Allegato tecnico
Progetto ShopChain

yakuzaishi.swe@gmail.com

Informazioni sul documento

Responsabile	Francesco Bugno
Redattori	Dario Furlan
	Michele Filosofo
Verificatori	Luca Busacca
	Matteo Midena
Uso	Esterno
Destinatari	Prof. Tullio Vardanega
	Prof. Riccardo Cardin
	Ing. Fabio Pallaro - Sync Lab S.r.l.
Versione	1.0.0

Sommario

Questo documento racchiude tutte le scelte architetturali che il gruppo *Yakuzaishi* ha preso nella realizzazione del prodotto *ShopChain*. Per la descrizione del prodotto sono utilizzati i diagrammi delle classi e di sequenza.

Registro delle Modifiche

Versione	Data	Autore	Ruolo	Descrizione
1.0.0	2022/05/14	Francesco Bugno	Responsabile	Approvato per il rilascio
0.2.0	2022/05/05	Matteo Midena	Verificatore	Verifica generale del documento
0.1.2	2022/05/02	Dario Furlan Matteo Midena	Progettista Verficatore	Stesura §3 e verifica
0.1.1	2022/04/25	Dario Furlan Luca Busacca	Progettista Verificatore	Aggiunti diagrammi di sequenza e verifica
0.1.0	2022/04/21	Luca Busacca	Verificatore	Verifica generale del documento
0.0.3	2022/04/20	Michele Filosofo Matteo Midena	Progettista Verificatore	Aggiunti diagrammi delle classi e verifica
0.0.2	2022/04/12	Dario Furlan Luca Busacca	Progettista Verificatore	Inizio stesura §2 e verifica
0.0.1	2022/04/10	Michele Filosofo Matteo Midena	Progettista Verificatore	Creata struttura documento, stesura §1 e verifica

Contenuti

Elenco delle tabelle

Elenco delle figure

1 Introduzione

1.1 Scopo del documento

Lo scopo di questo documento è quello di descrivere e motivare le scelte architetturali che il gruppo *Yakuzaishi* ha deciso di prendere in fase di progettazione e codifica del prodotto. Vengono quindi riportati i diagrammi delle classi e di sequenza per descrivere architettura e funzionalità principali del prodotto. Infine è presente una sezione dedicata ai requisiti che il gruppo *Yakuzaishi* è riuscito a soddisfare, così da avere un' ampia visione sullo stato di avanzamento del lavoro.

1.2 Scopo del prodotto

L'obiettivo richiesto dall'azienda proponente è la realizzazione di una web app$_G$ che permetta la gestione dei pagamenti per una piattaforma e-commerce$_G$, tramite l'uso di smart contracts$_G$ basati sulla blockchain$_G$ Fantom$_G$. La decisione di sviluppare l'applicazione per la blockchain$_G$ Fantom$_G$ è stata presa dopo una attenta analisi delle possibili blockchain$_G$ adatte alla costruzione di smart contract$_G$. L'applicazione sarà inoltre dotata di una funzione MoneyBox$_G$ per gestire pagamenti da più utenti per uno stesso prodotto.

1.3 Riferimenti

1.3.1 Riferimenti normativi

- Capitolato d'appalto C2 - ShopChain: Exchange platform on blockchain.

  https://www.math.unipd.it/~tullio/IS-1/2021/Progetto/C2.pdf

1.3.2 Riferimenti informativi

- Slide P2 del corso di ingegneria del software - Diagrammi delle classi;

  https://www.math.unipd.it/~rcardin/swea/2021/Diagrammi%20delle%20Classi_4x4.pdf

- Slide P5 del corso di ingegneria del software - Diagrammi di sequenza.

  https://www.math.unipd.it/~rcardin/swea/2022/Diagrammi%20di%20Sequenza.pdf

2 Architettura del prodotto

2.1 Descrizione generale

Il pattern architetturale scelto dal gruppo per lo sviluppo del progetto è il Model-View-ViewModel. Il seguente pattern è tra i più diffusi nello sviluppo delle web application e permette di scrivere codice facilmente mantenibile e riusabile; questo è possibile grazie al forte disaccoppiamento che sussiste tra logica di presentazione e di business. Inoltre l'MVVM è risultato il più adatto per essere utilizzato con React, libreria impiegata per lo sviluppo dell'UI e che renderizza le componenti in base al loro stato interno.

- Model: questa porzione ricopre la logica di business dell'applicazione;
- ViewModel: qui viene effettuato il binding tra View e Model ed è contenuta la loro logica;
- View: questa porzione gestisce la presentazione tramite una specifica gerarchia di componenti; ciascun componente contiene la logica strettamente legata alla sua visualizzazione e necessaria al mantenimento del proprio stato interno.

Il passaggio dei dati dal Model alle varie componenti grafiche avviene attraverso l'utilizzo di un Context React, al quale viene passato un'istanza del ViewModel. L'utilizzo di un Context React ci permette di accedere al valore corrente del ViewModel in qualsiasi porzione della View, senza doverlo passare di componente in componente attraverso le props (ossia gli argomenti dei componenti che compongono la vista). Nella radice dell'applicazione viene infatti creata un'istanza del ViewModel, che viene passata ad un Context.Provider, che fa da contenitore per tutta la View. All'interno di tale contenitore ogni componente puo' utilizzare un hook per accedere al Context React ed utilizzare il valore più recente del ViewModel.

E' stato scelto di utilizzare un Context React per il passaggio dei dati in quanto la nostra applicazione è molto profonda e non risultava conveniente passare i dati per molti componenti rischiando, nel peggiore dei casi, di doverli utilizzare nell'ultimo della gerarchia. Per poter fare in modo che una componente della View si renderizzi non solo al cambiamento del suo stato interno ma anche al cambiamento dei dati nel Model, abbiamo utilizzato la libreria Mobx. Questa ci permette di implementare l'observer pattern, non supportato di default da React. A tale scopo, Mobx permette di segnare delle classi (o attributi di esse) come "observable" e di costruire dei componenti della View come "observer". Quest'ultimi vengono automaticamente ri-renderizzati al cambiamento di un qualsiasi attributo observable.

2.2 Diagramma delle classi

In questa sezione verrà mostrato il diagramma delle classi dell'applicazione, per una maggiore fruibilità i diagrammi della parte frontend sono stati divisi in quattro parti. Le classi colorate compaiono più volte nei vari diagrammi.

