



# STAND UP FOR CHANGE:

Una DApp per la firma di petizioni

Marika Spagna Zito 0522501519  
Corso di Sicurezza dei Dati

# INTRODUZIONE

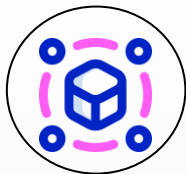
Firmare petizioni è un modo significativo e accessibile per esprimere il proprio **sostegno** o **dissenso** su questioni importanti.

**StandUpForChange** è una **DApp** che offre agli utenti una piattaforma per firmare e creare petizioni in modo sicuro e trasparente utilizzando la tecnologia blockchain.

Sviluppata sulla rete Ethereum, questa applicazione è progettata per facilitare la partecipazione attiva dei cittadini nelle questioni di interesse pubblico, consentendo loro di esprimere le proprie opinioni e sostenere cause importanti

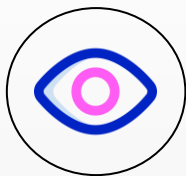


# VANTAGGI



## **Decentralizzazione:**

Non esiste un'autorità centrale che controlla il processo di creazione e firma delle petizioni. Ciò offre numerosi vantaggi in termini di trasparenza, resistenza alla censura, risparmio di costi e fiducia degli utenti.



## **Tracciabilità e trasparenza:**

Tutte le attività relative alle petizioni, inclusi i dettagli della petizione e le firme raccolte, sono pubbliche e verificabili sulla blockchain. Questo promuove la fiducia e la trasparenza nell'intero processo.



## **Immutabilità delle Firme:**

Una firma è rappresentata come una transazione che viene registrata sulla blockchain. Una volta che questa transazione è stata confermata ed inclusa in un blocco, diventa parte permanente del registro delle transazioni e non può essere cambiata retroattivamente.

# REQUISITI FUNZIONALI

## **FR 01** Visualizzazione delle petizioni

**Priorità Alta**

Tutti gli utenti potranno visualizzare l'elenco completo delle petizioni disponibili e le informazioni associate ad esse.

## **FR 02** Firma delle petizioni

**Priorità Alta**

Gli utenti potranno esprimere il loro sostegno alle petizioni esistenti firmandole attraverso l'applicazione.

## **FR 03** Creazione di una nuova petizione

**Priorità Alta**

Gli utenti avranno la possibilità di creare nuove petizioni inserendo il titolo della petizione le informazioni associate ad esse.

## **FR 04** Visualizzazione delle Petizioni Firmate **Priorità Media**

Gli utenti devono poter accedere a un elenco dettagliato delle petizioni alle quali hanno offerto il loro sostegno mediante la firma.

# REQUISITI FUNZIONALI

## **FR 05 Autenticazione degli utenti** **Priorità Alta**

Il sistema deve permettere all'utente di collegare un wallet Ethereum, garantendo che le firme sulle petizioni siano legate a identità verificate e immutabili.

