Graduado en Ingeniería Informática
Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros Informáticos

TRABAJO FIN DE GRADO

Queen’s University of Belfast Attendance Token

Autor: Carlos Vergara Chaparro
Director: Xavier Ferré

MADRID, JUNIO 2019
Resumen
Este documento cubre el proceso llevado a cabo para implementar el sistema de software denominado "Queen’s University of Belfast Attendance Token" (QUBAT) que propone una solución basada en tecnología blockchain para los problemas de asistencia de la Queen’s University of Belfast. Las siguientes páginas mostrarán el diseño del sistema, la implementación y los resultados de los experimentos realizados. Estos experimentos llevan a tratar el sistema de software QUBAT como un producto de mercado potencial que mejorará el funcionamiento de la Universidad al aumentar la motivación de los estudiantes para asistir a las clases.

Abstract
This dissertation covers the process carried out to implement the software system called “Queen’s University of Belfast Attendance Token” (QUBAT) which propose a solution to the Queen’s University of Belfast students lack attendance problem. The following pages will show the system design, implementation and results of the experiments done. These experiments lead to treat the QUBAT software system as potential market product which will improve the operation of the University by increasing the motivation of students on attending to classes.

The system code can be found in the repository of the develop branch at:

https://gitlab.eeeecs.qub.ac.uk/40250175/QUBCoin
Contents

1.0  Introduction and Problem Area................................................................................................................ 5

1.1  Problem area .................................................................................................................................................. 5

1.2  Problem statement .......................................................................................................................................... 5

1.3  Objectives ..................................................................................................................................................... 6

1.4  Solution approach ......................................................................................................................................... 6

1.5  State of the art ................................................................................................................................................. 7

2.0  Solution Description and System Requirements ............................................................................................. 8

2.1  Solution description ....................................................................................................................................... 8

2.2  Functional requirements .............................................................................................................................. 8

2.3  Non-functional requirements ......................................................................................................................... 9

2.3.1  Performance requirements ....................................................................................................................... 9

2.3.2  Security requirements ............................................................................................................................. 9

2.3.3  Quality requirements ................................................................................................................................ 10

2.3.4  Business rules requirements .................................................................................................................... 10

2.4  Functions definitions (class diagram) .......................................................................................................... 11

2.5  Interfaces required ....................................................................................................................................... 11

2.6  User characteristics ..................................................................................................................................... 12

3.0  Design .......................................................................................................................................................... 13

3.1  Architectural Description of the system ......................................................................................................... 13

3.2  User Interface Design .................................................................................................................................. 13

3.3  Software System Design .............................................................................................................................. 18

3.4  Discussion of design decisions ...................................................................................................................... 19

4.0  Implementation ............................................................................................................................................. 21

4.1  Choice of implementation languages and development environment ......................................................... 21

4.2  Use of software libraries ............................................................................................................................... 21

4.3  Key implementation decisions ..................................................................................................................... 22

4.4  Description of important functions implementation .................................................................................... 23
4.5 Description of each component implementation................................................. 24
5.0 Testing ................................................................................................................. 25
  5.1 Tests ................................................................................................................. 25
  5.2 Tests description .............................................................................................. 25
6.0 System Evaluation and Experimental Results ..................................................... 26
  6.1 Methodology ..................................................................................................... 26
  6.2 Experiments results ......................................................................................... 27
  6.3 Experiments results conclusions ................................................................. 31
  6.4 General conclusions ....................................................................................... 32
  6.5 Conclusion ....................................................................................................... 33
7.0 Appendices ........................................................................................................ 35
  7.1 Appendix A: Bibliography ............................................................................. 35
  7.2 Appendix B: Experiments form template ..................................................... 36
  7.3 Appendix C: Experiments results ................................................................. 37
1.0 Introduction and Problem Area

1.1 Problem area

As students progress in their academic trajectory, the attendance to classes and lectures of each individual falls into decline. Very often, those who reach university studies do not consider strictly mandatory to attend to lectures or labs and prefer self-studies or even not studying at all.

The academic system in regular universities is designed to teach students in classrooms. A teacher (or teachers) leads the class while the students participate either actively or passively. The fulfilment of both roles is necessary for the proper development of the system and will lead to producing the expected results. Students have the responsibility of attending to lectures and teachers of teaching the lecture. Any failure in these responsibilities would put the system in danger.

1.2 Problem statement

The Queen’s University of Belfast organisation knows about the importance of both roles’ responsibilities. Teachers are paid for their work and the existence of a contract makes them do their job. On the other hand, students pay their fees to have the right to attend to classes and because of this payment it could be taken for granted that they are going to attend. Of course, this is not the reality. It is known that not every student attend to every lecture and because of this, trying to solve the problem, the University has made many of the modules of different degrees a “mandatory attendance module”. In order to track this attendance some systems have been implemented such as signatures paper or attendance fob system (this last system in the Computer Science Building). If the student does not attend to lectures as is expected, he or she will be punished for it.

The student attendance is still a problem nowadays. The attendance tracking is not enough to resolve the situation. Despite students know about the punishment they will receive for not going to class, leaving apart the impact it causes in their studies progress, they decide to stay at home, library or just dedicate their time to other tasks.
Of course, justified absences are not a part of the problem to resolve but they could be a challenge to confront. If the solution proposed (and implemented) to solve the problem gets to reduce the number of justified absences (for example, a student changes a doctor appointment just to attend to a lecture because he or she is really motivated by the system) it would be an excellent achievement.