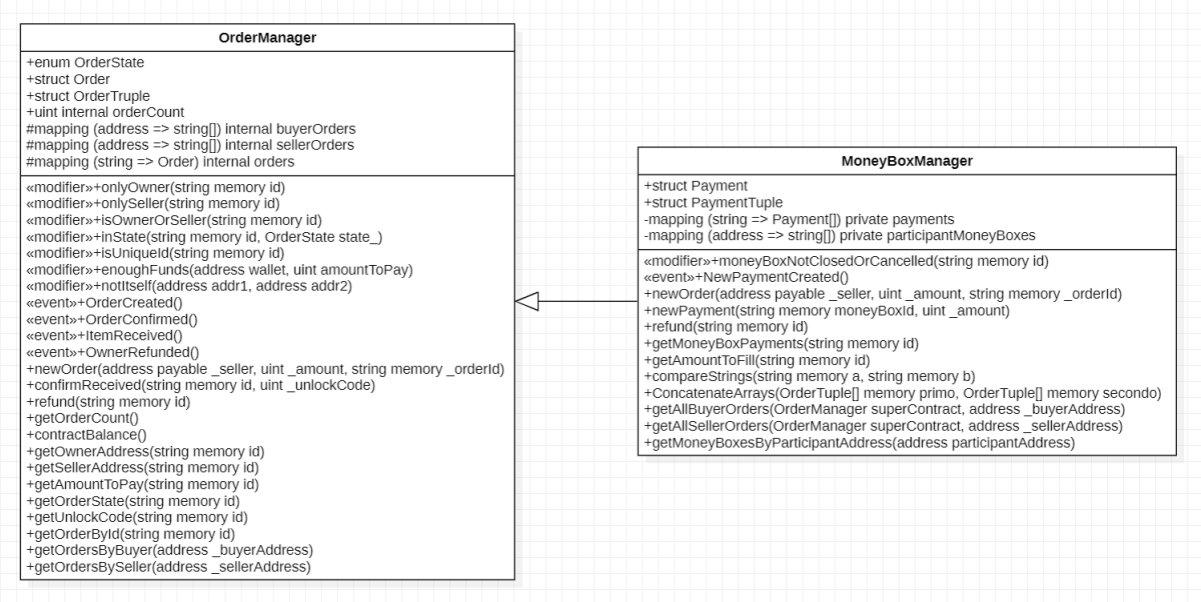

Figura 1: Classi Solidity

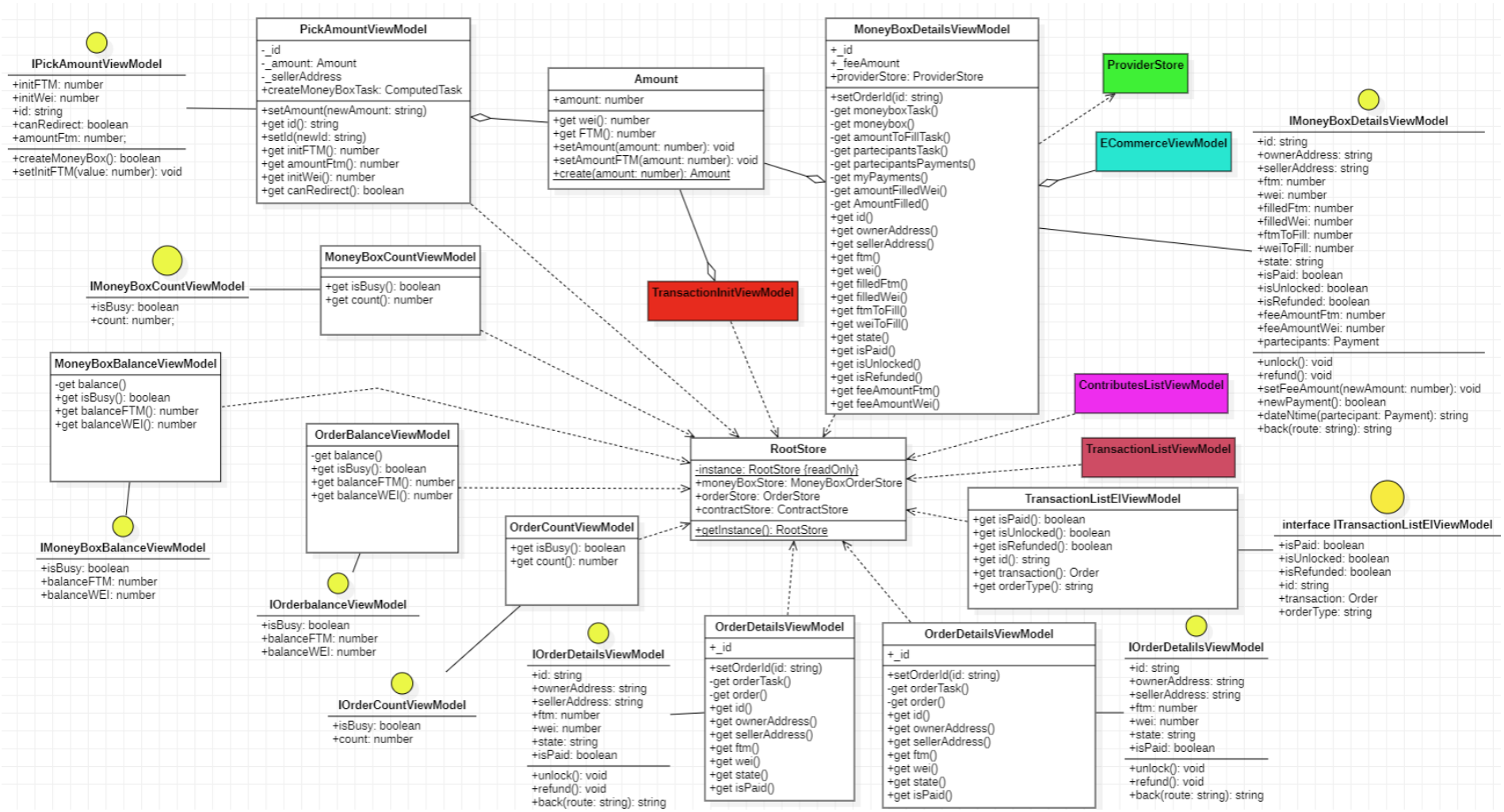

Figura 2: RootStore e classi correlate

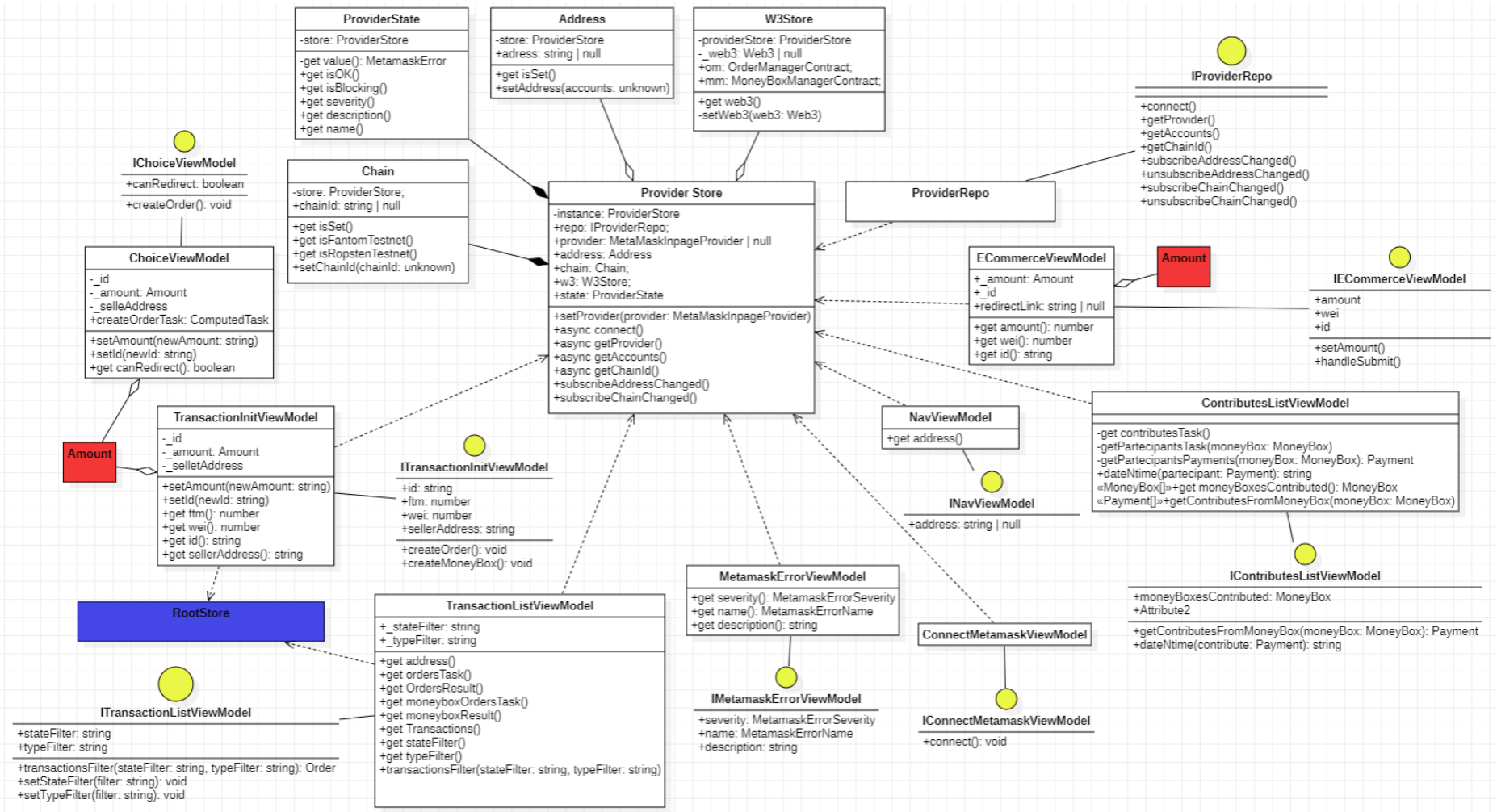

Figura 3: ProviderStore e classi correlate

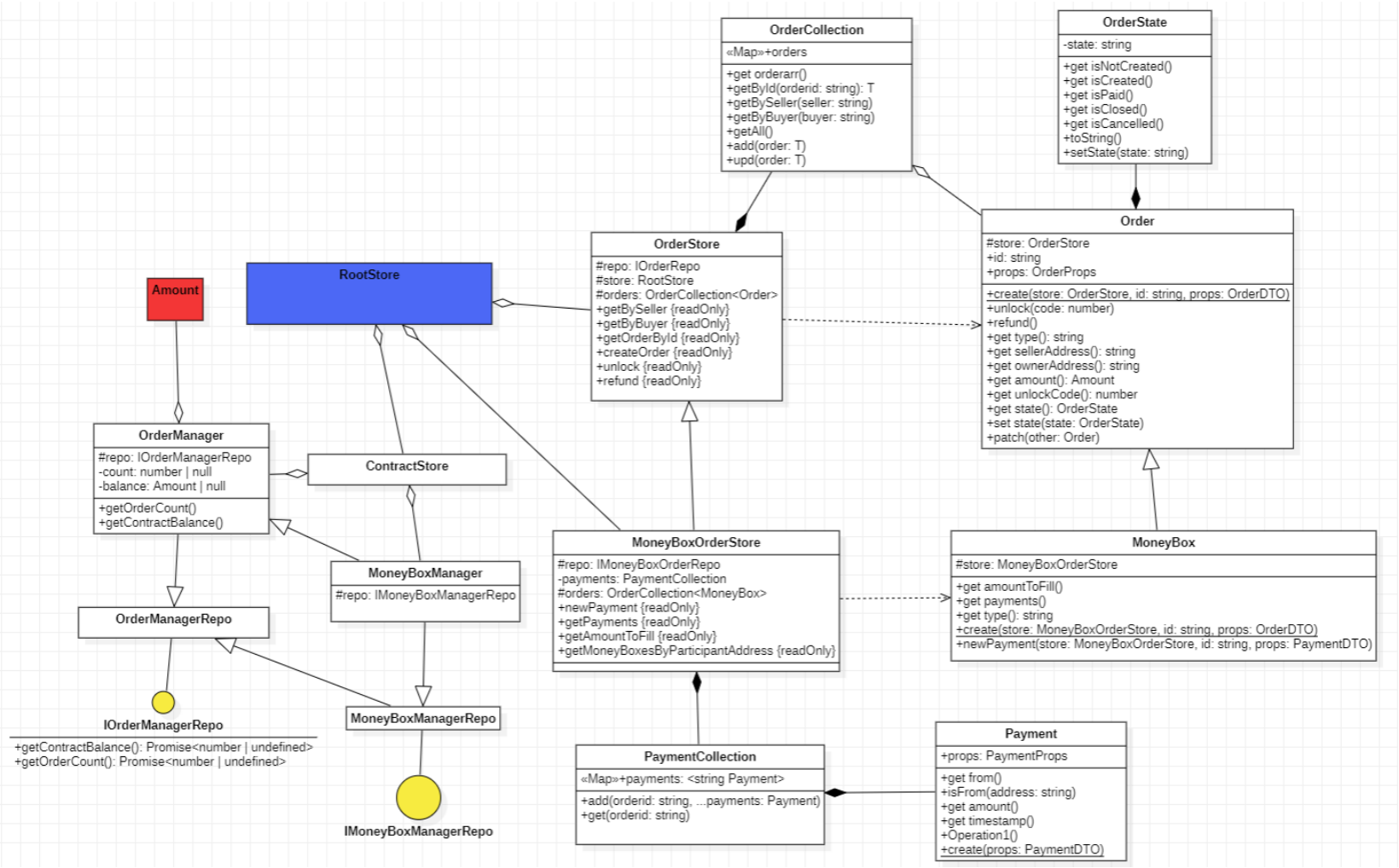

