



POLITECNICO DI BARI

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE
Corso di Laurea in Ingegneria Informatica

Relazione in Software Architecture and Pattern Design

Progettazione ed implementazione delle funzionalità relative alla sicurezza e alla ge- stione dei ruoli sulla piattaforma Filiera360

Professoressa
Prof. Ing. Marina Mongiello

Partecipanti
Cristian Corrado, 590470
Federico Piconese, 590295
Antonio Mineccia, 598553

Anno Accademico 2024 - 2025

Indice

1	Introduzione	1
2	Background e contesto	2
2.1	Introduzione alla tracciabilità alimentare	2
2.2	Il ruolo della tecnologia nella tracciabilità	2
2.3	Filiera360: un sistema di tracciabilità basato su blockchain	3
2.4	Limiti e sfide iniziali del sistema	3
3	Problematiche e scelte progettuali	4
3.1	Gestione della registrazione e dei ruoli	4
3.2	Sicurezza e protezione dei dati	4
3.3	Gestione dell'accesso alle funzionalità	5
3.4	Scalabilità e ottimizzazione delle prestazioni	5
3.5	Usabilità e accessibilità	5
4	Processo di sviluppo	7
4.1	Fasi del processo di sviluppo	7
4.2	Gestione del team e delle risorse	8
4.3	Difficoltà incontrate e soluzioni adottate	8
4.4	Strumenti utilizzati	9
5	Implementazione	10
5.1	Potenziamento del sistema di autenticazione	10
5.1.1	Autenticazione a due fattori (2FA)	10
5.1.2	Reimpostazione della password	11
5.2	Sviluppo del sistema di gestione ruoli e permessi	12
5.3	Funzionalità specifiche per ciascun ruolo	12
5.4	Implementazione della registrazione per i produttori	13
5.5	Gestione degli operatori da parte dei produttori	14
5.5.1	Recupero della lista degli operatori	14
5.5.2	Aggiunta di un operatore	14
5.5.3	Rimozione di un operatore	15

6	Validazione	16
6.1	Introduzione	16
6.2	Scelta del ruolo in fase di registrazione	16
6.3	Autenticazione a due fattori (OTP)	17
6.4	Differenziazione dell'interfaccia in base al ruolo	18
6.4.1	Interfaccia per l'utente (User)	18
6.4.2	Interfaccia per il produttore (Producer)	19
6.4.3	Interfaccia per l'operatore (Operator)	20
6.5	Utilizzo del token per la registrazione dei produttori	20
6.6	Gestione degli operatori da parte dei produttori	21
7	Conclusioni e sviluppi futuri	23

Capitolo 1

Introduzione

Il progetto "Filiera360" mira a rivoluzionare il sistema di tracciabilità dei prodotti, fornendo una piattaforma innovativa che integra tecnologie avanzate come blockchain, modelli 3D e codici QR. Attraverso l'utilizzo della blockchain per garantire la sicurezza e l'immutabilità dei dati, il sistema consente di tracciare ogni fase del ciclo di vita del prodotto, dalla produzione alla distribuzione. Con un'interfaccia web sviluppata in React e un backend basato su Flask, la piattaforma assicura trasparenza e autenticità, migliorando la gestione della supply chain e facilitando l'accesso alle informazioni per gli utenti finali.

Il ruolo che ci è stato assegnato riguarda l'implementazione di importanti miglioramenti, come l'integrazione di un sistema di autenticazione avanzato, con autenticazione a due fattori e reimpostazione sicura della password. Inoltre, è stato creato un sistema di registrazione per i produttori tramite inviti basati su token e un sistema di gestione dei ruoli e dei permessi, permettendo una gestione sicura e controllata dei vari utenti della piattaforma.

Capitolo 2

Background e contesto

2.1 Introduzione alla tracciabilità alimentare

La tracciabilità alimentare rappresenta un elemento chiave per garantire sicurezza, qualità e trasparenza nel settore agroalimentare. Con l'aumento della complessità delle catene di approvvigionamento e la crescente consapevolezza dei consumatori riguardo l'origine e la qualità dei prodotti, diventa sempre più importante adottare sistemi di tracciabilità efficienti e sicuri.

I sistemi tradizionali di tracciabilità si basano su database centralizzati e documentazione cartacea, risultando spesso vulnerabili a manipolazioni, perdite di dati e mancanza di trasparenza. Queste limitazioni hanno portato alla necessità di soluzioni innovative che garantiscano l'integrità e la verificabilità delle informazioni lungo l'intera filiera produttiva.