As mentioned above, the students’ study progress is affected by the absenteeism and, in consequence, the University as an institution. The teachers who receive less and less students in their classes could see their “teaching self-esteem” affected and get discouraged. Both, the poor academic results of the students and the demotivation of the teachers would affect the operation and prestige of the University. Find a solution to encourage attendance is completely necessary.

1.3 Objectives

The objective of this work is to find a solution to the absenteeism problem. Encouraging students to attend to lectures so finally they do it because they really want to, not because it is mandatory.

1.4 Solution approach

The solution approach for the attendance problem is based on the not fully satisfactory results of the current systems. They are implementing a tracking system so then, the students who do not achieve the minimum required can be punished. This punishment system seems to be not working.

Changing the way of thinking leads to develop a new system not based on punishment but in rewarding. If the students go to class not only for academic purpose but also for getting a reward, they will feel more motivated and even more excited about being in a lecture or in a practical.

The new attendance system must be related with something close to the students and what is closer to a university student than a smartphone? Therefore, it has been developed a software system, in particular, a mobile web application in which every student has an account and can get one token in this account every time he or she attends to a lecture, practical, lab, etc.
The application is based on blockchain technology so the token funds the students have cannot be corrupted. Taking advantage on the rise of this new technology is a good reason to guarantee the originality of the work done but also can be a problem because, being new, it undergoes constant changes and the existing documentation is not as extensive as it can be in other already stable technologies.

1.5 State of the art

The blockchain technology is used to store data. The difference between traditional databases and blockchains is that the last ones are decentralized. The blockchain technology made its first appearance on 2009 when the Bitcoin cryptocurrency was created. Blockchain was oriented to cryptocurrencies at the beginning but this changed with Ethereum development of the smart contract:

“The Ethereum, pioneered the next advancement to blockchains. It developed the smart contract, which allowed blockchains to be generated with computer programs working within and alongside them. This allowed financial institutions to create tools within the blockchain that represented loans and bonds.” (Biscontini, 2018)

Before the smart contracts exist, the only data represented in blockchains were the transactions. (Biscontini, 2018).

With smart contract arrival the crypto tokens appeared. While a cryptocurrency needs its own blockchain to exist, a token is built on top of a blockchain being defined by a smart contract. While a cryptocurrency represents a digital currency, tokens can represent assets, discount or even cryptocurrencies as well. The most used blockchain network to build tokens on top is the Ethereum blockchain (they were the developers who create the smart contracts) and the most used token standard is the ERC-20. (MiEthereum, 2018).
2.0 Solution Description and System Requirements

2.1 Solution description

QUBAT (Queen’s University of Belfast Attendance Token) System is the solution implemented. The system is a mobile web application built with NodeJS which connects with to a private Ethereum node to make tokens transactions.

The private Ethereum node runs locally, it is built using Go-Ethereum (geth) as a Proof-of-Work (PoW) system. The node oversees tokens transactions (QUBATs transactions) and Ether mining works. The smart contract which define the QUBAT (qubat.sol) is deployed on the Ethereum node using the framework Truffle. After the contract is deployed on the blockchain, Truffle creates a qubat.json file. The rest of the system will use this file to “talk” with the smart contract.

The NodeJS application uses different libraries (node-modules) to manage connections with the blockchain and the front end as well as middleware libraries. The application is implemented following an MVC (model views controller) architecture pattern. The library used to make connections between the application and the Ethereum node is Web3 (1.0 version).

The user model data storage is done in a database using MySQL server. Every user blockchain public account address is stored in the database too.

Every user must register in the application whether they are teachers or students. After registration and email address confirmation they can log in. After logging in, teachers will be redirected to the teacher screen as students to the student screen. This role splitting is important, teachers are only allowed to generate Attendance Codes and, on the other side, students are only allowed to capture the codes and spend the QUBATs they have earned. Each Attendance Code will be active just for two minutes, after two minutes the code could not be captured anymore. Also, teachers must wait for 15 minutes until they can generate a new code and students must wait 15 minutes until they can capture a new code.

2.2 Functional requirements
| FR01 | As a user of the QUBAT system I can register so that I can connect to the server |
| FR02 | As a registering user I can use my QUB email to register |
| FR03 | As a registering user I can choose my role as a student or as a teacher |
| FR04 | As a registered user I can confirm my account through my qmail |
| FR05 | As a confirmed user I can log in with my qmail and password |
| FR06 | As a student I can see the student main board after logging in |
| FR07 | As a teacher I can see the teacher main board after logging in |
| FR08 | As a student I cannot access to teacher’s main board |
| FR09 | As a teacher I cannot access to the student’s main board |
| FR10 | As a user I can log out from my main board |
| FR11 | As a user I can see my name and surname in my main to ensure that it is my account |
| FR12 | As a user I can go back to my main board from every screen |
| FR13 | As a teacher I can generate a new attendance code and show it in my screen |
| FR14 | As a student I can submit a new attendance code |
| FR15 | As a student I can see if the submitted code is valid or not |
| FR16 | As a student I can receive a QUBAT each time I submit a valid attendance code |
| FR17 | As a student I can spend the QUBATs I have earned |
| FR18 | As a student I can see my QUBATs funds when I want to spend them |
| FR19 | As a student I can spend as many QUBATs as I have |
| FR20 | As a student I can see my transaction code every time I spend some QUBATs |

2.3 Non-functional requirements

### 2.3.1 Performance requirements

| NFR1 | The gas cost of any transaction must not exceed the block gas limit of the blockchain, this gas block limit is targeted on 0xffffffff. |

### 2.3.2 Security requirements
2.3.3 Quality requirements

- **NFR4** The navigation over the system must be easy to understand. (Usable)
- **NFR5** The screens design must be related with the University website.

