



Winning Software Solution

winningssoftwaresolution@gmail.com

ShopChain

SyncLab

Specifiche Architettureali

Informazioni

<i>Redattori</i>	Giovanni Cocco
<i>Revisori</i>	Federico Marchi
<i>Responsabili</i>	Giovanni Cocco
<i>Versione</i>	1.0.0
<i>Uso</i>	esterno

Descrizione

Architettura del progetto

Versione	Data	Persona	Attività	Descrizione
1.0.0	23/2/2022	Giovanni Cocco	Redazione	Creazione del documento

Contents

1	Introduzione	3
1.1	Scopo del documento	3
2	Tecnologie/Linguaggi/Librerie	3
2.1	Solidity	3
2.2	Typescript	3
2.3	Express	3
2.4	MariaDB	3
2.5	Python	3
2.6	Web3	3
2.7	MetaMask	3
3	Architettura	4
3.1	Server	4
3.1.1	Diagramma delle classi	4
3.1.2	Design pattern: Constructor injection	4
3.2	Smart contract	5
3.2.1	Diagramma delle classi	5
3.2.2	Pattern architetturale adottato	5
3.3	Web app	5
3.3.1	Diagramma delle classi	5
3.3.2	Pattern architetturale adottato	5
3.4	Script di messa in vendita	6
3.4.1	sell_item	6
4	Diagrammi di sequenza	6
4.1	Main	6
4.2	Ascolto eventi del contratto	7
4.3	Inizializzazione server web	8
4.4	Nuovo oggetto in vendita	9
4.5	Nuova transazione	9
4.6	Cambio di stato di una transazione	10
4.7	Pagina transazioni in entrata	10

1 Introduzione

1.1 Scopo del documento

Il documento illustra le scelte architettureali e illustra l'architettura.

2 Tecnologie/Linguaggi/Librerie

2.1 Solidity

- **Versione:** 0.8.13
- **Documentazione:** <https://docs.soliditylang.org/en/v0.8.13/>

2.2 Typescript

- **Versione:** 4.6.3
- **Documentazione:** <https://www.typescriptlang.org/docs/>

2.3 Express

- **Versione:** 4.17.2
- **Documentazione:** <https://devdocs.io/express/>

2.4 MariaDB

- **Versione:** 10.7.3
- **Documentazione:** <https://mariadb.com/kb/en/documentation/>

2.5 Python

- **Versione:** 3.8
- **Documentazione:** <https://docs.python.org/3.8/>

2.6 Web3

- **Versione:** 1.7.1
- **Documentazione:** <https://web3js.readthedocs.io/en/v1.7.1/>