2.2 Il ruolo della tecnologia nella tracciabilità

Per superare queste limitazioni, negli ultimi anni si è assistito all'introduzione di nuove tecnologie volte a migliorare la gestione della tracciabilità. Tra queste, la blockchain si è affermata come una delle soluzioni più promettenti grazie alla sua capacità di garantire immutabilità, sicurezza e decentralizzazione dei dati.

La blockchain consente di registrare ogni evento della filiera in un registro distribuito, verificabile da tutti gli attori coinvolti e immune da alterazioni. Questo approccio permette di ridurre il rischio di frodi e di aumentare la fiducia nei prodotti alimentari, offrendo ai consumatori la possibilità di accedere a informazioni dettagliate sull'origine e sul percorso degli alimenti.

2.3 Filiera360: un sistema di tracciabilità basato su blockchain

Il progetto Filiera360 nasce con l'obiettivo di rivoluzionare il settore della tracciabilità alimentare attraverso l'uso della tecnologia blockchain. La blockchain è una tecnologia decentralizzata e immutabile che consente di registrare ogni transazione in modo sicuro e trasparente, eliminando la necessità di intermediari e riducendo il rischio di frodi.

Filiera360 permette agli attori della filiera agroalimentare (produttori, operatori e consumatori) di accedere a dati certificati e verificabili, migliorando la fiducia nel sistema e garantendo una maggiore sicurezza alimentare. L'uso di contratti intelligenti (smart contract) automatizza molte delle operazioni legate alla gestione della tracciabilità, riducendo errori e costi operativi.

2.4 Limiti e sfide iniziali del sistema

Nonostante i vantaggi offerti da Filiera360, la versione originale del progetto presentava alcune criticità che ne limitavano l'efficacia e l'usabilità. Tra i principali problemi identificati vi erano:

- Gestione degli accessi poco sicura: il sistema di autenticazione iniziale non garantiva un livello di sicurezza adeguato, esponendo la piattaforma a potenziali attacchi informatici e accessi non autorizzati.
- Ruoli e permessi poco strutturati: la piattaforma non offriva una gestione chiara dei diversi livelli di accesso, rendendo complessa l'attribuzione dei permessi a utenti, produttori e operatori.
- Processo di registrazione non ottimizzato: la registrazione dei produttori avveniva senza un meccanismo di verifica robusto, aumentando il rischio di accessi non autorizzati e compromettendo l'affidabilità del sistema.

Questi limiti hanno reso necessario un intervento mirato per migliorare l'infrastruttura della piattaforma, ottimizzandone la sicurezza e l'organizzazione.

Capitolo 3

Problematiche e scelte progettuali

3.1 Gestione della registrazione e dei ruoli

Inizialmente, la piattaforma non aveva un sistema strutturato per differenziare gli utenti in base ai ruoli. Questo portava ad una gestione inefficace delle funzionalità, dove tutti gli utenti avevano accesso alle stesse opzioni, indipendentemente dalle loro esigenze o autorizzazioni. Un utente che doveva solo visualizzare informazioni poteva accidentalmente accedere a funzionalità di modifica, causando potenziali errori o modifiche non autorizzate. Inoltre, senza un meccanismo di controllo, chiunque poteva registrarsi come produttore, minacciando l'integrità della piattaforma e la qualità dei dati. Questo predisponava l'accesso alla piattaforma di aziende non autorizzate, che potevano comportare una rappresentazione inaccurata del mercato e potenziali problemi legali.

Oltre ad un meccanismo di controllo completamente inesistente, l'accesso alle sezioni della piattaforma non era adeguatamente protetto, permettendo a utenti non autorizzati di accedere a dati sensibili. Di conseguenza, un utente standard poteva tranquillamente accedere a funzionalità riservate/non fruibili in base alla tipologia di utente loggato, mettendo a rischio la sicurezza dei dati.

3.2 Sicurezza e protezione dei dati

Un'altra problematica critica affrontata durante lo sviluppo della piattaforma è stata la sicurezza. L'assenza di un sistema di autenticazione avanzato rendeva la piattaforma vulnerabile ad accessi non autorizzati. Senza un'autenticazione a due fattori (2FA) o un meccanismo di recupero password sicuro, gli utenti erano esposti a possibili attacchi informatici, come il furto di credenziali o l'accesso non autorizzato ai dati sensibili.

Per risolvere questa problematica, è stata migliorata l'autenticazione basata su JSON Web Token (JWT), con l'aggiunta della 2FA per garantire un ulteriore livello di sicurezza.