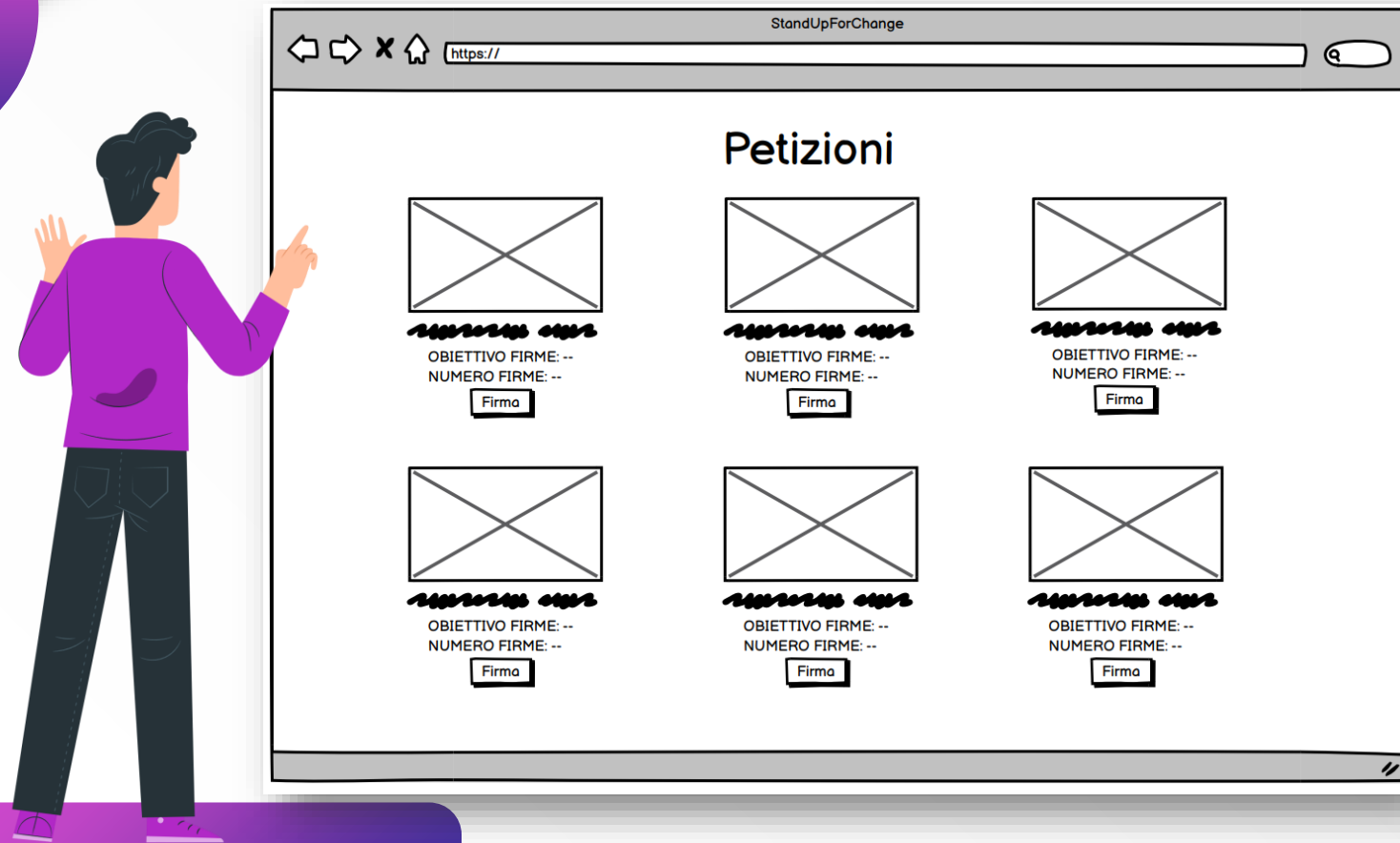
## **FR 06 Verifica delle firme** **Priorità Alta**

Il sistema deve tenere traccia del numero di firme per ciascuna petizione e rendere questa informazione accessibile agli utenti.

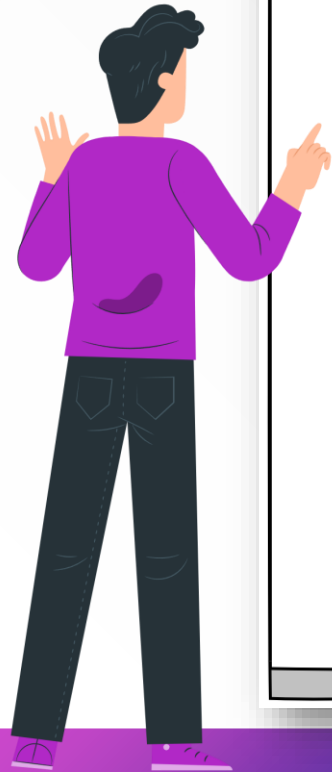
## **FR 07 Aggiornamento dello Stato della Petizione:** **Priorità Media**

Il sistema deve automaticamente aggiornare lo stato delle petizioni in funzione del numero di firme raccolte, indicando, ad esempio, la condizione di «Attiva» o «Completata».