2.3.4 Business rules requirements

- **NFR6** Only teachers can generate attendance codes
- **NFR7** Only students can submit attendance codes
- **NFR8** A teacher must wait 15 minutes between code generations
- **NFR9** A student must wait 15 minutes between code submissions
- **NFR10** A generated code must be active just for two minutes
2.4 Functions definitions (class diagram)

2.5 Interfaces required

The server provides the user a Graphical User Interface consisting of a series of screens between which the user can move through the system. The html screens are generated with the Embedded Java Script (EJS) language which allows to show JS variables (as an account balance or a user forename) if needed.

The users’ data are stored in a database using MySQL Server. The server connects to MySQL Server using Sequelize Node module. The exchange of users’ data is made following a defined data model:

'user',{  
qmail: {type: Sequelize.STRING, primaryKey: true,},  
forename: {type: Sequelize.STRING, },  
surname: {type: Sequelize.STRING, },  
password: {type: Sequelize.STRING, },  
active: {type: Sequelize.BOOLEAN, },  
student: {type: Sequelize.BOOLEAN, },
The QUBATs transactions and funds are stored in the blockchain. The private Ethereum node is connected to the system using the Web3 library. This Web3 library needs a provider so the application can be run in browsers. Every time the Ethereum node is run, geth creates a file named geth.ipc which Web3 uses to connect the back end with the blockchain. The library Net of NodeJS is necessary so Web3 can use the IPC Provider method.

The QUBAT smart contract is also deployed on the blockchain. The system can call methods of the smart contract by using the library web3.eth. Web3.eth can have an instance of QUBAT contract thanks to the ABI file which can be found within the qubat.json file the Truffle framework generated when the contract was deployed.

### 2.6 User characteristics

The system will be used by two kind of user depending on the role they play. This role will determine the web pages the user can access. Both types of user are non-expert users:

- **Teacher:** this role can be play by Queen’s University teachers of different study fields, they will have access to a desktop or laptop and will be able to generate and show an attendance code to their students.
- **Student:** Queen’s University students will be present in lectures to see an attendance code and submit it.
3.0 Design

3.1 Architectural Description of the system

3.2 User Interface Design

The Welcome screen redirects users to Login or Register pages. Simple and easy design. Big buttons for making feel the users comfortable by giving them the idea of a “simple to use” system. The red coloured background and the Queen’s University logo on the left top corner are the common design pattern on every screen:
Register and Login screens, easy to understand:

Teacher users. The following screens will show what a teacher is going to find after logging in.

Teacher main board, just a button to activate an Attendance Code and a few instructions below about the code. The name of the user appears in every main board, both teacher’s and student’s:

The code is generated and appears in a big font so it can be seen from every corner of the classroom:
If the teacher tries to generate a new code less than 15 minutes later this screen will appear:

**Student users.** The following screens will show what a student is going to find after logging in. Students are supposed to log in the system with their smartphones so the next screenshots will be from a smartphone.

Student main board. The name of the student appears and only two buttons are shown for two different functionalities: Submit a code and going to Spend QUBATs:
If the student submits a valid code this screen will be redirected to:

If the code is invalid:
After clicking on “Spend my QUBATs” the user could see the QUBATs funds and be able to spend a specific amount:

If trying to spend more QUBATs than owned:

After the transaction is recorded the user will see:
As mentioned above, all the screens have the QUB logo and the red background. Every text shown is in white color except for those which are navigable (black color). Every button in the application has the same grey color. Both, the red background color and the grey were selected from the QUB website (requirement NFR05).

The screens design has been kept as simple as possible so the users can use the system quickly and comfortably (requirement NFR04).

### 3.3 Software System Design

The software system design is made on relation to the requirements. The FR01, FR02, FR03, FR04 and FR05 are implemented by building a registration-login system. This r-l system has a User model, a local strategy (for the login authentication) implemented with passport and two GET and two POST methods on main.js and users.js respectively.

For requirements FR06, FR07, FR08 and FR09 in the /user GET each user is redirected to its proper screen depending on the attribute “student” of the user model, if it is true, the user is a student, if it is false is a teacher. NFR02 and NFR03 are related to user authentication too. The fulfilment of these two requirements is made by the ensureAuthenticated function and the user redirection implemented in each GET function (NFR06 and NFR07).
A log out button is shown on both main boards which will end in the login screen and shut down the user’s account until new login (FR10). The user’s name can be seen in main boards by passing the attributes forename and surname to the .ejs files (FR11).

There cannot be any screen in which the user gets stuck, so every page has navigation texts to go back to main board or the previous page. This functionality is implemented in the .ejs files (FR12).

