Sprint Retrospective, Iteration # 7

Context Project: Blockchain
Group: BullsBearsWolves

Introduction

This week started off with the leftovers of the conflict that happened last week. So we needed to increase the ambiance in our group. We got the feedback on the requirements from the TA and made a backlog based on these. This is also the reason why the backlog was late. The main objective was to finish the refactor which was already planning, along with features like Keystore and Transaction.

User Story	Issue #	Task	Members worked on	Estimated Effort per Task *(in hours)*	Actual Effort per Task	Done? (yes/no)	Notes
As a developer, I want my program to have a clear structure. Reviewing the UML together and edit if needed before fully implementing it.	190	Reviewing the “Objects Package” section	All	1 / person	1	Yes	
	190	Reviewing the “Controllers” Package section	All	1 / person	1	Yes	
	190	Reviewing the “Database Package” section	All	1 / person	1	Yes	

As a developer, I want to have a library interface (API), which hides implementations.	170	Implementation	Ashay	4	6	Yes	Because the functions I needed from the Database came slowly, I had to implement it step by step which took longer than if I did it with everything available.
	170	Testing	Ashay	4	4	yes	
	170	Documentation	Ashay	2	2	yes	
As a developer, I want the huge BlockController to be split into multiple classes, along with updating outdated methods which still uses code.	140	Implementation of BlockController	Ashay	3	3	yes	
	140	Implementation of TrustValueController	Ashay / Jasper	4	3	yes	
	140	Implementation of BlockVerificationController	Ashay	2	1	yes	
	140	Testing of BlockController	Ashay	2	1	yes	
	140	Testing of TrustValueController	Ashay	1	1	yes	
	140	Testing of BlockVerificationController	Ashay	1	1	yes	
	140	Documentation	Ashay	1	1	yes	
As a developer,	169	Implementation + Testing + documentation of Acquaintance.java	Luat	1	1	yes	

I want to have a structured blockchain implementation for the application according to the UML	172	Implementation + Testing + documentation of User.java	Luat	3	3	yes	
	163	Implementation + Testing + documentation of BlockData.java	Naqib	4	4	yes	
	166	Implementation + Testing + documentation of BlockType.java	Naqib	1	1	yes	
	174	Implementation + Testing + documentation of Chain.java	Luat	4	4	yes	
	167	Implementation + Testing + documentation of Hash.java	Naqib, Luat	2	1	yes	
	168	Implementation + Testing + documentation of Block.java	Naqib, Luat	4	4	yes	
As a developer, I want to spread responsibility in the Queries by creating more query subclasses because I do not want to have a God class.	183	Implementation of UserExist query	Ymte	1	1.5	yes	
	183	Testing the UserExist query	Ymte	1	1.5	yes	Took slightly longer due to bugs in the newly created user table
	184	Implementation of BlockAdd query	Ymte	1	1	yes	

(Refactoring the “Database” section)	184	Testing of BlockAdd query	Ymte	1	2	yes	The block class was heavily modified and needed to be reintegrated
	185	Implementation of ChainSize query	Ymte	1	1/2	yes	
	185	Testing of the ChainSize query	Ymte	1	1/2	yes	
	186	Implementation of the getChain query	Ymte	1	1/2	yes	
	186	Testing of the getChain query	Ymte	1	1	yes	
	187	Implementation of the getUser query	Ymte	1	1	yes	
	187	Testing of the getUser query	Ymte, Luat	1	2	yes	
	187	Documentation	Ymte	1	1	yes	
As a developer, I want to clean the project folder of unnecessary folders and files.	158	Cleaning the project	Jasper	1	1/2	yes	
As a developer, I want to find a solution for the failing jobs on Travis due to a timeout.	159	Reading	Jasper	2	2	yes	
	159	Implementation	Jasper	1	8	yes	The travis documentation said that you should just add the ‘travis_wait’ property to a job. However, this documentation was so vague that it took a

							long time to check where it went wrong in the travis build and where exactly it was needed to add the ‘travis_wait’ attribute. Next to that, the travis builds took really long.
As a developer, I want to use the Android KeyStore to save the public and private keys.	160	Testing	Jasper	2	3	yes	The testing of the Android Keystore was written. However, the test could not be run, because the private/ public keypair generated using the ED25519 protocol, could not be saved into the Android keystore; you had to generate a X509Certificate using the public and private key pair. And following the Android documentation, you should have used the BouncyCastle library. Although the BouncyCastle library supported Elliptic curve protocols, it did not support our particular one: the edwards curve. So after trying a lot of other solutions, it did not work out. So instead of the Android Keystore, we stored the private and public keys as seperate files, encrypted using the AES-CTR protocol with the hex

							representation of the public key as password.
	160	Implementation	Jasper	2	5	yes	See above
	160	Documentation	Jasper	1	1/2	yes	
As a developer, I want to use the Android KeyStore to read the public and private keys.	161	Testing	Jasper	2	2	yes	
	161	Implementation	Jasper	2	2	yes	
	161	Documentation	Jasper	1	1	yes	
As a developer, I want to sign the transaction upon sending it to another user.	165	Testing	Jasper	1.5	1	yes	
	165	Implementation	Jasper	2	2	yes	It is implemented in: https://github.com/deepweeb/B-B-W/commit/d8920fccc8fdbafe6800d77f987dead1f04b05e4 However, the view package is removed, so it is not used anymore.
	165	Documentation	Jasper	0.5	0.5	yes	