2.7 MetaMask

- **Versione:** 10.11.3
- **Documentazione:** <https://docs.metamask.io/guide/>

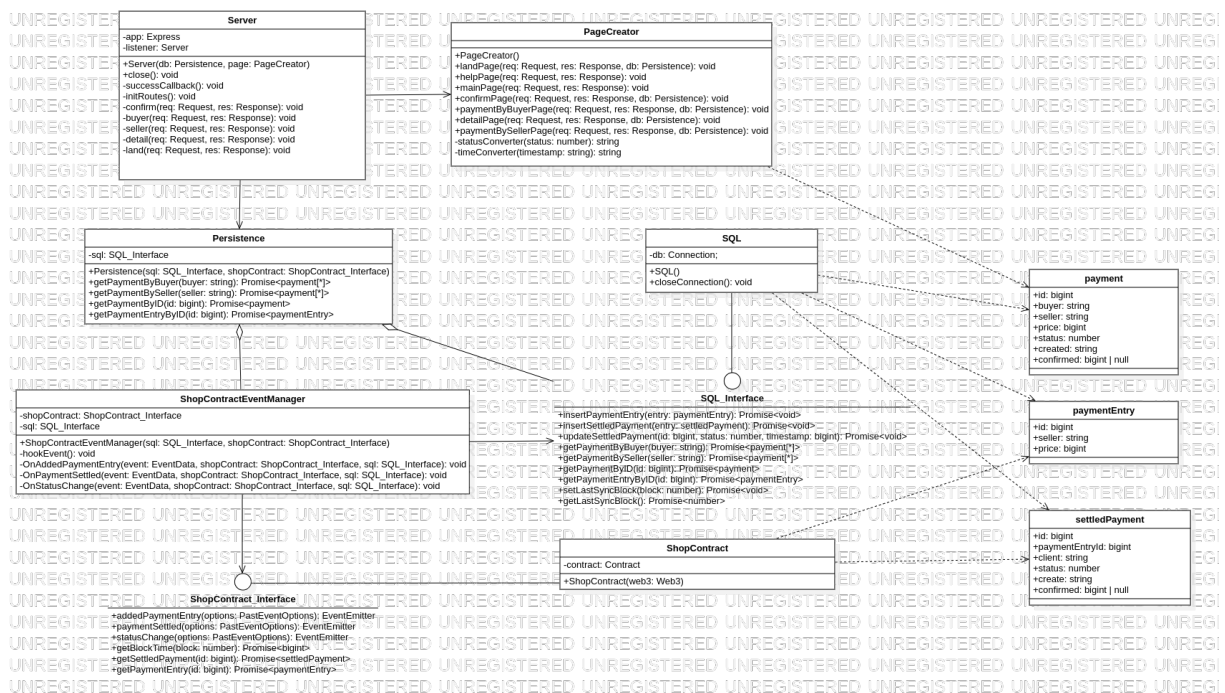
3 Architettura

Il progetto si compone di 4 macro parti:

- Server
- Smart contract
- Web app
- Script di messa in vendita

3.1 Server

3.1.1 Diagramma delle classi



Commenti

La classe ShopContract andrà a interfacciarsi con il contratto in blockchain.
La classe PageCreator andrà a interfacciarsi con la WebApp.

3.1.2 Design pattern: Constructor injection

Descrizione

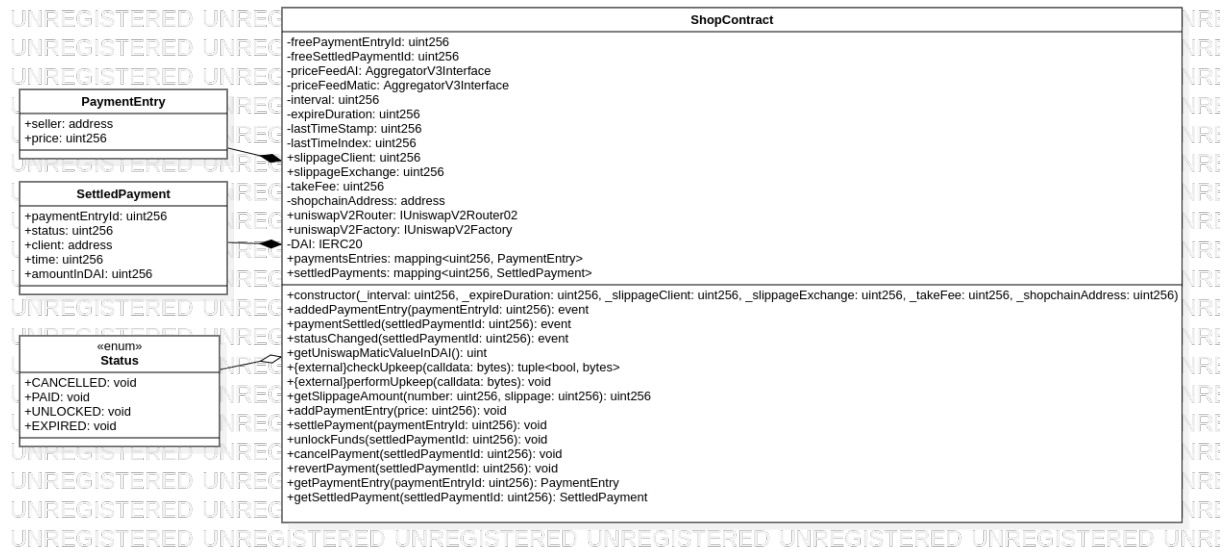
Le dipendenze sono tracciate e passate agli oggetti tramite il costruttore.