For the FR13, FR14 and FR15 it has been used a system with Java Web Token (JWT). The Attendance Code generated is a random six digits string which is provided with a time limit (two minutes) by the JWT (NFR10). If the student submits this code within these two minutes a QUBAT will be transferred to the user’s blockchain account (FR16) by calling the QUBAT smart contract method “transfer”.

The attribute time in the User model is created for NFR08 and NFR09. The teacher must wait 15 minutes between codes generations so it cannot become a QUBAT factory. The student must wait 15 minutes between codes submissions in order to avoid two things: submitting the same code several times and “QUBATs hunting” (a student could go from classroom to classroom just submitting codes).

FR17, FR18, FR19 and FR20 are fulfilled by implementing a transaction system where the students can see the QUBAT balance they have in their accounts by calling the method balanceOf of the QUBAT smart contract. After seeing the balance, they can spend as much QUBATs as they have, if the funds are enough, a transaction will be made by calling the method transfer of the QUBAT smart contract. After the transaction is registered, the hash code will be shown in the student’s screen by passing the returned value of the transfer method to the .ejs file.

3.4 Discussion of design decisions

In the registration-login system the POSTs are separated in users.js from the GETs which are in the main.js. This decision was made to leave clearer what happens when a new user is registered or logs in. In registration some communication must be established with blockchain (will be explained in section 4). The login POST main functionality is implemented in
passport.js as a local strategy, this decision was made in order to manage users’ authentication handled by passport middleware.

The teacher-student redirection is implemented this way because an API was not wanted to be implemented for this system consequently, both types of user share the same URI /user. This may be a problem so some user could type the URI in the URL bar and be in the wrong screen but thanks to the ensureAuthenticated function in auth.js this cannot happen, this function will ensure the user is trying to access to the URI is authenticated.

The two minutes expiration time is established by trying to avoid any kind of cheat on the students’ side. A student in the classroom could send a text message to a student at home with the Attendance Code on it so the second one could cheat. Two minutes is enough for the student in the classroom to submit the code but not too much so this student can send the code a classmate. Anyway, this expiration time can be reduced if necessary.

The 15 minutes rules are set in 15 minutes because this is more less the time taken between lectures. If a teacher decides to generate a code at the end of a lecture and the next teacher decides to do it at the begging, 15 minutes or more will be the time between these two events. Anyway, this waiting time can be modified if necessary.

The transaction hash appears so if in the future is needed to confirm a payment to the shop manager user.
4.0 Implementation

4.1 Choice of implementation languages and development environment

In order to work with server connections, the entire system has been developed on Ubuntu 16.04 OS.

Three main blocks are treated in the QUBAT system; the blockchain, the NodeJS application and the Database:

The blockchain is a private Ethereum node that can be run from a Linux console. The QUBAT smart contract is deployed on top of this blockchain and its code is programmed in Solidity language. Solidity is the language used for smart contracts, so this choice was not hard to make.

The NodeJS application is implemented using Java Script language. Python and Java Script are the most used languages to work with smart contracts. Java Script was the final choice because of the many Node modules that NodeJS offers to work with Ethereum blockchain and smart contracts. Also, there are plenty of Node modules like Passport, Bcrypt, Express… which have been used to implement the web application. These modules will be seen on detail in section 4.2.

For both blocks, the blockchain and the application, the development environment used is Visual Studio Code. It offers libraries and tools that facilitate both Solidity language programming and Java Script programming. It also offers a comfortable user interface and a debugger. The latter has been crucial in solving the problems found during the project development.

The database is created on MySQL Workbench programmed in SQL language. There were other options for the database as non-relational databases but in perspective to the future the relational database option was chosen although there is only one entity used in the system.

4.2 Use of software libraries
A list of all the packages used in the project can be found on the package.json file, the documentation of each package has not been added considering it can be easily searched. The main packages used are ("name": "version"):

"bcrypt": "^3.0.6" for hashing and unhashing passwords.

"ejs": "^2.6.1" for writing the html screens and render them.

"express": "^4.16.4" for the NodeJS infrastructure.

"jquery": "^3.4.0" for querying the users’ data to database.

"jsonwebtoken": "^8.5.1" for managing the attendance code expiration time and the email confirmation.

"net": "^1.0.2" for using an IPC Provider for the Web3 library.

"nodemailer": "^6.1.1" for connecting to the email host, preparing and sending the confirmation email.

"nodemon": "^1.18.11" for running the system locally.

"passport": "^0.4.0" a middleware for the accounts’ authentication management, it provides the useful req.user object so user data can be accessed on every GET-POST method.

"passport-local": "^1.0.0" for creating the login strategy to manage the user logins.

"sequelize": "^5.6.1" for connecting to the database server.

"web3": "^1.0.0-beta.52" for connecting to the blockchain and working with the smart contracts’ instances.

### 4.3 Key implementation decisions

The first decision to make it has been how to implement the QUBAT smart contract. Because the token is following the ERC-20 standard it has been decided to use a template supported by a MIT license, the details are in the token contract qubat.sol, and change the parameters required for the project.

Once the QUBAT smart contract is prepared a blockchain must be chosen to deploy the contract. To run a private Ethereum node with geth was decided for being the most used
blockchain technology for working with token smart contracts, having open code and being easy to run. The existence of the Web3.js library to connect NodeJS application to the blockchain was also decisive.