3.3 Gestione dell'accesso alle funzionalità

La piattaforma inizialmente non aveva una chiara separazione delle funzionalità tra i diversi ruoli. Questa lacuna ha richiesto una ristrutturazione completa del sistema di autorizzazione per garantire che ogni utente potesse accedere solo alle funzionalità pertinenti al proprio ruolo. Per esempio:

- Gli utenti standard possono solo visualizzare i prodotti e salvare preferiti.
- I produttori possono registrare nuovi prodotti, modificarli e gestire gli operatori.
- Gli operatori possono gestire i lotti di prodotti, ma sempre sotto l'autorizzazione del produttore associato.

Per implementare questa gestione dei ruoli, sono stati definiti specifici permessi per ogni tipologia di utente, verificando i privilegi di accesso tramite JWT in ogni richiesta.

3.4 Scalabilità e ottimizzazione delle prestazioni

Con la crescita del numero di utenti e produttori sulla piattaforma, è stato necessario affrontare le problematiche legate alla scalabilità. L'architettura iniziale non era ottimizzata per gestire un elevato numero di richieste simultanee, causando rallentamenti e possibili colli di bottiglia.

3.5 Usabilità e accessibilità

Un'altra problematica emersa nelle prime fasi di sviluppo riguardava l'interfaccia utente, che risultava poco intuitiva e complessa da navigare. La disposizione degli elementi non seguiva una logica chiara, e alcune funzionalità erano difficili da trovare o richiedevano troppi passaggi per essere eseguite. Questo causava frustrazione negli utenti e aumentava il rischio di errori operativi.

Per risolvere questo problema, è stato necessario ridefinire l'organizzazione dei componenti dell'interfaccia, migliorando la distribuzione degli elementi e semplificando l'accesso alle funzionalità principali. Sono state adottate scelte progettuali che favoriscono un'esperienza utente più fluida e intuitiva, come la riorganizzazione

dei menu, la riduzione del numero di clic necessari per completare un'operazione e una maggiore coerenza visiva tra le diverse sezioni della piattaforma.

Questi miglioramenti hanno contribuito a rendere la piattaforma più accessibile ed efficiente, riducendo il tempo necessario per svolgere le operazioni e minimizzando il rischio di errori.

Capitolo 4

Processo di sviluppo

Il processo di sviluppo del sistema è stato organizzato seguendo un approccio iterativo e incrementale. Inizialmente, il team ha dedicato del tempo all'apprendimento della piattaforma esistente, esaminando il codice e comprendendo le funzionalità di base. Successivamente, si è optato per la suddivisione in due gruppi, con l'obiettivo di affrontare in parallelo le diverse task assegnate.

Il team iniziale era composto da cinque membri; in particolare, il nostro gruppo si è occupato del potenziamento del sistema di autenticazione, della definizione dei ruoli e dell'implementazione dei permessi associati a ciascun ruolo. Gli altri due membri del team si sono concentrati su altre aree del progetto.

4.1 Fasi del processo di sviluppo

Il processo di sviluppo è stato suddiviso nelle seguenti fasi principali:

- **Analisi dei requisiti:** In questa fase iniziale, sono stati esaminati i requisiti del progetto, basandosi sul sistema esistente. L'obiettivo era quello di comprendere le funzionalità di base e le aree in cui era necessario implementare le migliorie citate in precedenza.
- **Progettazione:** Durante la fase di progettazione, sono stati definiti i flussi di lavoro e le interfacce utente per ogni nuova funzionalità. Le specifiche tecniche sono state redatte in modo che ciascun gruppo potesse lavorare in modo indipendente ma coerente con gli altri sviluppatori.
- **Implementazione:** Seguendo un ordine cronologico, le task che sono state sviluppate riguardano:
 - Potenziamento del sistema di autenticazione, con l'integrazione dell'autenticazione a due fattori e l'implementazione della reimpostazione sicura della password.

- Implementazione della registrazione per i produttori, con un sistema di inviti basato su token.
- Sviluppo del sistema di gestione dei ruoli e permessi, definendo e implementando ruoli distinti per utente, produttore e operatore.
- Implementazione delle funzionalità specifiche per ciascun ruolo.

Man mano che i pezzi di codice venivano implementati, questi sono stati revisionati dai nostri tutor. La revisione del codice era un passaggio cruciale per garantire la qualità e la coerenza delle implementazioni. Utilizzando Git, abbiamo integrato i lavori svolti dal nostro gruppo e dal gruppo parallelo, unendo il lavoro di entrambi i team in modo efficace.