As a developer, I want to verify a block upon receiving the transaction from someone else.	164	Testing	Jasper	1.5	1	yes	
	164	Implementation	Jasper	2	2	yes	This has been implemented. However, since the activities have not been adjusted yet, or have been deleted, it did not work yet.
	164	Documentation	Jasper	0.5	0.5	yes	
As a developer, I want to create a method which represents sending a block to another user.	165	Testing	Jasper	2	1	yes	
	165	Implementation	Jasper	3	2	yes	
	165	Documentation	Jasper	1	1	yes	
As a developer, I want to have a database class on which I can perform my custom queries.	188	Implementation	Ymte	3	4	yes	A new user table had to be created which caused slight delay
	188	Testing	Ymte	3	0	yes	The database is tested not through itself but through the queries.
	189	Integration	Ymte	4	0	no	The queries are all individually integrated into the database
As a developer, I want to integrate the	177	Implementation	Luat	4	6	yes	The code of BlockChain package should be ready before the integration, but

refactored Database Package with the Blockchain Component Package.							there were many bugs that needed to be fixed according to the updated UML. Also some requested methods from various members have been added too.
	177	Adjusting the tests	Luat, Ymte	3	10	yes	Much of the existing code and tests were used in parallel with the pre refactor code and are not yet compatible with the new UML. Tests would fail and we needed to debug it.
	177	Documentation	Luat	2	2	yes	
As a developer, I want to integrate the refactored Database Package with the Controller/Handlers Package.	173	Implementation	Ashay	2	4	yes	
	173	Adjusting the tests	Ashay	1	1	yes	
	173	Documentation	Ashay	1	1/2	yes	
As a developer, I want to integrate the refactored Blockchain Components Package with the	175	implementation	Ashay	4	1	yes	
	175	Adjusting the tests	Ashay	1	1	yes	
	175	Documentation	Ashay	1	1/2	yes	

Controller/Handlers Package.							
As a developer, I want to do the assignment about Project skills (due Friday		Doing the assignment	All	2 / person	2	yes	
As a developer, I want to refactor the activities in such a way that it only uses the API.	170	Refactoring MainPageActivity	Naqib	1	2	yes	
	170	Adjust tests in the MainActivity	Naqib	1	5	no	We tried again to test this. A label 'help wanted' was added, but after a discussion we decided to let this go for now. On monday we tried the test, so this task was for our previous GUI. We have decided to minimize the activities such that we could test it in another way. This would also mean that a lot of lines will be removed and the overall test coverage will be higher.
	176	Refactoring ContactPageActivity	Naqib	1	1	yes	
	176	Testing the refactored ContactsPageActivity	Naqib	1	1	no	We tried to test this on monday, because it had other methods. Still no coverage.

	178	Refactoring FriendsPageActivity	Naqib	1	1	yes	
	178	Testing the refactored FriendsPageActivity	Naqib	1	0	no	No time left for this task.
	179	Refactoring the PairPageActivity	Naqib	1	1	yes	
	179	Testing the refactored PairPageActivity	Naqib	1	0	no	No time left for this task.
	180	Refactoring the TransactionPageActivity	Naqib	1	1	yes	
	180	Testing the refactored TransactionPageActivity	Naqib	1	0	no	No time left for this task.
	182	Refactoring DisplayContactFriendsPageActivity	Naqib	1	1	yes	
	182	Testing the refactored DisplayContactFriendsPageActivity	Naqib	1	0	no	No time left for this task.
As a developer I want to study the TU Delft TrustChain documentation as Johan Pouwelse has implied.	181	Reading: TrustChain: A sybil-resistant Scalable Blockchain	Naqib	4	4	yes	
	181	Analysing: How we can implement this in our application (might be documented)	Naqib, Luat	4	2	yes	