# MOCK-UP



# MOCK-UP



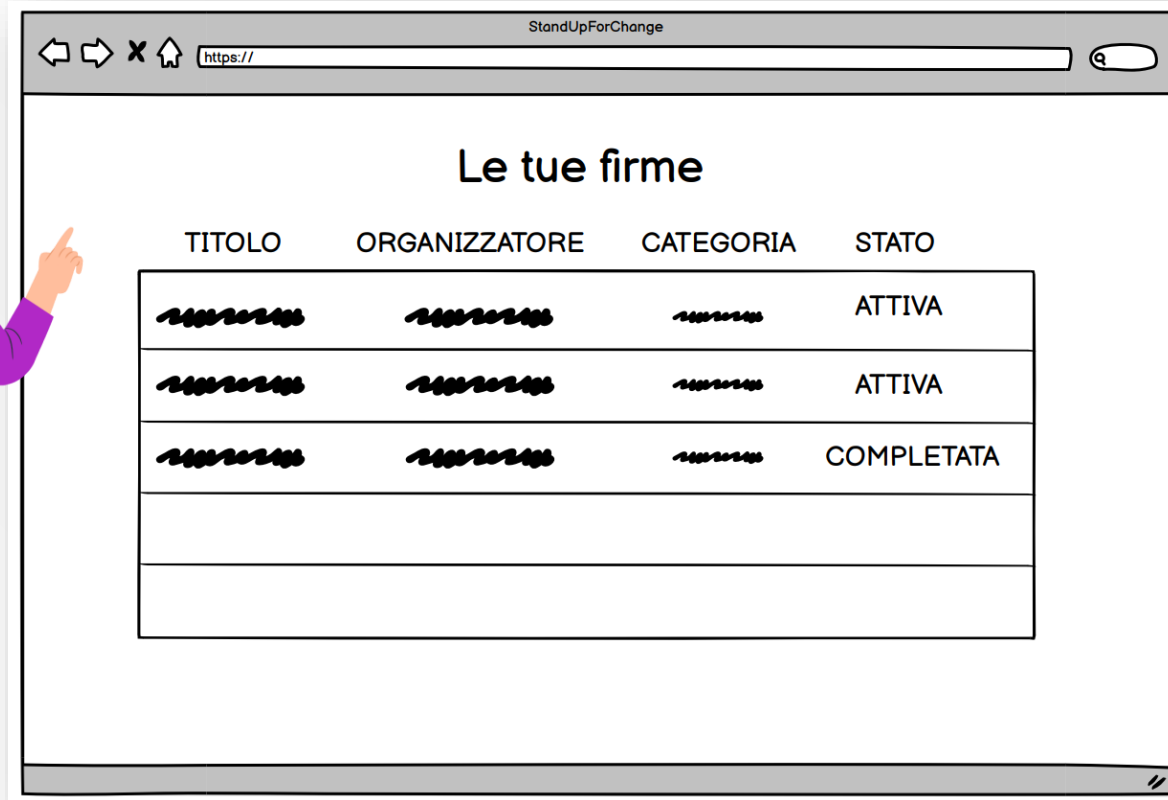
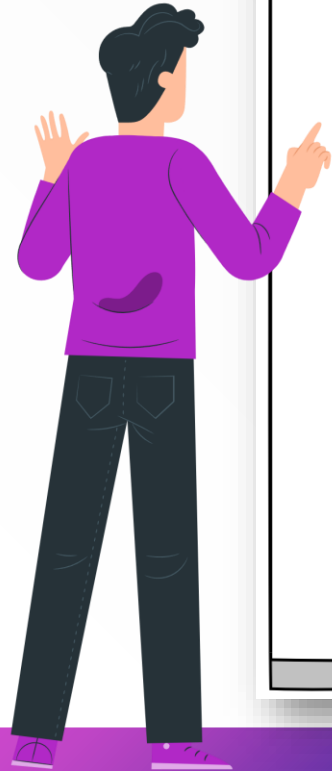
StandUpForChange