Motivazioni

Facilita il tracciamento delle dipendenze e agevola il mocking in fase di test.

3.2 Smart contract

3.2.1 Diagramma delle classi



Commenti

3.2.2 Pattern architetturale adottato

Descrizione

Motivazioni

3.3 Web app

3.3.1 Diagramma delle classi

[grafico]

Commenti

3.3.2 Pattern architetturale adottato

Descrizione

Motivazioni

3.4 Script di messa in vendita

3.4.1 sell_item

Parametri

price: float - il prezzo in dollari della entry di pagamento.

Valore Restituito

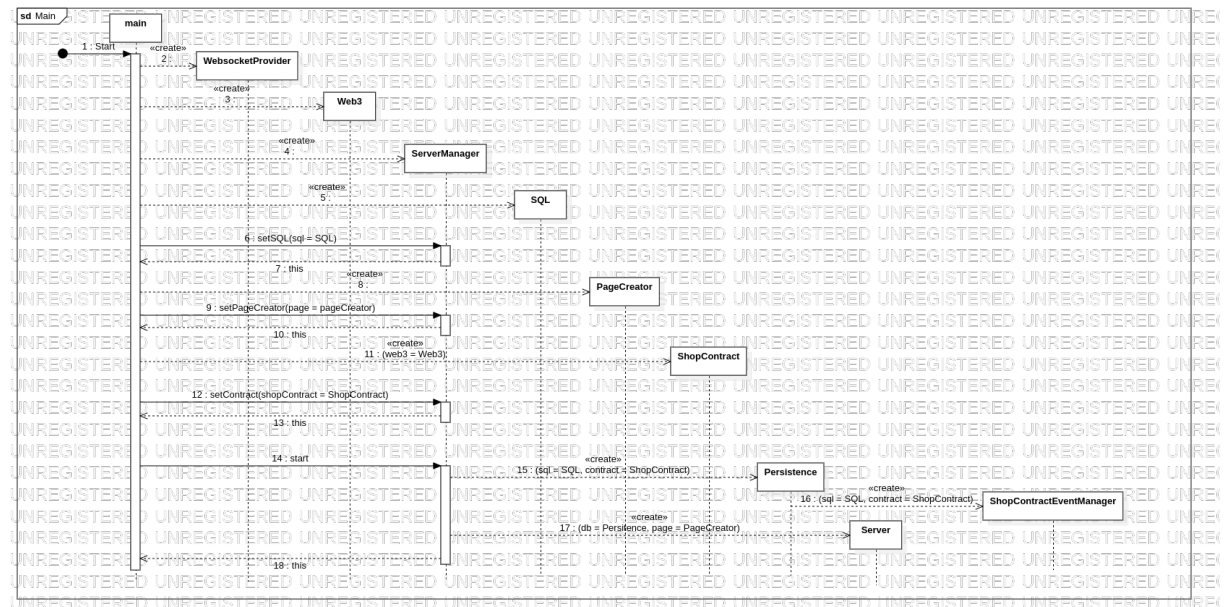
entry id: int - l'id dell'entry inserita in blockchain (-1 in caso di errore).

Comportamento

Il metodo inserisce una nuova entry di pagamento in blockchain con il prezzo specificato. Sia in caso di successo che di errore tutte le informazioni vengono salvate nel file *sell.log*.