4.2 Gestione del team e delle risorse

La comunicazione tra i membri del team è avvenuta principalmente tramite piattaforme di messaggistica istantanea e riunioni settimanali per la pianificazione dei task e il monitoraggio dell'avanzamento del progetto.

Per garantire la corretta gestione delle risorse, è stato utilizzato un sistema di versionamento del codice tramite Git, con repository ospitate su GitHub. Ogni task implementato è stato gestito come una branch separata, per ridurre il rischio di conflitti e migliorare la collaborazione tra i membri del team. I membri di ciascun gruppo hanno lavorato in parallelo e regolarmente hanno integrato il lavoro degli altri gruppi, creando un flusso di lavoro continuo e senza interruzioni.

Per la gestione e il monitoraggio dell'andamento delle task, si è scelto di utilizzare *Jira*, uno strumento di project management che permette di visualizzare il progresso delle attività, assegnare task specifiche e tracciare le scadenze. Ogni task è stata suddivisa in sotto-attività, monitorando costantemente l'avanzamento del lavoro e le eventuali problematiche emerse durante lo sviluppo.

Inoltre, per definire e tracciare il workflow di ciascun componente, è stato usato *Notions*. Questo strumento ci ha permesso di creare documenti condivisi in cui definire i dettagli tecnici e funzionali per ogni task, nonché per ogni gruppo di lavoro. È stato utile anche per mantenere una visione chiara delle dipendenze tra le varie componenti del progetto, facilitando la comunicazione tra i membri del team.

4.3 Difficoltà incontrate e soluzioni adottate

Durante lo sviluppo, sono emerse diverse difficoltà che hanno richiesto l'adozione di soluzioni specifiche:

- Integrazione dell'autenticazione a due fattori: Una delle difficoltà riscontrate è stata integrare un sistema di autenticazione a due fattori compatibile con l'architettura esistente. Dopo aver esplorato diverse librerie, è stato scelto un framework che potesse garantire elevati livelli di sicurezza, pur mantenendo la compatibilità con il sistema.
- Gestione dei permessi: La definizione dei ruoli e dei permessi ha richiesto una revisione attenta della struttura del database. È stato necessario progettare un sistema flessibile che consentisse una gestione granulare dei permessi per ciascun tipo di utente.
- Gestione dei token di invito: La gestione dei token di invito per la registrazione dei produttori ha comportato delle sfide legate alla sicurezza e alla validità dei token. È stato implementato un sistema che garantisse la scadenza dei token dopo un periodo predefinito, migliorando la sicurezza complessiva della piattaforma.

4.4 Strumenti utilizzati

Per lo sviluppo del sistema, sono stati utilizzati i seguenti strumenti:

- Linguaggi di programmazione: *Python*, *JavaScript*
- Framework: *Flask* per il backend, *React* per il frontend
- Project Management: *Jira* per la gestione delle task e il monitoraggio dei progressi
- Documentazione e Workflow: *Notion* per la definizione del workflow e la documentazione condivisa

Il processo di sviluppo ha portato a una piattaforma robusta, sicura e facilmente scalabile, in grado di supportare le esigenze degli utenti, produttori e operatori coinvolti nel sistema.

Capitolo 5

Implementazione

L'implementazione sulla piattaforma *Filiera360* si è concentrata sul potenziamento della sicurezza e della gestione degli accessi. In particolare, sono stati sviluppati i seguenti moduli:

- Autenticazione avanzata con supporto per autenticazione a due fattori (2FA).
- Sistema di reimpostazione della password sicura.
- Gestione avanzata di ruoli e permessi.
- Funzionalità specifiche per ciascun ruolo della piattaforma.
- Registrazione sicura dei produttori tramite inviti basati su token.
- Gestione degli operatori da parte dei produttori.

5.1 Potenziamento del sistema di autenticazione

5.1.1 Autenticazione a due fattori (2FA)

Per aumentare la sicurezza degli accessi, è stata implementata l'autenticazione a due fattori (2FA). Il sistema verifica la presenza di un codice OTP generato tramite *pyotp* e inviato all'utente.

```
@app.route('/login', methods=['POST'])
def login():
    data = request.json
    email = data.get("email")
    password = data.get("password")

    users = load_users()
```

```
user = users.get(email)

if not user or not bcrypt.checkpw(password.encode('utf-8'),
    ↪ user['password'].encode('utf-8')):
    return jsonify({"message": "Invalid email or password"}), 401

if user.get('two_factor_enabled', False):
    secret = user.get('2fa_secret', None)
    if not secret:
        secret = pyotp.random_base32()
        user['2fa_secret'] = secret
        save_users(users)

    otp = pyotp.TOTP(secret).now()
    return jsonify({"message": "2FA required", "otp": otp})

token = create_access_token(email)
return jsonify({"message": "Login successful", "access_token": token,
    ↪ "role": user['role']})
```

