

UNIVERSITÀ DEGLI STUDI DI NAPOLI “FEDERICO II”



UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

PROGETTO DI INGEGNERIA DEL SOFTWARE A.A. 2023/2024

DIETIDEALS24
BIDHUB

CANDIDATI

GIORDANO VINCENZO N86003039

LEONE GIANLUCA N86003082

MIGLIORISI ANDREA N86002979

ingsw_dietideals24_app

Questa pagina è stata lasciata intenzionalmente vuota.

SOMMARIO

Introduzione	4
Documento dei Requisiti Software	5
Analisi dei Requisiti	5
Modellazione dei casi d'uso	6
Individuazione del target degli utenti	10
Descrizioni testuali di quattro casi d'uso	14
Mock-up dell'interfaccia utente	22
Valutazione dell'usabilità a priori	63
Glossario	66
Specifica dei Requisiti	67
Classi, oggetti e relazioni di analisi	67
Diagrammi di sequenza di analisi	77
Prototipazione funzionale e progettazione degli event-based statechart	80
Documento di Design del sistema.....	84
Descrizione dell'architettura	84
Motivazione delle scelte tecnologiche adottate.....	86
Diagramma delle classi di design	87
Diagrammi di sequenza di design	91
Strumenti di Versioning	95
Testing e valutazione sul campo dell'usabilità	99
Unit Testing.....	99
Valutazione dell'usabilità sul campo	114
Obiettivo della valutazione	114
Metodologia	114
Risultati della Valutazione	115
Conclusioni della Valutazione	115
Analisi dei File di Log	116
Obiettivo dell'Analisi	116

ingsw_dietideals24_app

Configurazione e Raccolta dei Log	116
Estratto del File di Log	116
Analisi dei Dati di Log.....	117
Conclusioni dell'Analisi	117
Allegati	117
Conclusioni Generali	117

INTRODUZIONE

BidHub è una piattaforma di aste online innovativa, progettata per rivoluzionare l'esperienza di chi ama collezionare oggetti rari o scovare affari unici. Con un'attenzione particolare all'accessibilità e alla semplicità d'uso, BidHub mira a diventare il compagno ideale per ogni appassionato di aste, i "BidHunter", offrendo loro un accesso facile e intuitivo a un mondo di opportunità attraverso il proprio dispositivo mobile.

L'interfaccia di BidHub combina estetica e funzionalità, garantendo un'esperienza utente piacevole e immediata sia per i neofiti sia per i veterani delle aste. La registrazione è rapida e permette di accedere con pochi clic, anche tramite Google, per iniziare subito a partecipare alle aste.

La piattaforma si distingue per le sue funzioni personalizzabili, che consentono di filtrare le aste in base a parametri come nome, accessibilità, durata, autore e ubicazione geografica. Questo approccio dettagliato permette a ogni utente di trovare esattamente ciò che cerca in modo semplice ed efficiente.

Ogni asta è accompagnata da descrizioni complete, che offrono informazioni dettagliate e una narrazione coinvolgente per aiutare gli utenti a fare scelte informate. Inoltre, BidHub promuove la comunicazione diretta tra gli utenti, creando un ambiente che favorisce lo scambio di informazioni e la costruzione di relazioni basate su interessi condivisi.

La visione di BidHub è unire la tradizione delle aste fisiche alla comodità delle piattaforme digitali, creando uno spazio ricco di storie e possibilità. Il nostro impegno per l'innovazione continua si riflette nel costante sviluppo di nuove funzionalità, come itinerari d'asta, opzioni di condivisione sui social e recensioni dettagliate.

Il modello di sviluppo di BidHub combina la solidità dell'approccio a cascata nelle fasi iniziali con la flessibilità dei processi iterativi, garantendo una piattaforma robusta e reattiva. Seguendo i principi SOLID, ci assicuriamo che ogni componente del software sia progettato per essere flessibile, mantenibile e in grado di evolversi senza compromettere le fondamenta.

BidHub non è solo una piattaforma di aste: è una comunità. Invitiamo i nostri utenti a partecipare attivamente allo sviluppo della piattaforma, contribuendo con feedback e idee. Il nostro obiettivo è creare un ambiente che non sia solo uno strumento di compravendita, ma un luogo di condivisione e connessione.

Con uno sguardo rivolto al futuro, stiamo già sviluppando funzionalità basate sull'intelligenza artificiale per raccomandazioni personalizzate e strumenti di realtà aumentata per un'esperienza immersiva. BidHub è un viaggio che cresce insieme ai suoi utenti, unendo innovazione e pratiche etiche per costruire fiducia e trasparenza.

Unisciti oggi a BidHub e scopri il nuovo modo di vivere le aste. Che tu sia un cacciatore di tesori o un venditore di meraviglie, questa è la piattaforma che fa per te. Scopri BidHub: l'arte dell'asta a portata di clic!

DOCUMENTO DEI REQUISITI SOFTWARE

Il presente documento dei requisiti software descrive in modo dettagliato le funzionalità, le caratteristiche e i vincoli del sistema in fase di sviluppo. Esso rappresenta un riferimento fondamentale per garantire una comprensione condivisa tra i vari stakeholder e per guidare il team di sviluppo durante tutte le fasi del progetto.

Scopo del documento

L'obiettivo principale di questo documento è definire i requisiti funzionali e non funzionali del sistema, stabilendo una base contrattuale tra il cliente e il team di sviluppo. Inoltre, fornisce indicazioni per la verifica e la validazione del software, assicurando che il prodotto finale soddisfi le esigenze concordate.

Ambito del progetto

Il sistema in esame è progettato per la gestione di un'applicazione dedicata alle aste online. L'obiettivo è offrire agli utenti un'esperienza intuitiva e funzionale, supportandoli sia nella ricerca di prodotti di interesse da acquistare, sia nella pubblicazione di annunci per la vendita di oggetti.

Pubblico destinatario

Questo documento è rivolto a:

- **Clienti e stakeholder:** per verificare che i requisiti rappresentino fedelmente le loro necessità.
- **Team di sviluppo:** per comprendere le funzionalità da implementare e le priorità.
- **Tester:** per pianificare ed eseguire le attività di verifica e validazione.
- **Project manager:** per monitorare il progresso e garantire il rispetto degli obiettivi.

ANALISI DEI REQUISITI

L'analisi dei requisiti rappresenta una fase cruciale del processo di sviluppo software, in quanto consente di identificare e comprendere le necessità degli utenti e degli stakeholder, trasformandole in specifiche funzionali e non funzionali. In questa sezione del documento, vengono descritti i requisiti che il sistema dovrà soddisfare per garantire il successo del progetto, rispondendo alle esigenze e alle aspettative degli utenti finali.

L'analisi dei requisiti include la modellazione dei casi d'uso, che descrive i principali scenari di interazione tra gli utenti e il sistema, l'individuazione del target degli utenti, e una descrizione dettagliata dei casi d'uso. Inoltre, verranno presentati mock-up dell'interfaccia utente, per fornire una visione concreta dell'esperienza d'uso prevista, e sarà valutata l'usabilità a priori per garantire che il sistema sia facile e intuitivo da utilizzare.

Questa fase del progetto mira a stabilire una comprensione chiara e condivisa tra tutti i membri del team e gli stakeholder, riducendo il rischio di malintesi e garantendo che le funzionalità richieste siano effettivamente realizzabili.

MODELLAZIONE DEI CASI D'USO

L'**analisi dei casi d'uso** rappresenta uno strumento cruciale per modellare le funzionalità del sistema di gestione aste online e per comprendere le interazioni tra gli utenti e il sistema stesso. L'applicazione consente a ciascun utente registrato di assumere il ruolo di **venditore**, per mettere in vendita i propri articoli, o di **acquirente**, per partecipare alle aste e acquistare prodotti.

L'analisi si concentra sui principali casi d'uso, suddivisi in base ai diversi ruoli degli utenti e alle funzionalità offerte dal sistema. Ogni caso d'uso è stato progettato per garantire un'esperienza utente intuitiva ed efficace, evidenziando i flussi operativi chiave, come la **creazione di aste**, la **gestione delle offerte** e l'**interazione post-asta**.

I diagrammi dei casi d'uso, organizzati per ruolo e funzionalità, permettono di rappresentare in modo chiaro e completo le operazioni disponibili e le interazioni critiche tra utenti e sistema.

Per ciascun caso d'uso saranno descritti i seguenti elementi:

- **Attori:** Gli utenti che interagiscono con il sistema (es., venditore, acquirente, o utenti esterni per l'autenticazione).
- **Requisiti funzionali e non funzionali:** Le caratteristiche richieste per il corretto funzionamento e la scalabilità del sistema.
- **Precondizioni:** Gli stati iniziali richiesti per l'avvio del caso d'uso (es., un utente deve essere autenticato per partecipare a un'asta).
- **Post-condizioni:** Gli stati finali che risultano dal completamento del caso d'uso (es., un'offerta viene registrata correttamente o un'asta è completata con successo).

Tra i casi d'uso principali analizzati, suddivisi per ruolo e funzionalità, troviamo:

1. **Registrazione e Accesso al sistema:** Gli utenti possono registrarsi tramite credenziali personali o account di terze parti e accedere al sistema.
2. **Gestione del profilo utente:** Consente agli utenti di aggiornare le proprie informazioni personali, garantendo la personalizzazione e il controllo dei dati.
3. **Creazione di un'asta:** Il venditore configura un'asta specificando i dettagli del prodotto (descrizione, prezzo iniziale, durata) e sceglie il tipo di asta da utilizzare.
4. **Partecipazione a un'asta:** L'acquirente esplora le aste disponibili, filtra i risultati, visualizza i dettagli delle aste e presenta offerte in base alle regole del tipo di asta selezionato.
5. **Gestione di transazioni:** Si occupa delle interazioni tra acquirente e venditore post-asta, come il pagamento e la consegna del prodotto.

Questa analisi sistematica consente di strutturare in modo rigoroso lo sviluppo del sistema e offre una guida chiara per il suo utilizzo. La descrizione dettagliata facilita la comunicazione tra gli stakeholder e garantisce che le funzionalità rispondano pienamente alle esigenze degli utenti, assicurando un'esperienza coerente e fluida per tutte le tipologie di interazioni.

Questo diagramma si focalizza sui processi di registrazione dell'utente non registrato al sistema di gestione aste. I principali attori coinvolti sono:

- **Utente Non Registrato:** Può registrarsi al sistema attraverso due modalità principali:

1. **Registrazione diretta:** L'utente inserisce i propri dati manualmente (come email, password, ecc.) per creare un account.
2. **Registrazione tramite account terzi:** L'utente utilizza un account esistente su piattaforme esterne, come Facebook, Google, GitHub o LinkedIn. In questo caso, i dati come nome, email e foto profilo potrebbero essere sincronizzati automaticamente. Questa funzionalità è modellata come un'azione inclusa («include») all'interno del caso d'uso principale.

Precondizione: l'utente non deve già possedere un account registrato.

Nota: l'integrazione con account terzi garantisce un accesso rapido e semplificato.

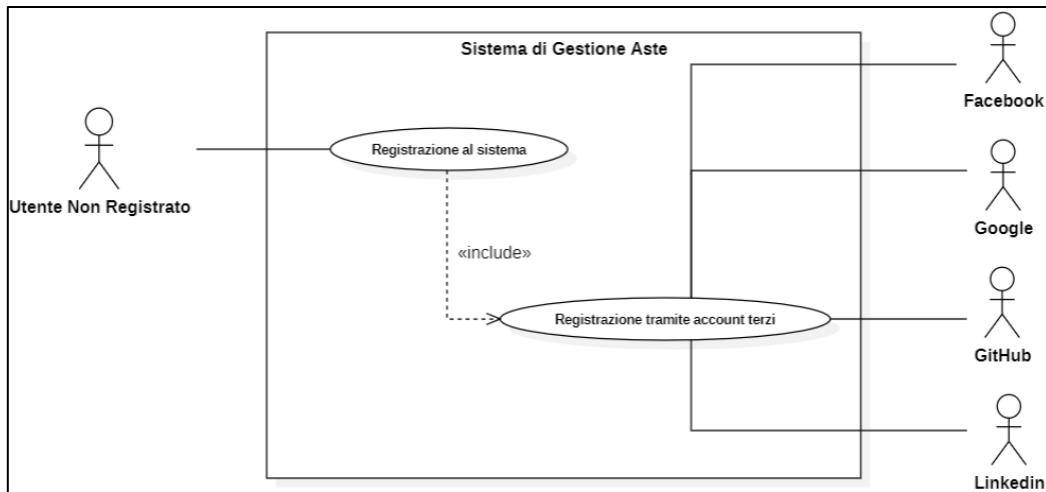


Figure 1: Use Case Diagram - Registrazione e Login

Questo diagramma descrive le operazioni disponibili per due categorie di utenti del sistema:

- **Utente Registrato:**

1. **Accesso al sistema:** Permette l'autenticazione al sistema utilizzando credenziali create in fase di registrazione o account terzi.
2. **Gestione del profilo:** Consente all'utente di aggiornare informazioni personali come email, password o dettagli aggiuntivi.

- **Utente Venditore** (specializzazione dell'utente registrato): Ha accesso a funzionalità specifiche legate alla vendita:

1. **Creazione Asta:** Permette di configurare un'asta scegliendo tra diversi tipi:
 - **Asta all'inglese** (relazione «extend»): Asta con offerte crescenti.
 - **Asta al ribasso** (relazione «extend»): Asta con offerte decrescenti.
 - **Asta silenziosa** (relazione «extend»): Le offerte non sono visibili agli altri partecipanti.
 - **Asta a tempo fisso** (relazione «extend»): L'asta ha una scadenza temporale fissa.
2. **Gestione offerte ricevute:** Permette di visualizzare e rispondere alle offerte fatte dagli utenti sulle aste attive.

3. **Ricerca aste attive:** Include funzionalità per filtrare le aste per categoria («include») e visualizzare i dettagli di una specifica asta («include»).
4. **Visualizza profilo venditore:** Consente di mostrare pubblicamente informazioni utili sul venditore.

Precondizione: il venditore deve aver effettuato l'accesso al sistema.

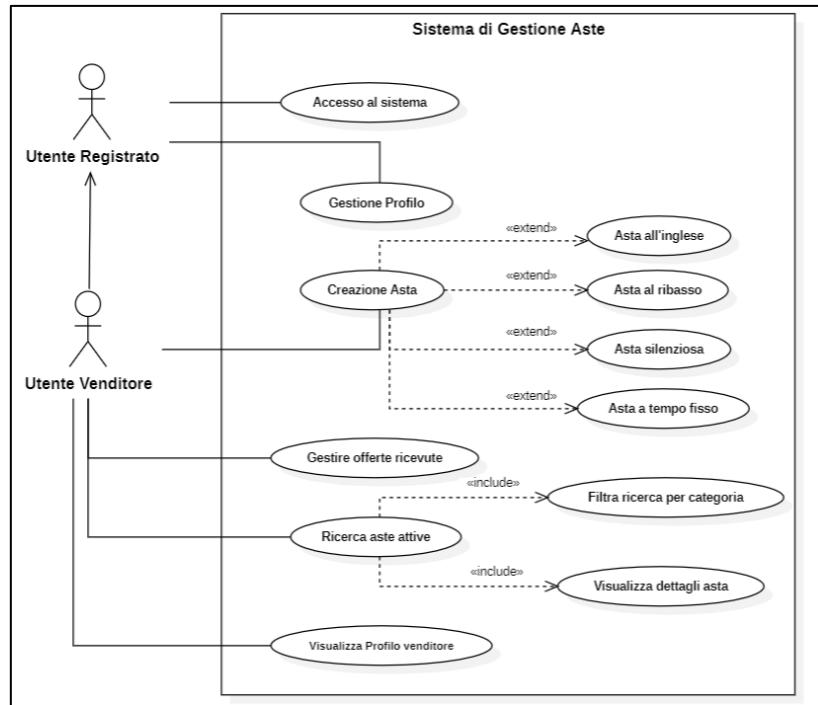


Figure 2: Use Case Diagram - Profilo venditore per creazione asta e gestione offerte

Questo diagramma illustra le operazioni disponibili per gli utenti registrati che svolgono il ruolo di acquirenti. Gli attori principali sono:

- **Utente Registrato:**
 - **Accesso al sistema:** Permette l'autenticazione per accedere alle funzionalità di base.
 - **Gestione del profilo:** Analogamente al venditore, consente di aggiornare i dati personali.
- **Utente - Acquirente** (specializzazione dell'utente registrato): Ha accesso a funzionalità specifiche per l'acquisto:
 1. **Ricerca aste attive:** Consente di esplorare le aste disponibili con due funzionalità incluse:
 - **Filtrare ricerca per categoria** («include»): Permette di restringere la ricerca ad aste di specifico interesse.
 - **Visualizza dettagli asta** («include»): Fornisce informazioni dettagliate su una specifica asta, come regole e partecipanti.
 2. **Presenta offerta in base al tipo di asta:** Permette all'acquirente di partecipare attivamente a un'asta seguendo le regole previste per il tipo di asta selezionato.
 3. **Visualizza profilo venditore:** Fornisce dettagli sull'affidabilità del venditore e informazioni utili per la valutazione dell'acquisto.

Precondizione: l'utente deve essere autenticato nel sistema.

Nota: il processo di offerta è differenziato in base al tipo di asta, rispettando le regole specifiche di ciascun formato (ad esempio, rilanci crescenti o offerte invisibili).

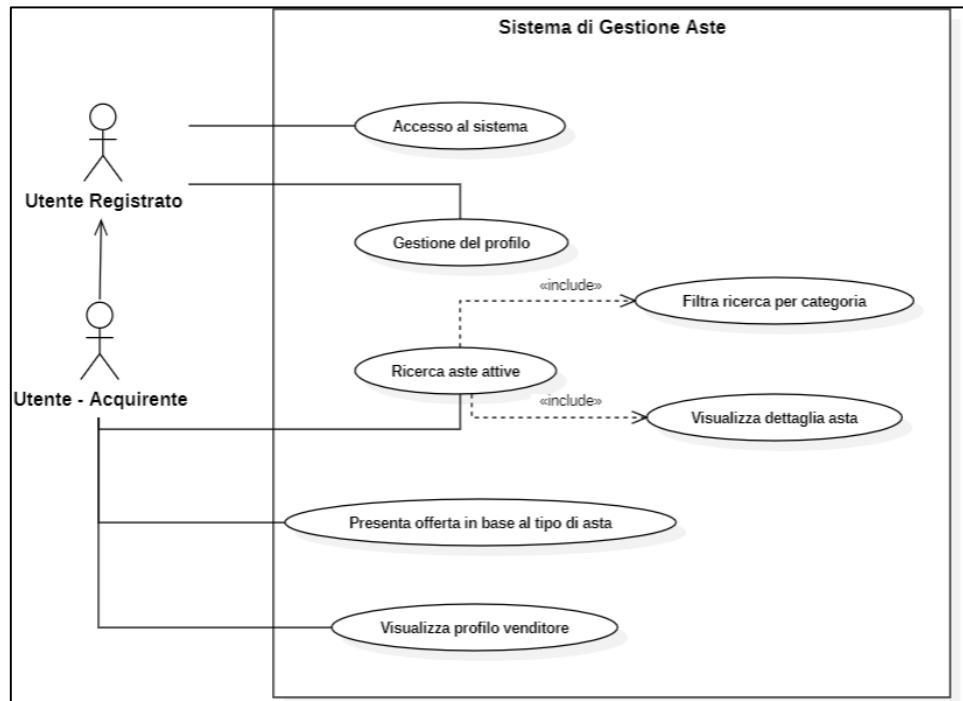


Figure 3: Use Case Diagram - Profilo acquirente per invio offerte e ricerca aste

INDIVIDUAZIONE DEL TARGET DEGLI UTENTI

Uno degli aspetti fondamentali nello sviluppo di un progetto software è l'identificazione del **target degli utenti finali**, poiché da essa dipendono sia le scelte progettuali che le priorità funzionali del sistema. Questo processo, condotto attraverso un'approfondita ricerca di mercato e tecniche come interviste con stakeholder, sondaggi e analisi dei competitor, consente di creare un prodotto che risponda efficacemente ai bisogni reali degli utenti.