https://

## Crea una petizione

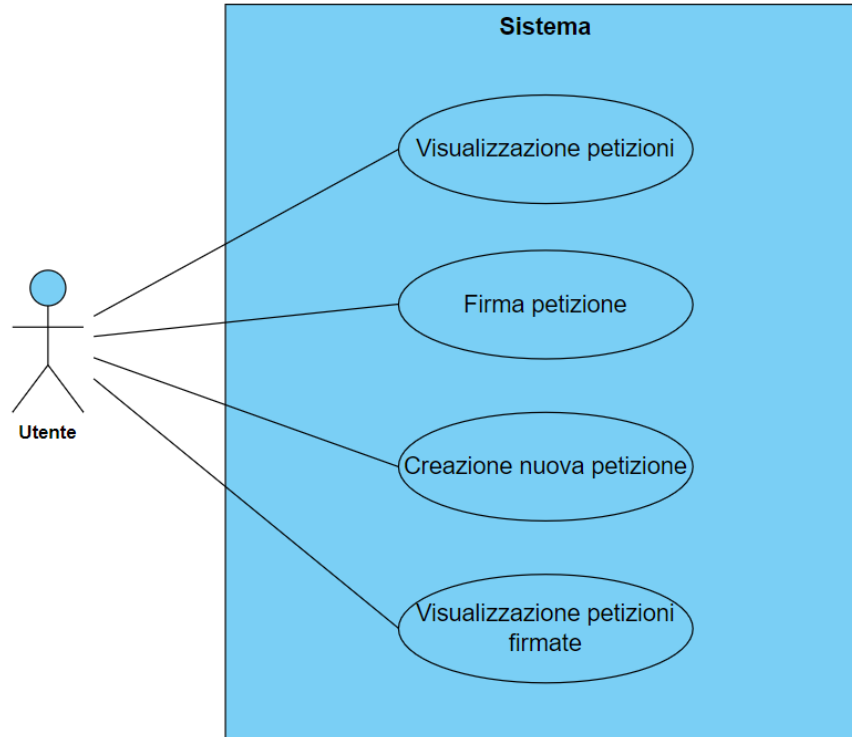
TITOLO	ORGANIZZATORE
<input type="text"/>	<input type="text"/>
CATEGORIA	OBIETTIVO FIRME
<input type="text"/>	<input type="text"/>
LINK IMMAGINE	
<input type="text"/>	
<input type="button" value="Crea"/>	

# MOCK-UP





# USE CASES MODEL



# USE CASES

## *UC-01: Creazione nuova petizione*

**Attore:** Utente

**Entry condition:** L'utente si trova nella pagina home della dApp e ha collegato il suo portafoglio Metamask.

**Flusso degli eventi:**

1. L'utente naviga verso la sezione "Crea una nuova petizione"
2. Visualizza un form che richiede l'inserimento dei seguenti campi:

**Titolo della petizione:** Un breve e descrittivo titolo che identifica la petizione.

**Organizzatore:** chi ha organizzato la petizione

**Categoria:** La categoria a cui appartiene la petizione (es. Ambiente, Diritti Umani, Salute, ecc.).

**Obiettivo firme:** Il numero minimo di firme necessarie per raggiungere l'obiettivo della petizione.

**Link immagine:** Immagine della petizione

3. L'utente compila i campi e sottomette il form

4. Il sistema verifica che tutti i campi sono stati compilati

5. Il sistema mostra una schermata per confermare la transazione.

6. L'utente autorizza la transazione.

**Exit condition (On Success):** Il sistema crea la petizione e notifica all'utente che la petizione è stata correttamente creata.

**Exit condition (On failure):** Il Sistema notifica che la petizione non è stata creata.

# USE CASES

## UC-02: Firma petizione

**Attore:** Utente

**Entry condition:** L'utente si trova nella pagina home della dApp e ha collegato il suo portafoglio Metamask.

**Flusso degli eventi:**

1. L'utente naviga attraverso l'elenco delle petizioni disponibili sulla dApp. Ogni petizione presenta le seguenti informazioni: Immagine della petizione, titolo della petizione, organizzatore, categoria, il totale delle firme raccolte fino a quel momento, il numero di firme che l'organizzatore si prefigge di raggiungere.
2. L'utente seleziona una delle petizioni attive.
3. L'utente clicca sul pulsante "Firma" posizionato sotto le informazioni della petizione scelta.
4. Il Sistema controlla se l'utente ha già firmato quella petizione e se la petizione è ancora attiva.
5. Il sistema mostra una schermata per confermare la transazione.
6. L'utente autorizza la transazione.

**Exit condition (On Success):** Il sistema firma la petizione e notifica all'utente che la petizione è stata correttamente fermata. Inoltre incrementa il numero di firme della petizione.