4 Diagrammi di sequenza

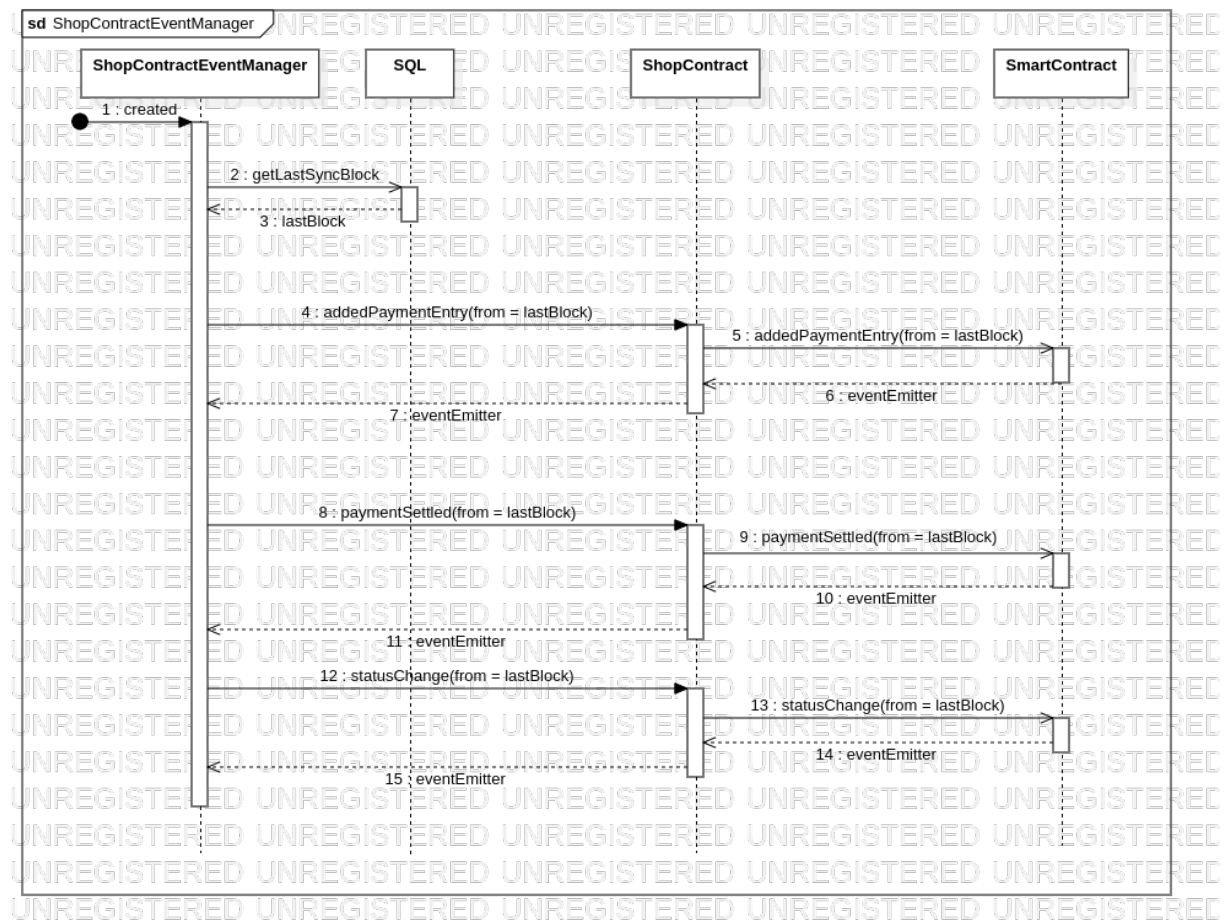
4.1 Main



Commenti

Mostra cosa avviene all'avvio del server.

