

Endbericht Projekt Kyas Coin

Auftraggeber:

- Name: Andreas Villscheider
- Email Adresse: andreas.villscheider@schuele.suedtirol.it
- Telefonnummer: +39 334 804 2516
- Adresse: Unterdrittelgasse 16, Brixen

Auftragnehmer:

- Alexander Plaikner
 - Email Adresse: alexander02plaikner.ap@gmail.com
 - Telefonnummer: +39 340 127 8474
 - Adresse: Im Peuren 28, St.Sigmund, Kiens
- Matteo Planker
 - Email Adresse: plankermatteo23@gmail.com
 - Telefonnummer: +39 331 871 0241
 - Adresse: Daunei 11, Wolkenstein
- Peter Mantinger
 - Email Adresse: mantingerpeter2@gmail.com
 - Telefonnummer: +39 366 115 0413
 - Adresse: Gufidaun 122

Inhaltsverzeichnis

1	Code Analyse	3
1.1	Python Masternode	3
1.2	Python Client	3
1.3	JavaFx Client	3
1.4	Android Wallet	3
2	Zeitaufwand	4
3	Rückmeldung d. Auftraggebers	5
3.1	Final Sprint Review mit Produktübergabe an den Kunden	5
3.2	Produktinkrements	5
3.3	Die Android Applikation	5
3.4	Die Java Applikation	5
3.5	Offene Punkte Liste	6

1 Code Analyse

1.1 Python Masternode

Es wurden bei der Entwicklung des Masternode-Codes andere Blockchains (Bitcoin, Monero, Chainlink) als Vorlage verwendet, also verlief die Implementierung der Grundlegende Struktur des Masternodes relativ reibungslos. Die Planung und Entwicklung der Blockchain-spezifischen Use-Cases, wie Coin Burn und Sign-Up Bonus waren hingegen sehr Zeitaufwendig.

1.2 Python Client

Beim Python Client handelt es sich um eine CLI (Command-Line Interface), welche vielleicht weniger Benutzerfreundlich ist, aber einfacher zu entwickeln ist. Insgesamt ist der Python Client eine Oberfläche um einfach API Calls zu machen und so mit der Blockchain interagieren zu können.

1.3 JavaFx Client

Die Entwicklung des JavaFx Clients verlief fast vollkommen reibungslos. Die Benutzeroberfläche wurde nach dem Beispiel des Bitcoin Wallets "BTC.com", ein Favorit unter Bitcoin-Enthusiasten, modelliert.

Es gab einige wenige Schwierigkeiten bei der Implementierung der API Calls, da die Responses mit Code zu interpretieren ohne externe Bibliotheken etwas komplex ist. In Zukunft könnte man RxJava verwenden, um automatisch Responses in POJO (Plain-Old Java Objects) zu konvertieren.

1.4 Android Wallet

Die grundlegende Struktur des Android Wallets lag dank eines alten Projekts teilweise schon vor, ansonsten wäre es nicht möglich gewesen in dieser Zeit eine App noch zu entwickeln.

Die App besteht aus einigen LiveData/Observer Stacks, d.h. dass die meisten Funktionen asynchron ablaufen und so relativ "bulletproof" ist. Die API-Calls in der App werden mit einem HTTP-Client (Retrofit2), Response-Converters (RxJava2CallAdapterFactory) und Datentyp-Converters (Gson-ConverterFactory) ausgeführt, was der Sauberkeit und "future-proofness" des Clients dienen.

2 Zeitaufwand

Alle Zeiten in IDEs wurden mit dem Plugin "Waka Time" aufgenommen, was inaktive Zeiten von der Gesamtzeit wegrechnet.

Aufgabe	Zeit (Stunden)	Auftragnehmer
Python Masternode		
Recherche und Planung	5	Team
Entwicklung (nano auf Server)	5	A. Plaikner
Entwicklung (PyCharm IDE)	25	A. Plaikner
Python Client		
Entwicklung (PyCharm IDE)	15	A. Plaikner
JavaFx Client		
Layout (Scenebuilder IDE)	3	P. Mantinger
Entwicklung (IntelliJ IDE)	10	A. Plaikner
Android Wallet		
Basics (vorhanden)	10	M. Planker
Layout (AndroidStudio IDE)	1.5	M. Planker
Entwicklung (AndroidStudio IDE)	9.5	M. Planker
Testing	2	M. Planker
Dokumentation		
Pflichtenheft (inkl. Diagramme)	10	Team
Endbericht	1.5	Team

3 Rückmeldung d. Auftraggebers

3.1 Final Sprint Review mit Produktübergabe an den Kunden

Meeting: https://meet.fallout.school/4AT	Datum: 29.04.2020
Startzeit: 14:00	Endezeit: 14:45
Teilnehmer Scrum Team: - Alexander Plaikner - Matteo Planker - Peter Mantinger	Teilnehmer Stakeholder: - Andreas Villscheider

In diesem letztem Sprint Review werden dem Stakeholder im weiteren Verlauf auch als Kunde bezeichnet, folgende Produktinkrements ausgeliefert.

- Graphische Oberfläche für den JAVA Client (JavaFX)
- Graphische Oberfläche für den Mobile Client (AndroidApp)

Diese Inkrements werden im Laufe des Meetings besprochen wie auch ausführlicher getestet.

3.2 Produktinkrements

Der Kunde aktualisiert sein lokales GIT-Repository und erhält somit Zugriff auf die neuesten Feature, welche im letzten Sprint erarbeitet wurden. Das Inkrement welches Gegenstand des ersten Test werden soll, ist die Android-Applikation, welche in Form einer APK Datei ausgeliefert wurde. Das zweite Augenmerk gilt der Java Anwendung, welche als IntelliJ-Projekt im Repository enthalten ist.

3.3 Die Android Applikation

Die APK lässt sich problemlos installieren und startet auch sehr schnell. Die App ist sehr minimalistisch gehalten, wirkt dadurch aber recht sauber und übersichtlich. Somit gelingt es dem Kunde, ohne weitere Erklärungen, eine neue Adresse erstellen. Leider gibt es aber dann ein Problem, so dass der versprochene Signup Bonus vom Masternode nicht überwiesen wird und deshalb auch keine Transaktionen getestet werden kann. Die Ursache des Problems kann nicht sofort geklärt werden aber das Entwicklerteam verspricht diesem sofort nach dem Meeting nachzugehen. Ersatzweise wird dem Kunden die Funktionalität der App in einem Emulator gezeigt, wo alles reibungslos funktionierte. Unter anderem konnten jetzt auch Transaktion getätigt werden, welche in der Blockchain dann auch aufscheinen.

3.4 Die Java Applikation

Der Java Client wurde von den Entwicklern als JavaFX Projekt realisiert und wirkt ebenfalls recht übersichtlich. Der Kunde hat mit diesem Client nun neben den Funktionalitäten, welche er bereits in der APP sah auch die Möglichkeit das Minen zu testen und sich einen Masternode auszusuchen. Alles scheint reibungslos zu funktionieren auch wenn es bisher nur einen Masternode gibt.

3.5 Offene Punkte Liste

Für eine endgültige Abnahme möchte der Kunde noch folgende offene Punkte abgeklärt wissen:

- Oben beschriebenes Problem der APK
- Ausbau des Netzwerks um weitere Masternodes zur Erhöhung der Sicherheit