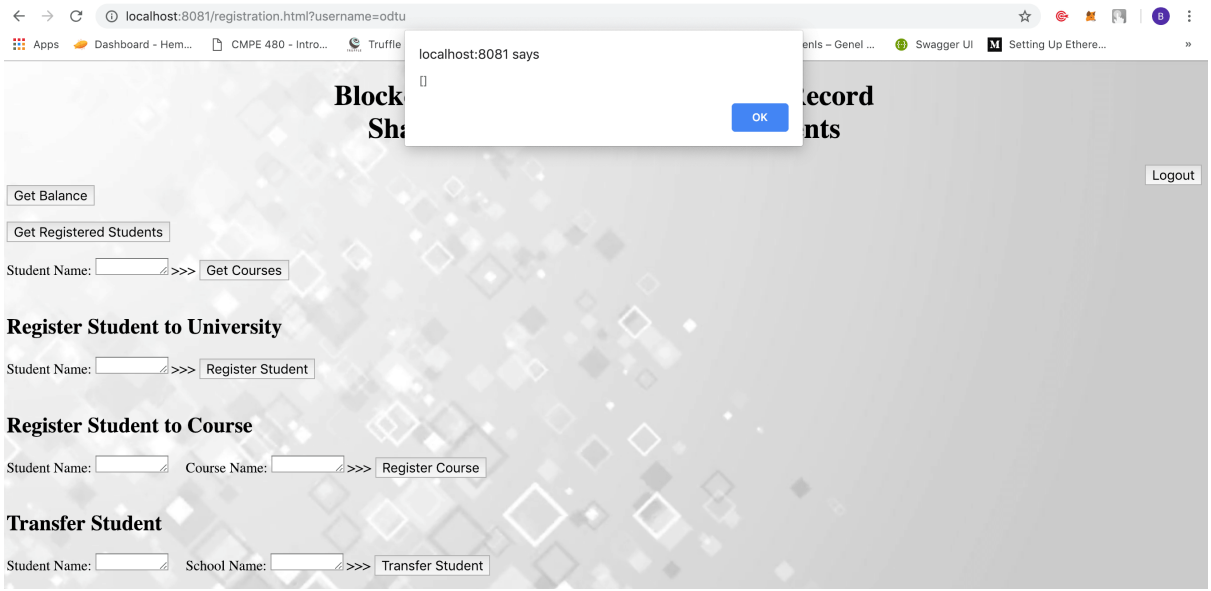

Figure 4: Get Registered Students Odtu

Login Bogazici Network again and transfer student Barın Özmen from university Bogazici to Odtu:

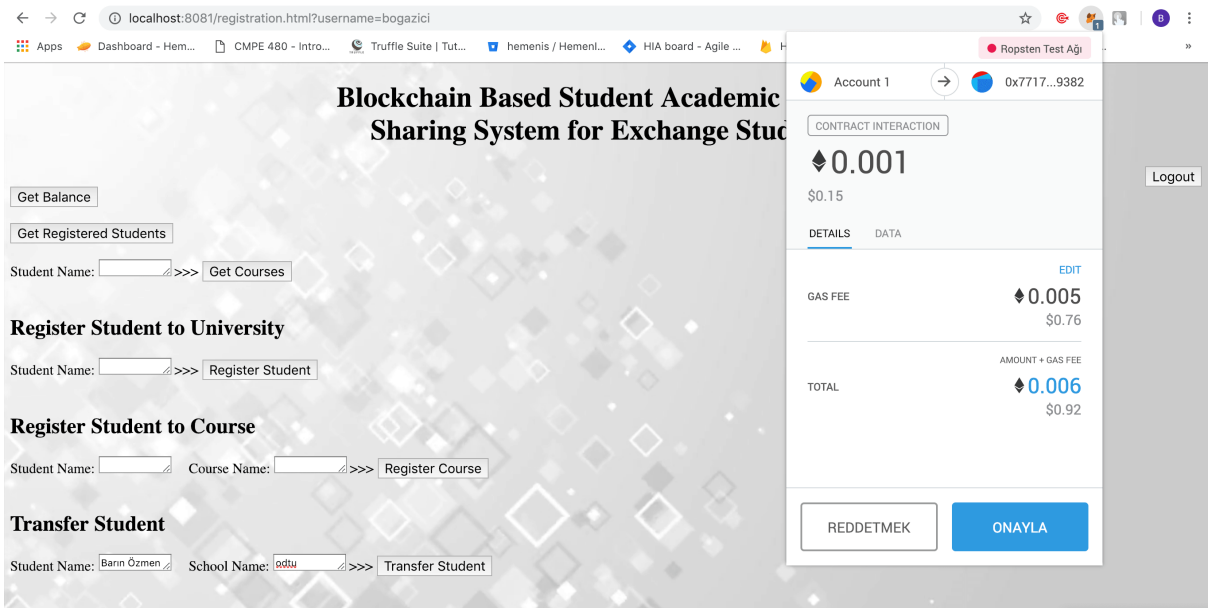

Figure 5: Transfer Student

Login Odtu Network. This time when we click Get Registered Students we can see Barın Özmen is registered to Odtu. Furthermore, when we clicked Get Courses, we see that all courses that Barın Özmen took can be seen