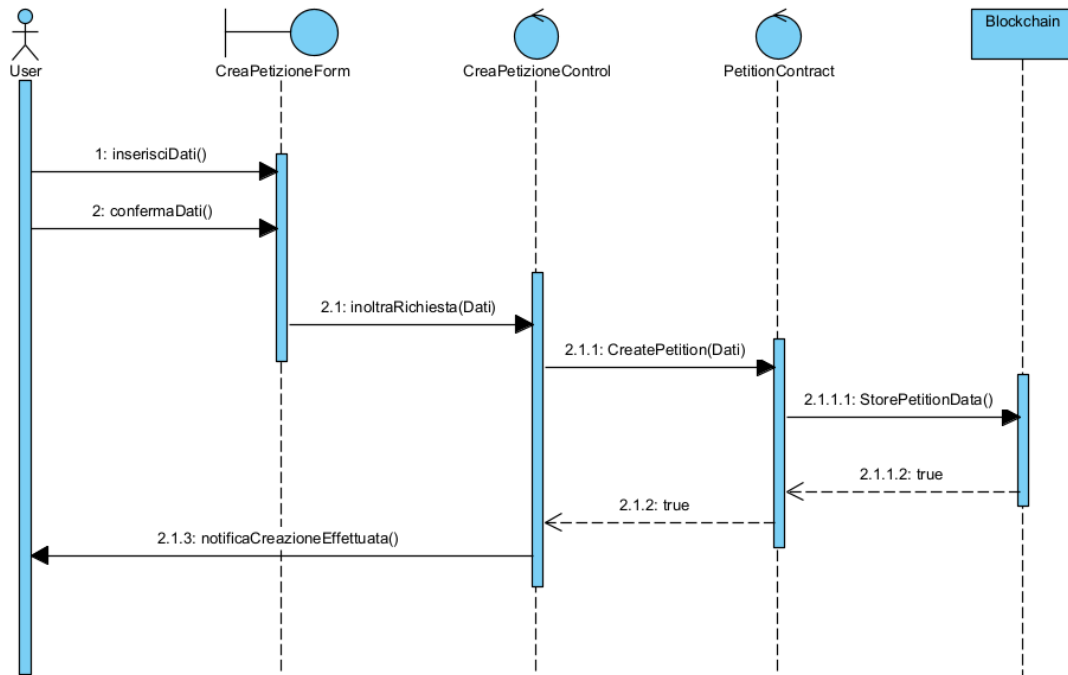
**Exit condition (On failure):** Il Sistema notifica che la firma non è stata correttamente effettuata.

**Flussi alternativi:**

- Se nel punto 4 il Sistema nota che l'utente ha già firmato la petizione, la transazione è annullata e mostra all'utente il seguente messaggio di errore "Hai già firmato questa petizione".
- Se nel punto 4 il Sistema nota che la petizione ha già raggiunto il numero di firme obiettivo mostra all'utente il seguente messaggio di errore " La petizione è stata completata".