L'Importanza della Segmentazione del Mercato

Segmentare il mercato significa suddividere il pubblico potenziale in gruppi omogenei, o **target group**, che condividono caratteristiche chiave come necessità, comportamenti e valori. Questa pratica permette di affinare il design del prodotto e di ottimizzare le strategie di marketing. Ogni segmento rappresenta una nicchia specifica su cui concentrare gli sforzi di sviluppo, riducendo il rischio di creare un prodotto generico e poco efficace.

Per individuare il target group, il team si è posto domande fondamentali:

- Chi accede al nostro prodotto?
- Quali bisogni degli utenti vogliamo soddisfare?
- Chi è interessato al suo utilizzo o acquisto?
- Come possiamo differenziarci dai competitor?

Questa analisi ha permesso di costruire un quadro chiaro del pubblico di riferimento, evidenziando caratteristiche comuni:

- **Ricerca di qualità e autenticità:** Gli utenti cercano prodotti autentici e di alta qualità.
- **Passione ed emozione nell'acquisto:** Gli acquisti sono guidati da motivazioni emotive, come il piacere di trovare un affare o l'eccitazione di ottenere un oggetto significativo.
- **Efficienza e comodità:** È forte la preferenza per soluzioni che risparmiano tempo e offrono semplicità, come processi di acquisto intuitivi.

Creazione di Personas: Strumento di Progettazione

Per rappresentare i target group, sono state create delle **Personas**, ovvero profili fittizi basati su dati reali. Le Personas non sono semplici descrizioni, ma strumenti fondamentali per guidare le decisioni progettuali, garantendo che ogni elemento del sistema sia sviluppato per soddisfare le esigenze di specifici segmenti di utenti.

Benefici dell'uso delle Personas:

- **Identificazione delle esigenze specifiche:** Ogni Persona rappresenta un segmento unico, evidenziandone obiettivi e sfide.
- **Simulazione di scenari realistici:** Durante i test di usabilità, le Personas permettono di valutare come diverse categorie di utenti interagiscono con il sistema.
- **Guida alle priorità progettuali:** Le caratteristiche delle Personas influenzano le funzionalità del sistema e il design dell'interfaccia.

L'uso delle Personas ha garantito un focus user-centric durante lo sviluppo, migliorando la qualità del prodotto e riducendo la curva di apprendimento.

Segmentazione degli Utenti

I risultati dell'analisi hanno identificato tre principali categorie di utenti:

1. Utenti esperti

- **Descrizione:** Professionisti con competenze avanzate e familiarità con strumenti simili.
- **Esigenze:** Funzionalità avanzate, personalizzazione e supporto a flussi complessi.
- **Obiettivo:** Ottimizzare l'efficienza operativa e mantenere il controllo.

2. Utenti intermedi

- **Descrizione:** Utenti con conoscenze di base o intermedie che usano il sistema in modo non quotidiano.
- **Esigenze:** Interfaccia intuitiva, accesso facilitato a funzionalità avanzate e documentazione chiara.
- **Obiettivo:** Semplificare le attività e risparmiare tempo.

3. Utenti occasionali o non specializzati

- **Descrizione:** Individui con accessi sporadici o per scopi secondari, spesso con competenze limitate.
- **Esigenze:** Interfacce semplici, flussi automatizzati e supporto integrato.
- **Obiettivo:** Completare attività rapidamente e senza difficoltà tecniche.

Approccio User-Centered Design (UCD)

Il processo di progettazione è stato guidato dai principi del **User-Centered Design (UCD)**, che pone l'utente al centro. Attraverso iterazioni continue di raccolta feedback, test di prototipi e validazione, il team ha progettato un sistema che bilancia funzionalità avanzate e usabilità.

Impatto sul Progetto

L'identificazione del target e l'utilizzo delle Personas hanno avuto un impatto significativo su:

- **Requisiti funzionali:** Adattati per soddisfare le esigenze delle diverse categorie di utenti.
- **Design dell'interfaccia:** Layout modulare e flessibile, progettato per utenti esperti e occasionali.
- **Strategie di implementazione:** Rilasci incrementali per raccogliere dati reali sull'utilizzo e ottimizzare il sistema.

Grazie a questo approccio, il prodotto finale è in grado di soddisfare un pubblico ampio, migliorando l'adozione iniziale, la soddisfazione e la fidelizzazione nel lungo termine.



Figura 4 : Personas - Julio Pen

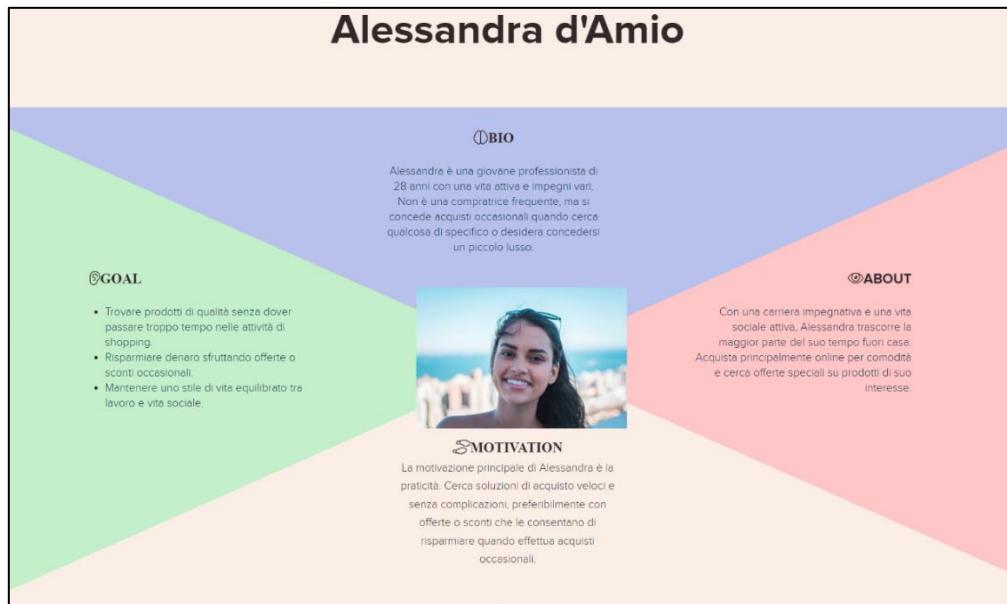


Figura 5: Personas - Alessandra d'Amio

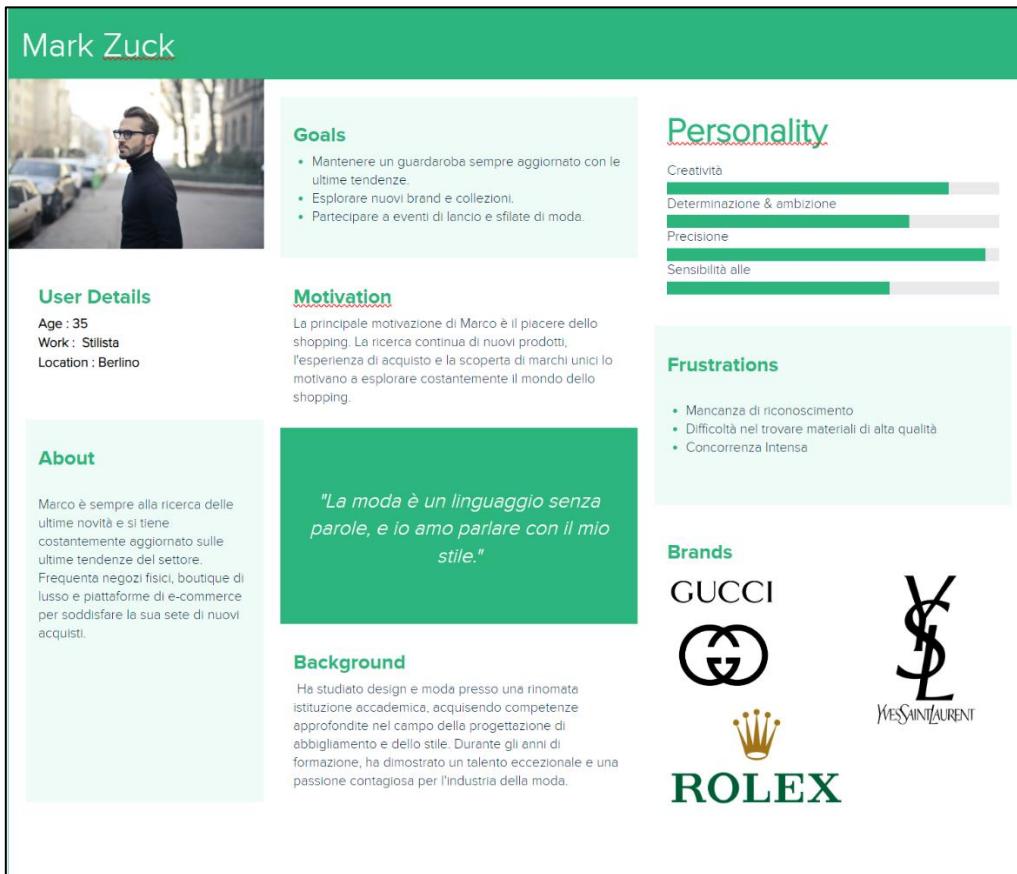


Figura 6: Personas - Mark Zuck

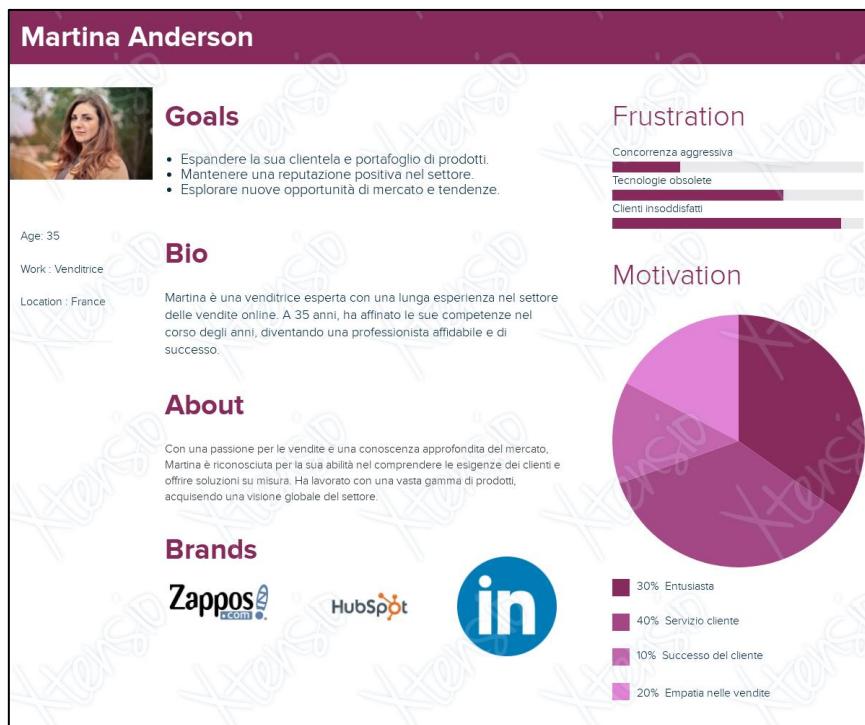


Figura 7: Personas - Martina Anderson

DESCRIZIONI TESTUALI DI QUATTRO CASI D'USO

Nell'ambito dello sviluppo software, i **diagrammi di Cockburn** (o diagrammi dei casi d'uso) rappresentano un elemento fondamentale per la modellazione dei requisiti funzionali di un sistema. Proposti da Alistair Cockburn, questi diagrammi si concentrano sulla definizione delle interazioni tra gli attori e il sistema, descrivendo in modo semplice e intuitivo le funzionalità principali che il sistema deve offrire.

Il loro obiettivo è fornire una visione chiara e condivisibile tra i membri del team di sviluppo, i committenti e gli stakeholder, facilitando la comprensione e la comunicazione dei requisiti. Attraverso questi diagrammi, è possibile:

- Identificare gli **attori** coinvolti (utenti, sistemi esterni, dispositivi).
- Delineare i **casi d'uso**, ovvero le funzionalità principali del sistema.
- Evidenziare le relazioni tra gli attori e i casi d'uso, mostrando le responsabilità del sistema rispetto alle esigenze degli utenti.

Questa sezione presenta i diagrammi di Cockburn relativi al progetto, fornendo una panoramica sulle principali interazioni e funzionalità previste.

1. CREAZIONE DI UN NUOVO TIPO DI ASTA

Il seguente diagramma di Cockburn rappresenta il caso d'uso dedicato alla creazione di un nuovo tipo di asta, una funzionalità fondamentale per i venditori registrati nel sistema. Questa operazione consente all'utente di configurare aste personalizzate, specificando dettagli rilevanti come nome, regole e parametri unici.

Obiettivo del Caso d'Uso

L'obiettivo principale è consentire all'utente di definire un nuovo tipo di asta in modo semplice e rapido.

Requisiti Preliminari

- L'utente deve essere autenticato come venditore nel sistema.
- La piattaforma deve essere operativa e predisposta per accettare nuovi tipi di aste.
- Devono essere disponibili categorie predefinite che l'utente può associare al nuovo tipo di asta.

Scenario Principale

1. L'utente accede alla sezione "Nuovo Tipo di Asta" tramite il pannello di controllo.
2. Il sistema mostra un modulo dedicato alla configurazione di un nuovo tipo di asta, comprendente campi per nome, descrizione, categoria e regole personalizzate.
3. L'utente compila i campi richiesti e invia i dati tramite il pulsante "Conferma".
4. Il sistema valida i dati forniti:
 - Verifica la completezza dei campi obbligatori.
 - Controlla che non esistano duplicati nel database.
5. In caso di successo, il sistema registra il nuovo tipo di asta nel database e notifica l'utente con un messaggio di conferma.

Variazioni e Estensioni

- **Eccezioni:**
 - **Dati non validi:** Se l'utente inserisce dati errati o incompleti, il sistema mostrerà un messaggio di errore con dettagli sui campi da correggere e tornerà al modulo per la modifica.
 - **Duplicati:** Se il nome del tipo di asta esiste già, il sistema avviserà l'utente e richiederà un nome unico.

Questa rappresentazione descrive in dettaglio il flusso di azioni e interazioni tra utente e sistema, garantendo un punto di riferimento chiaro per la progettazione e lo sviluppo della funzionalità.

USE CASE #1 NOME UC – CREAZIONE DI UN NUOVO TIPO DI ASTA

<i>Goal in Context</i>	L'utente desidera creare un'asta online in modo rapido e semplice, fornendo i dettagli necessari. Il sistema deve assicurarsi che i dati forniti siano validi e che l'asta sia correttamente registrata, notificando l'utente con una conferma chiara una volta completato il processo		
	<ol style="list-style-type: none"> 1. L'utente deve essere autenticato nel sistema 2. Devono esistere categorie predefinite nel database per la classificazione delle aste 		
<i>Success End Condition</i>	<p>L'asta viene creata con successo nel sistema:</p> <ul style="list-style-type: none"> • Il prodotto associato all'asta è registrato nel database • I dettagli dell'asta sono memorizzati correttamente • L'utente riceve una conferma visibile tramite l'interfaccia dell'applicazione 		
<i>DESCRIPTION</i>	Step n°	Attore 1 - Utente registrato al Sistema	Sistema
	1	Selezione il tipo di asta da creare da mockup "Create Auction"	
	2		Mostra Pagina di asta selezionata
	3		
	4	Inserisce dati dettagli asta relativi alla categoria d'asta selezionata	
	5	Clicca tasto "Conferma"	
	6		Mostra Messaggio di creazione d'asta con successo
	7		Invia notifica di creazione d'asta all'utente
	8		Il sistema registra correttamente la nuova asta creata dall'utente
<i>EXTENSIONS</i>	Step	Attore 1 - Utente registrato al Sistema	Sistema
	5.a	Inserisce dati non validi per l'asta di riferimento	
	6.a		Mostra messaggio di errore
	5.b	Clicca tasto "back arrow"	
	6.b		Mostra messaggio di errore e riprende da scenario 1
	5.c	Non inserisce tutti i dati obbligatori relativi al tipo di asta selezionata	
<i>Campi obbligatori mancati</i>	6.c		Mostra messaggio di errore e riprende da scenario 1

2. CANCELLAZIONE DI UN'ASTA ATTIVA

Il seguente diagramma di Cockburn descrive il caso d'uso relativo all'eliminazione di un'asta attiva da parte dell'utente. Questa funzionalità è cruciale per consentire agli utenti di gestire le proprie aste in modo efficace, mantenendo il controllo su quelle attualmente in corso.

Obiettivo del Caso d'Uso

Permettere all'utente autenticato di eliminare un'asta attiva creata in precedenza. L'eliminazione deve essere gestita in modo sicuro, con il sistema che verifica l'esistenza dell'asta e notifica l'utente dell'esito dell'operazione.

Requisiti Preliminari

- L'utente deve essere registrato nel sistema.
- Deve esistere almeno un'asta attiva associata all'utente.

Scenario Principale

L'utente avvia il processo di eliminazione navigando attraverso la propria area personale. Di seguito i passaggi principali:

1. L'utente accede alla pagina "Account".
2. Il sistema mostra la pagina con i dettagli del profilo utente.
3. Nella sezione "Vendi", l'utente seleziona l'opzione "All'asta".
4. Il sistema visualizza un elenco di aste attive create dall'utente (almeno una).
5. L'utente clicca sull'icona "trash" per eliminare un'asta specifica.
6. Il sistema mostra un messaggio di conferma dell'eliminazione avvenuta con successo.
7. Una notifica viene inviata all'utente per confermare l'operazione.
8. Il sistema registra correttamente la cancellazione dell'asta selezionata.

Variazioni e Estensioni

- **Eccezioni:**

- **Asta non trovata:**
Se l'asta selezionata è già stata eliminata o non esiste più, il sistema avvisa l'utente con un messaggio di errore e termina il processo.
- **Annullamento del processo:**
Se l'utente decide di annullare l'operazione cliccando sull'icona "back arrow", il sistema interrompe l'eliminazione e riprende dallo scenario principale.

USE CASE #2 NOME UC – CANCELLAZIONE DI UN'ASTA ATTIVA

<i>Goal in Context</i>	Permettere all'utente di eliminare un'asta attiva creata in precedenza. Il sistema deve gestire correttamente l'eliminazione, verificando che l'asta esista e notificando l'utente del successo dell'operazione	
	<ol style="list-style-type: none"> 1. L'utente deve essere autenticato nel sistema 2. Deve esistere almeno un'asta attiva associata all'utente 	
<i>Success End Condition</i>	<ul style="list-style-type: none"> • L'asta selezionata viene eliminata correttamente dal sistema • L'utente riceve una conferma visibile sull'interfaccia 	
<i>DESCRIPTION</i>	<p>Step n° Attore 1 – Utente registrato al Sistema Sistema</p> <p>1 Selezione pagina “Account”</p> <p>2</p> <p>3 Nella sezione “Vendi”, seleziona opzione “All’asta”</p> <p>4</p> <p>5 Clicca tasto “trash”</p> <p>6</p> <p>7</p> <p>8</p>	
		Mostra Pagina dettaglio account utente
		Mostra pagina con lista di aste attive (almeno una)
		Mostra Messaggio di eliminazione d'asta con successo
		Invia notifica di eliminazione d'asta all'utente
		Il sistema registra correttamente cancellazione dell'asta selezionata
<i>EXTENSIONS</i>	Step	Attore 1 - Utente registrato al Sistema Sistema
<i>Asta non Trovata</i>	5.a	L'asta è stata già eliminata o non esiste più
	6.a	Mostra messaggio di errore
<i>Annulloamento del processo</i>	5.b	Clicca tasto “back arrow”
	6.b	Mostra messaggio di errore e riprende da scenario 1

3. INVIO OFFERTA

Il seguente diagramma di Cockburn illustra il caso d'uso per l'invio di un'offerta su un'asta attiva, una funzionalità destinata agli acquirenti del sistema. Questo processo consente agli utenti autenticati di partecipare a un'asta, garantendo che l'offerte siano registrate correttamente e che il venditore venga notificato.