Figura 4: Gerarchia Order e MoneyBox

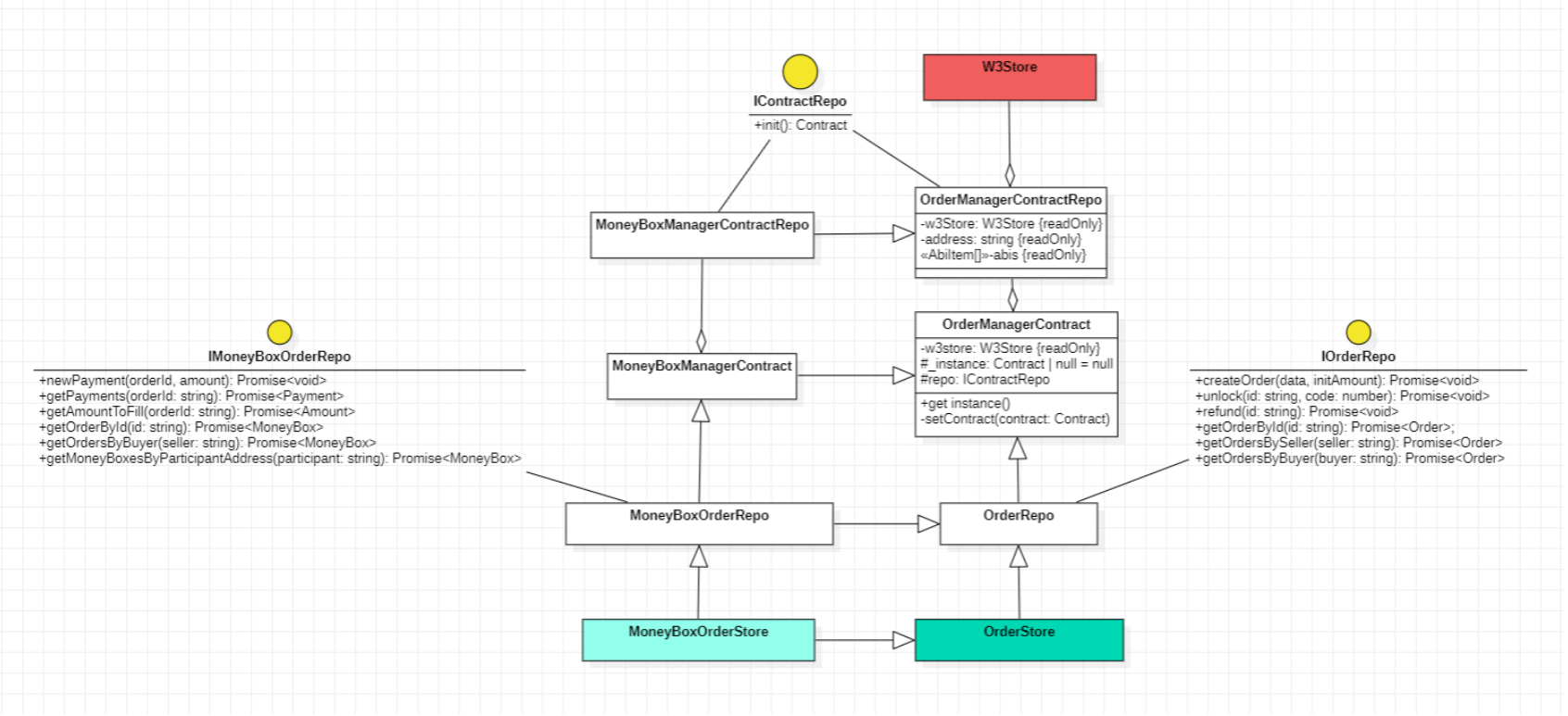

Figura 5: Passaggio da Web3Store a OrderStore

2.3 Diagrammi di sequenza

Vengono di seguito riportati i diagrammi di sequenza per le operazioni più importanti dell'applicazione.

