Team 16 - Biweekly Report 3

Due February 27th

Written by: Sadir Abdul Hadi, Kristelle Feghali and Alexandru Chiriac.

1. Overview

During the reading week, each team member has worked separately on their task. Sadir on the smart contract, Kristelle on the front-end, and Alex on the back-end. The basic part of these three elements is now ready, and we started integrating our work last week. Although we expected this part to be the most challenging, we did not have enough time during scenario week to overcome the problems we faced. Once the scenario week has ended, we continued our work and have now succeeded in integrating some parts. We're generally on track with our initial timeline, which says that next week we should "integrate the front end and the back end".

2. Meetings

Meeting 1 - Monday, February 13th 2017

Attendees: Sadir, Kristelle, Alex

As some team members were away during the reading week, we organised this skype meeting, where we confirmed how the work should we split and discussed the future plans.

Meeting 2 - Wednesday, February 15th 2017

Attendees: Andy Wallace (ATOS), Sadir, Alex

We discussed the project evolution, and demoed our smart contract. Andy has also given us tips for different aspects of the project, from innovation to planning.

Meeting 3 - Saturday, February 25th 2017

Attendees: Sadir, Kristelle

During this meeting, we mainly worked on integrating the front end with the MongoDB database, and building a meteor project which would bring these parts together. Furthermore, each of us demoed what they've done during reading week

Meeting 4 - Sunday, February 26th 2017

Attendees: Alex, Sadir

In this brief meeting Alex demonstrated the user authentication and file upload and showed how the data is stored in the MongoDB database while Sadir updated him on his progress with the smart contract and the issues with lag on the testnet.

Meeting 5 - Monday, February 27th 2017

Attendees: Hirsh, Sadir, Kristelle, Alex

This meeting was mainly for helping each other in our tasks, as some team members were facing some difficulties such as with the routing of the meteor project. We also met with Hirsh, a TA who has experience with ethereum. He told us that an Ethereum testnet will hopefully be provided to us.

3. Tasks Completed

Kristelle:

- Design of our website
- Learned Meteor
- Included my part in a meteor project
- Worked on login and register with meteor

Sadir:

- Smart Contract finalised and tested on TestRPC
- Learned Meteor to help my teammates facing problems
- Created the structure of the meteor project, with a routing system (fixed Kristelle's problem) and currently working on inserting chunks already coded into it.
- Built an authentication system, and currently building a MongoDB database for the website, after having designed it.

Alex:

- Built an authentication demo using the meteor accounts-ui package but also custom built one
- Built a file upload demo in meteor+blaze

Overall:

Overall, our project is starting to shape itself. We believe we are currently at the hardest stage: the integration of the backend and frontend which we hope to get done by the end of this week. Furthermore, the team found itself forced to learn new technologies to help each other.

4. Plan for the next two weeks

Although this might be a bit ambitious, we really hope to have a basic working website in two weeks, namely:

- Integrating the Front-End and Back-End (with its mongoDB and ethereum components).
- Make use of the UCL testnet, which is still under development.
- Research testing tools specifically tailored for the Blockchain

5. Difficulties we faced

- Debugging the routing system
- Deploying the smart contract on the testnet
- Debating among the team whether we should abandon REACT or not, as it’s not native to meteor

6. Individual section

Sadir

During the last two weeks, I finalised the smart contract that should be used. I also tested it and succeeded in uploading text files, hashing them, signing them, and checking their signatures, which constitute the set of main features we need. When I discovered that other team members were struggling with their part, I learned meteor by helping my teammates building an authentication system and a MongoDB database to store strings (links to the documents). In fact, the main issue isn’t the authentication system, but rather the rendering of its component, an issue which has been fixed right before submitting this report.
Another feature the team was struggling with was routing with Meteor, which I just fixed after following several tutorials, and I have now built the website structure using Meteor.
Although I was supposed to deploy the smart contract on the ethereum testnet, I had many difficulties syncing with the database, so I contacted a person who has experience (Hirsh), who told me he was deploying a test-net for UCL, which I’m waiting for. Other difficulties I faced and solved are: testing the smart contact on testRPC.

Kristelle

Throughout the past two weeks, I worked on the design of the website, the navigation and structure. I started from scratch with the html pages, the css pages, and the javascript pages. However, when I was done I added my work in a meteor project and faced a number of issues. First I had problems with routing. I am still working on this issue. Meteor renders the whole html pages and then the page the route should render. I faced this issue with two packages used for routing: iron:router and flow-router. Moreover, I tried to create a login and register form (with blaze, as the one Alex has used uses React, which we decided not to use).

It worked but I still have some problems to fix such as: the username and password appear in the url and no information is given to the user whether they successfully register/ login or not. My goal for the next two weeks is to fix the routing problem and have a register and login form ready.

Alex

Over the past two weeks I completed some advanced tutorials both on meteor and on react and built a fully functional (and secure) demo of an user accounts system (Users can register, sign in, view content of the website only if logged in, password are encrypted and session management is working properly) built with meteor + react, but this might have to be adapted to blaze depending on our approach as a team. I also learned how to upload files by using meteor + blaze but I am currently trying to make it work in a meteor+react application. Currently, I am facing some issues with imports but those should be fixed soon enough. There are various approaches to creating a meteor project, and we have to find common ground during the next couple of days so that we can easily link our components together.