Obiettivo del Caso d'Uso

L'obiettivo è permettere agli utenti autenticati di inviare un'offerta valida per un'asta attiva, assicurando che il sistema registri correttamente l'offerta e notifichi il venditore dell'asta.

Requisiti Preliminari

- L'utente deve essere autenticato nel sistema.
- Deve esistere almeno un'asta attiva nel sistema.

Scenario Principale

1. L'utente accede alla pagina delle aste attive e seleziona un'asta su cui fare un'offerta.
2. Il sistema mostra la pagina di dettaglio dell'asta selezionata, con informazioni sul prodotto e sull'attuale stato delle offerte.
3. L'utente inserisce l'importo dell'offerta nel campo dedicato.
4. L'utente preme il pulsante "Invia Offerta" per confermare.
5. Il sistema valida i dati (es. verifica che l'importo sia valido rispetto alle regole dell'asta).
6. Se l'offerta è valida, il sistema registra l'offerta nel database, mostra un messaggio di successo e notifica il venditore dell'asta.
7. L'utente visualizza un messaggio di conferma e rimane nella pagina di dettaglio dell'asta.

Variazioni e Estensioni

Eccezioni

- **Errore nella validazione dell'offerta:**
Se l'importo inserito dall'utente è inferiore al valore minimo richiesto o all'ultima offerta valida, il sistema mostra un messaggio di errore e consente all'utente di correggere i dati, riprendendo dal passo 3.
- **Annullamento del processo:**
Se l'utente decide di annullare l'offerta cliccando sul pulsante "Back" o navigando fuori dalla pagina, il sistema annulla il processo e riporta l'utente alla pagina delle aste attive senza registrare alcun dato.

USE CASE #3 NOME UC – INVIO OFFERTA

<i>Goal in Context</i>	L'utente vuole inviare una offerta per un prodotto all'asta	
	<ul style="list-style-type: none"> • L'utente è autenticato nel sistema • Esiste almeno un'asta attiva 	
	L'offerta viene registrata correttamente nel sistema, e il venditore (proprietario dell'asta) riceve una notifica dell'offerta	
<i>DESCRIPTION</i>	Step n°	Attore 1 – UTENTE VENDITORE
	1	L'utente seleziona l'asta a cui fare offerta dalla pagina delle aste attive
	2	
	3	L'utente inserisce l'importo dell'offerta
	4	L'utente clicca su "invia offerta"
	5	
	6	Invia una notifica al venditore dell'asta per avvisarlo dell'offerta
<i>EXTENSIONS</i>	Step	Attore 1 - VENDITORE
<i>Errore nella validazione dell'offerta</i>	4.a	Inserisce un importo minore rispetto all'ultima offerta disponibile
	5.a	
<i>L'utente annulla l'offerta</i>	4.b	L'utente ritorna alla pagina precedente e non emette alcuna offerta
	5.b	
		Riporta l'utente alla pagina delle aste attive

4. Accettazione di un'offerta per un'asta di tipo silenziosa**Obiettivo del Caso d'Uso**

L'obiettivo è consentire a un acquirente di presentare un'offerta per un'asta attiva e permettere al venditore di accettare l'offerta migliore, concludendo l'asta e creando un ordine per il vincitore.

Requisiti Preliminari

- Per l'acquirente:

- Deve essere autenticato nel sistema.
- Deve esistere almeno un'asta attiva e accessibile.
- **Per il venditore:**
 - Deve essere autenticato e proprietario dell'asta.
 - Deve essere presente almeno un'offerta registrata per l'asta in corso.

Scenario Principale

Fase 1: Invio dell'offerta (Acquirente)

1. L'acquirente seleziona un'asta attiva e accede alla pagina di dettaglio.
2. L'acquirente inserisce l'importo dell'offerta e conferma l'operazione.
3. Il sistema valida i dati dell'offerta (es. importo superiore all'offerta corrente) e registra l'offerta.
4. L'acquirente riceve un messaggio di conferma per l'invio dell'offerta.

Fase 2: Accettazione dell'offerta (Venditore)

5. Il venditore visualizza l'elenco delle offerte per la propria asta dalla pagina dedicata.
6. Il venditore seleziona l'offerta che intende accettare e conferma l'operazione.
7. Il sistema:
 - Conclude l'asta, aggiornandone lo stato a "completata".
 - Genera un ordine per l'acquirente e il venditore.
 - Notifica entrambe le parti della conclusione dell'asta.
8. Il venditore riceve una conferma della chiusura dell'asta.

USE CASE #4 NOME UC – Invio e accettazione di un'offerta per asta silenziosa

<i>Goal in Context</i>	L'obiettivo è consentire a un acquirente di inviare un'offerta per un'asta silenziosa attiva e permettere al venditore di accettare l'offerta migliore, concludendo l'asta e generando un ordine per l'acquirente.		
	<ul style="list-style-type: none"> • L'utente è autenticato nel sistema • Esiste almeno un'asta silenziosa attiva 		
<i>Success End Condition</i>	<ul style="list-style-type: none"> • L'offerta viene registrata correttamente. • Il venditore accetta l'offerta migliore. 		
	• L'asta viene conclusa e un ordine è generato per l'acquirente vincitore.		
<i>DESCRIPTION</i>	Step n°	Attore 1 – UTENTE ACQUIRENTE/VENDITORE	Sistema
	1	L'acquirente seleziona un'asta silenziosa dalla pagina delle aste attive.	
	2		Mostra i dettagli dell'asta selezionata.
	3	Inserisce l'importo dell'offerta e conferma.	
	4		Valida e registra l'offerta.
	5		Mostra un messaggio di successo.
	6		Notifica al venditore dell'offerta inviata.
	7	Il venditore visualizza le offerte registrate per l'asta.	
	8	Il venditore accetta un'offerta e conferma.	
	9		Conclude l'asta e crea un ordine per l'acquirente vincitore.
<i>EXTENSIONS</i>	Step	Attore 1 - VENDITORE	Sistema
<i>Errore nella validazione dell'offerta</i>	3.a	Inserisce un importo minore rispetto all'ultima offerta disponibile	
	4.a		Mostra un messaggio di errore e permette di correggere.
<i>Il venditore rifiuta l'offerta</i>	8.b	Il venditore decide di non accettare alcuna offerta.	
	9.b		L'asta rimane attiva fino alla scadenza del tempo massimo.

MOCK-UP DELL'INTERFACCIA UTENTE

I mockup presentati in questa sezione rappresentano una visualizzazione dettagliata e statica dell'interfaccia utente (UI) della piattaforma, con un alto grado di fedeltà. Essi offrono una visione chiara e precisa di come l'interfaccia dovrebbe apparire e comportarsi una volta sviluppata, senza la necessità di un'interazione dinamica. Questi mockup sono uno strumento essenziale per il processo di progettazione e sviluppo della piattaforma, poiché offrono una rappresentazione visiva delle caratteristiche principali e delle funzionalità previste.

Funzioni principali e obiettivi dei mockup:

1. Definire l'esperienza utente (UX):

I mockup permettono di visualizzare come gli utenti interagiranno con la piattaforma, offrendo un'anticipazione dei flussi di navigazione e delle operazioni più comuni. Questo aiuta a garantire che l'interfaccia sia user-friendly, intuitiva e coerente, minimizzando potenziali difficoltà nell'utilizzo. Gli utenti finali possono navigare in modo fluido, comprendendo facilmente dove si trovano e cosa devono fare per raggiungere i propri obiettivi.

2. Facilitare la comunicazione con gli stakeholder:

I mockup sono uno strumento ideale per comunicare in modo chiaro e visivo le caratteristiche della piattaforma agli stakeholder (clienti, investitori, team di marketing, ecc.). La possibilità di visualizzare l'aspetto finale della piattaforma, prima che venga effettivamente sviluppata, permette agli stakeholder di avere un'idea precisa delle funzionalità e del flusso dell'interfaccia. Ciò facilita la raccolta di feedback, l'identificazione di eventuali modifiche necessarie e la condivisione di visioni comuni.

3. Documentare il design per il futuro:

Oltre a supportare la fase attuale del progetto, i mockup sono anche uno strumento di documentazione per eventuali modifiche o migliorie future. Poiché i mockup rappresentano la versione più accurata della progettazione visiva e funzionale, costituiscono un punto di riferimento per eventuali revisioni o aggiornamenti della piattaforma.

Sezione Welcome Page / Login :

La **Welcome Page** rappresenta il punto di ingresso dell'applicazione, progettata per accogliere l'utente con una descrizione chiara e concisa delle funzionalità principali. La schermata è minimalista, con un focus sul pulsante "Inizia", che invita l'utente a proseguire verso la pagina di login.

Obiettivi principali:

- Introdurre l'applicazione in modo immediato e diretto.
- Guidare l'utente al primo step del flusso di autenticazione.

La **Pagina di Login** consente agli utenti già registrati di accedere ai propri account mediante email e password. La schermata presenta opzioni aggiuntive per il login tramite social network (Google, Facebook, GitHub, e LinkedIn), semplificando l'accesso per coloro che preferiscono metodi di autenticazione esterni.

Funzionalità principali:

1. **Autenticazione tradizionale:** Inserimento di email e password per l'accesso.
2. **Login con social network:** Possibilità di autenticarsi utilizzando account esterni.
3. **Accesso alle schermate secondarie:**
 - **Password dimenticata:** Permette di recuperare l'accesso in caso di smarrimento delle credenziali.
 - **Pagina di registrazione:** Consente la creazione di un nuovo account per utenti non registrati.

La **Pagina di Login** è progettata per offrire un'esperienza intuitiva e inclusiva, assicurando che ogni utente, indipendentemente dal metodo di autenticazione scelto, possa completare rapidamente l'accesso alla piattaforma.

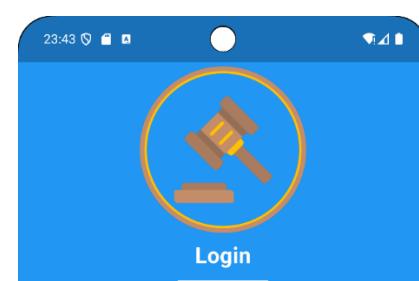


Figura 8: Pagina di welcome

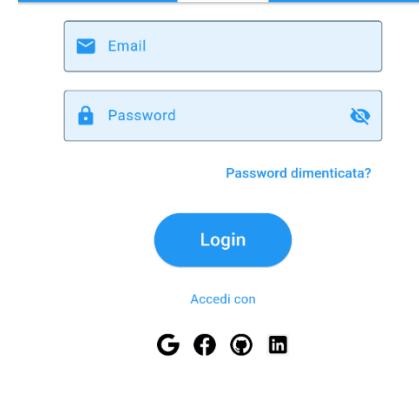


Figura 9: Pagina di login

Sezione Password Dimenticata /Registrazione :

La schermata **Password Dimenticata** è progettata per consentire agli utenti di recuperare l'accesso al proprio account in caso di smarrimento delle credenziali. Attraverso un'interfaccia semplice e intuitiva, l'utente può reimpostare la password seguendo questi passaggi:

1. **Inserimento email:** L'utente inserisce l'indirizzo email registrato.
2. **Nuova password e conferma:** L'utente immette una nuova password e ne conferma la corrispondenza.

Flusso di comportamento:

- Se l'email è valida e registrata nei sistemi, il processo di reimpostazione viene completato con successo, e l'utente può accedere con la nuova password.
- In caso contrario, un messaggio di errore informa l'utente che l'email inserita non è presente nei sistemi.

Obiettivo: Garantire un recupero delle credenziali rapido e senza complicazioni, mantenendo alti standard di sicurezza.

La schermata **Registrazione** consente agli utenti di creare un account personale per accedere all'applicazione. L'interfaccia guida l'utente attraverso i seguenti campi obbligatori:

- **Nome**
- **Cognome**
- **Data di nascita**
- **Indirizzo email**
- **Password**

Flusso di comportamento:

1. L'utente compila tutti i campi richiesti e conferma la registrazione.
2. Dopo la registrazione avvenuta con successo, l'utente viene automaticamente indirizzato alla schermata di login per accedere con le credenziali appena create.
3. In caso di utilizzo di un'e-mail già registrata, un messaggio di errore informa l'utente che l'indirizzo email è già presente nei sistemi, suggerendo di effettuare il login o di recuperare la password.

Obiettivo: Facilitare la creazione di un account in pochi passaggi, garantendo un'esperienza fluida e priva di ambiguità per l'utente.

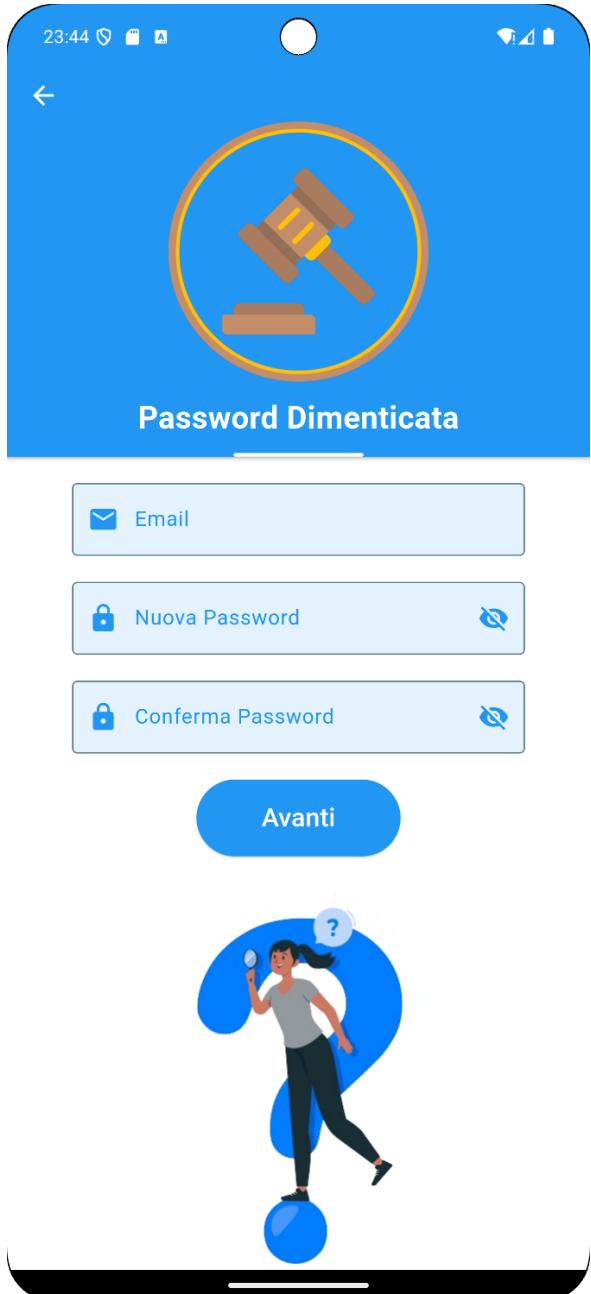


Figura 10: Pagina di recupero password

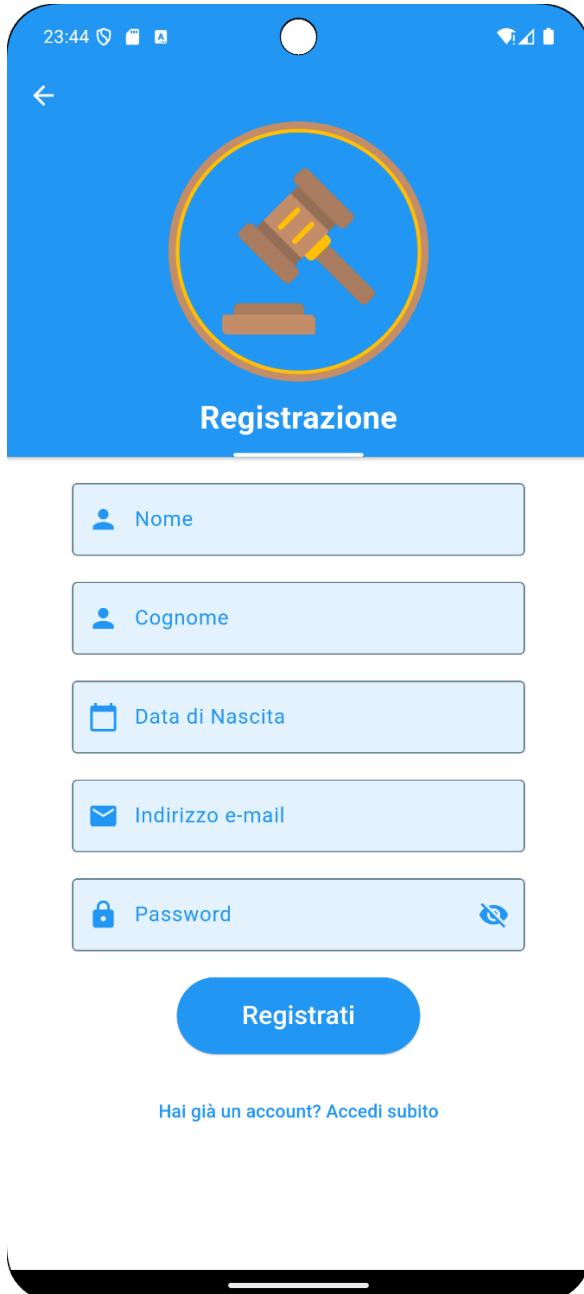


Figura 11: Pagina di registrazione

Sezione Homepage :

La **homepage** rappresenta il punto di accesso principale per l'utente, offrendo un'esperienza di navigazione fluida e intuitiva. La pagina è progettata per mettere in evidenza contenuti di interesse e fornire accesso rapido alle aste attive e alle categorie principali.

Carosello Principale: Nella sezione centrale della homepage è presente un carosello che mostra una selezione curata di aste relative a tre categorie principali:

- **Arte**
- **Gioielli**
- **Orologi da polso**

Dettagli funzionali:

- Ogni card nel carosello include un'anteprima del prodotto, un breve titolo, una breve descrizione e la data di termine dell'asta.
- Il carosello scorre automaticamente, mantenendo l'utente costantemente aggiornato sui contenuti più rilevanti.

Selezione Quotidiana di Aste: Questa sezione mostra una panoramica delle aste attive, limitata a 9 elementi visualizzati direttamente nella homepage.

- Pulsante "**Visualizza Tutto**": Consente agli utenti di accedere all'elenco completo delle aste attive nel sistema.
- **Aggiunta ai Preferiti:** Le aste possono essere aggiunte ai preferiti sia dalla homepage che dalla pagina con tutte le aste attive.

Obiettivo: Offrire agli utenti una rapida visibilità delle aste più interessanti e incoraggiare l'esplorazione dell'intero catalogo.

Slider delle Aste in Scadenza: Una sezione dedicata evidenzia le aste che stanno per terminare, con uno slider orizzontale che ne facilita la consultazione.

Categorie Consigliate: La homepage include una sezione con le categorie consigliate:

- Arte
- Gioielli
- Orologi da polso
- Moda
- Monete e francobolli
- Fumetti e animazione

Interazione:

- Ogni card di categoria conduce a una pagina dedicata, che mostra tutte le aste associate a quella categoria.
- Le categorie sono rappresentate da immagini accattivanti e brevi descrizioni per attirare l'attenzione dell'utente.

Obiettivo della Homepage:

La homepage è stata progettata per fornire una panoramica chiara e accattivante delle principali aste disponibili, offrendo percorsi di navigazione immediati verso i contenuti più rilevanti. La combinazione di caroselli, slider e

categorie consigliate consente agli utenti di scoprire facilmente le aste di loro interesse, garantendo un'esperienza dinamica e coinvolgente.