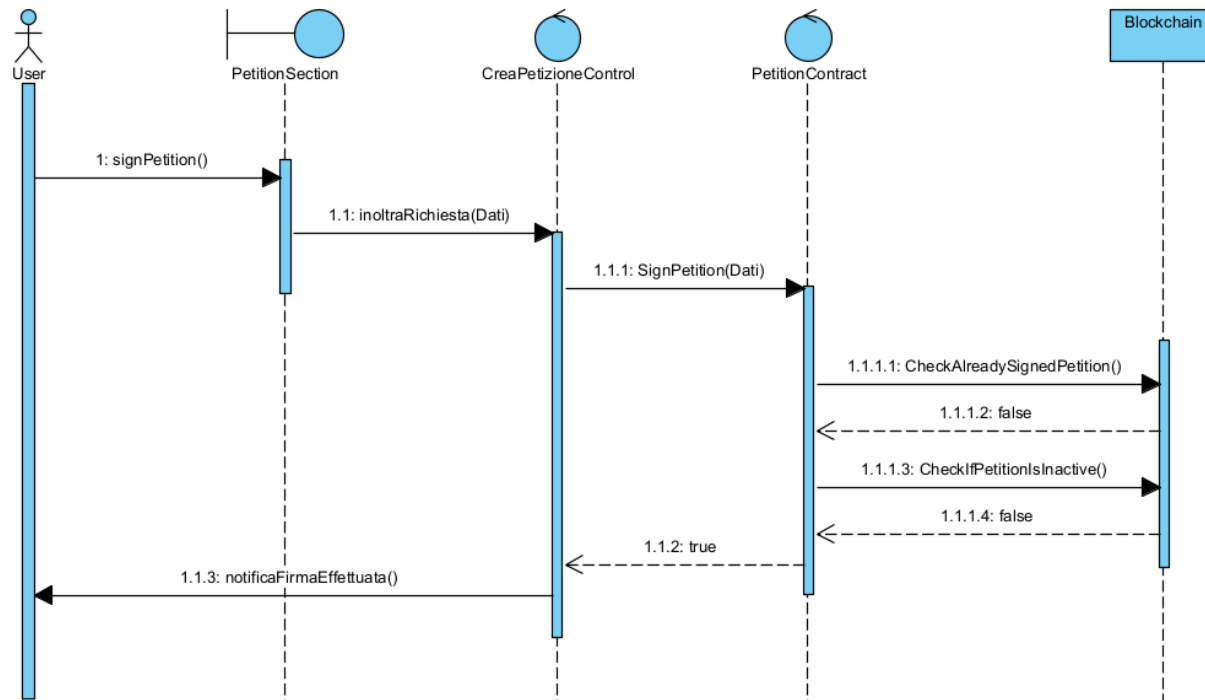
# SEQUENCE DIAGRAM

## *SD-01: Creazione petizione*



# SEQUENCE DIAGRAM

## *SD-02: Firma petizione*



# STRUMENTI PER LO SVILUPPO

## TRUFFLE

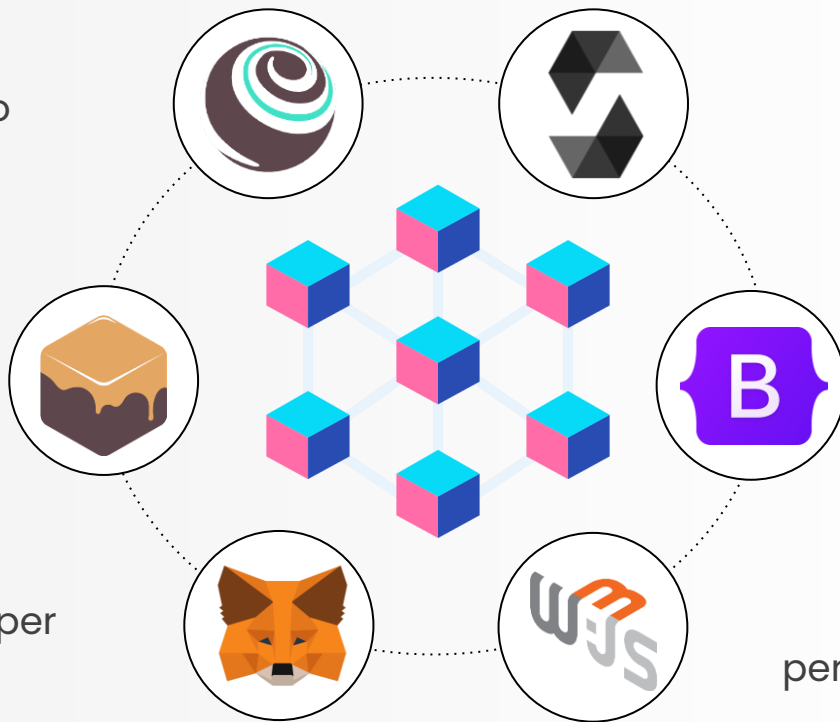
Framework lo sviluppo di smart contract su blockchain

## GANACHE

Simula l'ambiente blockchain di Ethereum

## METAMASK

Wallet di criptovalute per interagire con la blockchain



## SOLIDITY

Linguaggio degli smart contracts

## BOOTSTRAP

Framework di sviluppo front-end

## WEB3.js

Libreria JavaScript che permette l'interfacciamento con le blockchain

# CODICE CONTRATTO

```
contract PetitionContract {
    uint public counterPetition;

    // Definizione della struttura della petizione
    struct Petition {
        uint id;
        string title;
        string organizer;
        string img;
        string category;
        uint targetSignatures;
        uint numSignatures;
        bool isActive;
    }

    // Mapping degli ID delle petizioni alle loro struct Petition
    mapping(uint => Petition) public petitions;

    // Mapping degli indirizzi che hanno firmato petizioni agli ID delle petizioni firmate
    mapping(address => uint[]) public signedPetitions;

    // Evento emesso quando una petizione viene firmata
    event PetitionSigned(address indexed signer, uint indexed petitionId);
    //Evento emesso quando una petizione viene creata
    event PetitionCreated(uint indexed id, string title, string organizer, string img,
        string category, uint targetSignatures );
}
```