NodeJS lets to build a web application easily and quickly (for those who know how to use it), it provides lot of packages, tools and libraries which make easy the software development. After many documentation readings, the decision of using NodeJS was taken.

4.4 Description of important functions implementation

The code of the NodeJS has comments which lead to understand the implementation leaving no doubts. Even so some functionalities implementations are going to be explained in detail:

On registration POST, a new user is created following the User model. This user will be a student or a teacher depending on the form checkbox. After checking the user does not exist and that every form field is well filled (every field is filled, a qmail was provided, password fields match) the given password is hashed, a new Java Web Token is created and included in the URL the user is going to receive in the qmail address provided and, by last, the confirmation email with the URL is going to be sent. After this, if the user is a student, he or she will be given a new blockchain account which will have the same password as the user account. It is necessary to be done at this point to be able to use the same password for both accounts. On the other hand, teachers will be assigned with the coinbase account. Then, all user’s data are stored in the database including the time the account was created but with 15 minutes less. This 15-minute difference is implemented for letting the user to submit or generate a code the moment after is registered and logs in for the first time. After first code submission or generation the time will be updated.

On login POST, the passport local strategy is used. Within the implementation of this strategy the user is searched on the database, password provided is compared to the stored one and the user blockchain account is unlocked with the password stored if student user or the coinbase password if teacher user. This account unlocking is very important because if the coinbase account is locked, any transaction from this account cannot be made and the students will not receive any QUBAT when submitting a valid Attendance Code. On the students’ side, if the blockchain account is locked they will not be able to spend any QUBAT. The blockchain
accounts will be unlocked for more than four hours because no user is expected to be more than an hour using the application without leaving, every time they log in, the four hours will be reset.

4.5 Description of each component implementation

The NodeJS application has been described above so in this section it will be explained how to initiate and run the private Ethereum node.

First of all, NodeJS and NPM must be installed, after this, Go-Ethereum (geth) can be installed. Once geth is in the machine, the node can be initialized with a genesis block. The definition of this first block is on the genesis.json file. The blockchain data will be stored in an auto generated folder. After the blockchain is initialized it can be run with many different options but for this project the running command is:

```bash
geth --syncmode="fast" --cache=4096 --datadir=./db --targetgaslimit '0xffffffff' --rpc --rpcport 8545 --rpccorsdomain "*" --rpcapi "eth,web3,personal,net,miner" --ipcpath "/~/.ethereum/geth.ipc"
```

The first two parameters are not important (fast syncmode is not permitted when the blockchain has data). Datadir is the path where the blockchain data are stored. Targetgaslimit is the block gas limit the blockchain is trying to reach, if a transaction requires more gas than a block gas limit it cannot be done, therefore this parameter is set to a high number. RPC parameters are for defining the RPC endpoint and must be set to localhost:8545 so the data can be access only via localhost. Rpcapi import those libraries so can be used within the geth console. Ipcpath defines the path where the .ipc file is created every time the node is run so it can be used by Web3 library to connect to the NodeJS application.
5.0 Testing

5.1 Tests

The QUBAT smart contract has been tested using Truffle framework. The file qubat.js in the test folder of the repo shows the four blocks of test carried out. The qubat.sol file follow the ERC-20 standard and the code has been modified to be adapted to the QUBAT token. Therefore, some standard tests have been implemented to check if the two contract events are working, to check if the contract initialization goes well and if the initial supply of QUBATs goes to the coinbase account.

5.2 Tests description

The first block of tests checks that the smart contract is initialized with correct the values as name and symbol. The second block that the initial number of QUBATs is correct and that it has been added to the coinbase account. In the third block the Transfer event is tested by transferring a QUBAT from the coinbase to another account. Finally, the fourth block tests the other event, Approval event, by approving some tokens from the coinbase to be transferred.
6.0 System Evaluation and Experimental Results

6.1 Methodology

Because of the nature of the system, which is aimed at students, it has been opted for evaluating the use of it. For recovering information, the system will be tried for three different groups of QUB students. They will attend to one of the three different sessions and asked to register and use the system after a brief verbal explanation on what the system is made for:

“Thank you all for being here today. You will be asked to try a software system designed to reward the student’s attendance by giving you a token called QUBAT each time you catch a code provided by the teacher in a lecture. After that, you can buy this chocolate with the given token by following the required procedure. You only need to register in the next URL and use the system as you think is convenient. After the demonstration you will be asked to fill a form about the system tried.”

The form they will be provided is consisted on the next questions:

- Did you find any inconvenience to finish the task you were asked for? Which one? (a basic system must be usable and not cause a user paralysis).
- Did you try to login before registration? (the system must leave clear the user must register before being able to login).
- Did you find any difficulty on following the confirmation instructions given by the system? Which one? (as far as a user must confirm the qmail before being able to login it must be a clear process to follow).
- Once you were logged in, describe the first impression of the main screen design. (is good to know if the user finds the screen design messy, basic, easy to understand, etc.).
- Were you able to capture the attendance code given? (the main functionality of the system must be easy to complete).
- Did you find hard to buy the chocolate using the token? (this functionality is important for the system).
- Without watching your smartphone screen again, what is the name of the token you have spent? (Do the users get the name of the token?).
- What did you like the most?
What would you change? (two general questions to see if the global perspective of the system is good and what can be improved).