Codice 5.1: Implementazione della 2FA nel login

5.1.2 Reimpostazione della password

Un sistema di recupero password sicuro è stato implementato utilizzando token di reset. Il sistema invia un'email con un link per il reset.

```
@app.route('/forgot-password', methods=['POST'])
def forgot_password():
    email = request.json.get('email')

    if email not in users:
        return jsonify({"message": "Email not found"}), 404

    token = generate_reset_token(email)
    reset_url = f"http://localhost:3001/reset-password/{token}"

    msg = Message('Password Reset Request', sender='noreply@example.com',
    ↪ recipients=[email])
    msg.body = f"To reset your password, visit: {reset_url}"
    mail.send(msg)

    return jsonify({"message": "Password reset email sent"}), 200
```

Codice 5.2: Implementazione della reimpostazione della password

5.2 Sviluppo del sistema di gestione ruoli e permessi

Il sistema di selezione del ruolo consente agli utenti di scegliere tra tre opzioni principali (Producer, Operator, User), influenzando la visibilità dei campi nel modulo di registrazione e le funzionalità disponibili. Questo approccio permette di registrare e categorizzare accuratamente gli utenti, con il ruolo salvato nel database tramite il campo flags. Dopo la registrazione, gli utenti accedono automaticamente alle funzionalità compatibili con il proprio ruolo, migliorando l'esperienza utente, semplificando l'interfaccia e ottimizzando l'accesso alle sezioni pertinenti. La selezione dinamica dei ruoli migliora la chiarezza e riduce gli errori nel processo di registrazione.

```
# Aggiunta dell'utente con ruoli specifici
users[email] = {
  "manufacturer": manufacturer,
  "password": hashed_password,
  "role": role,
  "flags": {
    "producer": role == "producer",
    "operator": role == "operator",
    "user": role == "user"
  },
  "operators": []
}
```

Codice 5.3: Assegnazione di ruoli e permessi agli utenti

5.3 Funzionalità specifiche per ciascun ruolo

Ogni ruolo nella piattaforma ha accesso a funzionalità specifiche, progettate per soddisfare le esigenze di ciascun tipo di utente:

- **Utente (User):**
 - Può consultare i prodotti disponibili.
 - Può accedere alla cronologia delle proprie ricerche.
 - Può salvare e consultare una lista di prodotti preferiti.
- **Produttore (Producer):**
 - Può inserire le informazioni generali sui prodotti agricoli o alimentari.
 - Può visualizzare e modificare le proprie informazioni e i propri prodotti.

- **Operatore (Operator):**

- Ogni operatore è associato a un produttore specifico.
- Ha il permesso di registrare nuovi lotti di prodotti, modificarli e gestirne gli spostamenti, sempre a nome del produttore associato.

5.4 Implementazione della registrazione per i produttori

Per garantire che solo produttori autorizzati possano registrarsi, è stato implementato un sistema di inviti basato su token. Un aspetto cruciale nella fase di registrazione riguarda l'*inviteToken*, un codice obbligatorio per gli utenti che scelgono il ruolo di *Producer*. Questo meccanismo è stato introdotto per garantire che solo le aziende autorizzate possano accedere alla piattaforma in qualità di produttori. Il funzionamento dell'*inviteToken* si basa su specifiche condizioni:

- Obbligatorietà per i produttori: se l'utente sceglie il ruolo di *Producer*, il campo *inviteToken* diventa obbligatorio per completare la registrazione.
- Unicità: ogni *inviteToken* può essere utilizzato una sola volta. Una volta che un'azienda lo impiega con successo durante la registrazione, esso viene contrassegnato come usato nel database, impedendo ulteriori utilizzi.
- Scadenza: ogni *inviteToken* può essere configurato con una data di scadenza, determinata dai proprietari della piattaforma. Se l'utente tenta di registrarsi con un token scaduto, il sistema mostra un messaggio di errore e impedisce la registrazione.