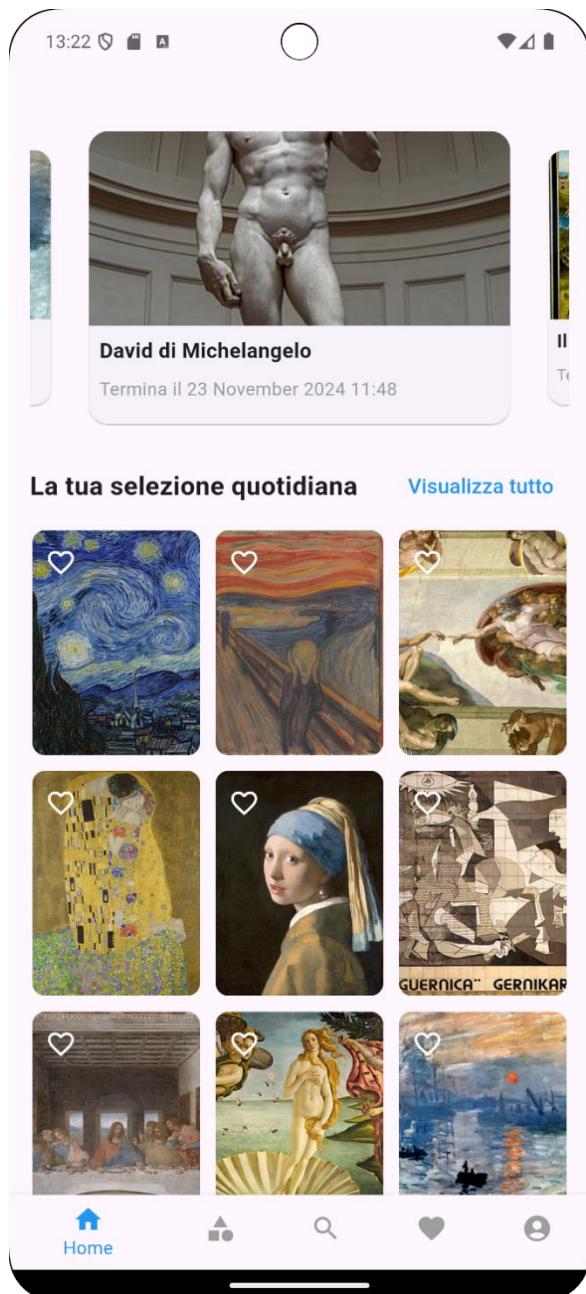


Figura 12: Homepage 1

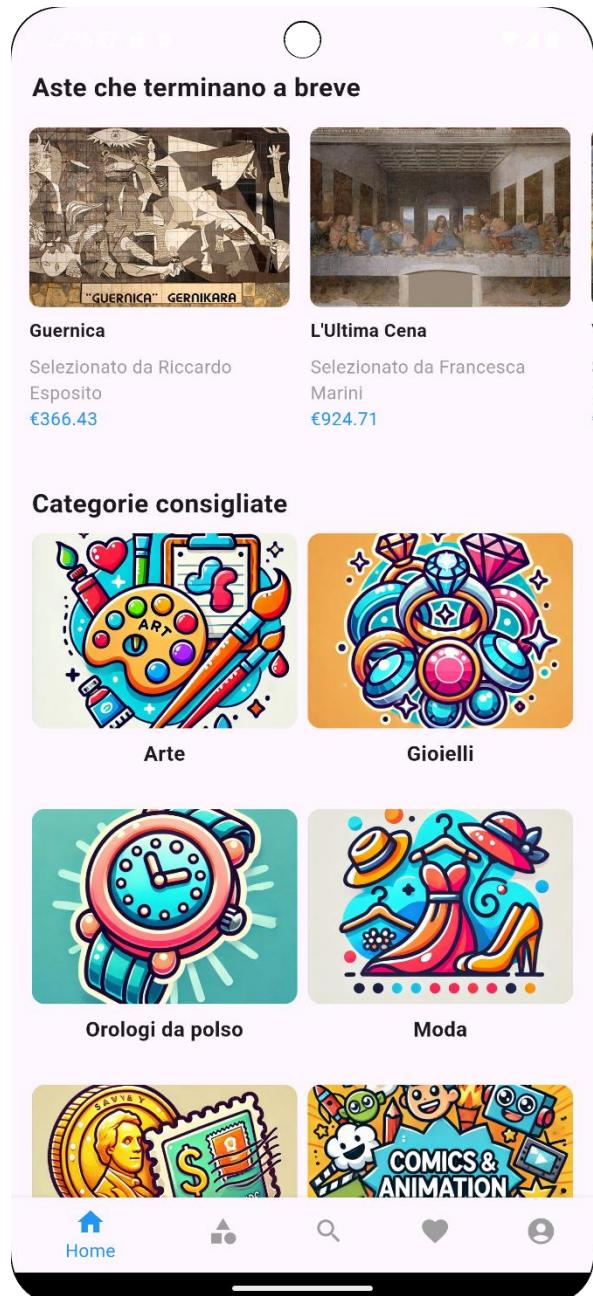


Figura 13: Homepage 2

Sezione « Visualizza Tutto » della Selezione Quotidiana :

Questa sezione offre agli utenti l'accesso completo all'elenco di tutte le aste attive. Questa funzionalità consente di esplorare l'intero catalogo di aste disponibili in modo organizzato e personalizzabile.

Gli utenti possono scegliere tra due modalità di visualizzazione:

1. Visualizzazione a Lista:

- Ogni asta è rappresentata da una card che include dettagli essenziali: immagine, titolo, descrizione breve e prezzo attuale.
- Disposizione verticale per una consultazione ordinata e dettagliata.

2. Visualizzazione a Griglia:

- Le aste sono presentate in un layout a griglia, includendo sempre immagine, titolo, descrizione e prezzo attuale, per un'esplorazione visiva più rapida.
- Ideale per chi preferisce un approccio più compatto e grafico.

Interazione:

- **Aggiunta ai Preferiti:** Icona dedicata su ogni card, disponibile in entrambe le modalità di visualizzazione, per aggiungere rapidamente le aste alla lista dei preferiti.
- **Navigazione Dettagliata:** Ogni card è cliccabile e reindirizza l'utente alla pagina dettagliata dell'asta, dove è possibile vedere ulteriori informazioni e partecipare con un'offerta.

Navigazione e Esperienza Utente

- Pulsante di Cambiamento Visuale: Un pulsante dedicato consente agli utenti di alternare rapidamente tra la modalità lista e la modalità griglia.
- Interfaccia Dinamica: La transizione tra le modalità è fluida e immediata, garantendo un'esperienza utente ottimale.

Obiettivo della Sezione "Visualizza Tutto":

Questa sezione è stata progettata per offrire una panoramica completa delle aste attive, mettendo a disposizione strumenti intuitivi per ordinare e personalizzare l'esplorazione. Le opzioni di visualizzazione a lista e a griglia

garantiscono flessibilità, soddisfacendo le preferenze di diversi tipi di utenti, mentre l'accesso immediato ai dettagli delle aste incoraggia un'interazione più profonda.

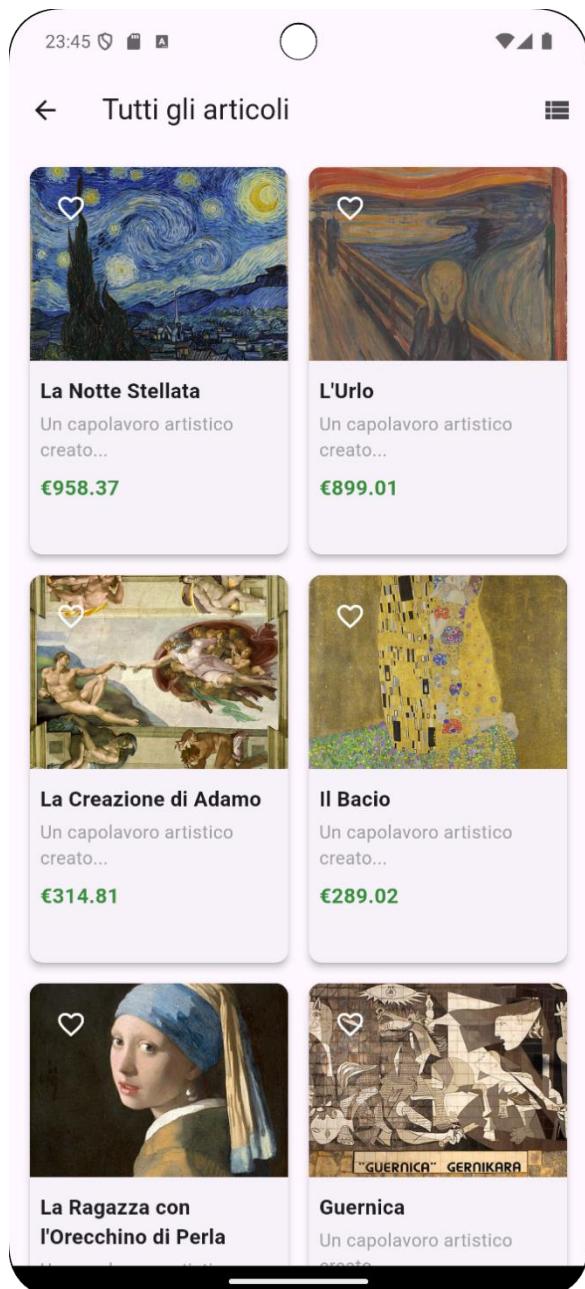


Figura 14: Pagina di tutti gli articoli (visualizzazione griglia)

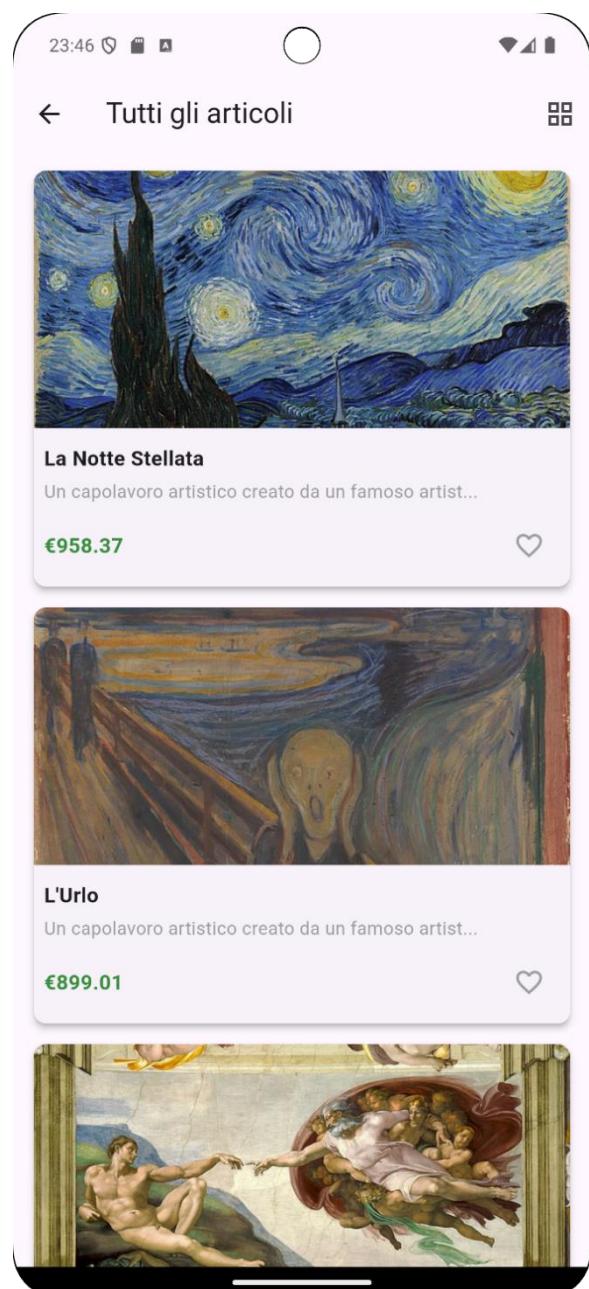


Figura 15: Pagina di tutti gli articoli (visualizzazione lista)

Sezione Categorie / Dettaglio Categorie :

Questa sezione rappresenta un punto di accesso centrale per esplorare tutte le categorie disponibili all'interno dell'applicazione. Con un design intuitivo e interattivo, gli utenti possono navigare tra le varie categorie e accedere ai rispettivi elenchi di aste.

Categorie Disponibili

- L'elenco include una vasta gamma di categorie per soddisfare gli interessi degli utenti:
 - **Arte**
 - **Gioielli**
 - **Orologi da Polso**
 - **Moda**
 - **Monete e Francobolli**
 - **Fumetti e Animazione**
 - **Auto e Moto**
 - **Carte Collezionabili**
 - **Giocattoli e Modellini**
 - **Archeologia**
 - **Sport**
 - **Musica, Film e Fotocamere**
 - **Libri e Cimeli Storici**

Interazione con le Categorie

- **Cliccabilità:**
Ogni categoria è rappresentata da una card o un elemento cliccabile che conduce l'utente alla pagina dedicata, contenente l'elenco delle aste corrispondenti.
- **Design Visuale:**
 - Ogni card include un'immagine rappresentativa, titolo, breve descrizione e prezzo attuale dell'asta.

Anche qui gli utenti possono scegliere tra le due modalità di visualizzazione:

- Visualizzazione a lista;
- Visualizzazione a Griglia

Entrambe prendono lo stesso concetto visto in precedenza.

Obiettivo della Sezione "Categorie":

La sezione "Categorie" è progettata per garantire un'esperienza di navigazione completa, offrendo agli utenti la libertà di esplorare le aste di loro interesse in modo personalizzato e fluido.

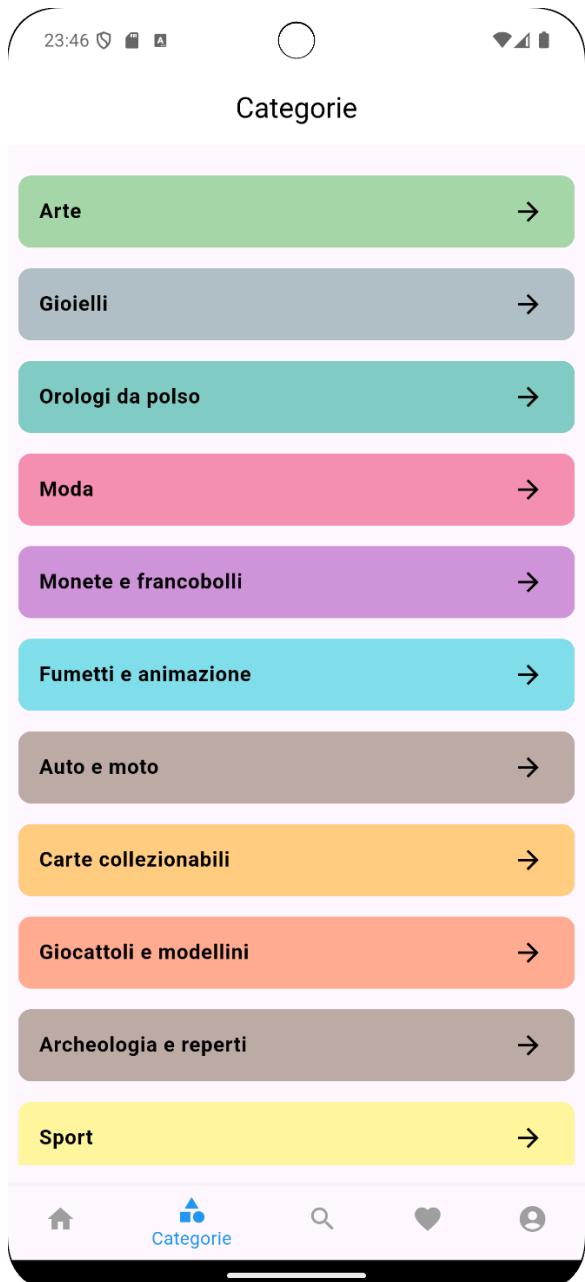


Figura 16: Pagina delle categorie

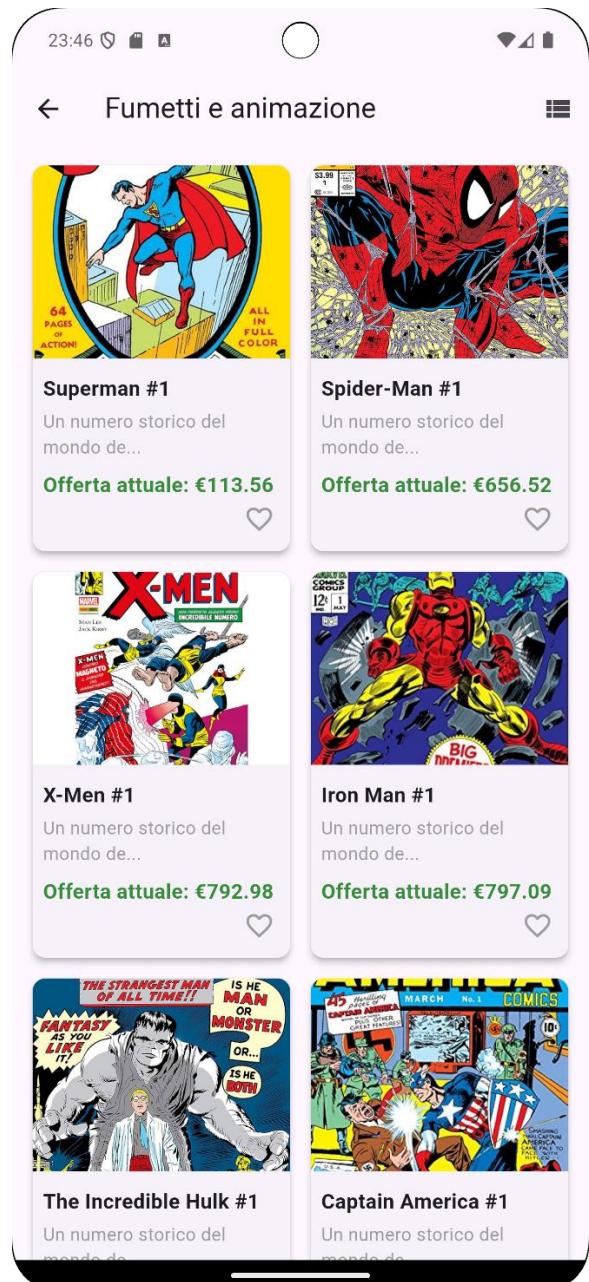


Figura 17: Pagina di una specifica categoria

Sezione Ricerca :

Questa sezione consente agli utenti di individuare rapidamente i prodotti all'asta, fornendo strumenti avanzati per affinare i risultati in modo semplice e intuitivo.

Funzionalità Principali

1. Ricerca tramite Parola Chiave:

- Nella parte superiore della pagina è presente una barra di ricerca prominente dove gli utenti possono inserire parole chiave relative al prodotto desiderato.
- La ricerca restituisce risultati rilevanti in tempo reale, basandosi su titoli e descrizioni

2. Filtri per Categoria:

- Un menu a tendina permette di filtrare i risultati per le categorie disponibili, elencate nella sezione precedente

Messaggi di Feedback

• Nessun Risultato:

- Se la ricerca non produce risultati, un messaggio dedicato guiderà l'utente a riformulare la ricerca o a modificare i filtri applicati.
- Esempio: "*Nessun risultato trovato per la parola chiave inserita. Prova con un termine differente o modifica i filtri.*"

Navigazione Fluida

- Accesso Intuitivo: La barra di ricerca è sempre visibile, consentendo agli utenti di affinare la ricerca in qualsiasi momento.
- Ritorno alla Home o ad Altre Pagine: Un menu di navigazione rapido permette di tornare facilmente alla Home o esplorare altre sezioni dell'app.