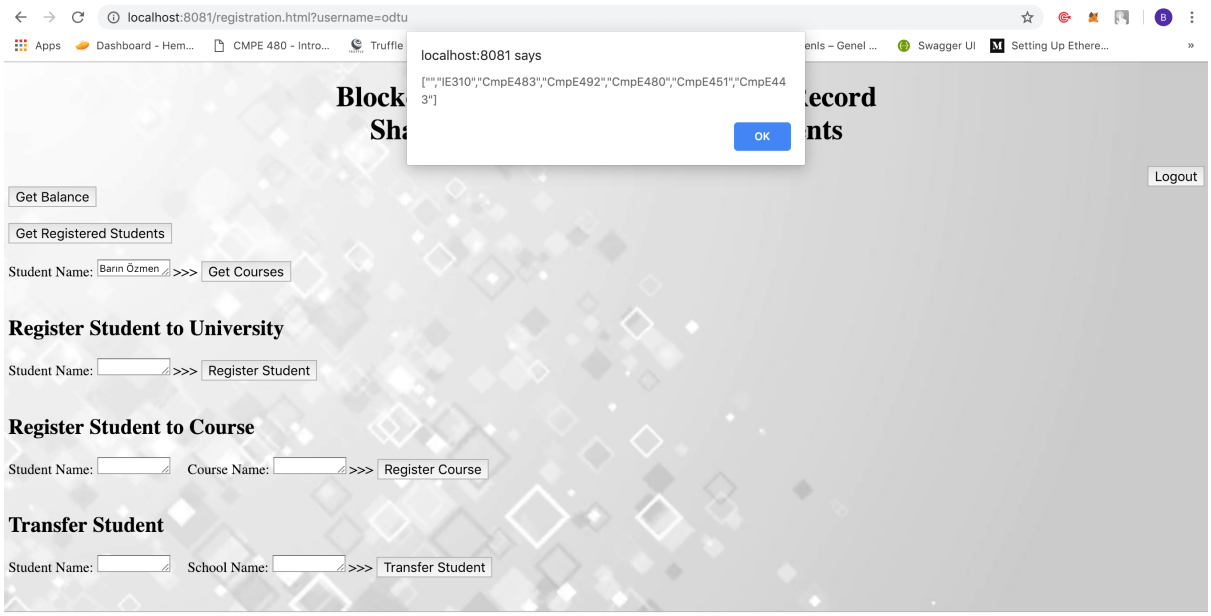

Figure 6: Get Courses of Student

First substance of SoA in this report says that an application that has been built as an interim solution that is not intended to provide full functionality or reuse for future applications. Yet, this software can be used in the future if it is well developed. Of course it can be upgraded in the following years but, universities can use this software as long as they want to. Before that software, which actually now, they are using only paper work and keep data in their machines. With this software, paperwork and storing data on machines will no longer exist.

Second one is emphasizing on the necessity of asynchronous communication. So i improved this project that asynchronous communication will not create any problem. Actually that is what a smart contract do. Smart contract does not let transaction without permission. Asynchronously accepted transaction will be no longer a problem.

Last one is about GUI based functionality. With integrating Metamask on web browser, the GUI for this project become simple and useful. Furthermore, because of this project does not need heavy data exchange which is service based, it will be easy to implement a GUI for this software.

CONCLUSION AND DISCUSSION

Our purpose is to transfer academic records of students from already registered university to another university where being an exchange or master student on that one. While doing it, the transaction must be private and permissioned. This actually what Quorum blockchain network does. In this project, there are 7 universities which all be in that contract(means that privateFor:7 universities when deployed). After registration of universities and students by mapping specific addresses, private transaction is needed. At that point, again we can make private transactions so that only the specific universities will be noticed and can get data from the contract. Therefore, this project can be done on Quorum Blockchain Network easily.

As we consider potential impact of this project in a societal context in economic and political way, there is no pros or cons on this project at all. Yet in environmental way, it will take away the waste of paper which can be called also tree demolition. Because there will be no need for paper work. All can be handled from that contract online. And also, as we consider social environment, nor students neither professors spend too much time on that transaction procedure, because whole system can be shown in digital environment by clicking few buttons

and transaction can be approved by clicking again few buttons.

Lastly, if we consider sustainability, again it will come as an advantage. Once the system built, all universities can use that system and there is no need for changing the structure in the future. We can see that for whole those years, universities used paper works and still they are using it. Try to image when this software came, no one can figure out when this system will be changed again.

FUTURE WORK

For now, i setup the quorum network and build 7 nodes on that network. I understand what the quorum blockchain network in case of private network does so far. The difference between public key and private key is the major point in that project. Because other universities cannot know all students academic information unless they permitted to. I made some transactions to figure out well.

Quorum network is built and contact is integrated with the web so it can be easily used from web. In the future, functionalities can be improved easily and html page can be designed according to that functionalities based on previous examples.

After all, exchange and master request are now available. Which means transaction between 2 or more universities are implemented. In the future, this system may also be used by Erasmus students, the system will be designed with the new European GDPR regulation in mind.

At last, GUI for this project is implemented with using web and it is ready to go live etherium network. In the future, this contract can be moved from local network to live network, so that it can be used from wide world.

REFERENCES

1. https://truffleframework.com/tutorials/building-dapps-for-quorum-private-enterprise-blockchains
2. https://piazza-resources.s3.amazonaws.com/jda1696qwm25l/jdhn6ovfev41fw/01intro.pdf?
3. https://www.jpmorgan.com/global/Quorum
4. http://conferences.inf.ed.ac.uk/EuroDW2018/papers/eurodw18-Stathakopoulou.pdf
5. https://eprint.iacr.org/2017/1093.pdf
6. http://www.exforsys.com/tutorials/soa/soa-disadvantages.html
7. https://www.overleaf.com