Se un utente inserisce un token non valido, già utilizzato o scaduto, viene visualizzato un messaggio di validazione accanto al campo di inserimento, evitando così confusione e migliorando l'esperienza d'uso.

```
@app.route('/signup', methods=['POST'])
def signup():
    data = request.get_json()
    email = data.get('email')
    manufacturer = data.get('manufacturer')
    password = data.get('password')
    role = data.get('role', 'user')
    invite_token = data.get('inviteToken', None)

    if not email or not manufacturer or not password:
        return jsonify({"message": "All fields are required"}), 400
```



```
if email in users:
    return jsonify({"message": "Email already exists"}), 409

if role == "producer":
    if not invite_token:
        return jsonify({"message": "The invite token is required for
        ↪ producers."}), 400

    is_valid, message = is_valid_invite_token(invite_token)
    if not is_valid:
        return jsonify({"message": message}), 403
```

Codice 5.4: Implementazione token d'invito

5.5 Gestione degli operatori da parte dei produttori

I produttori hanno la possibilità di gestire una lista di operatori, aggiungendoli o rimuovendoli secondo necessità.

5.5.1 Recupero della lista degli operatori

```
@app.route('/operators', methods=['GET'])
@jwt_required()
def get_operators():
    if not required_permissions(get_jwt_identity(), ['producer']):
        return jsonify({"operators": None})

    user = users.get(get_jwt_identity())
    operators = user.get("operators", [])

    return jsonify({"operators": operators})
```

Codice 5.5: Recupero della lista operatori

5.5.2 Aggiunta di un operatore

```
@app.route('/operators/add', methods=['POST'])
@jwt_required()
def add_operator():
    ...
```

Codice 5.6: Aggiunta di un operatore alla lista di un produttore

5.5.3 Rimozione di un operatore

```
@app.route('/operators/delete', methods=['POST'])
@jwt_required()
def remove_operator():
    ...
```

Codice 5.7: Rimozione di un operatore dalla lista di un produttore

Capitolo 6

Validazione

6.1 Introduzione

La fase di validazione ha avuto un ruolo fondamentale nel verificare il corretto funzionamento della piattaforma *Filiera360*. Sono stati effettuati diversi test per garantire che ogni funzionalità implementata rispondesse ai requisiti definiti. In particolare, sono stati analizzati i seguenti aspetti:

- La corretta assegnazione dei ruoli in fase di registrazione.
- Il funzionamento dell'autenticazione a due fattori (OTP) dopo il login.
- La differenziazione dell'interfaccia in base al ruolo assegnato.
- Il corretto utilizzo del token per la registrazione dei produttori.
- La gestione degli operatori da parte dei produttori.

6.2 Scelta del ruolo in fase di registrazione

Per garantire che ogni utente possa accedere solo alle funzionalità previste per il proprio ruolo, è stato introdotto un sistema di selezione del ruolo in fase di registrazione. L'utente può scegliere tra i ruoli disponibili: *Utente*, *Produttore* o *Operatore*.



Sign Up

Role:

User

Producer

Operator

User

Figura 6.1: Scelta del ruolo in fase di registrazione

6.3 Autenticazione a due fattori (OTP)

Durante la fase di login, il sistema richiede un codice OTP generato randomicamente, e inviato via mail. Questa misura di sicurezza impedisce accessi non autorizzati.

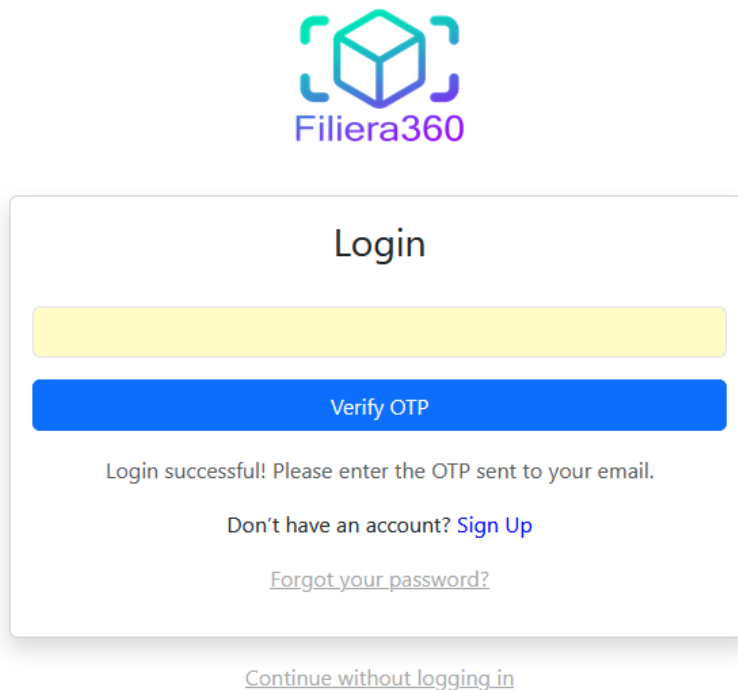


Figura 6.2: Inserimento del codice OTP dopo il login

6.4 Differenziazione dell'interfaccia in base al ruolo

Uno degli aspetti più importanti della piattaforma è la personalizzazione dell'interfaccia in base al ruolo dell'utente. Ogni tipologia di utente ha accesso a informazioni e funzionalità specifiche.

