



Winning Software Solution

winningsoftwaresolution@gmail.com

ShopChain

SyncLab

Analisi dei requisiti - v1.0.1

Informazioni

<i>Redattori</i>	Alberto Nicoletti Andrea Volpe
<i>Revisori</i>	Giovanni Cocco
<i>Responsabili</i>	Elia Scandaletti
<i>Versione</i>	1.0.1
<i>Uso</i>	esterno

Descrizione

Analisi dei requisiti del progetto ShopChain con casi d'uso e requisiti.

Versione	Data	Persona	Attività	Descrizione
1.0.1	01/03/2022	Giovanni Cocco	Redazione	Approfondimento requisiti funzionali
1.0.0	09/02/2022	Elia Scandaletti	Approvazione	Approvazione per RTB
0.1.2	07/02/2022	Andrea Volpe	Redazione	Modifica dei requisiti come concordato nell'incontro del 04/02/2022
0.1.1	05/02/2022	Alberto Nicoletti	Redazione	Adeguamento casi d'uso a tecnologie scelte
0.1.0	04/02/2022	Giovanni Cocco	Revisione	Correzioni varie
0.0.3	21/01/2022	Andrea Volpe	Redazione	Riorganizzazione requisiti
0.0.2	21/01/2022	Alberto Nicoletti	Redazione	Stesura requisiti
0.0.1	03/01/2021	Andrea Volpe	Redazione	Stesura casi d'uso
0.0.0	11/12/2021	Alberto Nicoletti	Redazione	Strutturazione del documento

1 Introduzione

1.1 Scopo del documento

Lo scopo del documento è raccogliere i risultati dell'attività di analisi dei requisiti. Contiene quindi la descrizione dei casi d'uso del prodotto software da sviluppare, ed i requisiti suddivisi per tipologia. Si vuole così dimostrare una completa comprensione del problema e delle aspettative della soluzione. I casi d'uso, ma soprattutto i requisiti saranno tenuti in considerazione nelle fasi di progettazione, di verifica e di validazione.

2 Descrizione del prodotto

L'azienda *SyncLab* propone, attraverso il capitolato C2: *ShopChain - Exchange Platform on Blockchain*. L'obiettivo è sviluppare un sistema di pagamento sicuro e *super partes* per E-commerce che trattenga i fondi durante la spedizione e che li sblocchi all'arrivo del pacco. Ciò consiste nella realizzazione su blockchain di un contratto digitale che si incarichi di ricevere l'ammontare in criptovaluta, lo trattenga, e lo consegni al venditore solo quando il pacco viene recapitato all'acquirente.

2.1 Scopo del prodotto

Il progetto consiste nello sviluppo di una piattaforma su blockchain con lo scopo di rendere automatizzato e sicuro lo smistamento dei fondi da clienti a E-commerce. Il processo di trasferimento del denaro avviene seguendo queste fasi:

1. caricamento dei dati dell'ordine di acquisto su contratto digitale;
2. trasferimento del denaro dal wallet dell'acquirente al contratto;
3. notifica al venditore dell'avvenuto pagamento;
4. conferma di ricezione del pacco da parte dell'acquirente tramite scannerizzazione di un QR Code sul pacco del prodotto acquistato;
5. invio del denaro dal contratto al wallet del venditore.

2.2 Parti del prodotto

Il prodotto software è composto dalle seguenti parti:

- smart contract nella blockchain per la gestione di tutte le fasi del processo di trasferimento del denaro;
- script per la messa in vendita automatizzata su contratto da integrare nel backend del venditore;
- landing page per il pagamento da parte dell'acquirente;
- piattaforma web per la visualizzazione e gestione delle transazioni da parte di sia venditore sia acquirente;
- webapp che consente lo sblocco dei fondi dal contratto.

2.3 Caratteristiche utenti

Gli utenti di *ShopChain* possono essere suddivisi in due categorie:

- venditore: gli amministratori di un sito di e-commerce che vogliono aggiungere *ShopChain* come metodo di pagamento;
- acquirente: I clienti di un sito di e-commerce che scelgono di utilizzare *ShopChain* come metodo di pagamento per i prodotti da acquistare.

Tutti gli utenti devono essere in possesso di un wallet compatibile con la blockchain scelta per questo Progetto. Non potendo prevedere con accuratezza quanti e quali e-commerce decideranno di utilizzare *ShopChain* altre considerazioni sulle caratteristiche di utenza sono superflue. Il prodotto deve essere facilmente integrabile in quante più tipologie di e-commerce possibili.

2.4 Vincoli e preferenze

Il proponente non impone vincoli nella scelta delle tecnologie, ma ci sono comunque dei suggerimenti da considerare:

- utilizzo di blockchain pubblica;
- utilizzo di Java e Angular per lo sviluppo delle parti di Back-end e di Front-end della componente web application del sistema;
- utilizzo di database PostgreSQL.

Per il completamento del progetto il proponente richiede che siano ottenuti i seguenti risultati:

- server, completo di UI;
- test che dimostrino il corretto funzionamento dei servizi e delle funzionalità previste, con una copertura minima dell'80% correlata di report;
- documentazione su scelte implementative e progettuali effettuate, le relative motivazioni, i problemi aperti e le eventuali soluzioni proposte da esplorare.

3 Casi d'uso

3.1 Attori

- **Venditore con wallet connesso:** venditore che ha già connesso il proprio wallet con Metamask;
- **venditore con wallet non connesso:** venditore che non ha ancora connesso il proprio wallet con Metamask;
- **acquirente con wallet connesso:** acquirente che ha già connesso il proprio wallet con Metamask;
- **acquirente con wallet non connesso:** acquirente che non ha ancora connesso il proprio wallet con Metamask;
- **e-commerce:** backend dell'e-commerce;
- **metamask:** applicativo di terze parti per la gestione dei wallet;
- **database:** persistenze SQL interna al nostro sistema.

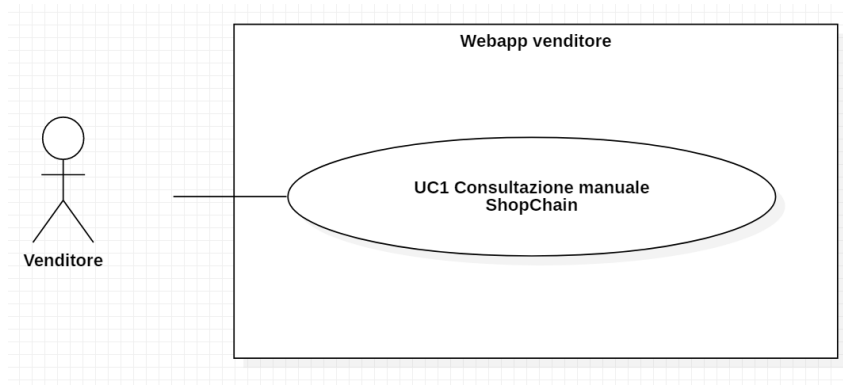


Figure 1: Caso d'uso UC1

UC1 - Consultazione manuale ShopChain

Attori primari: Venditore con wallet connesso.