# CODICE CONTRATTO

```
// Funzione per creare una nuova petizione
function createPetition(
  uint id,
  string memory title,
  string memory organizer,
  string memory img,
  string memory category,
  uint targetSignatures,
  uint numSignatures
) public {
  require(petitions[id].id == 0, "Petition with same ID already exists"); // Verifica se esiste già una petizione con lo stesso ID

  petitions[id] = Petition(
    id,
    title,
    organizer,
    img,
    category,
    targetSignatures,
    numSignatures,
    true
  );
  counterPetition++;
  // Emetti l'evento PetitionCreated
  emit PetitionCreated(id, title, organizer, img, category, targetSignatures);
}
```



# CODICE CONTRATTO

```
// Funzione per firmare una petizione
function signPetition(uint petitionId) public {
    Petition storage petition = petitions[petitionId];
    require(petition.isActive, "Petition is not active");

    // Controllo per verificare se l'utente ha già firmato
    require(!hasSignedPetition(msg.sender, petitionId), "You have already signed this petition");

    petition.numSignatures++;
    signedPetitions[msg.sender].push(petitionId);

    if (petition.numSignatures >= petition.targetSignatures) {
        petition.isActive = false;
    }
    emit PetitionSigned(msg.sender, petitionId);
}
```

```
// Funzione per verificare se un utente ha già firmato una petizione
function hasSignedPetition(address signer, uint petitionId) internal view returns (bool) {
    uint[] storage signedIds = signedPetitions[signer];
    for (uint i = 0; i < signedIds.length; i++) {
        if (signedIds[i] == petitionId) {
            return true;
        }
    }
    return false;
}
```

# TESTING

Sono stati svolti test in JavaScript utilizzando il framework di testing per smart contracts di Ethereum chiamato **Mocha**, insieme alla libreria di assert di Node.js per le asserzioni.

Per far partire i test bisogna digitare da terminale il seguente comando:

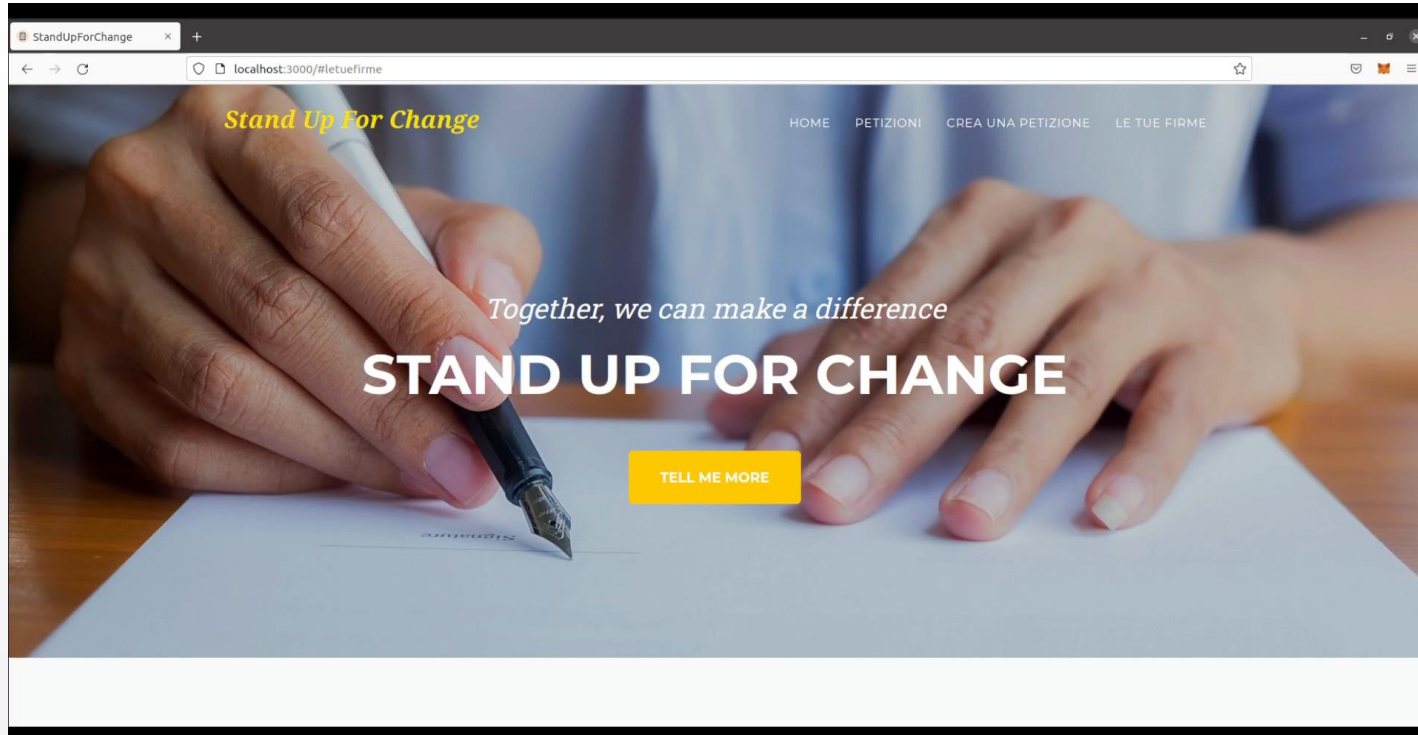
```
> truffle test
```

```
Contract: PetitionContract
```