Obiettivo della Pagina di Ricerca:

La pagina di ricerca è stata progettata per offrire un'esperienza di esplorazione rapida e precisa. Con strumenti intuitivi come la ricerca per parola chiave e i filtri avanzati, gli utenti possono trovare facilmente i prodotti di loro interesse, migliorando l'efficienza e la soddisfazione durante l'uso dell'applicazione.



Figura 18: Pagina di ricerca



Figura 19: Pagina di ricerca con selezione filtro

ingsw_dietideals24_app

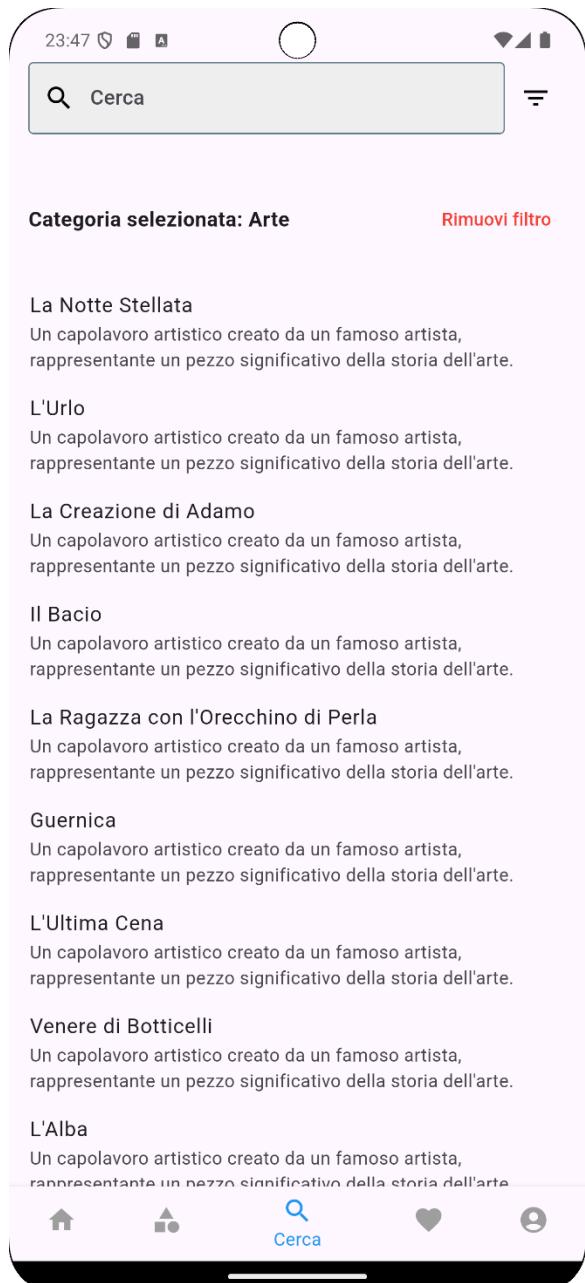


Figure 20: Pagina di ricerca con filtro inserito

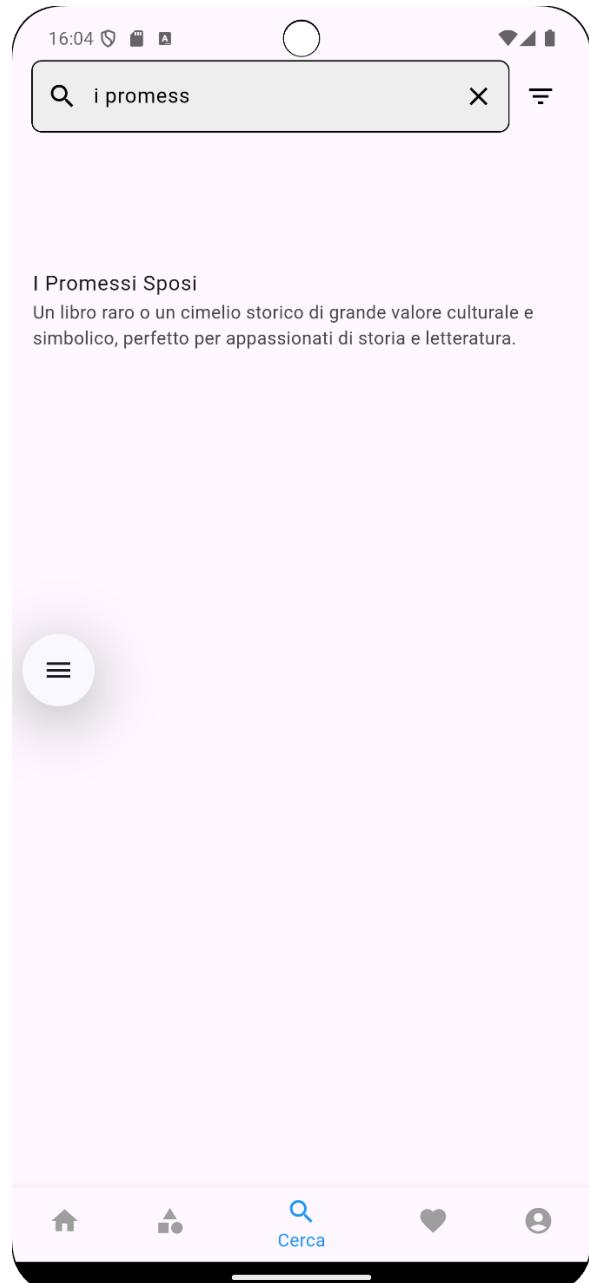


Figura 21: Pagina di ricerca con parola chiave

Sezione Preferiti :

Questa sezione raccoglie tutti i prodotti che l'utente ha selezionato come preferiti, offrendo un punto di accesso rapido e organizzato agli oggetti di maggiore interesse.

Funzionalità Principali

1. Visualizzazione dei Preferiti:

- I prodotti selezionati come preferiti vengono presentati in un layout uniforme e intuitivo, con i seguenti dettagli:
 - Immagine del prodotto: Visualizzazione di anteprima chiara e accattivante.
 - Titolo: Nome dell'asta o del prodotto.
 - Descrizione: Breve riepilogo del prodotto, evidenziando i dettagli principali.
 - Prezzo: Prezzo attuale o ultima offerta registrata.

2. Rimozione dai Preferiti:

- Un'icona a forma di cestino è disponibile accanto a ciascun prodotto per rimuoverlo dalla lista.
- Al clic, il prodotto viene eliminato immediatamente e la lista si aggiorna dinamicamente.

3. Messaggio di Feedback:

- Se non ci sono prodotti nella lista dei preferiti, viene visualizzato un messaggio informativo:
 - *"Nessun preferito trovato."*

Design e Interattività

• Layout Pulito:

- Ogni prodotto è presentato in una card chiara e ben organizzata.

• Interazione Utente:

- Cliccando su una card, l'utente viene reindirizzato alla pagina dettagliata del prodotto per ulteriori informazioni o per partecipare all'asta.

Obiettivo della Pagina dei Preferiti:

La pagina dei preferiti è progettata per offrire un'esperienza personalizzata, permettendo agli utenti di monitorare con facilità i prodotti che trovano più interessanti. La possibilità di rimuovere gli oggetti inutilizzati e di accedere rapidamente ai dettagli delle aste rende questa pagina uno strumento pratico e altamente funzionale.

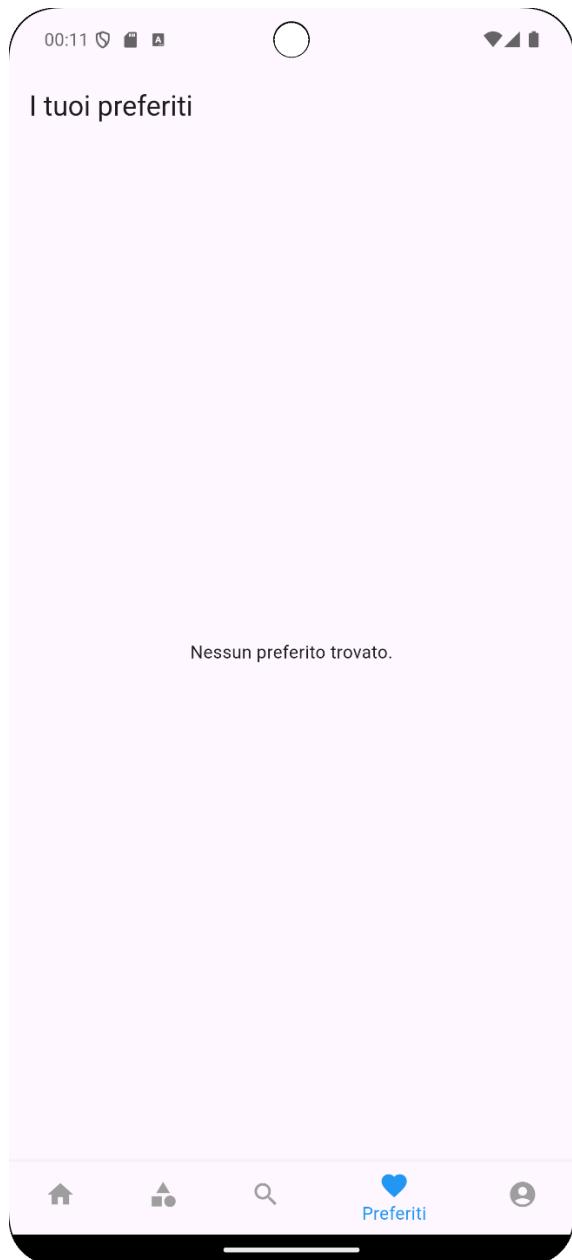


Figura 22: Pagina dei preferiti 1

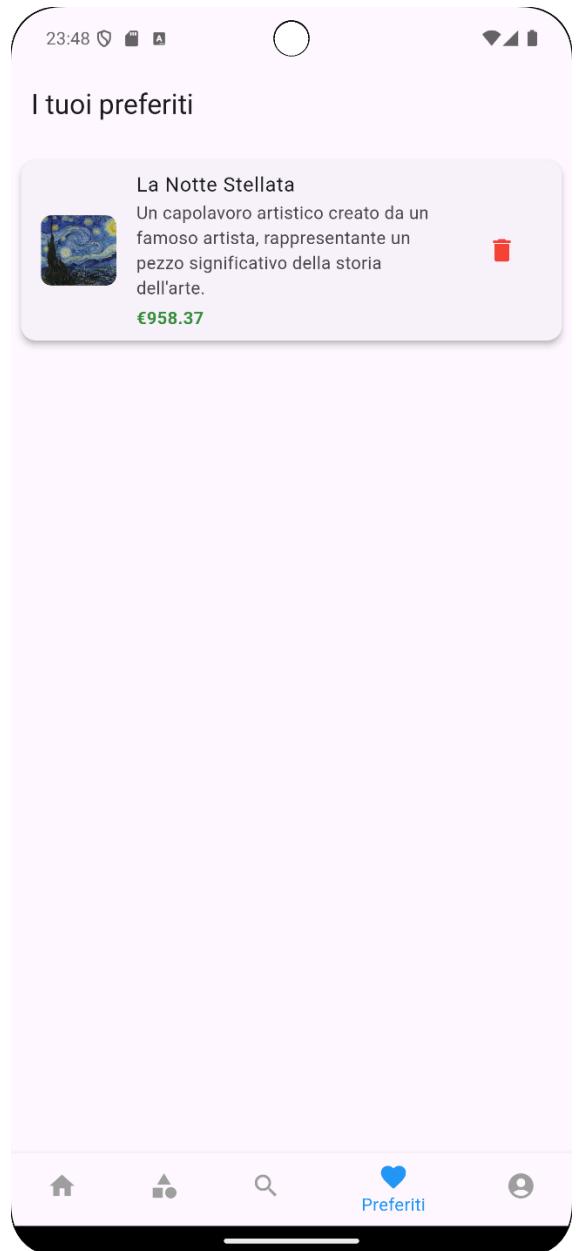


Figura 23: Pagina dei preferiti 2

Sezione Account / Profilo Utente :

Questa sezione funge da hub centrale per l'utente, consentendogli di navigare facilmente tra diverse sezioni e gestire le proprie attività all'interno dell'applicazione.

La pagina è suddivisa in sezioni principali:

1. Acquista:

- *Ordini:* Permette di visualizzare l'elenco degli ordini relativi alle aste vinte.
- *Le mie offerte:* Mostra una panoramica dettagliata di tutte le offerte effettuate dall'utente.

2. Vendi:

- *Crea asta:* Consente all'utente di creare nuove aste, configurando i dettagli del prodotto.
- *Prodotti all'asta:* Visualizza l'elenco degli oggetti attualmente in asta creati dall'utente.
- *Prodotti venduti:* Elenco degli oggetti che sono stati venduti con successo.
- *Prodotti non venduti:* Raccolta degli oggetti che non hanno ricevuto offerte o non sono stati venduti.

3. Posta in arrivo:

- Una sezione dedicata alla gestione delle notifiche, tra cui:
 - *Aste non vendute.*
 - *Aste vinte.*
- Le notifiche possono essere lette, archiviate o eliminate per mantenere l'organizzazione.

4. Account:

- *Impostazioni dell'account:*
 - Permette di modificare informazioni personali come avatar, nome, cognome, password, bio, località, sito web e social collegati.
 - Opzione per eliminare l'account, rimuovendo l'utente e tutti i relativi dati (aste, offerte, ordini) dai sistemi.
- *Indirizzi:*
 - Consente di aggiornare l'indirizzo di spedizione e/o fatturazione.
- *Pagamenti:*
 - Possibilità di inserire, visualizzare o rimuovere carte di credito/debito come metodi di pagamento.

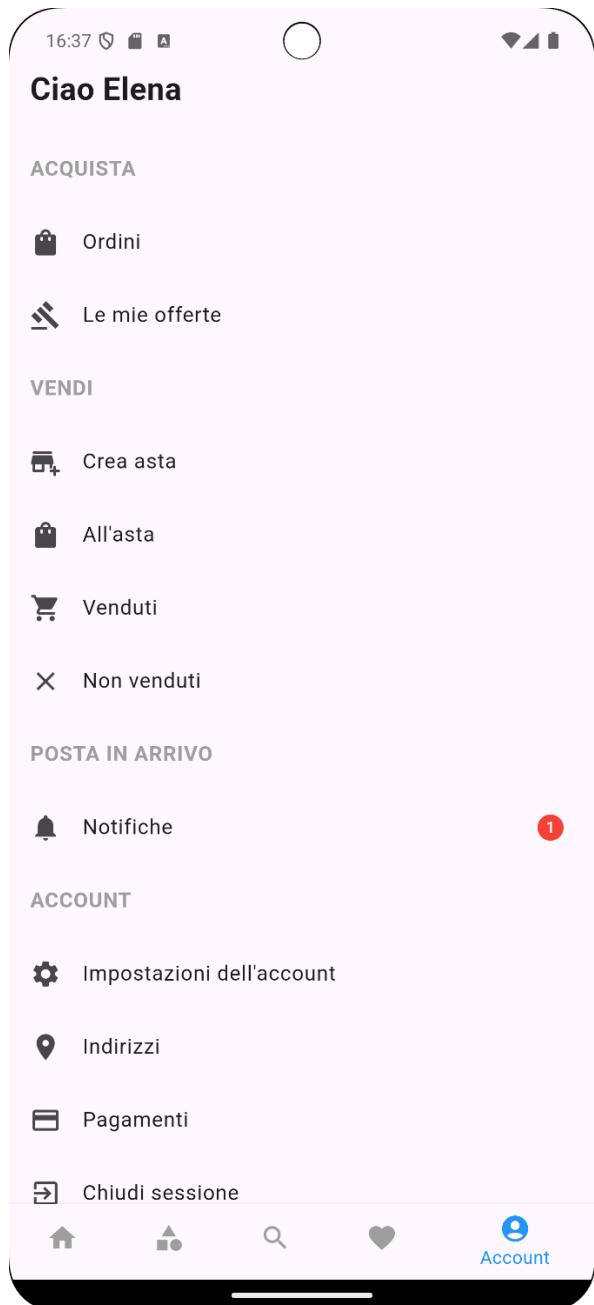


Figure 24: Pagina account

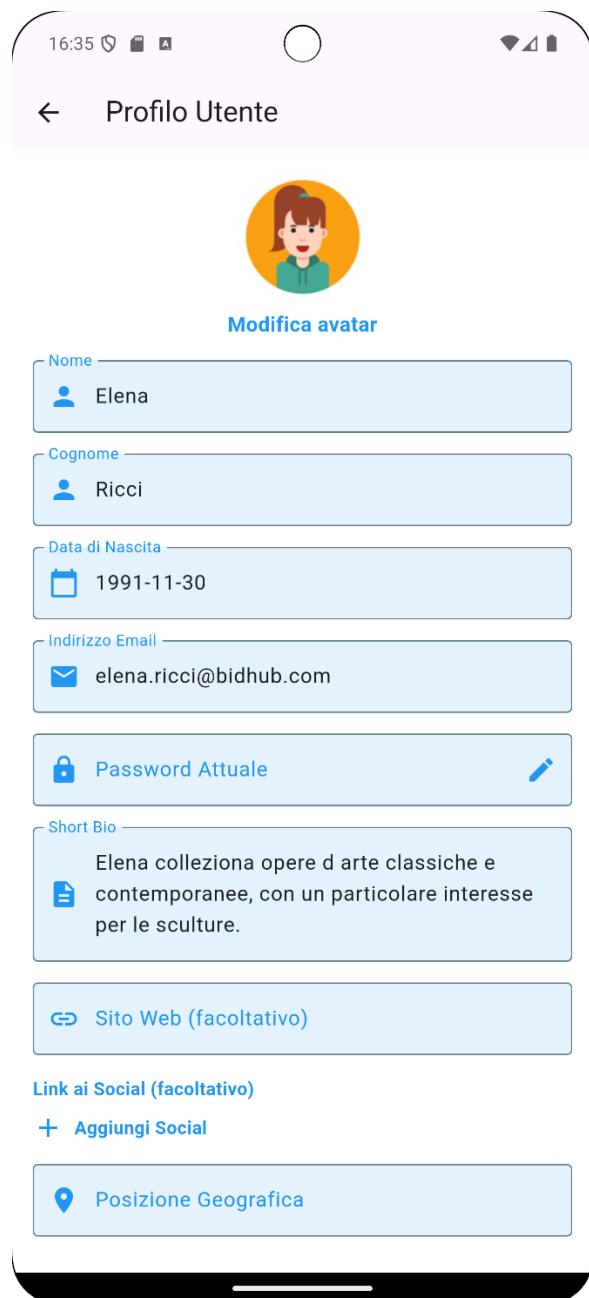


Figura 25: Pagina del profilo utente

Sezione Indirizzi / Pagamenti :

La pagina **Indirizzi**, accessibile dalla sezione "**Account**", consente all'utente di gestire i propri indirizzi di spedizione e fatturazione in modo semplice e veloce.

Funzionalità Principali

1. Inserimento Indirizzo di Spedizione:

- L'utente può compilare un campo dedicato per specificare il proprio indirizzo di spedizione.

2. Indirizzo di Fatturazione:

- Se l'indirizzo di fatturazione è uguale a quello di spedizione, l'utente può selezionare una checkbox che sincronizza automaticamente i due campi.
- Se diverso, è possibile compilare l'indirizzo di fatturazione manualmente.

3. Salvataggio delle Modifiche:

- Un pulsante consente di confermare e salvare gli indirizzi inseriti o aggiornati.

Interfaccia Utente

- Checkbox Intuitiva:

- La checkbox per indicare la corrispondenza tra gli indirizzi è posizionata in modo prominente per essere facilmente accessibile.

La pagina **Pagamenti**, sempre accessibile dalla sezione "**Account**", permette all'utente di gestire i propri metodi di pagamento in modo sicuro.

Funzionalità Principali

1. Inserimento Carta di Credito/Debito:

- L'utente può aggiungere una nuova carta inserendo:
 - Nome e cognome dell'intestatario.
 - Numero della carta.
 - Data di scadenza (formato MM/AAAA).
 - Codice CVC (non salvato nei sistemi per motivi di privacy).