Precondizioni: Il venditore usa il servizio ShopChain e vorrebbe avere più informazioni sul suo utilizzo.

Postcondizioni: Il venditore consulta il manuale utente di ShopChain.

Scenario principale:

1. Al venditore viene fornito il manuale utente già dall'acquisizione del prodotto ShopChain.
2. Il venditore è libero di consultare il manuale in ogni momento.

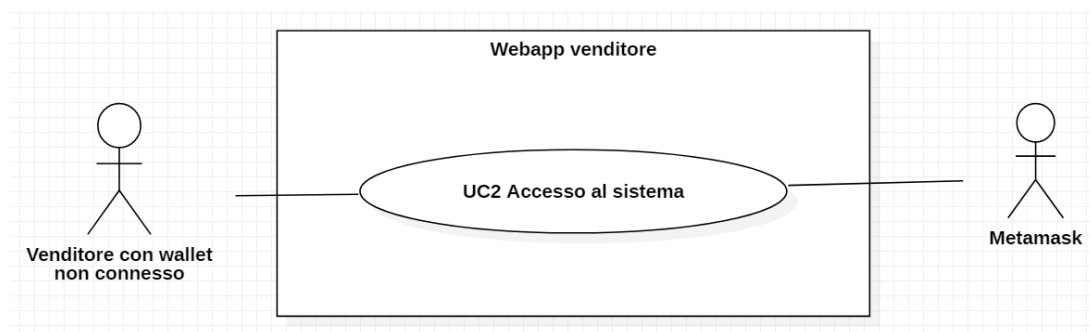


Figure 2: Caso d'uso UC2

UC2 - Accesso al sistema (webapp venditore)

Attore primario: Venditore con wallet non connesso.

Attore secondario: Metamask.

Precondizioni: Il venditore vuole accedere alla webapp.

Postcondizioni: Il venditore si è connesso alla webapp con Metamask.

Scenario principale: Il venditore usando Metamask, si connette alla webapp con il proprio wallet.

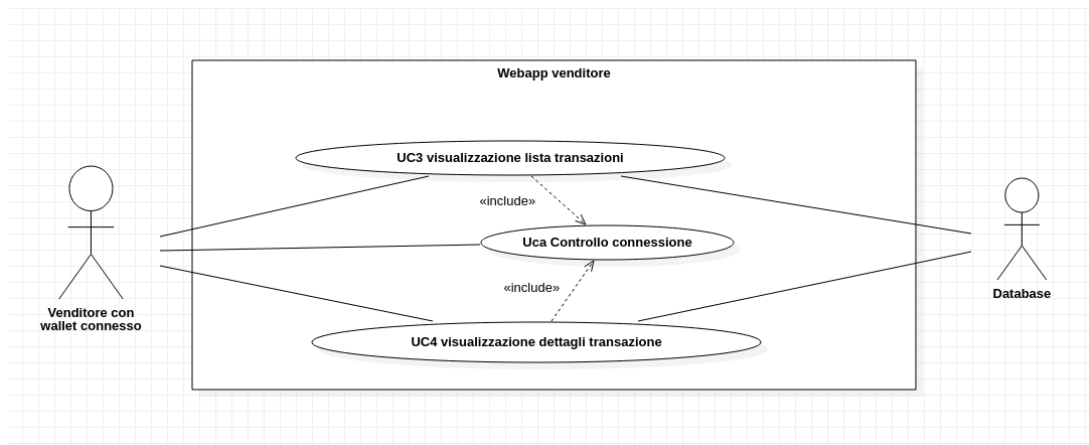


Figure 3: Casi d'uso UC3 e UC4

UC3 - visualizzazione lista transazioni

Attore primario: Venditore con wallet connesso.

Attore secondario: Database.

Precondizioni: Il venditore ha effettuato l'accesso al sistema.

Postcondizioni: Il venditore vede la lista delle transazioni.

Scenario principale:

1. Il venditore vede la lista delle transazioni;
2. Il venditore può scegliere che tipo di transazioni vedere (tutte, completate, in attesa);
3. Il sistema mostra al venditore l'elenco delle transazioni richieste.

UC4 - visualizzazione dettaglio transazione

Attore primario: Venditore con wallet connesso.

Attore secondario: Database

Precondizioni: Il venditore ha acceduto alla funzionalità di visualizzazione della lista delle transazioni.

Postcondizioni: Il venditore vede i dettagli di una singola transazione.

Scenario principale:

1. Il venditore seleziona una delle transazioni dall'elenco;
2. Il venditore visualizza i dettagli della singola transazione.

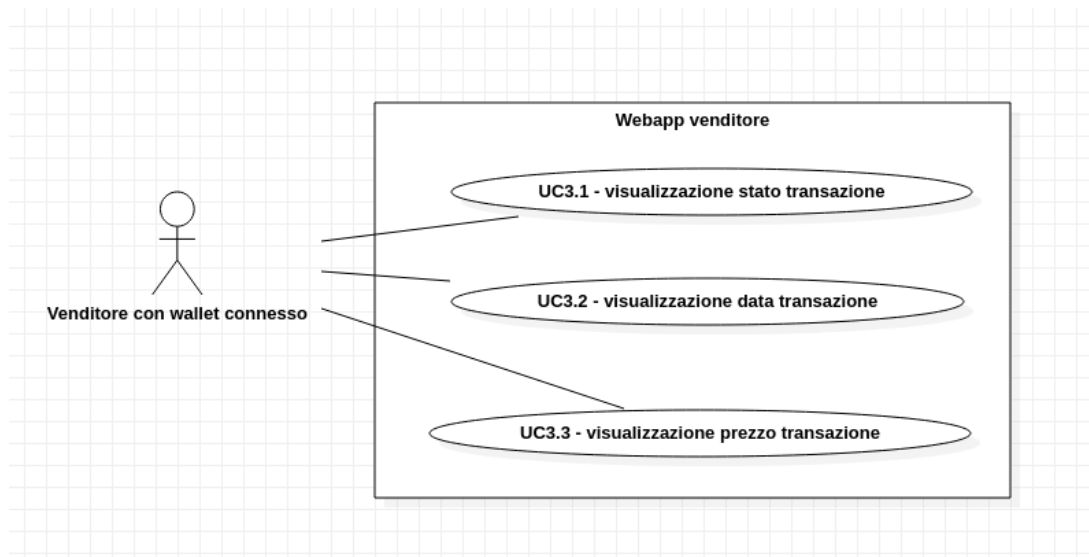


Figure 4: Casi d'uso UC3.1, UC3.2 ed UC3.3

UC3.1 - Visualizzazione stato transazioni

Attore primario: Venditore con wallet connesso.