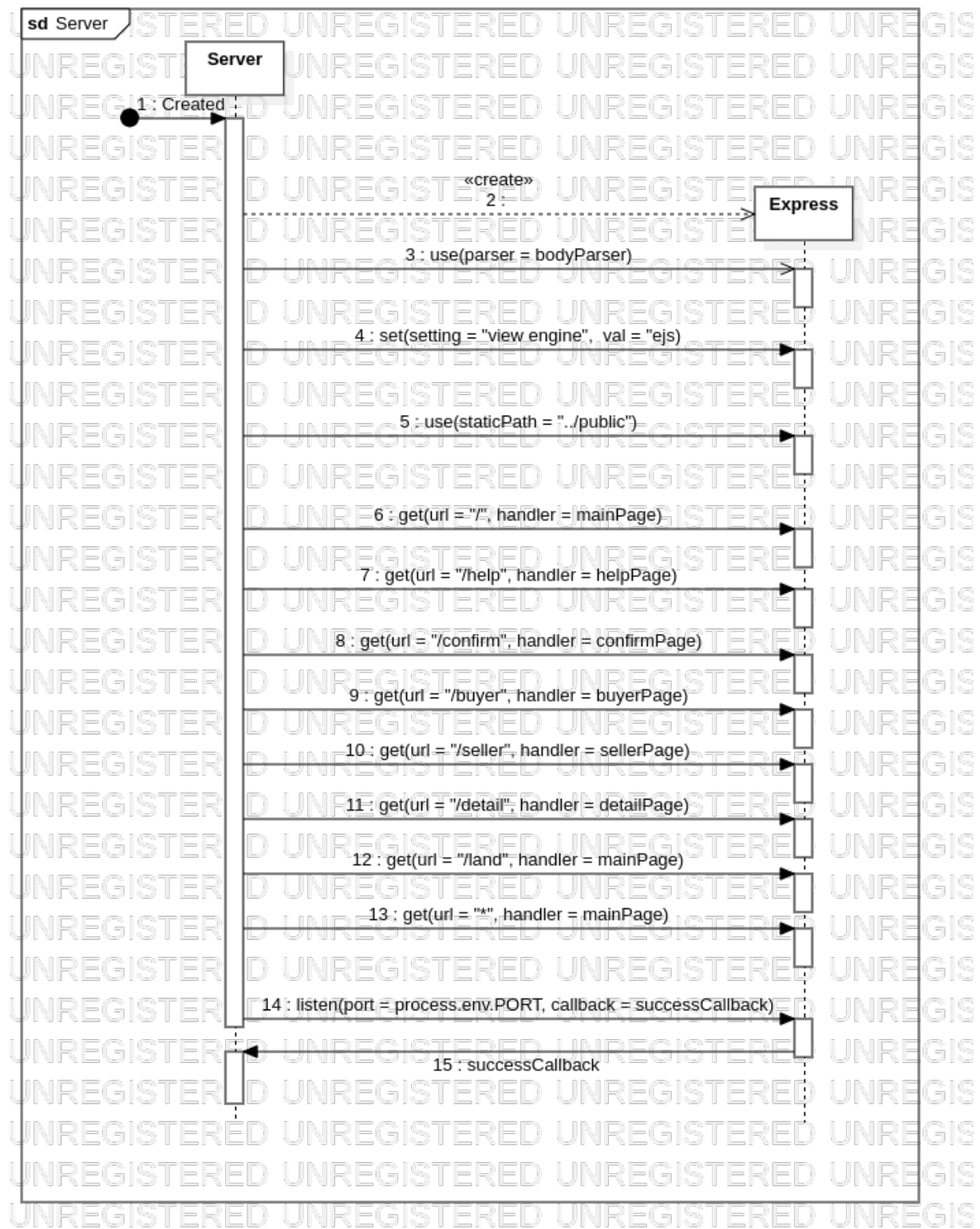
4.2 Ascolto eventi del contratto



Commenti

Mostra la sottoscrizione degli eventi del contratto.