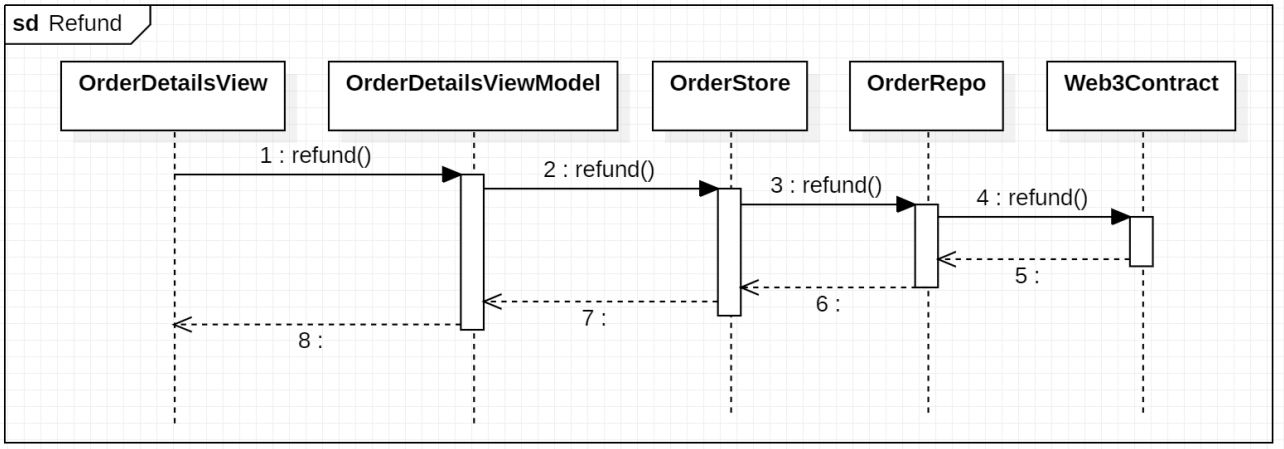

Figura 6: Diagramma di sequenza della funzione di Refund

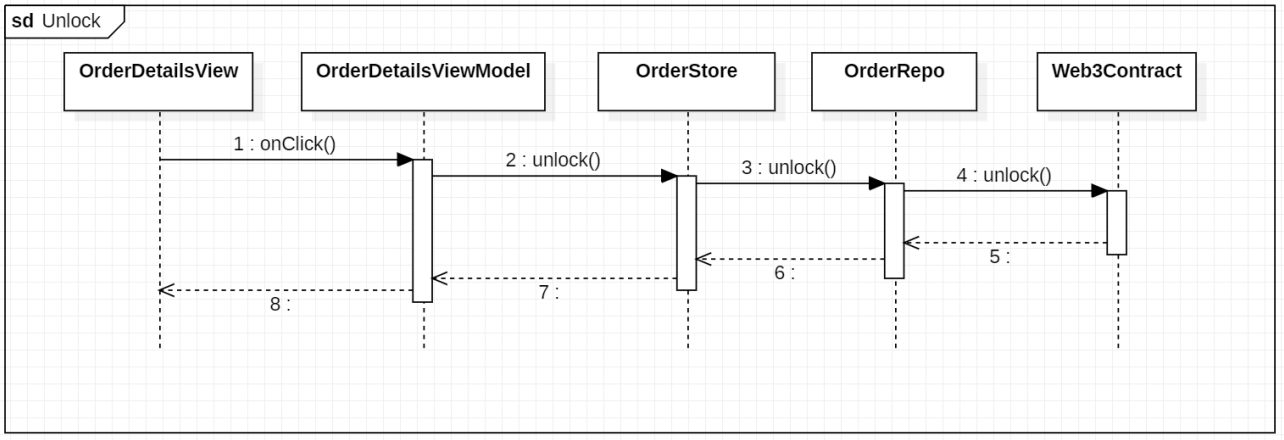

Figura 7: Diagramma di sequenza della funzione di Unlock

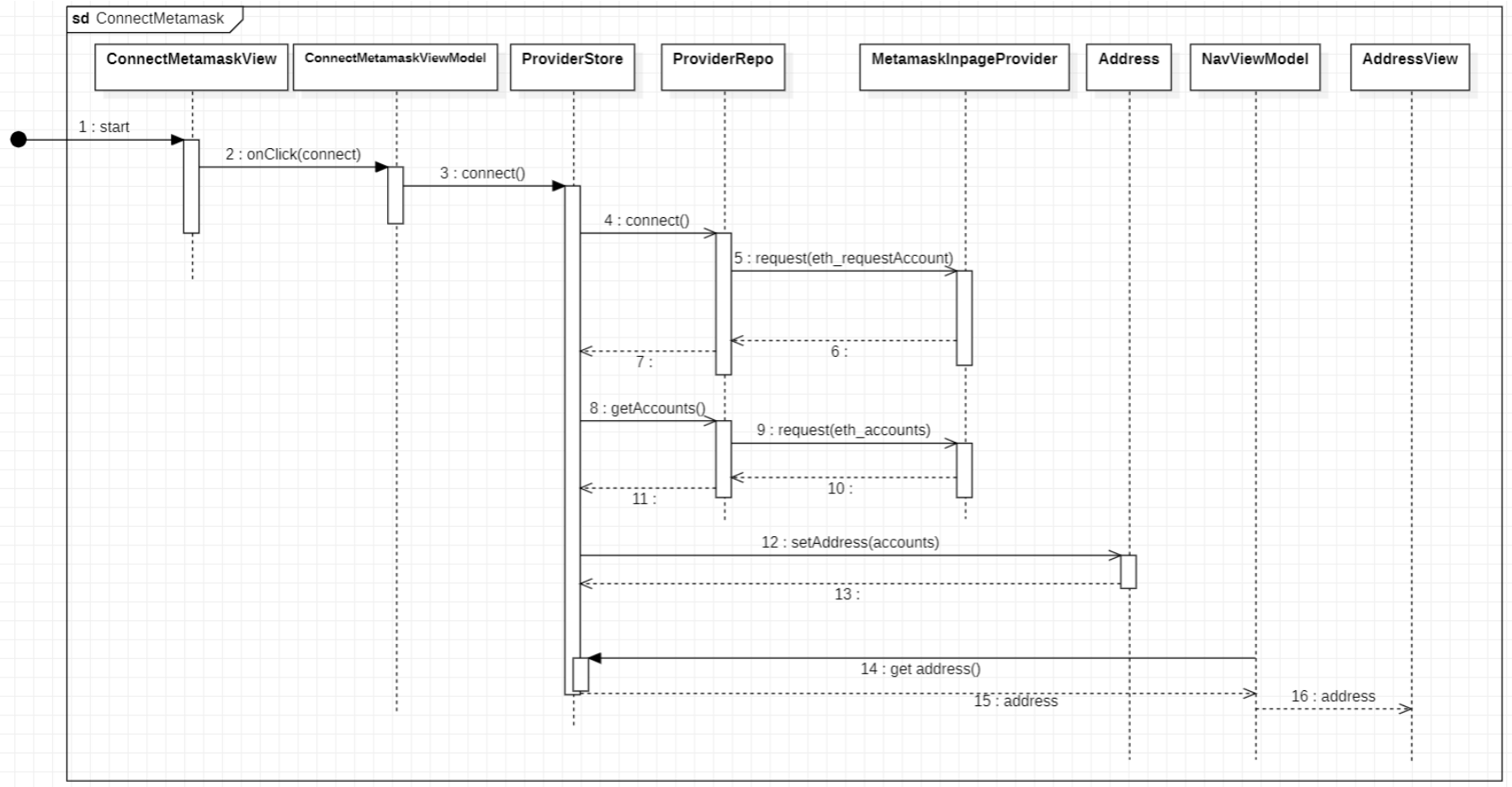

Figura 8: Diagramma di sequenza della funzione di connessione a Metamask

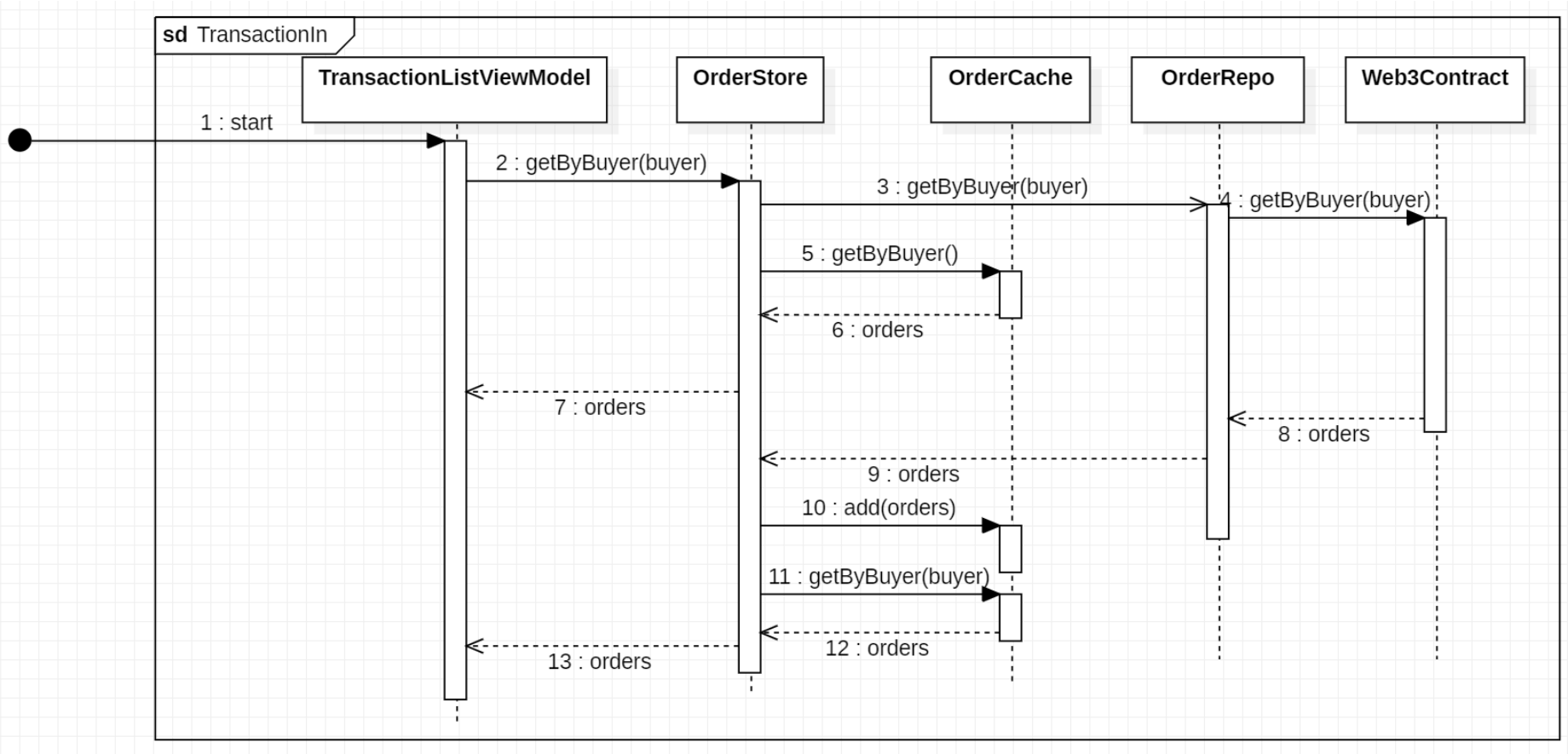

Figura 9: Diagramma di sequenza della selezione delle transazioni in entrata

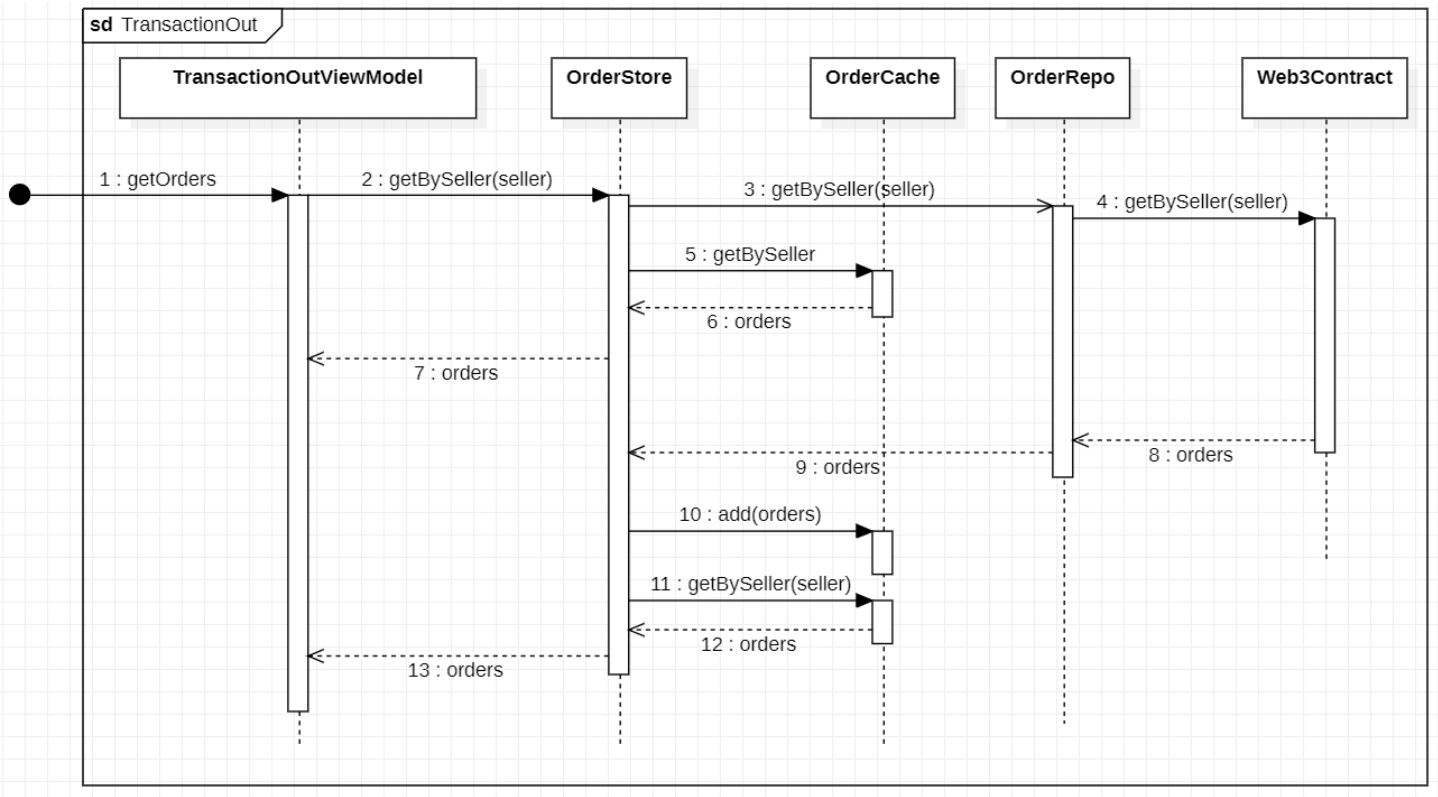

Figura 10: Diagramma di sequenza della selezione delle transazioni in uscita

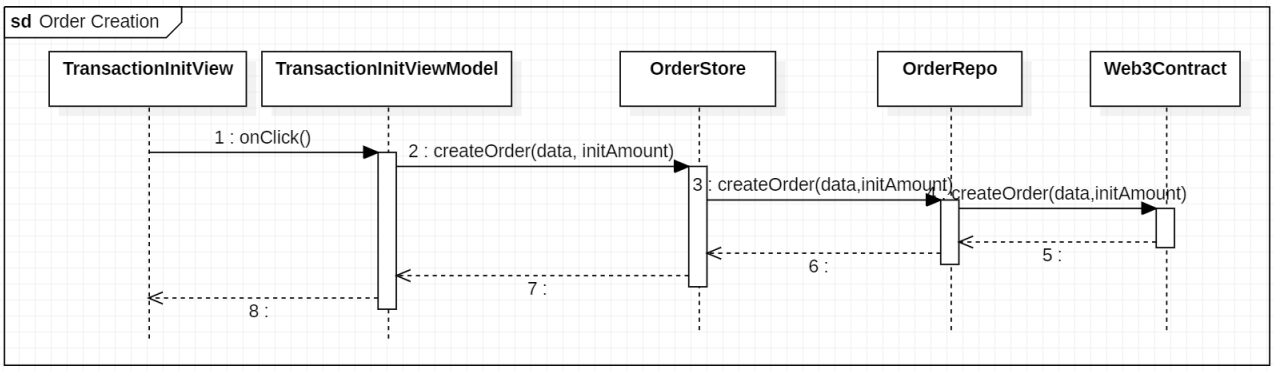

Figura 11: Diagramma di sequenza della funzione di creazione di un nuovo ordine

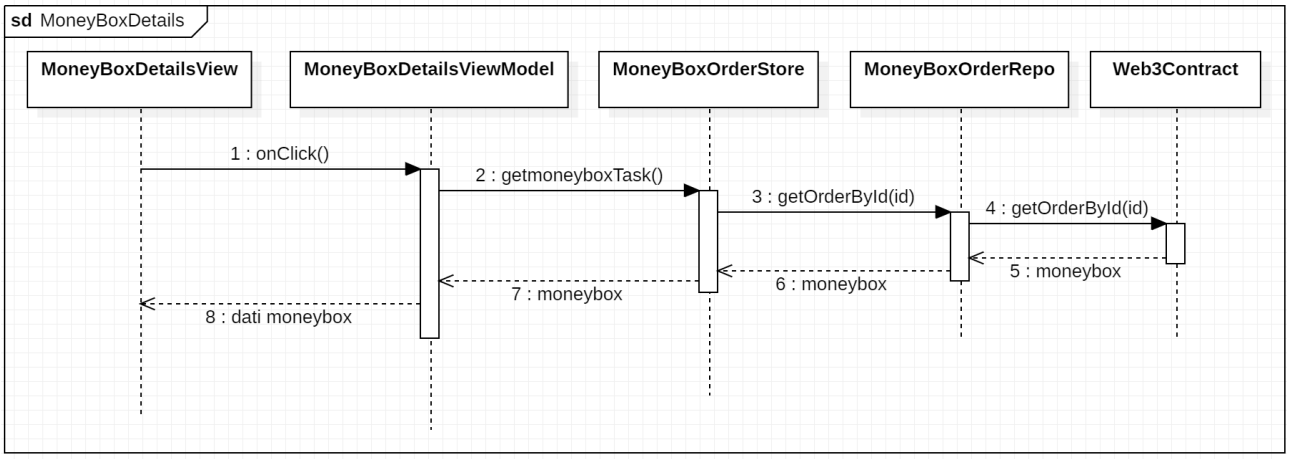

Figura 12: Diagramma di sequenza della funzione di visualizzazione dei dettagli di una Moneybox

2.4 Design pattern

Sono stati utilizzati due design pattern nello sviluppo dell'applicazione:

- Singleton: è un design pattern che consente di garantire che una classe abbia una sola istanza, fornendo al contempo un punto di accesso globale a questa istanza;
- Repository Pattern: un repository incapsula un insieme di oggetti memorizzati nel database e le operazioni che possono essere eseguite su di essi.

Il design singleton è stato usato per avere un'unica fonte detentrice dello stato di ogni modello. Il repository pattern ci ha permesso di racchiudere tutte le interazioni con sorgenti esterne in un oggetto unico, mantenendo la separazione tra implementazione e interfaccia.

3 Requisiti soddisfatti