Precondizioni: Il venditore vuole visualizzare lo stato delle transazioni.

Postcondizioni: Il venditore visualizza lo stato delle transazioni.

Scenario principale: Viene visualizzato lo stato delle transazioni.

UC3.2 - Visualizzazione data transazioni

Attore primario: Venditore con wallet connesso.

Precondizioni: Il venditore vuole visualizzare la data delle transazioni.

Postcondizioni: Il venditore visualizza la data delle transazioni.

Scenario principale: Viene visualizzato la data delle transazioni.

UC3.3 - Visualizzazione prezzo transazioni

Attore primario: Venditore con wallet connesso.

Precondizioni: Il venditore vuole visualizzare il prezzo delle transazioni.

Postcondizioni: Il venditore visualizza il prezzo delle transazioni.

Scenario principale: Viene visualizzato il prezzo delle transazioni.

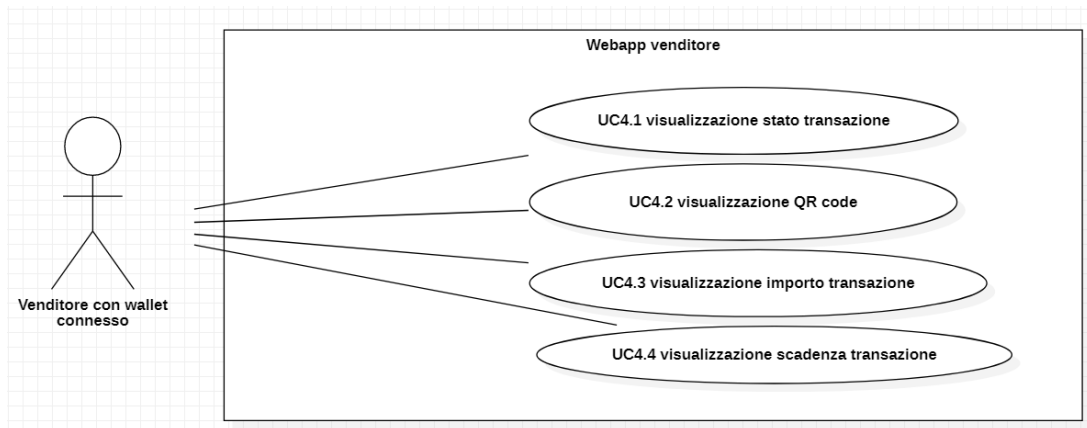


Figure 5: Casi d'uso UC4.1, UC4.2, UC4.3 ed UC4.4

UC4.1 - Visualizzazione stato transazione

Attore primario: Venditore con wallet connesso.

Precondizioni: Il venditore vuole visualizzare lo stato di una transazione.

Postcondizioni: Il venditore visualizza lo stato di una transazione.

Scenario principale: Viene visualizzato lo stato della transazione.

UC4.2 - Visualizzazione QR Code

Attore primario: Venditore con wallet connesso.

Precondizioni: Il venditore vuole ottenere il QR Code di una transazione in attesa.

Postcondizioni: Il venditore è in possesso del QR Code da applicare sul pacco del prodotto.

Scenario principale:

1. Viene visualizzato il QR Code della transazione in attesa selezionata;
2. Il venditore può stampare il QR Code applicarlo sul pacco del prodotto.

UC4.3 - Visualizzazione importo transazione

Attore primario: Venditore con wallet connesso.

Precondizioni: Il venditore vuole visualizzare l'importo di una transazione.

Postcondizioni: Il venditore visualizza l'importo di una transazione.

Scenario principale: Viene visualizzato l'importo della transazione in dollari.

UC4.4 - Visualizzazione scadenza transazione

Attore primario: Venditore con wallet connesso.

Precondizioni: Il venditore vuole visualizzare la scadenza di una transazione in attesa.

Postcondizioni: Il venditore visualizza la scadenza di una transazione in attesa.

Scenario principale: Il venditore vede la data della scadenza di una transazione in attesa, oltre la quale l'acquirente riceverà indietro la valuta bloccata.

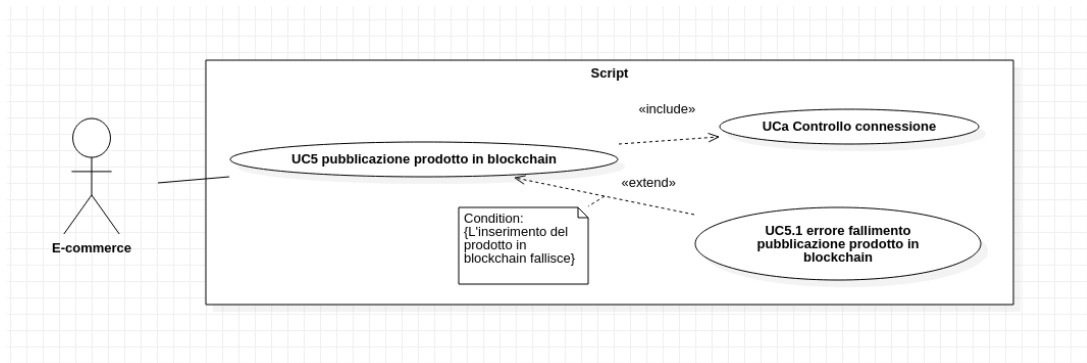


Figure 6: Casi d'uso UC5 e UC5.1

UC5 - Pubblicazione prodotto in blockchain

Attore primario: E-commerce.

Precondizioni: L'e-commerce ha integrato lo script nel proprio backend.

Postcondizioni: Lo script nell'e-commerce ha pubblicato il nuovo prodotto in blockchain.

Scenario principale:

1. Il venditore inserisce un nuovo prodotto nell'e-commerce;
2. Tramite lo script il prodotto viene pubblicato in blockchain;
3. Il prodotto è pubblico in blockchain e utilizzabile per fare nuove transazioni.