Would you feel extra motivated on attending to lectures if you could get these tokens there? (and then be able to spend them). (this is the most important question).

As can be seen, students are not asked for their age, gender or study field because the system must be usable for any kind of user (expert and non-expert).

(Go to Appendix B to find the form to be filled).

6.2 Experiments results

1. Did you find any inconvenience to finish the task you were asked for? Which one?

User 1: “No it was simple to follow”

User 2: “No, I didn’t.”

User 3: “No inconvenience”

User 4: “I didn’t find any inconvenient.”

User 5: “No”

User 6: “NO”

User 7: “No.”

2. Did you try to login before registration?

User 1: “No”

User 2: “No, I didn’t.”

User 3: “No”

User 4: “Yes.”

User 5: “Yes.”

User 6: “NO”
3. Did you find any difficulty on following the confirmation instructions given by the system? Which one?

User 1: “No it was straight forward, really creative system, fits well with rest of Qub websites.”
User 2: “No, I didn’t.”
User 3: “No”
User 4: “No, I didn’t find any difficulty. Everything is clear.”
User 5: “No difficulty”
User 6: “No”
User 7: “No”

4. Once you were logged in, describe the first impression of the main screen design.

User 1: “Neat and organized layout”
User 2: “It’s a very minimalist design where you can find everything clear”
User 3: “Easy to understand”
User 4: “It reminds me to Queen’s University webpage.”
User 5: “The main screen is easy to navigate”
User 6: “I thought it was quite simple but also very easy to navigate”
User 7: “A little bit basic”

5. Were you able to capture the attendance code given?

User 1: “Yes”
User 2: “Yes”
User 3: “Yes”
User 4: “Yes, but on the first attempt I confused the capital ‘i’ with a ‘l’ lowercase.”
User 5: “Yes”
User 6: “Yes, pretty easy”
User 7: “Yes”

6. Did you find hard to buy the chocolate using the token?

User 1: “No”
User 2: “No, it was easy”
User 3: “Easy transaction process”
User 4: “No, I didn’t”
User 5: “No, it is a simple and easy way to buy”
User 6: “NO”
User 7: “No”

7. Without watching your smartphone screen again, what is the name of the token you have spent?

User 1: “Qubat”
User 2: “QUBAT”
User 3: “QUBAT”
User 4: “QuBat”
User 5: “QoBbad”
User 6: “QUBT”
User 7: “QUBT”

8. What did you like the most?
User 1: “The reward for attending class, and incentive to go to class”

User 2: “The idea of creating a new coin for the university avoiding giving money to the banks”

User 3: “The chocolate. The name of the coin is so catchy!”

User 4: “Spending the Qubat, because I find this idea really original”

User 5: “It is easy of navigation and a good strategy for rewarding students.”

User 6: “It was very easy to navigate”

User 7: “The main screen is easy to navigate”

9. What would you change?

User 1: “For certain classes you could provide multiple Qubats for guest lectures, etc. to encourage students to participate more.”

User 2: “I would change the verification pay method because the code is too long.”

User 3: “Different rewards to choose from (different prizes cost a certain amount of QUBATs”

User 4: “The sale number, because it is really long and I find it not very aesthetic.”

User 5: “Code -> It should be shorter.”

User 6: “The design of the page.”

User 7: “The design of the main screen.”

10. Would you feel extra motivated on attending to lectures if you could get these tokens there?

User 1: “Yes if there were cool prizes.”

User 2: “Of course, It’s a really nice idea.”

User 3: “Genuinely, yes”

User 4: “Yes, for sure. Being rewarded for attending class is motivating.”

User 5: “Yes.”
User 6: “Yes.”
User 7: “Yes.”

(Go to Appendix C to find the experiments results).

6.3 Experiments results conclusions

1. **Did you find any inconvenience to finish the task you were asked for? Which one?**

No user found hard to achieve any task of the experiment. This means the system is usable.

2. **Did you try to login before registration?**

Two users tried to log in before registration, but the system did not allow them. Finally, they were able to complete the registration process. Nothing must be changed.

3. **Did you find any difficulty on following the confirmation instructions given by the system? Which one?**

No problems found in any user confirmation process. It seems that students are used to complete this kind of email confirmation.

4. **Once you were logged in, describe the first impression of the main screen design.**

The conclusion after gathering all the answers is that the students’ main screen is very easy to understand and very clear. In the future other functionalities could be added to this screen by following the design pattern.

5. **Were you able to capture the attendance code given?**
Just one user had an issue with this matter, but the user was able to submit the valid code anyway. The main functionality of the system seems to be intuitive and easy to complete. It is a great achievement for the system design.

6. Did you find hard to buy the chocolate using the token?

The spending QUBATs process is easy to follow and to complete.

7. Without watching your smartphone screen again, what is the name of the token you have spent?

Four of seven users could remember and write the name of the token. QUBAT seems to be easy to remember considering that more than a half of the students could wrote it down in a paper after just using the system once.

8. What did you like the most?

It seems that the ideas of being rewarded and being able to spend the tokens after are the most liked characteristics of the system.

9. What would you change?

The users did not like the transaction code and web page design. A user proposes to add more prizes for being bought.

10. Would you feel extra motivated on attending to lectures if you could get these tokens there?

Again, some users thought about the prizes to get and all the students agree on be rewarded for attending to class so they can spend the tokens later.