2. Gestione Carte Salvate:

- Le carte salvate sono elencate con informazioni essenziali (es. numero carta parzialmente oscurato per sicurezza, intestatario e scadenza).
- È possibile rimuovere una carta utilizzando l'**icona del cestino** accanto alla carta desiderata.

Interfaccia Utente

- Form Intuitivo:

- Ogni campo è chiaramente etichettato e posizionato per guidare l'utente nella compilazione.

- Sicurezza e Privacy:

- Il codice CVC non viene salvato per rispettare le norme sulla protezione dei dati sensibili.

- I numeri delle carte vengono visualizzati con cifre oscurate, mostrando solo le ultime 4 cifre.
- **Feedback Utente:**
 - Notifiche di successo o errori (es. carta non valida) sono visualizzate in tempo reale.

Obiettivi della Pagina degli Indirizzi e dei Pagamenti

Le **pagine Indirizzi e Pagamenti** sono progettate per semplificare la gestione delle informazioni personali e dei metodi di pagamento degli utenti. Con un'interfaccia sicura e intuitiva, l'utente può aggiornare i propri dati con facilità, avendo la certezza che le informazioni sensibili sono trattate con la massima riservatezza.

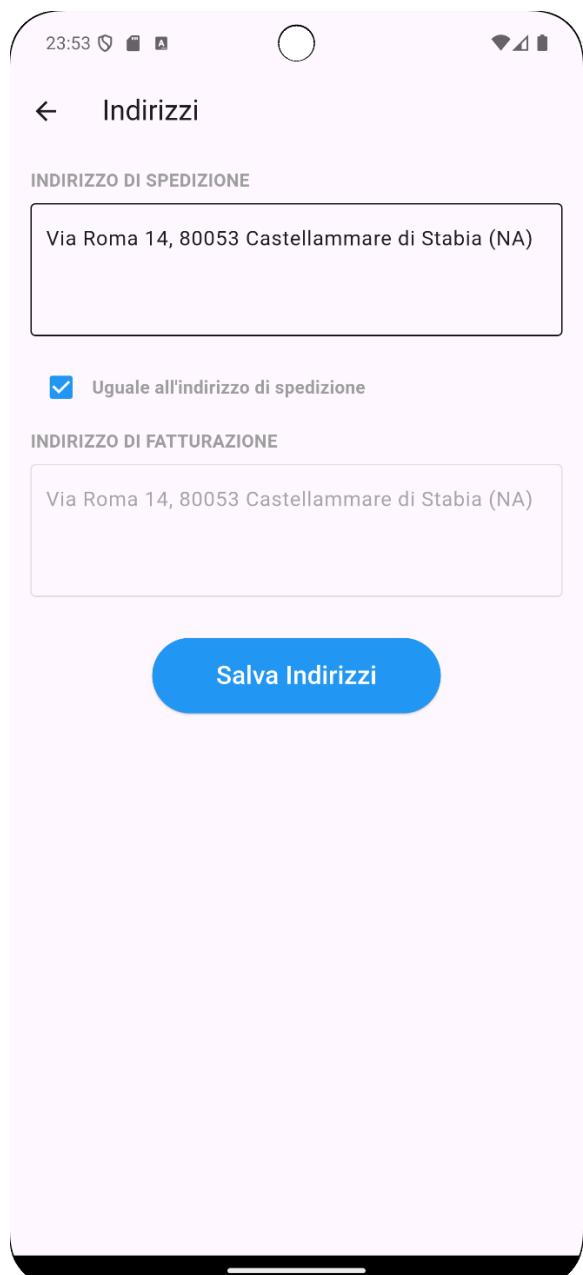


Figura 26: Pagina degli indirizzi



Figura 27: Pagina dei pagamenti 1

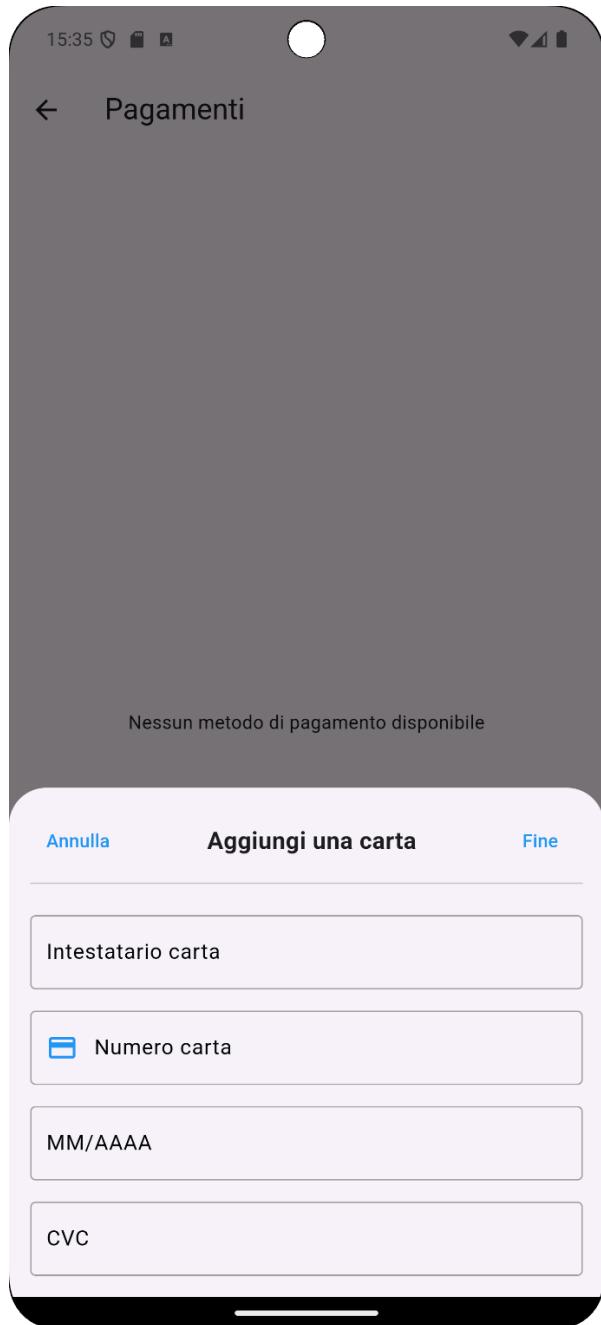


Figure 28: Pagina dei pagamenti 2



Figure 29: Pagina dei pagamenti 3

Sezione Notifiche :

Questa sezione è accessibile dalla pagina "**Account**", consente agli utenti di monitorare gli eventi e aggiornamenti legati alle proprie aste. Questa pagina è progettata per fornire informazioni in modo organizzato e intuitivo, garantendo un'interazione rapida ed efficace con le notifiche ricevute.

Tipologie di Notifiche

1. Offerta Ricevuta per un Prodotto:

- L'utente riceve una notifica quando viene fatta un'offerta per uno dei propri prodotti all'asta.

2. Offerta per un'Asta Silenziosa:

- Notifica che invita l'utente a decidere se accettare o rifiutare un'offerta ricevuta per un prodotto inserito in un'asta silenziosa.

3. Asta Persa:

- L'utente viene informato che un'asta è terminata senza offerte ricevute.
- Oppure viene notificato che non è stato raggiunto il prezzo minimo richiesto per concludere la vendita.

Funzionalità Principali

1. Visualizzazione Notifiche:

- Tutte le notifiche sono elencate in ordine cronologico.
- Notifiche non lette sono evidenziate per attirare l'attenzione dell'utente.

2. Gestione delle Notifiche:

- *Segna come Letta:*
 - L'utente può contrassegnare una singola notifica come letta per archiviazione.
- *Elimina Notifiche:*
 - Opzione per cancellare una notifica specifica tramite un'icona dedicata.
 - Possibilità di eliminare tutte le notifiche contemporaneamente per una gestione più rapida.

3. Feedback Visivo:

- Notifiche lette cambiano stato visivo, passando da evidenziate a uno stile meno prominente.
- Conferme visive sono fornite per le azioni di cancellazione o modifica dello stato delle notifiche.

Interfaccia Utente

• Design Pulito e Organizzato:

- Ogni notifica è presentata come una scheda con:
 - Messaggio: Dettagli sull'evento (es. "Offerta ricevuta per il prodotto X").
 - Azioni Disponibili: Icone per segnare come letta o eliminare la notifica.
- Le notifiche non lette sono evidenziate con un colore di sfondo diverso.

• Gestione Azioni:

- Un'azione globale consente di eliminare tutte le notifiche o di segnarle tutte come lette, direttamente dalla barra superiore.

Obiettivi della Pagina delle Notifiche

La pagina delle Notifiche è stata progettata per migliorare l'esperienza utente, consentendo un rapido accesso agli aggiornamenti essenziali sulle aste. L'interfaccia è orientata all'efficienza, con opzioni chiare per la gestione delle notifiche e un focus sull'accessibilità e la facilità d'uso.



Figure 30: Pagina delle notifiche

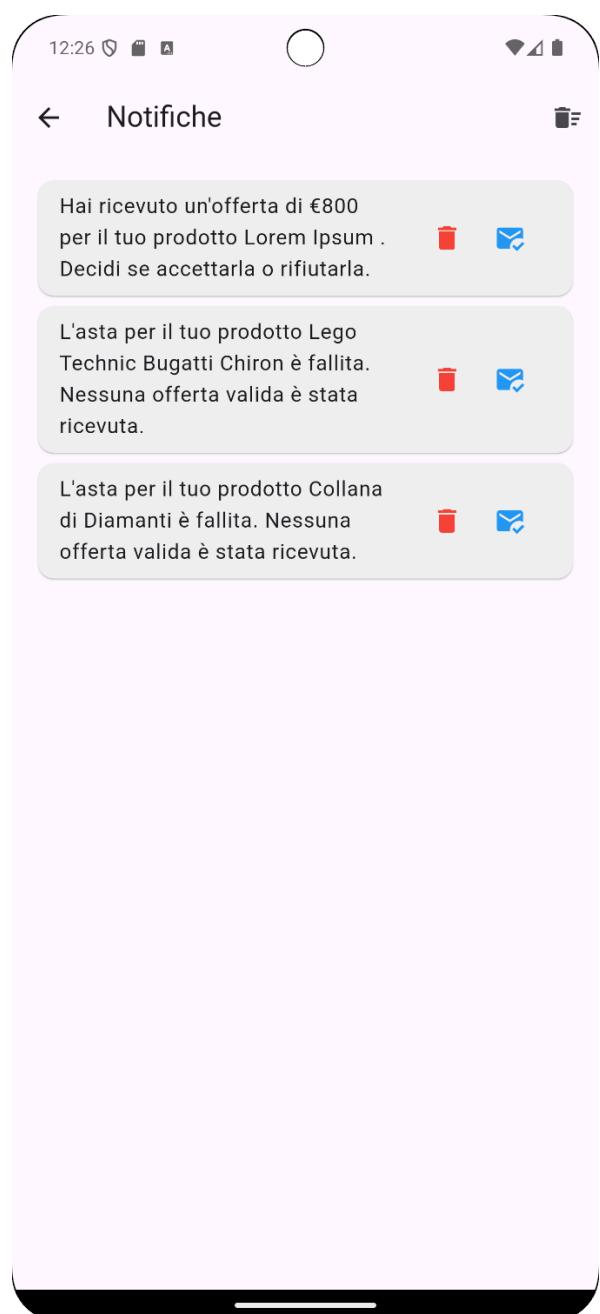


Figure 31: Pagina delle notifiche del venditore



Figure 32: Accettare o rifiutare offerta per asta silenziosa



Figure 33: Pagina delle notifiche dell'acquirente

Sezione Creazione Asta :

Questa sezione è accessibile dalla sezione "**Vendi**" all'interno della pagina "**Account**", consente agli utenti di creare nuove aste scegliendo tra diverse tipologie disponibili. L'interfaccia è progettata per essere intuitiva e flessibile, con campi specifici per ciascun tipo di asta e feedback immediato per garantire un'esperienza fluida.

Tipologie di Asta Disponibili

1. Asta a Tempo Fisso:

- Ideale per vendite standard con una durata definita.
- Campi obbligatori:
 - Titolo
 - Descrizione
 - Prezzo di vendita (iniziale)
 - Prezzo minimo (segreto)
 - CATEGORIA
 - Data e ora di scadenza
- Campo facoltativo:
 - Immagine del prodotto

2. Asta all'Inglese:

- Perfetta per aste competitive basate su offerte crescenti.
- Campi obbligatori:
 - Titolo
 - Descrizione
 - Base d'asta
 - Soglia di rialzo (valore predefinito: 10€)
 - Intervallo di tempo in ore (valore predefinito: 1h)
 - CATEGORIA
- Campo facoltativo:
 - Immagine del prodotto

3. Asta al Ribasso:

- Ideale per attrarre acquirenti tramite un prezzo decrescente nel tempo.
- Campi obbligatori:
 - Titolo
 - Descrizione
 - Prezzo iniziale (preferibilmente elevato)

- Decremento di prezzo
- Prezzo minimo (segreto)
- Timer di decremento (valore predefinito: 1h)
- CATEGORIA
- Campo facoltativo:
 - Immagine del prodotto

4. Asta Silenziosa:

- Pensata per vendite private con un prezzo iniziale e senza offerte visibili.
- Campi obbligatori:
 - Titolo
 - Descrizione
 - Prezzo iniziale
 - Data e ora di scadenza
 - CATEGORIA
- Campo facoltativo:
 - Immagine del prodotto

Funzionalità della Pagina

1. Navigazione e Selezione del Tipo di Asta:

- Gli utenti possono selezionare la tipologia di asta da un elenco o tramite schede intuitive.

2. Form Dinamico:

- La pagina adatta i campi visualizzati in base alla tipologia di asta selezionata.

3. Feedback in Tempo Reale:

- *Messaggio di successo:*
 - Mostrato al completamento della creazione, confermando l'avvenuto inserimento dell'asta nei sistemi.
- *Messaggio di errore:*
 - Visualizzato se mancano campi obbligatori, con indicazioni su quali dati devono essere completati.

4. Validazione dei Campi:

- Controlli lato client per assicurare che i campi obbligatori siano compilati e che i valori inseriti rispettino i criteri richiesti (es. prezzo minimo maggiore di zero).

5. Opzioni di Personalizzazione:

- Inserimento di immagini facoltative per rendere più accattivante l'asta.
- Configurazione di parametri specifici come intervalli di tempo e valori di soglia.

Interfaccia Utente

- Design Pulito e Organizzato:
 - L'interfaccia guida l'utente attraverso i passaggi necessari per completare la creazione dell'asta.
- Focus sui Campi Essenziali:
 - I campi obbligatori sono evidenziati, mentre quelli facoltativi sono separati visivamente per evitare confusione.

Obiettivi della Pagina

La pagina è progettata per:

- Consentire agli utenti di creare aste personalizzate in modo rapido ed efficiente.
- Offrire un'interfaccia semplice ma potente, adatta a diverse esigenze di vendita.
- Garantire che i dati inseriti siano validi e completi, riducendo errori e semplificando la gestione delle aste.



Figure 34: Pagina della creazione del tipo d'asta

Figure 35: Creazione asta a tempo fisso

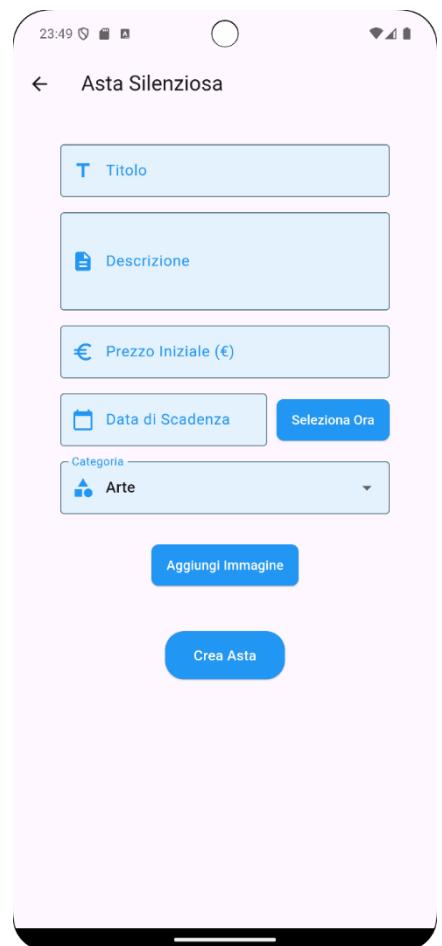
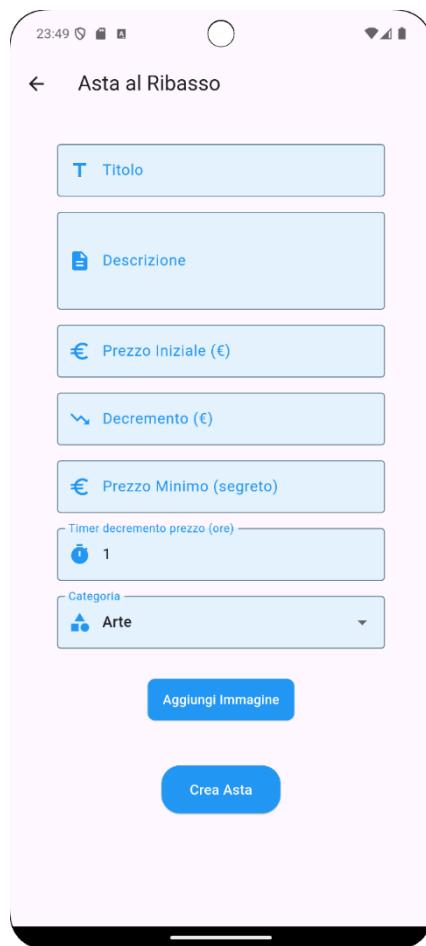
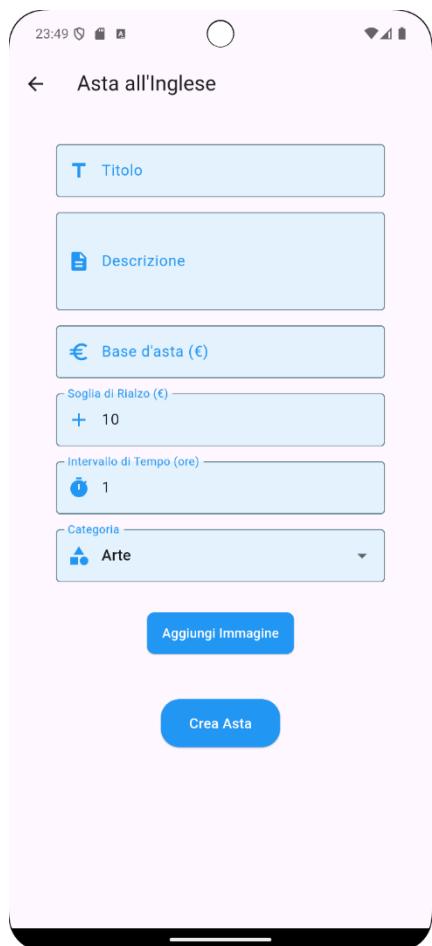


Figure 36: Creazione asta all'inglese

Figure 37: Creazione asta al ribasso

Figure 38: Creazione asta silenziosa

Sezione Aste Attive :

La sezione **Aste Attive**, raggiungibile dalla pagina "**Account**" nella sezione "**Vendi**", consente agli utenti di visualizzare e gestire le proprie aste attualmente in corso. Questa sezione è fondamentale per i venditori, offrendo un controllo completo sulle proprie inserzioni attive, con un'interfaccia chiara e funzionalità essenziali.

Funzionalità Principali

1. Elenco delle Aste Attive:

- a. Ogni asta viene visualizzata sotto forma di scheda o lista con i seguenti dettagli:
 - i. Immagine del prodotto: Mostra l'immagine caricata dall'utente per rappresentare l'asta.
 - ii. Titolo: Il nome dell'asta per una rapida identificazione.
 - iii. Descrizione: Una breve panoramica del prodotto all'asta.
 - iv. Prezzo Iniziale: Indica il valore di partenza per l'asta.