Estensione: UC5.1 Errore fallimento pubblicazione prodotto in blockchain: lo script salva in un file di log il messaggio di errore.

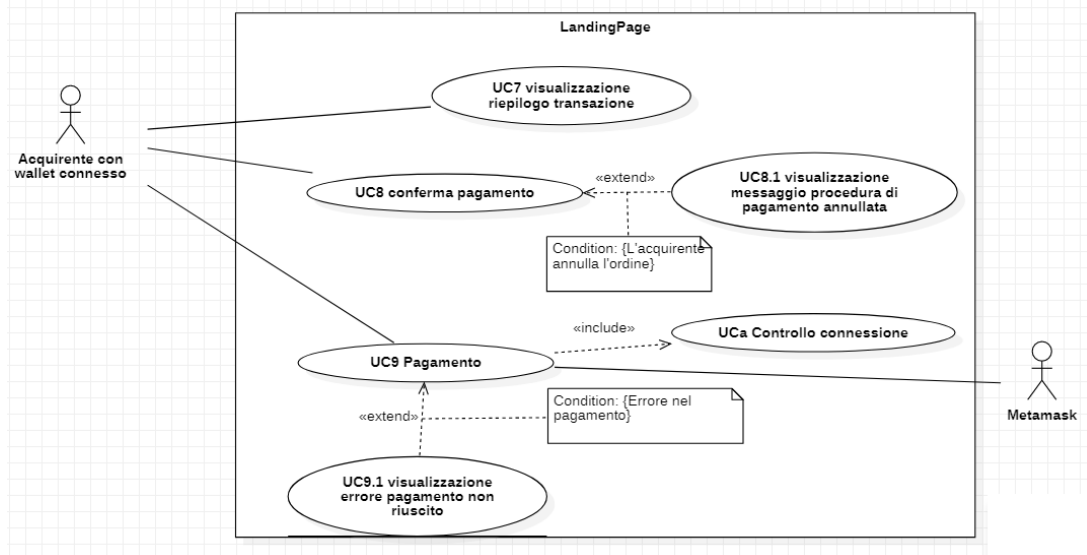


Figure 7: Casi d'uso UC7, UC8, UC8.1, UC9 e UC9.1

UC7 - Visualizzazione riepilogo transazione

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'acquirente con wallet connesso vuole effettuare un acquisto usando ShopChain.

Postcondizioni: Il sistema mostra il riepilogo della transazione.

Scenario principale: Viene mostrato all'acquirente il riepilogo della transazione con il prezzo in dollari da pagare.

UC8 - Conferma pagamento

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'acquirente sta effettuando un acquisto e il sistema mostra il riepilogo della transazione.

Postcondizioni: L'acquirente ha confermato il pagamento.

Scenario principale:

1. Il sistema mostra il riepilogo della transazione;
2. L'acquirente conferma l'ordine.

Estensione: UC8.1 Visualizzazione messaggio procedura di pagamento annullata:

1. visualizzazione messaggio procedura di pagamento annullata;
2. cancellazione sessione di pagamento.

UC9 - Pagamento

Attore primario: Acquirente con wallet connesso.

Attore secondario: Metamask.

Precondizioni: L'acquirente ha confermato il pagamento.

Postcondizioni: Il pagamento è stato completato con successo.

Scenario principale:

1. L'acquirente avvia la procedura di pagamento;
2. L'acquirente effettua il pagamento tramite Metamask;
3. L'ordine è stato pagato.

Estensioni: UC9.1 Visualizzazione errore pagamento non riuscito:

1. Il pagamento non è stato completato con successo;
2. Cancellazione sessione di pagamento.

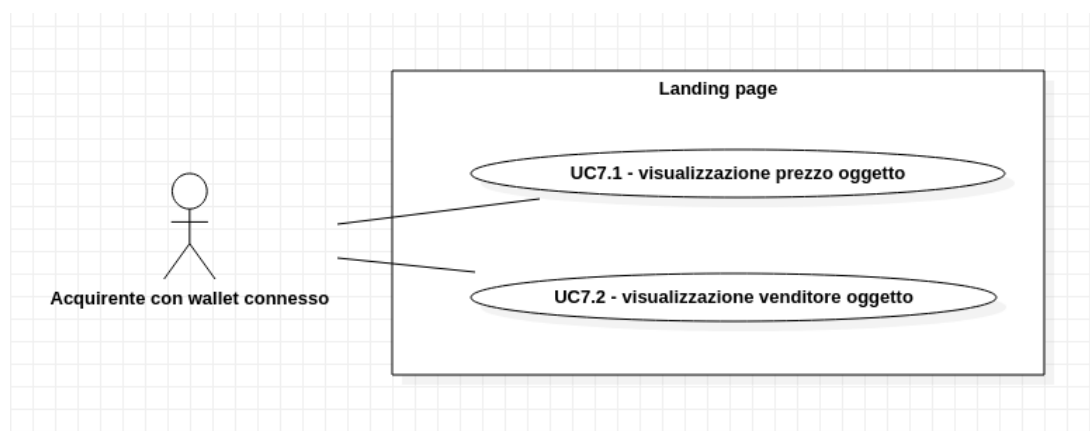


Figure 8: Casi d'uso UC7.1 e UC7.2

UC7.1 - Visualizzazione prezzo oggetto

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'Acquirente vuole visualizzare il prezzo dell'oggetto.

Postcondizioni: L'Acquirente visualizza il prezzo dell'oggetto.

Scenario principale: Viene visualizzato il prezzo dell'oggetto.

UC7.2 - Visualizzazione il venditore dell'oggetto

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'Acquirente vuole visualizzare il venditore dell'oggetto.

Postcondizioni: L'Acquirente visualizza il venditore dell'oggetto.

Scenario principale: Viene visualizzato il venditore dell'oggetto.

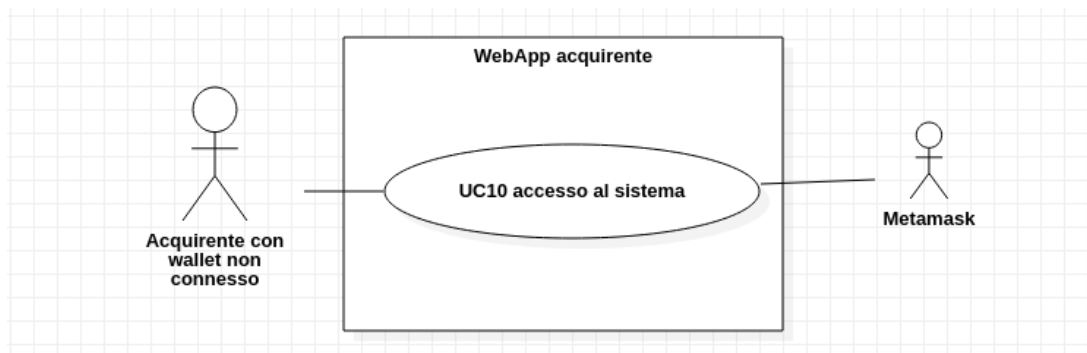


Figure 9: Caso d'uso UC10

UC10 - Accesso al sistema (WebApp Acquirente)

Attore primario: Acquirente con wallet non connesso.