- ✓ should create a new petition (508ms)
- ✓ should sign a petition (785ms)
- ✓ should not allow duplicate petitions with the same ID (1615ms)
- ✓ should not allow signing an inactive petition (1102ms)
- ✓ should get signed petitions for an address (1486ms)
- ✓ should get signed petition IDs for an address (1287ms)
- ✓ should not allow signing an already signed petition (664ms)

```
7 passing (31s)
```

# DEMO DAPP



# DEMO FIRMA PETIZIONE

StandUpForChange

localhost:3000/#petizioni

Stand Up For Change

HOMEPETIZIONI CREA UNA PETIZIONE LE TUE FIRME

PETIZIONI

Firma oggi, cambia il domani.

**Ambiente**

**Basta plastica monouso:  
Agiamo Ora!**

*GreenEarth Italia*

OBBIETTIVO FIRME: 13000  
NUMERO FIRME: 70

Firma

**Diritti Umani**

**Stop alle discriminazioni  
sul lavoro**

*YourVoice*

OBBIETTIVO FIRME: 20000  
NUMERO FIRME: 8000

Firma

**Animali**

**No ai Test su Animali:  
Cosmetici Cruelty-Free**

*CrueltyFree Beauty*

OBBIETTIVO FIRME: 8000  
NUMERO FIRME: 7999

Firma

**Salute Mentale e Istruzione**

**Sosteniamo la salute  
mentale studentesca**

*Fondazione ScuolaSana*

OBBIETTIVO FIRME: 10000  
NUMERO FIRME: 758

Firma

# DEMO CREA PETIZIONE

StandUpForChange

localhost:3004/#petizioni

HOME PETIZIONI CREA UNA PETIZIONE LE TUE FIRME

## PETIZIONI

*Firma oggi, cambia il domani.*

**Ambiente**

**Basta plastica monouso:  
Agiamo Ora!**

*GreenEarth Italia*

OBIETTIVO FIRME: 13000  
NUMERO FIRME: 70

Firma

**Diritti Umani**

**Stop alle discriminazioni  
sul lavoro**

*YourVoice*

OBIETTIVO FIRME: 20000  
NUMERO FIRME: 8000

Firma

**Animali**

**No ai Test su Animali:  
Cosmetici Cruelty-Free**

*CrueltyFree Beauty*

OBIETTIVO FIRME: 8000  
NUMERO FIRME: 7999

Firma

**Salute Mentale e Istruzione**

**Sosteniamo la salute  
mentale studentesca**

*Fondazione ScuolaSana*

OBIETTIVO FIRME: 10000  
NUMERO FIRME: 758

Firma

# POSSIBILI SVILUPPI FUTURI

- **Notifiche e aggiornamenti in tempo reale:** Fornire notifiche agli utenti quando nuove petizioni vengono create o quando vengono raggiunti determinati obiettivi di firma.
- **Integrazione di strumenti di comunicazione social:** Consentire agli utenti di condividere petizioni tramite piattaforme di social media e facilitare la diffusione della petizione attraverso reti di contatti.
- **Integrazione di criptovalute:** Consentire agli utenti di contribuire finanziariamente alle petizioni o di offrire incentivi per la firma attraverso l'uso di criptovalute.
- **Integrazione di sistemi di IA per analisi e filtraggio dei contenuti:** L'IA può essere utilizzata per analizzare il testo delle petizioni al fine di identificare errori grammaticali, incoerenze semantiche o ambiguità, ma anche per filtrare e rimuovere eventuali contenuti offensivi, diffamatori o inappropriati.





**GRAZIE PER  
L'ATTENZIONE**

---