Secondo quanto riportato nel *Piano di progetto v2.0.0*, il gruppo *Yakuzaishi* è riuscito a soddisfare tutti i requisiti obbligatori. Arrivati a questo punto si è avviata anche la codifica delle funzionalità opzionali e desiderabili che proseguirà nel periodo successivo.

3.1 Tabella requisiti soddisfatti

Segue una tabella con l'elenco di tutti i requisiti funzionali soddisfatti.

Codice	Classificazione	Descrizione	Stato
R1F1	Obbligatorio	Si deve processare una richiesta di checkout da un e-commerce$_{\mathbf{G}}$.	Soddisfatto
R1F2	Obbligatorio	L'utente deve poter scegliere la tipologia di pagamento.	Soddisfatto
R1F2.1	Obbligatorio	L'utente deve poter scegliere la tipologia di pagamento unico.	Soddisfatto
R2F2.2	Desiderabile	L'utente deve poter scegliere la tipologia di pagamento MoneyBox$_{\mathbf{G}}$.	Soddisfatto
R2F2.2.1	Desiderabile	Si deve poter visualizzare lo stato di completamento della MoneyBox$_{\mathbf{G}}$.	Soddisfatto
R2F2.2.2	Desiderabile	Si deve poter copiare l'invito di partecipazione alla MoneyBox$_{\mathbf{G}}$.	Soddisfatto
R3F2.2.3	Opzionale	L'utente deve poter visualizzare una traduzione visiva dell'indirizzo MoneyBox$_{\mathbf{G}}$.	Non Soddisfatto
R2F2.2.4	Desiderabile	Si deve poter chiudere la MoneyBox$_{\mathbf{G}}$ restituendo i soldi.	Soddisfatto
R2F2.2.5	Desiderabile	L'utente deve poter visualizzare l'elenco delle transazioni partecipanti alla MoneyBox$_{\mathbf{G}}$.	Soddisfatto
R1F3	Obbligatorio	L'utente deve poter visualizzare il totale dell'ordine.	Soddisfatto
R1F4	Obbligatorio	L'utente deve poter effettuare la connessione a Metamask$_{\mathbf{G}}$.	Soddisfatto