Attore secondario: Metamask.

Precondizioni: L'acquirente nella webapp acquirente non è connesso con Metamask.

Postcondizioni: L'acquirente accede al sistema.

Scenario principale: L'acquirente usando Metamask, si connette con il proprio wallet.

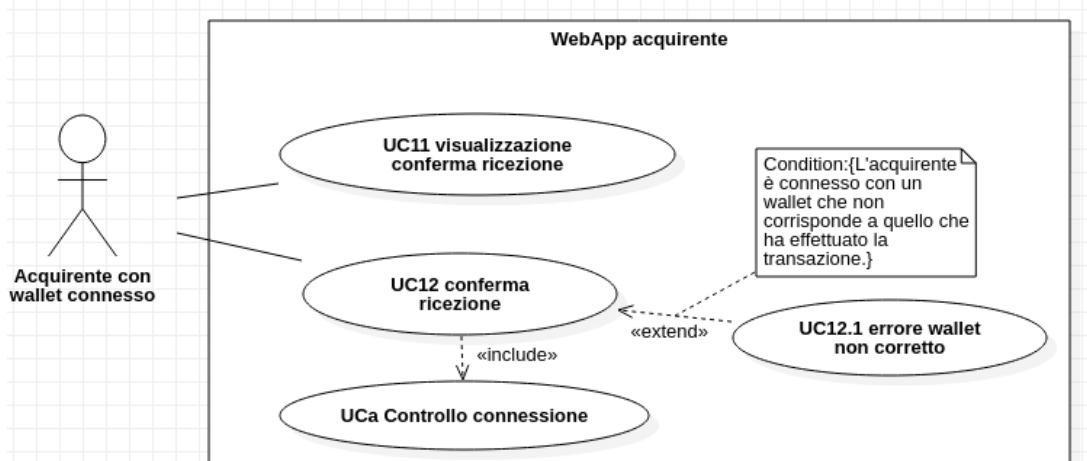


Figure 10: Casi d'uso UC11, UC12 e UC12.1

UC11 - Visualizzazione conferma ricezione

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'acquirente è connesso con Metamask ed è arrivato alla pagina di conferma ricezione scansionando il QR Code sul pacco del prodotto.

Postcondizioni: L'acquirente visualizza la pagina della transazione relativa al pacco appena ricevuto.

Scenario principale: Vengono mostrati i dati relativi alla transazione e un pulsante per confermare la ricezione del pacco.

UC12 - Conferma ricezione

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'acquirente è connesso con Metamask alla pagina di conferma ricezione e conferma la ricezione del pacco.

Postcondizioni: L'acquirente ha confermato la ricezione del pacco, i fondi sono stati sbloccati.

Scenario principale:

1. L'acquirente controlla sia tutto corretto e conferma la ricezione del pacco;
2. Il denaro viene sbloccato e mandato nel wallet del venditore.

Estensione: UC12.1 Errore wallet non corretto:

1. Viene mostrato un messaggio di errore spiegando che il wallet con cui è connesso l'acquirente non è corretto.
2. Si invita a cambiare wallet e riprovare.

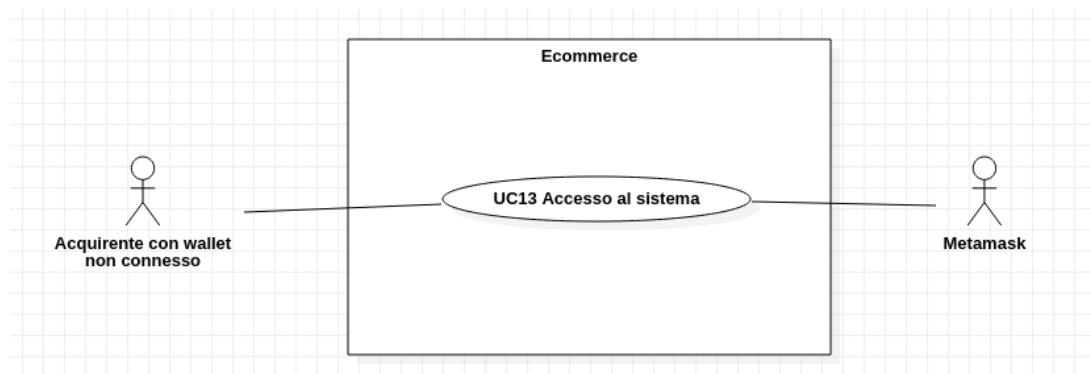


Figure 11: Caso d'uso UC13

UC13 - Accesso al sistema (pagina lista transazioni acquirente in e-commerce)

Attore primario: Acquirente con wallet non connesso.

Attore secondario: Metamask.

Precondizioni: L'acquirente, nella pagina con la sua lista delle transazioni in e-commerce, non è connesso con Metamask.

Postcondizioni: L'acquirente accede alla pagina con la sua lista delle transazioni in e-commerce.

Scenario principale: L'acquirente usando Metamask, si connette con il proprio wallet.

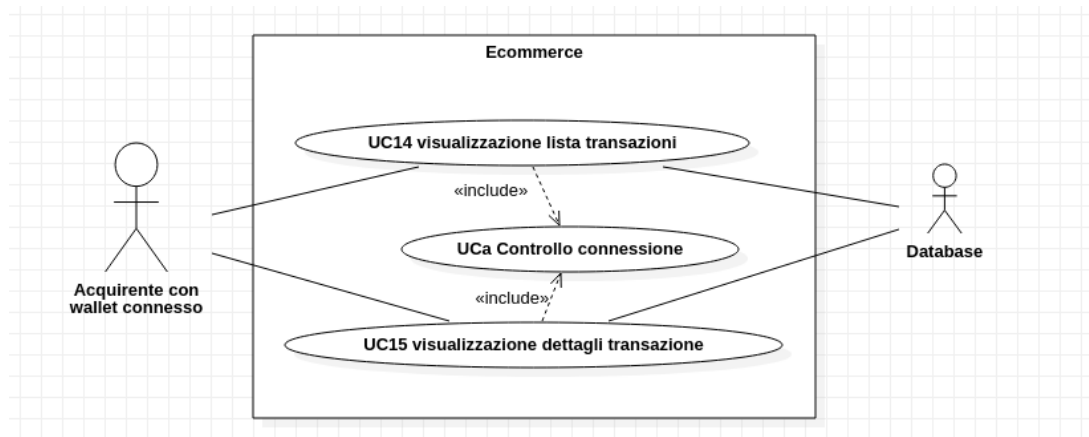


Figure 12: Casi d'uso UC14 e UC15

UC14 visualizzazione lista transazioni

Attore primario: Acquirente con wallet connesso.