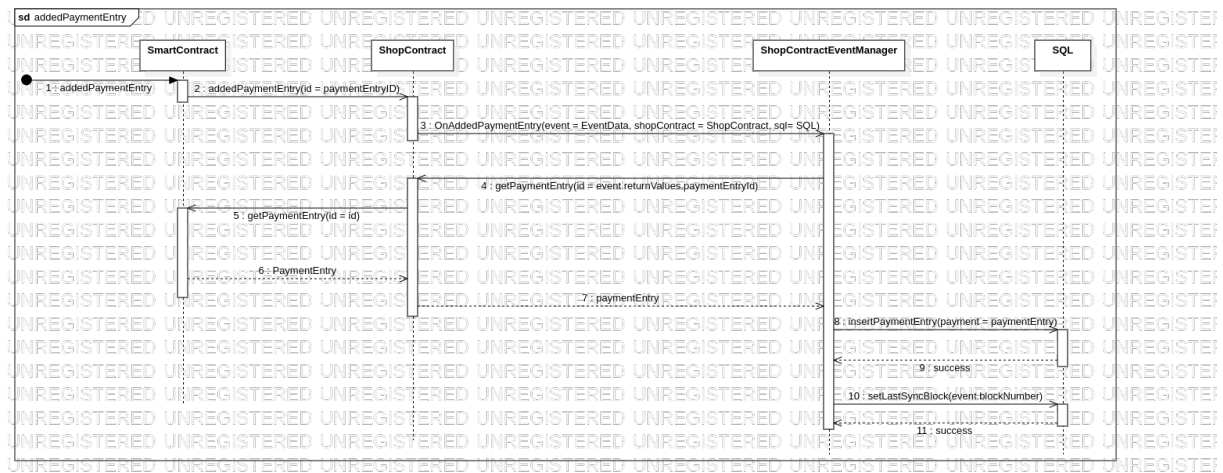
4.3 Inizializzazione server web



Commenti

Mostra l'inizializzazione delle routes per express.

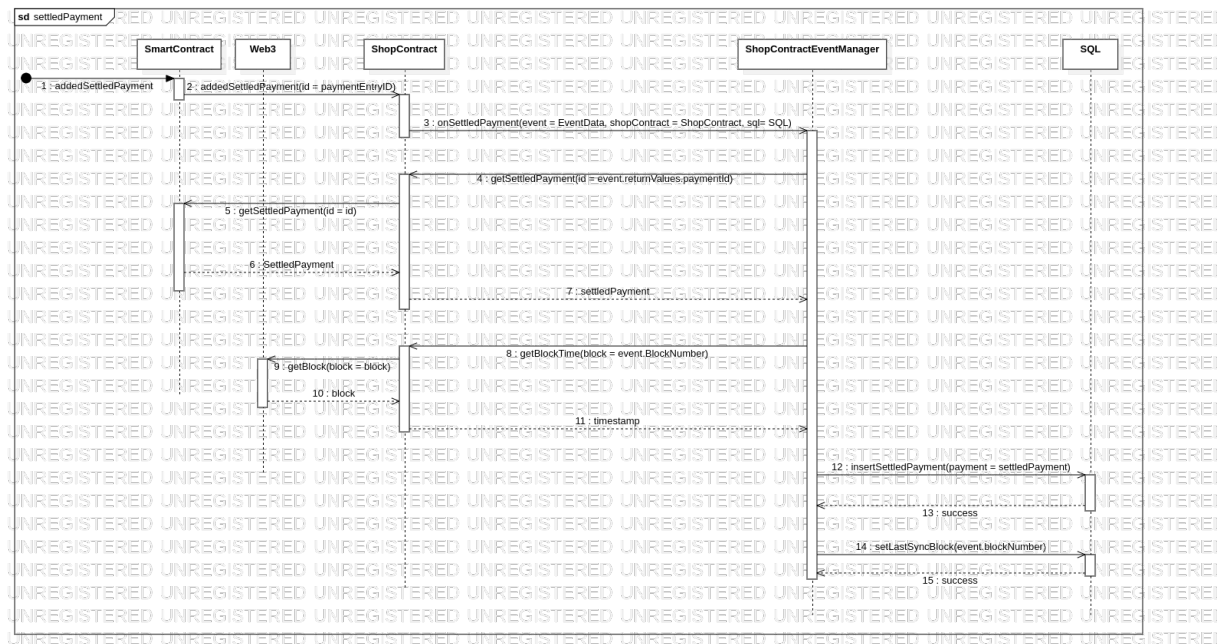
4.4 Nuovo oggetto in vendita



Commenti

Mostra cosa succede quando viene inserito una nuova entry di pagamento da parte di un e-commerce.