What went well

1. The PMD / findbugs and checkstyle errors were fixed while working in your own branch. This way we did not have to use someone else's code full of errors.
2. We did not merge as long the build did not pass. This way some PR’s were open for four days (Travis took a long time for each build). But this should result in a better grade.
3. The pull requests were well reviewed. The comments made sense and this resulted in code improvement. When the formatting was not perfect there was just one comment ‘Reformat your code’ instead of commenting for every ‘error’.
4. Everyone was almost everyday one time and nobody left before 12.30.
5. Some tasks were very dependent on each other and the responsible members cooperated well in order to make it work.
6. The ‘kut punten system’ was updated correctly. This way (most of) the members were sharp.
7. Almost every task is done according to the sprint backlog. We have a good feeling about this week.
8. We made a lot of code change. The old code is very much improved. The new code have a proper structure and is written according to the UML.
9. We wrote the retrospective together so this way everyone knows the status of the project.
10. All the members worked professionally together. The members respected each other. When one of the member did not agree with another one, he told the person instead of keeping it to himself and get frustrated.
11. We still made sure that person that creates a pull request is not the same person that merges the changes into develop. This way everyone was responsible for the changes and the last person that approved had to merge. This person had to check whether the changes were made. This resulted in not merging when for example the build did not pass.

Which improvements from the previous sprint did we apply?

1. Two members of our team were not present during a meeting. One of the member was too late. We had the penalty of working for the next two days from 9 am till 7 pm. Those members were truly present between these hours.
2. Every members updated the EAD according to their changes.
3. We did not merge until every member fixed their own PMD / checkstyle/ findbugs.
4. Our main priority was refactoring the whole code according to point 7 from the previous retrospectief.
5. The XML files were not changed, but that is because we are going to change the activities (*see sprint backlog week 8)*.

What went wrong

1. Before creating a PR the formatting was not always correct. This way a member had to make a comment about this.
2. At the end of the previous sprint (Friday, 09/06) there was ‘fight’ between two members. On monday one member was not present and left a message that he had contacted Martijn and this would be a matter of continuing with four or five members. On tuesday the members knew that we would continue with all members. Contacting Martijn was not necessary as we could have solved this internally.
3. Not all tasks were finished before thursday. This way we could not completely test the end version of the API via the activities (issue #170, 178, 176, 179, 180 and 182). There is a chance that all the commits on the apiActivities branch will not make the master this week. This way it would look like one member did not contribute a lot to the code. We would recommend to look at the overall commits.
4. Travis took a really long time to run the tests. Almost every running task took around 2 hours to finish and even then, some jobs failed, because the android emulator would run out of time (see images below). So we had to restart the jobs again. So we had to wait a really long time for the travis task to finish and thus also wait a really long time for a pull request to merge. Although the code was correct. At the time of writing, an attempt to fix this issue have been available (seems valid too) & we are giving Travis time to rebuild with the fix.

```
2326
2327 No output has been received in the last 10m0s, this potentially indicates a stalled build or something wrong with the build itself.
2328 Check the details on how to adjust your build configuration on: https://docs.travis-ci.com/user/common-build-problems/#Build-times-out-because-no-output-was-received
2329
2330 The build has been terminated
2331
```

```
136
137 Still running (20 of 20): ./gradlew build connectedCheck check lint test
138 Timeout (20 minutes) reached. Terminating "./gradlew build connectedCheck check lint test"
139
140
141 The command ./gradlew build connectedCheck check lint test exited with 137.
```

5. Some pull requests contained still more tickets than were allowed (4 or 5).
6. The deadline for ‘final PR’ on thursday 6 pm was not completely lived by. This way we do not know whether the builds could be fixed. As you can see in issue #159 the building was more time consuming than we expected. If all the members had lived by the rule we would have a bigger chance to make the build pass.

How should we improve

1. If one of the members is more than an half hour too late, without a legit reason, he must work at school for the next day from 9 am to 7 pm.
2. If a member is absent because of his/her own personal reason during regular meeting hours (from 9:30 to 12:30), for example a personal appointment, this member will have to give a reasonable reason to the group, and will have to be present from 9 am to 7 pm at the university for the next two (2) working days (any dishonest will NOT be tolerated if found) or should have finished their tasks, which have been set at the daily sprint.
3. The pull requests should be split up to address 1 or max 2 issues (*see also bullet 8*).
4. Since the GUI is not our primary task and it contains a lots of activities we should refactor it such that it only displays the features that we have with a minimalistic amount of methods. This way our tests will also grow.
5. The EAD should not be updated on a detail level. Martijn is not interested in the packages and classes. It is more about the components. This is a huge priority, since next friday the final version should be submitted.
6. The refactor of the main code should have the utmost priority. Almost all new implementations are dependent on the large refactor.

7. If someone does not have finished his task before the weekly deadline and does not have a good reason (or reasonable amount of invested effort) for it, he will be in trouble. A letter will be written to the TA with the proposal to exclude this member out of the group.
8. The issue of oversized pull requests roared its ugly head again for Luat and Ymte. The Travis build did not succeed so that in the beginning relatively small pull requests could not be merged. They made the mistake of continuing in the same branch (and thus the same pull request) when further developing their code, while they should have moved to a different branch that was branched off their original branch. This also happened because of the delay in approval of the pull requests.
9. We should never submit an empty sprint backlog again. Even is this would mean that we have to change the course on monday.