Attore secondario: Database.

Precondizioni: L'acquirente ha effettuato l'accesso al sistema.

Postcondizioni: L'acquirente vede la lista delle transazioni.

Scenario principale:

1. L'acquirente vede la lista delle transazioni;
2. L'acquirente può scegliere che tipo di transazioni vedere (tutte, completate, in attesa);
3. Il sistema mostra all'acquirente l'elenco delle transazioni richieste.

UC15 - visualizzazione dettaglio transazione

Attore primario: Acquirente con wallet connesso.

Attore secondario: Database.

Precondizioni: L'acquirente ha acceduto alla funzionalità di visualizzazione della lista delle transazioni.

Postcondizioni: L'acquirente vede i dettagli di una singola transazione.

Scenario principale:

1. L'acquirente seleziona una delle transazioni dall'elenco;
2. L'acquirente visualizza i dettagli della singola transazione.

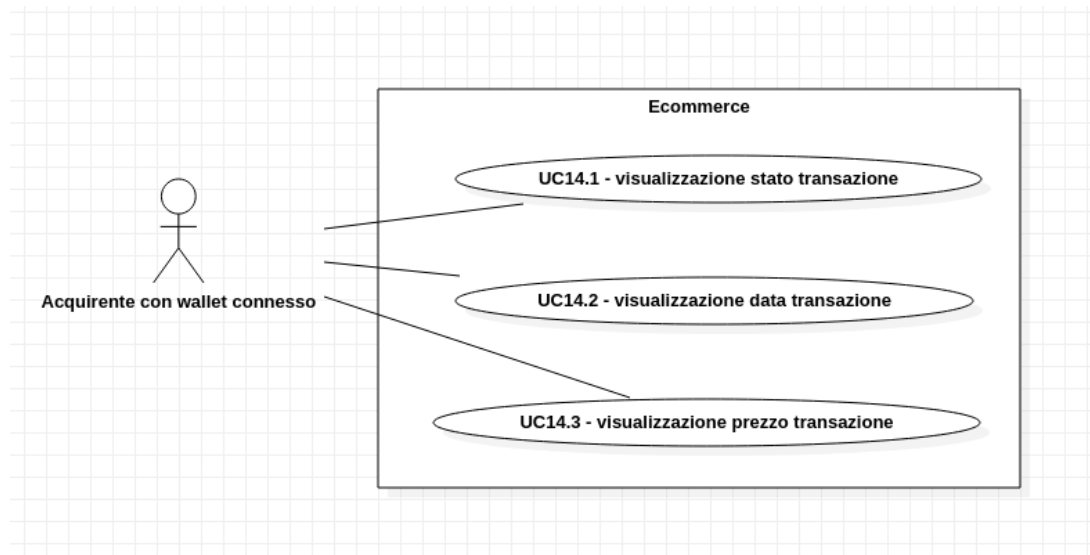


Figure 13: Casi d'uso UC14.1, UC14.2 e UC14.3

UC14.1 - Visualizzazione stato transazioni

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'Acquirente vuole visualizzare lo stato delle transazioni.

Postcondizioni: L'Acquirente visualizza lo stato delle transazioni.

Scenario principale: Viene visualizzato lo stato delle transazioni.

UC14.2 - Visualizzazione data transazioni

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'Acquirente vuole visualizzare la data delle transazioni.

Postcondizioni: L'Acquirente visualizza la data delle transazioni.

Scenario principale: Viene visualizzato la data delle transazioni.

UC14.3 - Visualizzazione prezzo transazioni

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'Acquirente vuole visualizzare il prezzo delle transazioni.

Postcondizioni: L'Acquirente visualizza il prezzo delle transazioni.

Scenario principale: Viene visualizzato il prezzo delle transazioni.

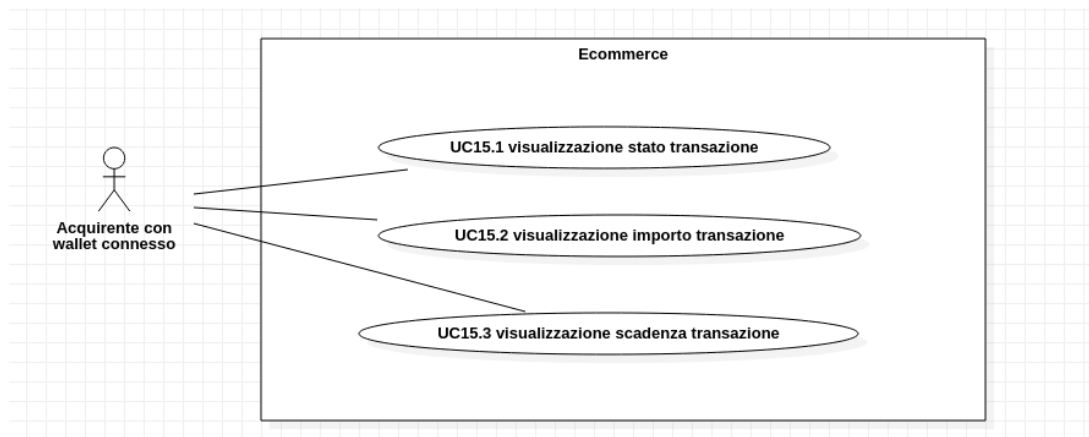


Figure 14: Casi d'uso UC15.1, UC15.2 e UC15.3

UC15.1 - Visualizzazione stato transazione

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'acquirente vuole visualizzare lo stato di una transazione.

Postcondizioni: L'acquirente visualizza lo stato di una transazione.

Scenario principale: Viene mostrato lo stato della transazione: in attesa o completata.

UC15.2 - Visualizzazione importo transazione

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'acquirente vuole visualizzare l'importo di una transazione.

Postcondizioni: L'acquirente visualizza l'importo di una transazione.

Scenario principale: Viene mostrato l'importo della transazione in dollari.