6.4.1 Interfaccia per l'utente (User)

Gli utenti standard possono consultare i prodotti disponibili, visualizzare la cronologia delle proprie ricerche e salvare i prodotti preferiti.

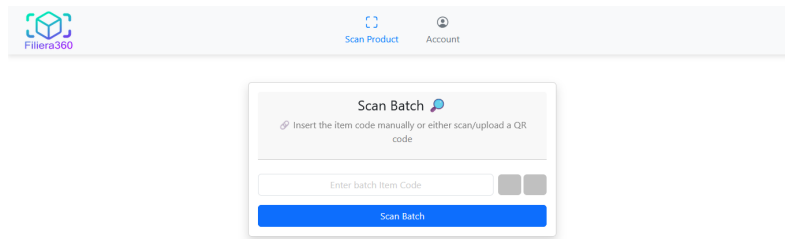


Figura 6.3: Interfaccia riservata agli utenti

6.4.2 Interfaccia per il produttore (Producer)

I produttori possono inserire e gestire i propri prodotti e i lotti, visualizzarli, e gestire la lista dei propri operatori.

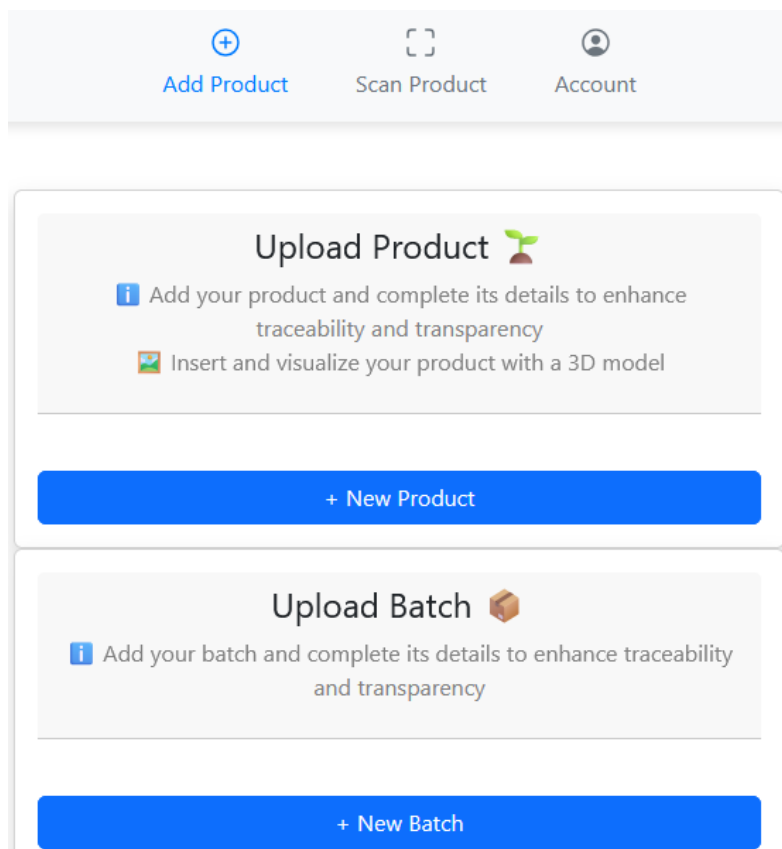


Figura 6.4: Interfaccia riservata ai produttori

6.4.3 Interfaccia per l'operatore (Operator)

Gli operatori, assegnati a un produttore specifico, possono registrare nuovi lotti di prodotti e modificarli.