R1F5	Obbligatorio	L'utente deve poter pagare.	Soddisfatto

Continua nella pagina successiva

Codice	Classificazione	Descrizione	Stato
R1F5.1	Obbligatorio	L'utente che ha scelto il metodo di pagamento unico deve pagare interamente la somma richiesta.	Soddisfatto
R2F5.2	Desiderabile	L'utente che partecipa al pagamento $\text{MoneyBox}_{\mathbf{G}}$ deve poter pagare una parte della somma richiesta.	Soddisfatto
R1F5.4	Obbligatorio	L'utente deve poter visualizzare un messaggio d'errore nel caso in cui la transazione fallisca.	Soddisfatto
R1F6	Obbligatorio	Il proprietario dell'ordine o il venditore devono poter richiedere il rimborso della transazione.	Soddisfatto
R1F7	Obbligatorio	Il proprietario dell'ordine deve poter confermare la ricezione e quindi sbloccare i fondi dallo smart $\text{contract}_{\mathbf{G}}$.	Soddisfatto
R1F7.1	Obbligatorio	Il proprietario dell'ordine deve poter visualizzare il codice di sblocco.	Soddisfatto
R1F8	Obbligatorio	L'utente deve poter visualizzare le transazioni.	Soddisfatto
R2F8.1	Desiderabile	Il venditore deve poter visualizzare le transazioni in entrata pagate.	Soddisfatto
R2F8.1.1	Desiderabile	Il venditore deve poter visualizzare le transazioni in entrata pagate non sbloccate.	Soddisfatto
R2F8.1.2	Desiderabile	Il venditore deve poter visualizzare le transazioni in entrata pagate e sbloccate.	Soddisfatto
R2F8.1.3	Desiderabile	Il venditore deve poter visualizzare le transazioni in entrata pagate ma cancellate.	Soddisfatto