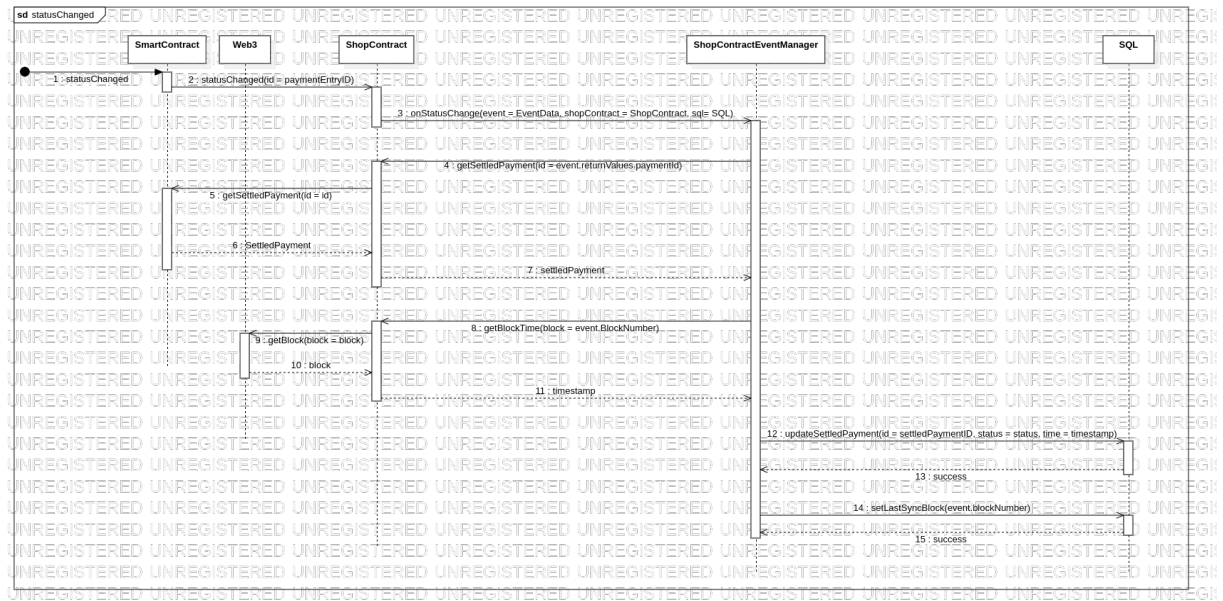
4.5 Nuova transazione



Commenti

Mostra cosa succede quando viene create una nuova transazione.

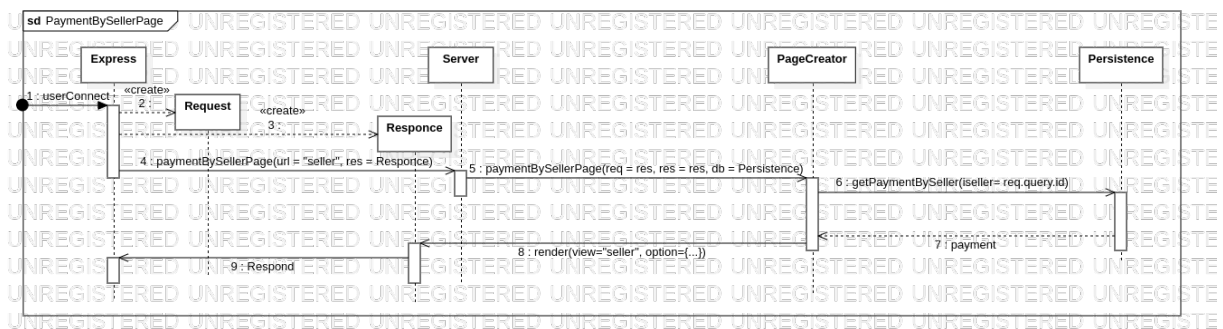
4.6 Cambio di stato di una transazione



Commenti

Mostra cosa succede quando una transazione cambia di stato.

4.7 Pagina transazioni in entrata



Commenti

Mostra cosa succede quando un utente richiede la pagina delle transazione in entrata. Le altre pagine utilizzano lo stesso modello e dunque si preferisce evitare diagrammi di sequenza ridondanti.