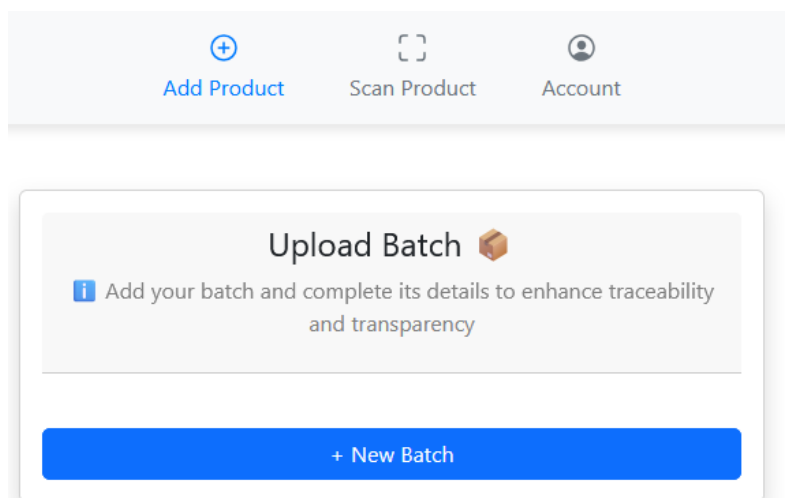



Figura 6.5: Interfaccia riservata agli operatori

6.5 Utilizzo del token per la registrazione dei produttori

Per evitare registrazioni non autorizzate come produttori, è stato introdotto un sistema basato su inviti. Un utente può registrarsi come produttore solo se dispone di un token valido.



Sign Up

Produttore1

producer123@email.com

•••••

Role:

Producer

Token di Invito


Sign Up

Already have an account? [Login](#)

Figura 6.6: Utilizzo del token in fase di registrazione del produttore

6.6 Gestione degli operatori da parte dei produttori


I produttori possono visualizzare, aggiungere e rimuovere operatori associati alla propria azienda. Questo consente un controllo preciso sugli utenti che possono modificare i prodotti e gestire i lotti.


Account 

Manufacturer: **Test0**
email: **filiera360@gmail.com**
role: **producer**

Reset password

Logout

Operators 



cri@gmail.com




Figura 6.7: Gestione degli operatori da parte dei produttori

Capitolo 7

Conclusioni e sviluppi futuri

Il progetto *Filiera360* ha introdotto un sistema innovativo per la tracciabilità dei prodotti, migliorando la sicurezza e la gestione degli utenti grazie all'implementazione di autenticazione avanzata, registrazione sicura per i produttori e un efficace sistema di gestione dei ruoli e permessi. L'integrazione con la blockchain Hyperledger Fabric ha garantito la trasparenza e l'immutabilità dei dati, mentre l'interfaccia web ha reso l'esperienza utente intuitiva e accessibile.

Tuttavia, ci sono ancora diversi aspetti che possono essere migliorati e ampliati per rendere la piattaforma ancora più efficiente e sicura. Tra gli sviluppi futuri da considerare:

- Memorizzazione sicura delle credenziali: Attualmente, le credenziali degli utenti vengono gestite in maniera funzionale, ma un miglioramento chiave sarebbe l'integrazione di un database sicuro per la loro gestione, evitando l'archiviazione in file JSON e adottando soluzioni come la crittografia delle password e la protezione degli accessi con metodi avanzati.
- Hashing di OTP e token di invito: Per aumentare la sicurezza, sarebbe opportuno applicare algoritmi di hashing agli OTP (One-Time Password) e ai token di invito prima di salvarli nei file JSON o nel database. Questo ridurrebbe il rischio di compromissione dei dati in caso di accesso non autorizzato.
- Design responsive: Sebbene la piattaforma sia pienamente funzionale su desktop, l'introduzione di un design completamente responsive migliorerebbe l'esperienza utente su diversi dispositivi. Ciò garantirebbe una migliore fruibilità su smartphone e tablet, ampliando il bacino di utenti che possono accedere al servizio senza limitazioni.
- Test approfonditi su dispositivi mobili: Un'ottimizzazione mirata per dispositivi mobili richiederebbe test approfonditi su diversi sistemi operativi e modelli di smartphone. L'obiettivo è garantire un'interfaccia fluida, tempi di risposta rapidi e una navigazione intuitiva anche su schermi più piccoli.

- Gestione della revoca degli account operatore: In caso di cessazione del rapporto di lavoro, dovrebbe essere implementata una funzionalità per la revoca dell'account operatore, garantendo un controllo efficace sugli accessi alla piattaforma.

L'implementazione di questi miglioramenti non solo aumenterebbe la sicurezza e l'usabilità della piattaforma, ma renderebbe *Filiera360* un sistema ancora più affidabile e scalabile, capace di adattarsi alle future esigenze del settore.