R1F8.2	Obbligatorio	Il proprietario dell'ordine deve poter visualizzare le transazioni in uscita.	Soddisfatto

Continua nella pagina successiva

Codice	Classificazione	Descrizione	Stato
R2F8.2.1	Desiderabile	Il proprietario dell'ordine deve poter visualizzare le transazioni in uscita non pagate.	Soddisfatto
R2F8.2.2	Desiderabile	Il proprietario dell'ordine deve poter visualizzare le transazioni in uscita pagate ma non sbloccate.	Soddisfatto
R2F8.2.3	Desiderabile	Il proprietario dell'ordine deve poter visualizzare le transazioni in uscita pagate e sbloccate.	Soddisfatto
R2F8.2.4	Desiderabile	Il proprietario dell'ordine deve poter visualizzare le transazioni in uscita cancellate.	Soddisfatto
R3F9	Opzionale	Si deve convertire l'ammontare depositato sullo smart contract$_{\mathbf{G}}$ in stable coin$_{\mathbf{G}}$.	Non Soddisfatto
R3F10	Opzionale	La piattaforma deve trattenere una piccola somma in percentuale di un ordine destinata al fondo ShopChain.	Non Soddisfatto
R1F11	Obbligatorio	L'utente deve poter visualizzare il suo indirizzo wallet$_{\mathbf{G}}$.	Soddisfatto
R1F11.1	Obbligatorio	L'utente deve poter visualizzare il suo indirizzo wallet$_{\mathbf{G}}$ in forma testuale.	Soddisfatto
R1F11.2	Obbligatorio	L'utente deve poter visualizzare un avviso della mancata connessione a Metamask$_{\mathbf{G}}$.	Soddisfatto
R3F11.3	Opzionale	L'utente deve poter visualizzare il suo indirizzo wallet$_{\mathbf{G}}$ sotto forma di stringa di emoji.	Non Soddisfatto
R3F12	Opzionale	Gli utenti partecipanti devono ricevere una notifica al completamento della MoneyBox$_{\mathbf{G}}$.	Non Soddisfatto