6.4 General conclusions
Three experiments sessions have been carried out. The first session attendants were User 1 and User 3, the second sessions attendants were Users 2 and 4 and Users 5, 6 and 7 attended to the third session. After the three experiment sessions some conclusions came out: the login-register system is well implemented and does not need to be changed; some users tried to log in before registration, but they were redirected. The students main board design is simple but also understandable, as mentioned above, if any functionality needs to be added to the student role it can be added in this page so it would not affect any other. On the other hand, the transaction code shown after a transaction is done was not liked, some users said it was confusing and too long to be there, consequently it has been deleted and replaced by the date and time of the transaction.

The users shown real enthusiasm about the general concept of the system; they would really like to have it in their lectures. They were proposing new ideas about the prizes the could buy with the QUBATs and after each session some of them started to discuss with me any kind of way of cheating. So far, every way of cheating proposed was already avoided.

**6.5 Conclusion**

It has been a hard project in which to work individually. The concepts of web application and blockchain were completely new for me at the beginning of the module. The Blockchain technology and particularly, Ethereum, are quite new concepts and therefore very changeable, the documentation that can be accessed is not as extensive as that of other more settled technologies and is often outdated due to the constant and rapid change it undergoes. The clearest and most direct example I have faced during the project has been the Web3 library, responsible for connecting the NodeJS part with the blockchain. When it came to obtain a provider to work with, many documents recommended doing so using the HttpProvider method of Web3 0.x.x versions. This method is deprecated and in the new version of Web3 (1.0.0) the method does not exist. In substitution, the IPC Provider method is used (implemented in the QUBAT system) which initially caused me a problem which took three days to solve. The satisfaction of fixing this problem led me to continue with the project with much more motivation.

Despite having found another series of problems during the development of the system, in the end, the main functionalities of this have been successfully achieved. Users will not find bugs
or failures when using it, although certain aspects of the software can be improved. Security aspects related to having deployed the application in a web server will have to be considered and implemented if a commercial use of the system is wanted. Some efficiency aspects in the Ethereum node can be improved since the hired power of the server is not much. If wanted, in the future, the application and the blockchain to be in different servers, security aspects in the exchange of data between them should be considered by taking care of the IPC Provider connection so that no data can be captured during an exchange.

As for the results, the experiments carried out show that the idea and the project can perfectly be considered to have a commercial use. Although it was only a test, the students were excited about the idea of receiving a token and then be able to spend it. This enthusiasm can be used to improve attendance levels throughout the university and thus improve the experience of both teachers and students. The shop system is to be established by the university if it is wanted. The QUBAT system is perfectly compatible and, in the future, could integrate a new type of user that was the manager of the shop with a series of product lists and their prices. This system has not been implemented during this project because it is considered outside the objectives of it, although some lines are left open to be implemented in the future, for example, the hashes of the transactions are accessible each time a student spends their QUBATs. These codes could be used to confirm the transaction by the shop manager user.

Finally, at a personal level, the development of the project has been an academic challenge like never. Because of not being English my native language, the effort required, which is high in a project at this level, has had to be even greater and for that reason the satisfaction obtained after having finished has also been much greater. I think that this type of project is completely necessary for our training and to be able to face the demands of the labour market in the very near future.
7.0 Appendices

7.1 Appendix A: Bibliography


7.2 Appendix B: Experiments form template

Please, fill the next form and if you found any inconvenient in any of the tasks asked in the questions explain why you found it. Thank you.

- Did you find any inconvenient to finish the task you were ask for? Which one?
- Did you try to login before registration?
- Did you find any difficulty on following the confirmation instructions given by the system? Which one?
- Once you were logged in, describe the first impression of the main screen design.
- Were you able to capture the attendance code given?
- Did you find hard to buy the chocolate using the token?
- Without watching your smartphone screen again, what is the name of the token you have spent?
- What did you like the most?
- What would you change?
- Would you feel extra motivated on attending to lectures if you could get these tokens there? (and then be able to spend them).
7.3 Appendix C: Experiments results

User 1:

Please, fill the next form and if you found any inconvenient in any of the tasks asked in the questions explain why you found it. Thank you.

- Did you find any inconvenient to finish the task you were ask for? Which one?
  No, it was simple to follow

- Did you try to login before registration?
  No

- Did you find any difficulty on following the confirmation instructions given by the system? Which one?
  No, it was straightforward, really creative system, fits in well with rest of our websites.

- Once you were logged in, describe the first impression of the main screen design.
  Neat and organized layout

- Were you able to capture the attendance code given?
  Yes

- Did you find hard to buy the chocolate using the token?
  No

- Without watching your smartphone screen again, what is the name of the token you have spent?
  Qubat

- What did you like the most?
  The reward for attending class, and incentive to go to class

- What would you change?
  For certain classes you could provide multiple Qubats for guest lectures, etc. To encourage students to participate more.

- Would you feel extra motivated on attending to lectures if you could get these tokens there? (and then be able to spend them).
  Yes, if there were cool prizes.
Please, fill the next form and if you found any inconvenient in any of the tasks asked in the questions explain why you found it. Thank you.

- Did you find any inconvenient to finish the task you were ask for? Which one?
  No, I didn’t.

