



Verbale esterno del 2023-04-14 Progetto Trustify

pentasoftswe@gmail.com

Informazioni sul documento

Responsabile	Stefano Meneguzzo
Redattori	Marco Brugin
Verificatori	Nicola Lazzarin
Uso	Interno/Esterno
Destinatari	Gruppo <i>PentaSoft</i>

Sommario

Vengono riportati gli argomenti discussi durante la riunione del 14 Aprile 2023 con l'azienda Sync Lab S.r.L.

Contenuti

1	Generale	1
1.1	Informazioni sulla riunione	1
1.2	Ordine del giorno	1
2	Svolgimento	2
3	Tracciamento delle decisioni	3

1 Generale

1.1 Informazioni sulla riunione

- **Luogo:** Videochiamata Google Meet;
- **Ora di inizio:** 10:00;
- **Ora di fine:** 11:00;
- **Partecipanti Interni:** Nicola Lazzarin, Pietro Lauriola, Stefano Meneguzzo, Marco Brugin, Marco Rosin, Luca Marcato;
- **Partecipanti Esterni:** Matteo Galvagni;

1.2 Ordine del giorno

L'incontro effettuato è servito per discutere ed espandere i seguenti argomenti:

- stato di avanzamento sull'implementazione della componente di **backend**;
- discussione sulla progettazione della componente di **front-end**;
- discussione sulla progettazione della componente di **API**;
- discussione sulla modalità d' implementazione dei test sulla componente di **API** attraverso la libreria *Web3jG*.

2 Svolgimento

1. Stato di avanzamento componente di backend

D: Per quanto riguarda l'eliminazione delle recensioni, l'eliminazione si deve ripercuotere direttamente anche sullo *Smart Contract_G*?

R: In sè l'eliminazione non ha molto senso, perchè comporta un costo in termini di computazione all'interno della *Blockchain_G*, quindi è meglio continuare a restituire tutte le recensioni eventualmente andandole a marcare con un flag, per poi discriminare a livello di **frontend** le recensioni che sono o meno state eliminate.

2. Progettazione e implementazione componente di frontend

D: Per quanto riguarda l'esecuzione di un pagamento come si può migliorare l'esperienza utente in termini di approvazione e spesa di una determinata quantità di *token_G*?

R: Per migliorare l'esperienza utente è bene andare ad inserire un check che verifichi se l'utente abbia già approvato una quantità sufficiente al pagamento, in caso contrario dovrà essere richiesta l'approvazione. Inoltre nel momento in cui si va a effettuare una *transazione_G*, dato che si tratta di una operazione asincrona è inserire un pop-up che blocchi eventuali altre operazione dell'utente e gli segnali cosa sta effettivamente accadendo fino a *transazione* effettivamente eseguita. Per il resto l'interfaccia può andare bene, altri filtri oltre a quello per indirizzo potremmo eventualmente pensarli più avanti.

3. Progettazione e implementazione componente API

D: Per quanto riguarda le *REST-API_G* ha senso restituire anche un errore nell'eventualità che il range di recensioni richieste non sia corretto?

R: Si ha senso, è bene però che l'errore restituito sia parlante nei confronti dell'utente e che si vadano a gestire anche gli errori *HTTP_G*.

4. Modalità d' implementazione dei test sulla componente API attraverso la libreria *Web3j*

D: Per quanto riguarda le *REST-API_G* ci risulta difficile effettuare i test su tale componente?

R: La libreria in sè non è strutturata in modo ottimale, vi consiglio, dato che le *REST-API* utilizzeranno gli stessi metodi dello *Smart Contract_G*, di andare prima a concludere in modo definitivo quest'ultimo per poi andare a effettuare il deploy sulla *testnet_G* ed effettuare i test da lì.

3 Tracciamento delle decisioni

ID	Decisione
VE_2023_04_14-1	Implementazione gestione visuale recensioni eliminate
VE_2023_04_14-2	Implementare check di verifica sulla approvazione dei $token_G$
VE_2023_04_14-3	Implementare check di verifica sull'effettivo completamento della $transazione_G$
VE_2023_04_14-4	Implementare eventuale schermata di attesa per l'utente in caso di attese
VE_2023_04_14-5	Restituire attraverso la $REST-API_G$ anche eventuali errori parlanti all'utente