Continua nella pagina successiva

Codice	Classificazione	Descrizione	Stato
R1F14	Obbligatorio	L'utente deve poter visualizzare lo stato di connessione a Metamask$_{\mathbf{G}}$.	Soddisfatto
R1F14.1	Obbligatorio	L'utente deve poter visualizzare una conferma di connessione corretta a Metamask$_{\mathbf{G}}$.	Soddisfatto
R1F14.2	Obbligatorio	L'utente deve poter visualizzare un errore nel caso non sia installato Metamask$_{\mathbf{G}}$.	Soddisfatto
R1F14.3	Obbligatorio	L'utente deve poter visualizzare un errore nel caso in cui la blockchain selezionata non sia corretta.	Soddisfatto
R1F14.4	Obbligatorio	L'utente deve poter visualizzare un errore nel caso in cui non abbia connesso un account a ShopChain.	Soddisfatto
R1F15	Obbligatorio	L'utente deve poter visualizzare un messaggio di avviso nel caso in cui la transazione sia già presente in blockchain$_{\mathbf{G}}$.	Soddisfatto
R1F16	Obbligatorio	L'utente deve poter visualizzare una pagina con i dettagli dell' ordine pagato.	Soddisfatto
R1F16.1	Obbligatorio	L'utente deve poter visualizzare i dettagli dell'ordine pagato.	Soddisfatto
R1F16.1.1	Obbligatorio	L'utente deve poter visualizzare l'id dell'ordine pagato.	Soddisfatto
R1F16.1.2	Obbligatorio	L'utente deve poter visualizzare l'indirizzo del venditore dell'ordine pagato.	Soddisfatto
R1F16.1.3	Obbligatorio	L'utente deve poter visualizzare l'ammontare dell'ordine pagato.	Soddisfatto
R1F16.1.4	Obbligatorio	L'utente deve poter visualizzare lo stato dell'ordine pagato.	Soddisfatto
R1F16.1.5	Obbligatorio	L'utente deve poter visualizzare la data dell'ordine pagato.	Soddisfatto

Tabella 2: Requisiti funzionali

3.2 Grafici requisiti soddisfatti

Per quanto riguarda i requisiti funzionali il gruppo *Yakuzaishi* è riuscito a soddisfarne 43 su 48, arrivando ad una copertura del 90%.

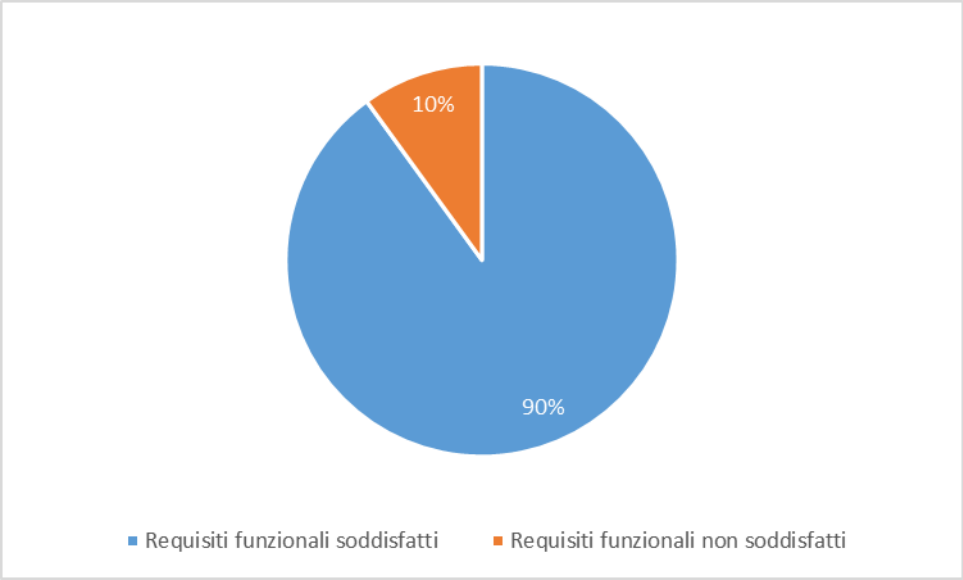

Figura 13: Requisiti funzionali soddisfatti

Riguardo invece i requisiti obbligatori siamo arrivati ad una copertura del 100%.

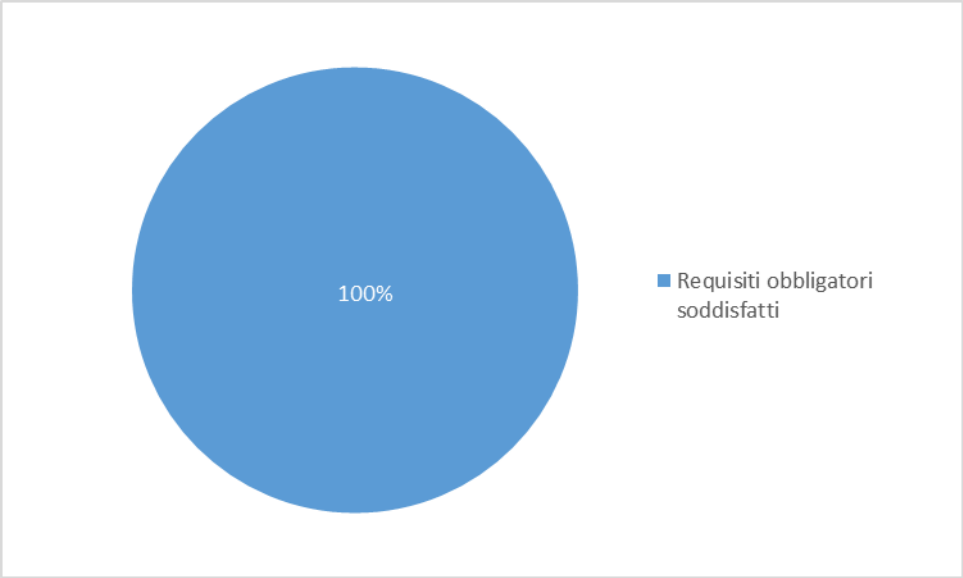

Figura 14: Requisiti obbligatori soddisfatti