UC15.3 - Visualizzazione scadenza transazione

Attore primario: Acquirente con wallet connesso.

Precondizioni: L'acquirente vuole visualizzare la scadenza di una transazione in attesa.

Postcondizioni: L'acquirente visualizza la scadenza di una transazione in attesa.

Scenario principale: Viene visualizzata la data della scadenza di una transazione in attesa, oltre la quale l'acquirente riceverà indietro la valuta bloccata.

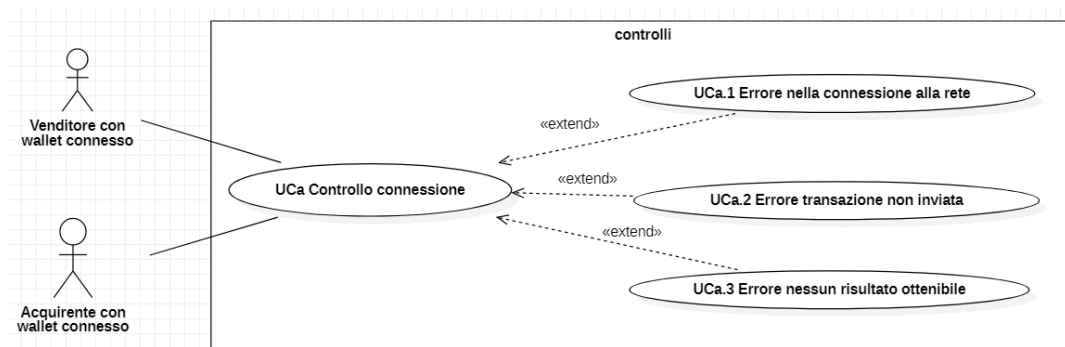


Figure 15: Casi d'uso UCa, UCa.1, UCa.2 e UCa.3

UCa Controllo connessione

Attori primari: Venditore con wallet connesso, acquirente con wallet connesso.

Precondizioni: Il sistema vuole interagire con la blockchain.

Postcondizioni: Il sistema può interagire con la blockchain senza problemi.

Scenario principale: Vengono fatti una serie di controlli per la corretta interazione con la blockchain.

Estensioni:

1. UCa.1 Errore nella connessione alla rete:
Il venditore o l'acquirente è connesso alla rete sbagliata
2. UCa.2 Errore transazione non inviata:
La transazione non è stata inviata correttamente.
3. UCa.3 Errore nessun risultato ottenibile:
La transazione è stata correttamente inviata, ma non è stato ottenuto nessun risultato.

4 Requisiti

Ogni requisito è identificato da un codice univoco composto da: R(per requisito)+F/N(per la tipologia: funzionale, non funzionale)+O/D/F(per la rilevanza: obbligatorio, desiderabile, facoltativo)+x(un numero univoco a due cifre).

4.1 Requisiti funzionali

Requisiti funzionali			
Codice	Descrizione	Rilevanza	Fonti
RFO01	Per accedere alla webapp venditore il venditore con wallet non connesso deve connettere il proprio wallet tramite Metamask.	Obbligatorio	UC2
RFO02	Un venditore con wallet non connesso non può accedere alla webapp venditore.	Obbligatorio	Capitolato, decisione interna
RFO34	La webapp deve comunicare il wallet del venditore al server.	Obbligatorio	UC2
RFO03	Il venditore deve poter visualizzare la lista delle transazioni nella webapp venditore.	Obbligatorio	UC3
RFO35	Il server deve fornire una pagina con la lista delle transazioni.	Obbligatorio	UC3

Requisiti funzionali			
Codice	Descrizione	Rilevanza	Fonti
RFO36	La persistenza deve permettere di filtrare le transazioni per venditore.	Obbligatorio	UC3
RFO04	Il venditore deve poter visualizzare nella webapp venditore, per ogni transazione, il suo stato (attesa o completata).	Obbligatorio	UC4.1
RFO05	Il venditore deve poter visualizzare nella webapp venditore, per ogni transazione, il QR Code da applicare sul pacco.	Obbligatorio	UC4.2
RFO51	La webapp venditore deve poter generare data una stringa un QR Code.	Obbligatorio	UC4.2
RFO06	Il venditore deve poter visualizzare nella webapp venditore, per ogni transazione, l'importo in dollari.	Obbligatorio	UC4.3
RFO07	Il venditore deve poter visualizzare nella webapp venditore, per ogni transazione, la scadenza della transazione.	Obbligatorio	UC4.4
RFO37	Il server deve fornire una pagina con i dettagli di un transazione (prezzo, stato, data di creazione, data di chiusura, acquirente e venditore).	Obbligatorio	UC4
RFO38	La persistenza deve tener traccia dei dettagli di un transazione (prezzo, stato, data di creazione, data di chiusura, acquirente e venditore).	Obbligatorio	UC4
RFO39	La persistenza deve poter associare a ogni transazione il suo blocco in blockchain.	Obbligatorio	UC4.4
RFO40	La persistenza deve poter associare ad ogni blocco il proprio timestamp.	Obbligatorio	UC4.4

Requisiti funzionali			
Codice	Descrizione	Rilevanza	Fonti
RFO40	La persistenza deve poter fornire i dati di una particolare transazione su richiesta del server.	Obbligatorio	UC4
RFO08	L'acquirente deve ricevere indietro la valuta bloccata con cui ha pagato per la transazione, quando la transazione scade.	Obbligatorio	Decisione interna, UC4.4
RFO41	Lo smart contract deve rimborsare la valuta al compratore se sono passati 14gg dall'apertura della transazione	Obbligatorio	Decisione interna, UC4.4
RFO42	Lo smart contract deve esporre pubblicamente un metodo che esegue il rimborso di tutte le transazioni scadute	Obbligatorio	Decisione interna, UC4.4
RFO09	Il prodotto deve essere pubblicato automaticamente in blockchain, quando il venditore inserisce un nuovo prodotto nell'e-commerce, grazie al nostro script incluso dal venditore nel suo e-commerce.	Obbligatorio	UC5
RFO10	Il nostro script deve salvare un messaggio di errore in un file di log, se la pubblicazione del prodotto in blockchain dall'e-commerce (grazie allo script) fallisce.	Obbligatorio	UC5.1
RFO43	Lo script deve ritornare l'id del prodotto inserito in blockchain.	Obbligatorio	UC5
RFD44	Lo script essere usabile sia come modulo python che come script python standalone.	Desiderabile	UC5
RFO11	Per accedere alla landing page l'acquirente con wallet non connesso deve connettere il proprio wallet tramite Metamask.	Obbligatorio	UC6, UC15