2. Navigazione alla Pagina di Dettaglio:

- a. Cliccando su una scheda d'asta, l'utente viene indirizzato alla pagina di dettaglio del prodotto, dove può visualizzare informazioni più approfondite e monitorare le offerte ricevute.

3. Eliminazione delle Aste:

- a. Ogni asta attiva include un'icona a forma di cestino.
- b. Cliccando sull'icona, l'utente può eliminare l'asta dai sistemi, con una conferma per evitare cancellazioni accidentali.

4. Stato delle Aste:

- a. Le aste vengono ordinate in base alla loro attivazione o scadenza, consentendo una gestione più semplice delle inserzioni.

Interfaccia Utente

- Layout Pulito e Intuitivo:
 - Presentazione chiara delle informazioni principali di ogni asta.
 - Immagini e testi bilanciati per una visualizzazione agevole anche su dispositivi mobili.
- Azioni a Portata di Mano:
 - Icona del cestino sempre visibile per facilitare l'eliminazione.
 - Navigazione immediata alla pagina di dettaglio tramite un semplice clic.

Obiettivi della Pagina delle Aste Attive

Questa pagina è stata progettata per :

- Consentire agli utenti di monitorare rapidamente lo stato delle proprie aste.
- Offrire un accesso diretto ai dettagli e alla gestione di ciascuna asta.
- Semplificare la gestione delle aste attive, riducendo il tempo necessario per visualizzare, controllare o rimuovere un'inserzione.

Questa sezione rappresenta un punto centrale per i venditori, garantendo un controllo semplice ma efficace sulle aste attive e migliorando l'esperienza di gestione complessiva.

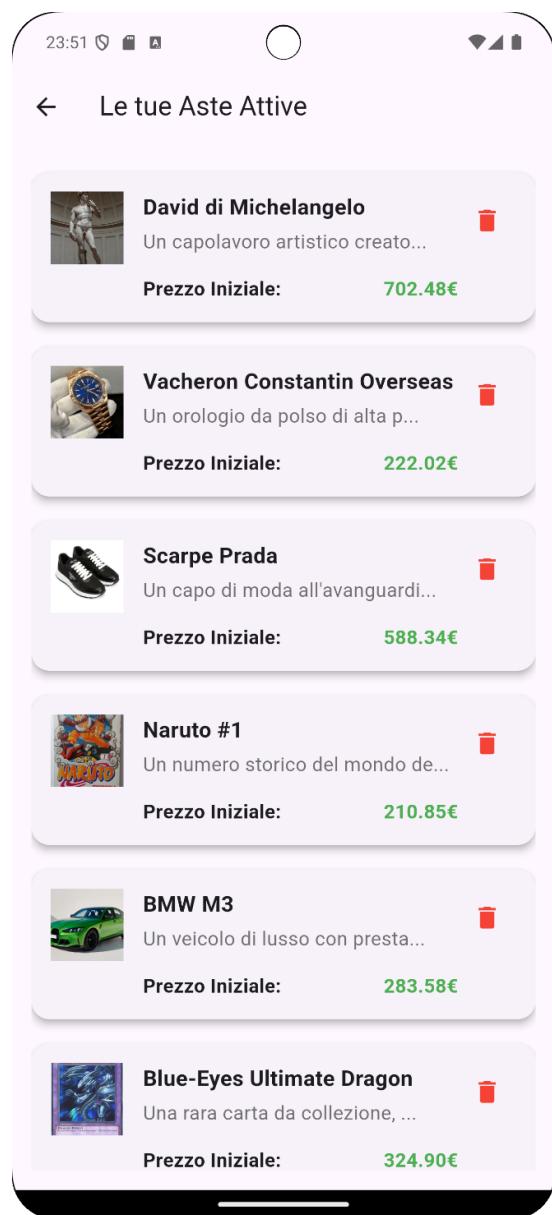


Figure 39: Pagina delle proprie aste attive 1

Figure 40: Pagina delle proprie aste attive 1

Pagina Dettaglio Prodotto :

La **Pagina Dettaglio del Prodotto** offre agli utenti una panoramica completa e interattiva su un prodotto all'asta. È accessibile cliccando su un'asta in qualsiasi elenco o sezione dell'applicazione (ad esempio, **Home**, **Categorie**, **Aste Attive**, ecc.). Questa pagina consente agli utenti di esplorare dettagli importanti sul prodotto, effettuare offerte o visitare la pagina profila del venditore, garantendo un'esperienza intuitiva e funzionale.

Funzionalità Principali

1. Informazioni sul Prodotto:

- **Immagine del prodotto:** Una visualizzazione chiara dell'immagine associata all'asta, caricata dal venditore.
- **Badge Categoria:** Indica la categoria del prodotto (es.: Arte, Gioielli).
- **Badge Tipo di Asta:** Specifica il formato dell'asta (es.: Asta a tempo fisso, Asta all'inglese).
- **Titolo:** Il nome del prodotto, utile per identificarlo facilmente.
- **Descrizione:** Una descrizione dettagliata fornita dal venditore.

2. Offerte e Preferiti:

- **Offerta più alta:** Visualizzazione dell'offerta attuale più alta, sempre aggiornata.
- **Cuore per i preferiti:** L'utente può aggiungere il prodotto ai propri preferiti cliccando sull'icona a forma di cuore.

3. Interazione con l'Asta:

- **Tempo rimanente:** Un timer mostra quanto tempo resta per effettuare offerte.
- **Campo per fare un'offerta:**
 - L'utente può inserire un prezzo maggiore dell'offerta attuale.
 - Se l'offerta è valida, il prezzo si aggiorna con il nuovo valore massimo.
 - Se l'offerta è inferiore o uguale, il sistema non accetta l'importo e mostra un messaggio di errore.
- **Bottone per confermare l'offerta:** Consente di inviare l'offerta, previa verifica della validità dell'importo.

4. Informazioni sul Venditore:

- **Nome del venditore:** Mostra chi sta vendendo il prodotto.
- **Link al profilo del venditore:** L'utente può accedere al profilo del venditore per maggiori dettagli.

5. Costi di Spedizione e Metodi di Pagamento:

- **Costi di spedizione:** Specificati dal venditore per questa asta.
- **Modalità di pagamento:** Elenco dei metodi di pagamento accettati dal venditore.

6. Gestione per il Venditore:

- Se l'utente loggato è lo stesso venditore che ha creato l'asta:
 - Il campo per effettuare un'offerta non viene mostrato, poiché il venditore non può fare offerte sui propri prodotti.

Interfaccia Utente

- **Layout Strutturato:**

La pagina è organizzata per offrire una visualizzazione chiara delle informazioni:

- L'immagine e i badge sono in primo piano.
- Offerte e interazioni sono posizionate sotto le informazioni del prodotto.
- Dettagli sul venditore e condizioni di spedizione/pagamento sono visibili nella parte inferiore.

- **Design Intuitivo:**

- Gli elementi interattivi (campo offerta, pulsante "Fai un'offerta", icona preferiti) sono facilmente accessibili.
- Un sistema di messaggi guida l'utente durante il processo di offerta.

Obiettivi della Pagina

La **Pagina Dettaglio del Prodotto** è progettata per:

- Fornire agli utenti tutte le informazioni necessarie per valutare un prodotto e partecipare all'asta.
- Garantire un'interazione fluida e chiara con il sistema di offerte.
- Offrire ai venditori una gestione semplice delle proprie aste.



Figure 41: Pagina dettaglio prodotto 1



Figure 42: Pagina dettaglio prodotto 2

Pagina Dettaglio Venditore :

La **Pagina Dettaglio del Venditore** consente agli utenti di esplorare le informazioni principali relative a un venditore specifico, insieme alla lista dei suoi prodotti attualmente in vendita. Questa pagina rappresenta una vetrina per il venditore, offrendo una panoramica chiara e organizzata delle sue attività. È accessibile cliccando sul nome o sull'avatar del venditore da una qualsiasi asta.

Funzionalità Principali

1. Informazioni sul Venditore:

- **Avatar del Venditore:** Mostra l'immagine del profilo del venditore, se presente.
- **Nome e Cognome:** I dati identificativi principali del venditore.
- **Bio:** Una breve descrizione fornita dal venditore per presentarsi.



2. Prodotti in Vendita:

- **Lista dei Prodotti:** Visualizza tutti i prodotti attualmente in vendita dal venditore.
 - Ogni prodotto è rappresentato con:
 - **Immagine del prodotto:** Una miniatura per identificare visivamente l'oggetto.
 - **Titolo:** Nome del prodotto in vendita.
 - **Descrizione:** Una breve descrizione del prodotto.
 - **Prezzo:** Il prezzo iniziale o attuale dell'asta.

Interfaccia Utente

• Layout Strutturato:

- La parte superiore della pagina è dedicata alle informazioni personali del venditore (avatar, nome, bio).
- La parte inferiore mostra la lista dei prodotti in vendita, con un design pulito e intuitivo per facilitare la navigazione.

• Design Intuitivo:

- Ogni prodotto è rappresentato con una card o un elemento ben delineato, per migliorare l'esperienza visiva.
- L'avatar e il nome del venditore sono chiaramente visibili nella parte superiore della pagina.

Obiettivi della Pagina

La **Pagina Dettaglio del Venditore** è progettata per:

- Permettere agli utenti di conoscere meglio il venditore e i suoi prodotti.
- Offrire al venditore un mezzo per presentare efficacemente il proprio profilo e gli oggetti in vendita.

Figure 43: Pagina dettaglio del venditore

Sezione Prodotti Non Venduti :

La **Pagina Prodotti Non Venduti** fornisce al venditore un elenco chiaro e dettagliato di tutte le aste concluse che non hanno portato alla vendita del prodotto. Questa pagina permette di identificare rapidamente i motivi per cui un'asta non ha avuto successo, aiutando il venditore a ottimizzare le future inserzioni.

Funzionalità Principali

1. Elenco dei Prodotti Non Venduti:

- Ogni prodotto viene presentato con le seguenti informazioni:
 - **Immagine del Prodotto:** Una miniatura per identificare visivamente l'oggetto.
 - **Titolo:** Nome del prodotto messo all'asta.
 - **Prezzo di Partenza:** Il prezzo iniziale fissato dal venditore al momento della creazione dell'asta.
 - **Motivo del Fallimento:** Una spiegazione del motivo per cui l'asta non ha portato alla vendita:
 - *"Nessuna offerta ricevuta."*
 - *"Prezzo minimo non raggiunto."*

2. Interattività:

- L'utente può cliccare su un prodotto per visualizzare ulteriori dettagli relativi all'asta, compresa una cronologia delle offerte (se presenti) e altre informazioni utili per migliorare future aste.

Interfaccia Utente

• Design Organizzato e Intuitivo:

- La pagina presenta i prodotti in un formato lista, con ogni prodotto ben delineato in una card o sezione separata per migliorare la leggibilità.

• Informazioni Chiare:

- I motivi per cui un prodotto non è stato venduto sono mostrati in modo evidente, consentendo al venditore di comprendere rapidamente le ragioni del fallimento dell'asta.

Obiettivi della Pagina

La **Pagina Prodotti Non Venduti** è pensata per:

- Offrire al venditore una panoramica completa degli oggetti invenduti.
- Fornire feedback immediato sulle aste concluse senza successo.
- Permettere al venditore di analizzare e migliorare le strategie per le aste future.

Nota

Questa pagina rappresenta uno strumento essenziale per il venditore, offrendo la possibilità di migliorare continuamente le performance delle aste grazie all'accesso a informazioni dettagliate e utili.



Figure 44: Pagina dei prodotti non venduti 1



Figure 45: Pagina dei prodotti non venduti 2

Sezione Prodotti Venduti :

La **Pagina dei Prodotti Venduti**, accessibile dalla sezione "**Vendi**" nella pagina "**Account**", consente al venditore di visualizzare tutti i prodotti che sono stati venduti attraverso le aste. Questa pagina offre un riepilogo dettagliato di ciascun prodotto venduto, fornendo trasparenza sulle transazioni concluse e sulle performance delle vendite.

Struttura della Pagina

La pagina è organizzata in una lista di prodotti venduti, ciascuno con i seguenti dettagli:

- **Immagine del Prodotto:** Una rappresentazione visiva del prodotto venduto.
- **Titolo del Prodotto:** Il nome del prodotto venduto.
- **Nome dell'Acquirente:** L'identità dell'utente che ha acquistato il prodotto.
- **Prezzo Finale di Vendita:** L'importo al quale il prodotto è stato venduto.
- **Margine di Guadagno:** Il margine netto ottenuto dal venditore, calcolato rispetto al prezzo di base o di riserva.

Funzionalità

- **Navigazione Intuitiva:**
La lista dei prodotti venduti è strutturata per offrire una panoramica immediata delle transazioni concluse.
- **Dettagli Chiaramente Visibili:**
Tutte le informazioni rilevanti (acquirente, prezzo, margine) sono presentate in modo leggibile e organizzato.

Obiettivi della Pagina

- **Monitoraggio delle Vendite:**
Consente al venditore di verificare le informazioni sulle vendite concluse.
- **Analisi dei Guadagni:**
Fornisce dati utili sul margine di guadagno per ogni prodotto, aiutando il venditore a valutare l'efficacia delle strategie di vendita.
- **Trasparenza:**
Promuove un'esperienza chiara e completa per il venditore, offrendo tutti i dettagli necessari per ogni transazione.



Figure 46: Pagina dei prodotti venduti 1



Figure 47: Pagina dei prodotti venduti 2

Sezione delle Proprie Offerte :

La **Sezione delle Proprie Offerte** consente agli utenti di visualizzare e gestire tutte le offerte effettuate su prodotti in asta. Questa pagina, accessibile tramite la sezione "Acquista" nella pagina "Account", è progettata per fornire un riepilogo chiaro e dettagliato delle offerte, offrendo anche la possibilità di rimuoverle.

Funzionalità Principali

1. Elenco delle Offerte Effettuate:

- Ogni offerta è rappresentata come una card che include:
 - **Immagine del Prodotto:** Una miniatura per identificare facilmente l'articolo.
 - **Titolo del Prodotto:** Il nome associato all'asta.
 - **Descrizione del Prodotto:** Una breve descrizione per fornire contesto sull'oggetto.
 - **Offerta Effettuata:** L'importo offerto dall'utente per l'articolo.

2. Rimozione delle Offerte:

- Gli utenti possono eliminare un'offerta cliccando sull'icona del cestino presente accanto alla card.

3. Interattività:

- Ogni card può essere cliccata per accedere alla pagina di dettaglio del prodotto, consentendo di visualizzare ulteriori informazioni sull'asta e il prodotto correlato.

Interfaccia Utente

• Struttura a Lista:

- Le offerte sono visualizzate in un formato lista per una facile navigazione.
- Gli elementi sono presentati con un design semplice e ben organizzato per garantire chiarezza.

• Icone e Azioni Intuitive:

- L'icona del cestino è ben visibile e chiaramente identificabile, rendendo l'azione di rimozione semplice e diretta.

Obiettivi della Pagina

La **Sezione delle Proprie Offerte** è progettata per:

- Offrire un riepilogo rapido e chiaro delle offerte effettuate dall'utente.
- Consentire agli utenti di gestire e, se necessario, rimuovere le offerte.
- Fornire accesso diretto alla pagina di dettaglio dei prodotti per un approfondimento.



Figure 48: Pagina delle proprie offerte 1



Figure 49: Pagina delle proprie offerte 2

Sezione Ordini Non Pagati / Pagati :

La **Sezione Ordini**, accessibile tramite la sezione "Acquista" nella pagina "**Account**", consente agli utenti di visualizzare e gestire gli ordini relativi alle aste vinte. Questa pagina è organizzata in due sottosezioni principali: **Non Pagato** e **Pagato**, offrendo un'esperienza utente chiara e intuitiva per completare i pagamenti e monitorare lo stato degli ordini.

Struttura della Pagina

1. Ordini Non Pagati

In questa sezione vengono visualizzate tutte le aste vinte ma ancora in attesa di pagamento. Ogni ordine è rappresentato con un riepilogo che include:

- **Titolo del Prodotto:** Il nome dell'oggetto vinto.
- **Descrizione:** Una breve descrizione del prodotto.
- **Stato dell'Ordine:** Mostra lo stato attuale, in questo caso "**In elaborazione**".
- **Totale dell'Offerta:** L'importo vincente dell'asta.

Funzionalità:

- **Pulsante "Paga":**
 - Permette di accedere alla pagina di **Checkout** per completare il pagamento.

2. Ordini Pagati

Questa sezione mostra tutti gli ordini già saldati. Ogni ordine è presentato con un riepilogo contenente:

- **Titolo del Prodotto:** Il nome dell'articolo acquistato.
- **Descrizione:** Una breve descrizione del prodotto.
- **Stato dell'Ordine:** Mostra lo stato attuale, in questo caso "**Pagato**".
- **Totale Pagato:** La somma del totale dell'asta e delle spese di spedizione.

Pagina di Checkout

Cliccando su "**Paga**" da un ordine nella sezione **Non Pagato**, l'utente viene reindirizzato alla **Pagina di Checkout**, dove è possibile completare il pagamento.

Dettagli della Pagina di Checkout:

- **Indirizzo di Spedizione:**
 - Viene precompilato con l'indirizzo di spedizione registrato dall'utente.
- **Costo di Spedizione:**
 - Una tariffa standard di €10 viene aggiunta al totale.
- **Totale da Pagare:**
 - La somma dell'importo vinto nell'asta e del costo di spedizione.
- **Pulsante "Effettua Pagamento":**
 - Simula il pagamento e aggiorna lo stato dell'ordine.

Conferma di Pagamento:

Una volta completato il pagamento, l'ordine viene spostato nella sezione **Pagato**, indicando all'utente che la transazione è avvenuta con successo.

Obiettivi della Pagina

La **Sezione Ordini** è progettata per:

- Fornire un'interfaccia semplice e intuitiva per gestire gli ordini derivanti dalle aste vinte.
- Consentire agli utenti di completare il pagamento degli ordini con chiarezza e facilità.
- Garantire la tracciabilità degli ordini grazie alla suddivisione in **Non Pagato** e **Pagato**.