- Did you try to login before registration?
  No, I didn’t.

- Did you find any difficulty on following the confirmation instructions given by the system? Which one?
  No, I didn’t.

- Once you were logged in, describe the first impression of the main screen design.
  It’s a very minimalistic design where you can find everything clear.

- Were you able to capture the attendance code given?
  Yes

- Did you find hard to buy the chocolate using the token?
  No, it was easy.

- Without watching your smartphone screen again, what is the name of the token you have spent?
  QUBAT

- What did you like the most?
  The idea of creating a new coin for the university avoiding giving money to the banks.

- What would you change?
  I would change the verification pay method because the code is too long.

- Would you feel extra motivated on attending to lectures if you could get these tokens there? (and then be able to spend them).
  Of course, It’s was a really nice idea.
Please, fill the next form and if you found any inconvenient in any of the tasks asked in the questions explain why you found it. Thank you.

- Did you find any inconvenient to finish the task you were ask for? Which one?
  No inconvenience

- Did you try to login before registration?
  No

- Did you find any difficulty on following the confirmation instructions given by the system? Which one?
  No

- Once you were logged in, describe the first impression of the main screen design.
  Easy to understand

- Were you able to capture the attendance code given?
  Yes

- Did you find hard to buy the chocolate using the token?
  Easy transaction process

- Without watching your smartphone screen again, what is the name of the token you have spent?
  QUBAT

- What did you like the most?
  The chocolate
  The name of the coin is very catchy!

- What would you change?
  Different reward to choose from
  (different pizzas cost a certain amount of QUBAT)

- Would you feel extra motivated on attending to lectures if you could get these tokens there? (and then be able to spend them).
  Genuinely, yes.
Please, fill the next form and if you found any inconvenient in any of the tasks asked in the questions explain why you found it. Thank you.

- Did you find any inconvenient to finish the task you were ask for? Which one?
  I didn't find any inconvenient.

- Did you try to login before registration?
  Yes

- Did you find any difficulty on following the confirmation instructions given by the system? Which one?
  No, I didn't find any difficulty, Everything is clear.

- Once you were logged in, describe the first impression of the main screen design.
  It reminds me to Queen's University webpage.

- Were you able to capture the attendance code given?
  Yes, but on the first attempt I used "the capital" i with a "1" lowercase.

- Did you find hard to buy the chocolate using the token?
  No, I didn't.

- Without watching your smartphone screen again, what is the name of the token you have spent?
  $2.95

- What did you like the most?
  Sending the quote, because I find it really original.

- What would you change?
  The scale number, because it is really long and I find it not very aesthetic.

- Would you feel extra motivated on attending to lectures if you could get these tokens there? (and then be able to spend them).
  Yes, for sure, being rewarded for attending class is motivating.
Please, fill the next form and if you found any inconvenient in any of the tasks asked in the questions explain why you found it. Thank you.

- Did you find any inconvenient to finish the task you were ask for? Which one?
  
  No.

- Did you try to login before registration? Yes.

- Did you find any difficulty on following the confirmation instructions given by the system? Which one?
  
  No difficulties.

- Once you were logged in, describe the first impression of the main screen design.

  The main screen is easy to read.

- Were you able to capture the attendance code given?

  Yes.

- Did you find hard to buy the chocolate using the token?

  No, it is a simple system to buy.

- Without watching your smartphone screen again, what is the name of the token you have spent?

  [Handwritten: [ ]

- What did you like the most?

  [Handwritten: It is easy to navigate and a good strategy for managing students.]

- What would you change?

  [Handwritten: Code a it should be stricter.]

- Would you feel extra motivated on attending to lectures if you could get these tokens there? (and then be able to spend them).

  [Handwritten: Yes.]
Please, fill the next form and if you found any inconvenient in any of the tasks asked in the questions explain why you found it. Thank you.

- Did you find any inconvenient to finish the task you were asked for? Which one?
  NO

- Did you try to login before registration?
  NO

- Did you find any difficulty on following the confirmation instructions given by the system? Which one?
  NO

- Once you were logged in, describe the first impression of the main screen design.
  I thought it was quite simple, but also very easy to navigate.

- Were you able to capture the attendance code given?
  Yes, pretty easy

- Did you find hard to buy the chocolate using the token?
  NO

- Without watching your smartphone screen again, what is the name of the token you have spent?
  QUIST

- What did you like the most?
  It was very easy to navigate

- What would you change?
  The design of the page.

- Would you feel extra motivated on attending to lectures if you could get these tokens there? (and then be able to spend them).
  Yes
Please, fill the next form and if you found any inconvenient in any of the tasks asked in the questions explain why you found it. Thank you.

- Did you find any inconvenient to finish the task you were asked for? Which one?
  No

- Did you try to login before registration?
  No

- Did you find any difficulty on following the confirmation instructions given by the system? Which one?
  No

- Once you were logged in, describe the first impression of the main screen design.
  A little bit basic

- Were you able to capture the attendance code given?
  Yes

- Did you find hard to buy the chocolate using the token?
  No

- Without watching your smartphone screen again, what is the name of the token you have spent?
  QUBTK

- What did you like the most?
  The main screen is easy to navigate

- What would you change?
  The design of the main screen.

- Would you feel extra motivated on attending to lectures if you could get these tokens there? (and then be able to spend them).
  Yes.