Requisiti funzionali			
Codice	Descrizione	Rilevanza	Fonti
RFO12	Un acquirente con wallet non connesso non può accedere alla landing page.	Obbligatorio	Capitolato, decisione interna
RFO48	La landing page deve tracciare l'id dell'oggetto venduto tramite metodo GET.	Obbligatorio	Capitolato, decisione interna
RFO13	L'acquirente con wallet connesso deve vedere il prezzo da pagare in dollari nella landing page.	Obbligatorio	UC7
RFO14	L'acquirente con wallet connesso poter confermare il pagamento nella landing page.	Obbligatorio	UC8
RFO15	Il sistema deve mostrare un messaggio "ordine annullato" se l'acquirente con wallet connesso annulla il pagamento sulla landing page.	Obbligatorio	UC8.1
RFO16	Il sistema deve cancellare la sessione di pagamento se l'acquirente con wallet connesso annulla il pagamento sulla landing page.	Obbligatorio	UC8.1
RFO17	L'acquirente con wallet connesso deve essere indirizzato automaticamente su Metamask per completare il pagamento, quando conferma il pagamento sulla landing page.	Obbligatorio	UC9
RFO47	La landing page deve, attraverso Metamask, chiamare il metodo di pagamento dello smart contract.	Obbligatorio	UC9
RFO18	Il sistema deve cancellare la sessione di pagamento, se il pagamento non va a buon fine quando l'acquirente con wallet connesso effettua il pagamento tramite Metamask.	Obbligatorio	UC9.1

Requisiti funzionali			
Codice	Descrizione	Rilevanza	Fonti
RFO19	Per accedere alla webapp acquirente, l'acquirente con wallet non connesso deve connettere il proprio wallet tramite Metamask.	Obbligatorio	UC10
RFO20	Un acquirente con wallet non connesso non può accedere alla webapp acquirente.	Obbligatorio	Capitolato, decisione interna
RFO21	L'acquirente con wallet connesso deve poter vedere il prezzo in dollari nella webapp acquirente, dopo essere atterrato nella webapp acquirente scansionando il QR Code sul pacco.	Obbligatorio	UC11
RFO22	L'acquirente con wallet connesso deve poter confermare la ricezione del pacco nella webapp acquirente, dopo essere atterrato nella webapp acquirente scansionando il QR Code sul pacco.	Obbligatorio	UC12
RFO45	La landing page deve attraversare Metamask chiamare il metodo di conferma dello smart contract.	Obbligatorio	UC12
RFO46	Lo smart contract deve permettere la chiamata del metodo di conferma solo all'acquirente (identificato dal suo wallet).	Obbligatorio	UC12
RFO23	Il sistema deve mostrare un messaggio di errore "wallet non corretto" nella webapp acquirente, nel caso in cui dopo aver confermato la ricezione del pacco nella webapp acquirente, il sistema rileva che il wallet non corrisponde a quello che ha effettuato la transazione.	Obbligatorio	UC12.1

Requisiti funzionali			
Codice	Descrizione	Rilevanza	Fonti
RFO24	Per accedere alla pagina con la sua lista delle transazioni in e-commerce, l'acquirente con wallet non connesso deve connettere il proprio wallet tramite Metamask.	Obbligatorio	UC13
RFO25	Un acquirente con wallet non connesso non può accedere alla pagina con la sua lista delle transazioni in e-commerce.	Obbligatorio	Capitolato, decisione interna
RFO48	La pagina delle transazioni deve comunicare il wallet del venditore al server.	Obbligatorio	UC13
RFO26	L'acquirente con wallet connesso deve poter visualizzare la lista delle sue transazioni in una pagina apposita, dedicata all'e-commerce.	Obbligatorio	UC14
RFO49	La persistenza deve permettere di filtrare le transazioni per acquirente.	Obbligatorio	UC14
RFO27	L'acquirente con wallet connesso deve poter visualizzare lo stato di ogni sua transazione (attesa o completata) in una pagina apposita, dedicata all'e-commerce.	Obbligatorio	UC15.1
RFO28	L'acquirente con wallet connesso deve poter visualizzare l'importo di ogni sua transazione in una pagina apposita, dedicata all'e-commerce.	Obbligatorio	UC15.2
RFO29	L'acquirente con wallet connesso deve poter visualizzare l'importo in dollari di ogni sua transazione in una pagina apposita, dedicata all'e-commerce.	Obbligatorio	UC15.2

Requisiti funzionali			
Codice	Descrizione	Rilevanza	Fonti
RFO30	L'acquirente con wallet connesso deve poter visualizzare la scadenza di ogni sua transazione in una pagina apposita, dedicata all'e-commerce.	Obbligatorio	UC15.3
RFO31	Il sistema deve mostrare un messaggio di errore "Errore nella connessione alla rete" se l'utente (acquirente o venditore) è connesso alla rete sbagliata.	Obbligatorio	UCa.1
RFO50	Il sistema deve poter interrogare Metamask su quale sia la rete attualmente usata dall'utente.	Obbligatorio	UCa.1
RFO32	Il sistema deve mostrare un messaggio di errore "Errore transazione non inviata" se la transazione non è stata inviata in maniera corretta.	Obbligatorio	UCa.2
RFO33	Il sistema deve mostrare un messaggio di errore "Errore nessun risultato ottenibile" se la transazione è stata inviata in maniera corretta ma non è stata ottenuto alcun risultato di risposta.	Obbligatorio	UCa.3

Requisiti non funzionali				
Codice	Descrizione	Rilevanza	Fonti	Tipologia
RNO01	Utilizzo di blockchain pubblica.	Obbligatorio	Capitolato, decisione interna	Vincolo
RND02	Utilizzo di database PostgreSQL.	Desiderabile	Capitolato	Vincolo
RNO03	Il venditore deve disporre di un manuale utente per l'utilizzo del sistema.	Obbligatorio	UC1, decisione interna	Vincolo
RNO04	Test che dimostrino il corretto funzionamento dei servizi e delle funzionalità previste, con una copertura minima dell' 80% correlata di report.	Obbligatorio	Capitolato	Qualità
RNO05	Documentazione su scelte implementative e progettuali effettuate, le relative motivazioni, i problemi aperti e le eventuali soluzioni proposte da esplorare.	Obbligatorio	Capitolato	Qualità
RNO06	Deve essere reso disponibile uno script python scaricabile dai venditori da integrare nel loro e-commerce per la pubblicazione automatica dei nuovi prodotti in blockchain.	Obbligatorio	UC5, decisione interna	Vincolo