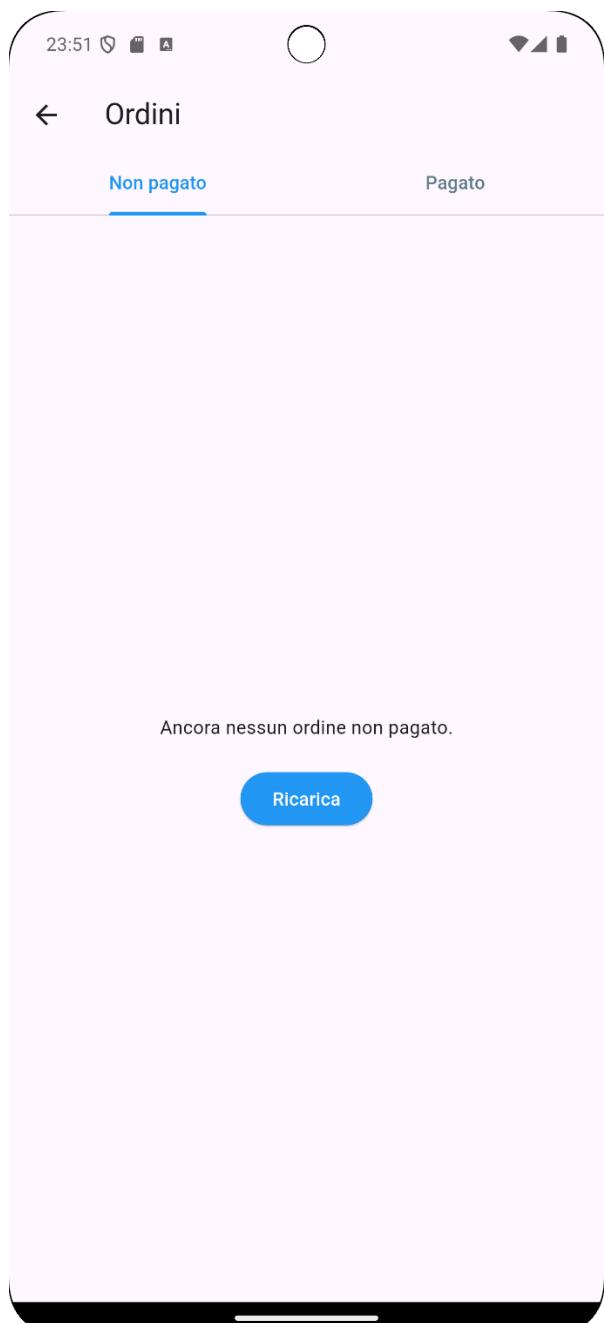


Figure 50: Pagina degli ordini non pagati 1

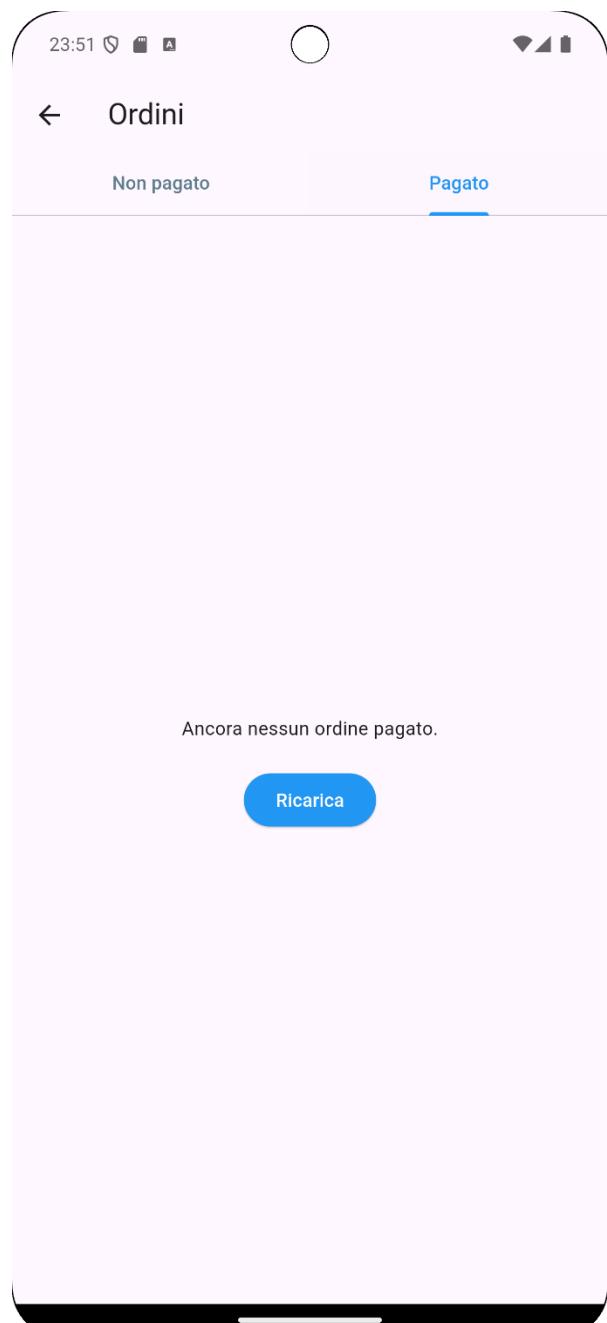


Figure 51: Pagina degli ordini pagati 1



Figure 52: Pagina degli ordini non pagati 2

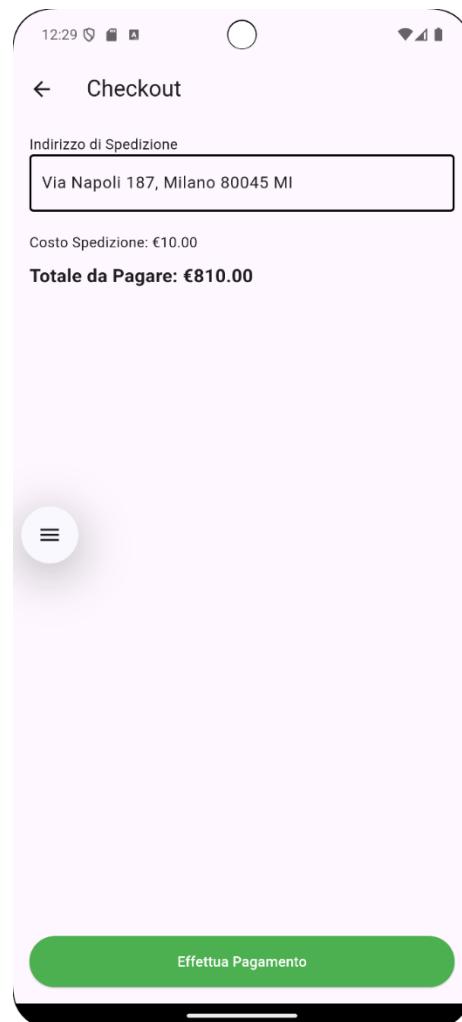


Figure 53: Pagina di checkouts dopo aver cliccato su Paga

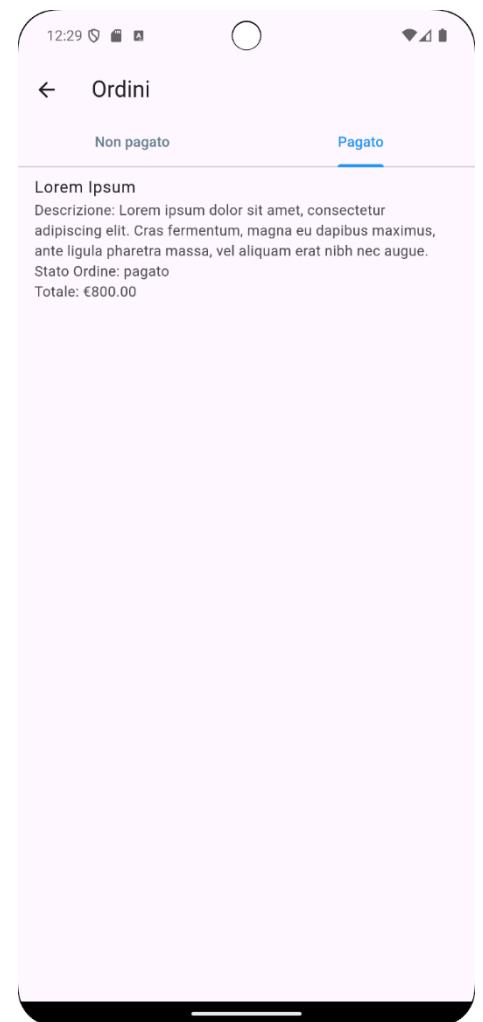


Figure 54: Pagina degli ordini pagati 2

Nota: Nel caso in cui le immagini presentate risultino essere poco chiare, è possibile reperirle in alta qualità, ed eventualmente in formato pdf, nella repository pubblica del progetto sia su Figma sia su GitHub, disponibile al seguente link: [FIGMA](#), [GitHub](#)

VALUTAZIONE DELL'USABILITÀ A PRIORI

Il processo di ingegnerizzazione dell'usabilità stabilisce in anticipo e con precisione le caratteristiche che dovrà avere il software al suo completamento. Questa metodologia si appoggia su una base scientifica e regole ben definite per garantire che il prodotto finale soddisfi i requisiti in modo efficace.

Il termine "ingegnerizzazione" implica l'uso di un metodo sistematico e basato su standard, mentre "usabilità" pone l'accento sull'importanza di progettare un'interfaccia utente intuitiva e accessibile.

Per assicurare un'accurata valutazione dell'usabilità del software, si seguono due fasi fondamentali e complementari:

1. **Valutazione euristica:** un'analisi preliminare condotta dagli esperti di usabilità che fanno parte del team di sviluppo. Questa revisione si basa su criteri prestabiliti di usabilità e si effettua in assenza di feedback diretto da parte degli utenti o degli stakeholder, focalizzandosi sui prototipi di design.
2. **Test di usabilità sul campo:** una valutazione pratica effettuata con un gruppo rappresentativo degli utenti finali. Durante questa fase, gli utenti interagiscono con il software in un ambiente monitorato dagli esperti di usabilità, che osservano e raccolgono le reazioni e i feedback.

Queste due fasi sono essenziali per testare le interfacce utente durante la fase di sviluppo, anche se il prodotto non è ancora finito, permettendo così di valutare l'effettiva implementazione delle funzionalità nei prototipi.

	Registrazione alla piattaforma	Autenticazione alla piattaforma	Cambio password	Effettua ricerca aste attive	Crea un nuovo tipo di asta	Segnalare offerte per tipo di asta
MockUp	X	X	X	X	X	X
Prototipo Interattivo su Figma	X	X	X	X	X	X
Statecharts	X	X	X	X	X	X
Prima implementazione	X	X	X	X	X	X
Beta testing	X	X	X	X	X	X
Applicazione Finale	X	X	X	X	X	X

Nell'ambito dello sviluppo del software, la valutazione della funzionalità complessiva dei prototipi è cruciale. Gli specialisti in usabilità esaminano i prototipi durante la fase di design, utilizzando diagrammi di stato per comprendere le interazioni del sistema. Essi controllano anche la conformità del prodotto a standard riconosciuti di usabilità, come le otto regole fondamentali di Ben Shneiderman o le dieci euristiche di Nielsen.

Per esempio, nel valutare un caso d'uso in cui un utente registrato crea una nuova asta, la consistenza e lo standard dell'interfaccia sono critici. Elementi familiari, come icone di navigazione universali e l'uso intuitivo del colore, semplificano l'esperienza dell'utente. La chiarezza dei messaggi di errore e la prevenzione degli stessi attraverso campi obbligatori aiutano a guidare l'utente senza intoppi.

Valutazione del caso d'uso: un utente registrato crea una nuova asta

- **Coerenza e Standardizzazione dell'Interfaccia:** La familiarità degli utenti con elementi comuni di interfaccia, come icone a forma di freccia per indicare la navigazione "indietro", facilita l'uso dell'applicazione. L'uso intuitivo dei colori, con il verde che segnala operazioni riuscite e il rosso che evidenzia errori, contribuisce a un'esperienza utente fluida e comprensibile.
- **Supporto all'utente per la diagnosi e correzione degli errori:** I messaggi di errore appaiono quando l'utente prova a caricare una foto con un formato non supportato. Questi messaggi sono espressi in un linguaggio chiaro e offrono indicazioni precise su come risolvere l'errore, consentendo all'utente di procedere senza intoppi.
- **Prevenzione degli errori:** L'applicazione impedisce in modo proattivo la finalizzazione della creazione di una nuova asta se tutti i campi obbligatori, come il titolo dell'asta, la foto, e il prezzo base, non sono stati compilati, prevenendo così errori di creazione.
- **Flessibilità ed Efficienza d'Uso:** Tuttavia, si nota una carenza nell'adattabilità dell'interfaccia per utenti avanzati, dovuta alla mancanza di scorciatoie per operazioni rapide. L'introduzione di funzionalità di auto-completamento durante l'immissione dei dati potrebbe aumentare significativamente l'efficienza per gli utenti esperti.

Tuttavia, si è notata una mancanza di funzionalità avanzate per gli utenti esperti, come le scorciatoie e l'auto-completamento, che potrebbero migliorare l'efficienza.

Valutazione del caso d'uso: un utente non registrato si iscrive alla piattaforma di aste online BidHub usando email e password.

- **Autonomia e controllo dell'utente:** Per assistere gli utenti meno esperti che potrebbero interagire accidentalmente con l'interfaccia, è stata apprezzata l'introduzione di pulsanti di navigazione che permettono di passare facilmente tra le schermate di login e registrazione, garantendo così un'esperienza utente confortevole e senza stress.
- **Adesione alla realtà nella rappresentazione del sistema:** I messaggi popup, sia di conferma sia di errore, che si manifestano durante il processo di registrazione sono formulati in modo chiaro e intuitivo, usando termini e concetti già noti all'utente, rispecchiando così un'eccellente pratica di design nell'interfaccia utente.
- **Design minimalista ed estetico:** Considerando che BidHub è una piattaforma di aste online, l'uso di un design pulito e di colori neutri, che evocano serietà e professionalità, risulta in linea con l'ambiente di aste online, rispettando i requisiti visivi e funzionali del prodotto.
- **Visibilità dello stato del sistema:** La piattaforma deve mantenere gli utenti pienamente informati durante ogni fase dell'interazione. Un punto debole riscontrato è la mancanza di indicazioni preliminari sulla necessità di una password complessa, come quella di otto caratteri con requisiti specifici. Per migliorare, sarebbe opportuno implementare istruzioni visibili o suggerimenti dinamici che guidino l'utente nella creazione di una password valida durante la fase di registrazione.

La fase di valutazione dell'usabilità si conclude con la creazione di un diagramma in cui ogni valutatore e i problemi di usabilità identificati sono mappati. L'asse verticale elenca i valutatori, mentre l'asse orizzontale organizza i problemi di usabilità in base alla loro evidenza. Un quadrato nero all'intersezione indica che un valutatore ha rilevato un problema, fornendo una visualizzazione chiara della distribuzione e della frequenza dei problemi riscontrati.

	•					ESPERTO 4	VALUTATORI
•		•			•	ESPERTO 3	
				•		ESPERTO 2	
	•	•	•			ESPERTO 1	
Errore generico durante la creazione dell'asta	L'utente non inserisce tutti i dati obbligatori	L'utente seleziona un file che non è una immagine	L'utente seleziona una data errata	L'utente inserisce un prezzo iniziale errato	Il sistema non riesce a caricare la nuova asta perché l'utente non ha concesso i permessi necessari		
PROBLEMI DI USABILITÀ'							

	•			•		ESPERTO 4	VALUTATORI
•		•		•		ESPERTO 3	
	•		•		•	ESPERTO 2	
			•			ESPERTO 1	
Errore generico durante la creazione dell'asta	L'utente non inserisce tutti i dati obbligatori	L'utente inserisce data di nascita non reale	L'utente inserisci indirizzo e-mail non valido	L'utente non seleziona l'opzione venditore o acquirente	L'utente inserisce una password poco sicura		
PROBLEMI DI USABILITÀ'							

GLOSSARIO

Questo glossario fornisce una panoramica dei termini tecnici e dei concetti chiave utilizzati all'interno della documentazione e della piattaforma di gestione delle aste online. L'obiettivo è di garantire una comprensione uniforme delle terminologie tra tutti gli utenti, dagli sviluppatori ai gestori del sistema e agli utenti finali.

Ogni termine è accompagnato da una definizione chiara e sintetica, con eventuali esempi o contesti applicativi per facilitare l'assimilazione del concetto. Il glossario è particolarmente utile per chi si approccia per la prima volta al sistema o per chi desidera approfondire il significato di specifici elementi funzionali.

Termine	Definizione
Asta	Una procedura in cui un prodotto è offerto in vendita al miglior offerente.
Ordine	Una richiesta effettuata da un acquirente per acquistare un prodotto vinto in un'asta.
Offerta (Bid)	Un valore proposto da un utente per partecipare a un'asta e acquistare un prodotto.
Checkout	Il processo di finalizzazione di un ordine, includendo il pagamento e l'indirizzo di spedizione.
Pagamento	L'azione di trasferire denaro per completare un ordine. Può avvenire tramite carte o altri metodi.
Provider	Una classe che fornisce dati e logica alle interfacce utente tramite il framework Provider di Flutter.
Notifica	Un messaggio informativo inviato all'utente, ad esempio per segnalare un'asta vinta o scaduta.
Stato dell'Ordine	Lo stato attuale di un ordine (es. "in elaborazione", "pagato").
Token	Un identificatore sicuro usato per autenticare un utente durante le richieste alle API.
API (Application Programming Interface)	Un'interfaccia che consente al frontend di comunicare con il backend per scambiare dati.
Utente	Una persona che interagisce con il sistema, sia per fare offerte, acquistare prodotti, che per vendere articoli durante un'asta. L'utente può avere diversi ruoli, come compratore, venditore.
BidHub	Piattaforma di aste online progettata per offrire un'esperienza moderna e accessibile a chiunque desideri partecipare ad aste globali.

BidHunter	Termine utilizzato per indicare gli utenti appassionati di aste che utilizzano la piattaforma BidHub per scoprire, seguire e aggiudicarsi oggetti unici o preziosi.
------------------	---

SPECIFICA DEI REQUISITI

La sezione "Specifiche dei Requisiti" ha l'obiettivo di fornire una descrizione dettagliata delle funzionalità e delle caratteristiche tecniche richieste dal sistema, traducendo le necessità emerse durante l'analisi in requisiti concreti da implementare. Questo documento è fondamentale per garantire che tutte le parti coinvolte nel progetto abbiano una comprensione comune delle aspettative e delle modalità di realizzazione del software.

In questa parte del documento, verranno analizzati i principali oggetti, classi e le loro relazioni, nonché le modalità di interazione con il sistema. Inoltre, saranno forniti diagrammi di sequenza che illustrano il flusso delle operazioni e la comunicazione tra gli attori e il sistema in vari scenari di utilizzo. La specifica dei requisiti servirà da guida per la fase di progettazione e sviluppo, assicurando che il sistema soddisfi appieno le necessità degli utenti e gli obiettivi del progetto.

La sezione si articola in:

- **Classi, oggetti e relazioni di analisi:** descrizione dei principali componenti del sistema e delle loro interazioni.
- **Diagrammi di sequenza di analisi:** illustrazione dei flussi di dati e delle operazioni che avvengono durante le interazioni tra gli utenti e il sistema.
- **Prototipazione funzionale e progettazione:** indicazioni su come le funzionalità saranno strutturate e realizzate.

CLASSI, OGGETTI E RELAZIONI DI ANALISI

La sezione "Classi, Oggetti e Relazioni di Analisi" si focalizza sull'identificazione delle entità principali del sistema e sulla definizione delle loro caratteristiche e relazioni, al fine di modellare in modo chiaro e strutturato il dominio applicativo. Questo lavoro rappresenta un passaggio fondamentale per tradurre i requisiti funzionali in una rappresentazione concettuale che possa guidare le successive fasi di progettazione e sviluppo.

Le classi e gli oggetti individuati rappresentano i concetti fondamentali del sistema, come utenti, prodotti, offerte e ordini, mentre le relazioni tra di essi descrivono le modalità di interazione e dipendenza. L'obiettivo principale di questa analisi è garantire che tutte le entità rilevanti siano correttamente rappresentate e che le loro interazioni supportino pienamente i casi d'uso previsti.

In particolare, i **casi d'uso** saranno analizzati e tradotti attraverso i diagrammi delle classi utilizzando il modello **Entity-Boundary-Controller (EBC)**. Questo approccio consente di suddividere logicamente le responsabilità delle classi:

- **Entity:** rappresenta i dati e la logica di business associata.
- **Boundary:** gestisce l'interazione tra il sistema e gli attori esterni, come utenti o sistemi terzi.
- **Control:** coordina le operazioni tra boundary ed entity per realizzare il flusso dei casi d'uso.

Questa sezione include:

- **Identificazione delle classi principali:** descrizione delle classi fondamentali, con i loro attributi e comportamenti principali.

- **Relazioni tra le classi:** analisi delle associazioni, aggregazioni, composizioni ed ereditarietà per rappresentare le dipendenze tra le entità del sistema.
- **Diagrammi EBC per i casi d'uso:** rappresentazione grafica che illustra come le responsabilità vengono distribuite tra entity, boundary e controller per implementare le funzionalità richieste.
- **Modello concettuale del dominio:** una vista d'insieme che sintetizza le entità, le loro relazioni e il contesto operativo del sistema.

Questo approccio metodologico assicura una chiara separazione delle responsabilità tra i vari componenti, promuovendo un design modulare e facilmente manutenibile, oltre a garantire che il sistema sia pienamente conforme ai requisiti emersi durante la fase di analisi.

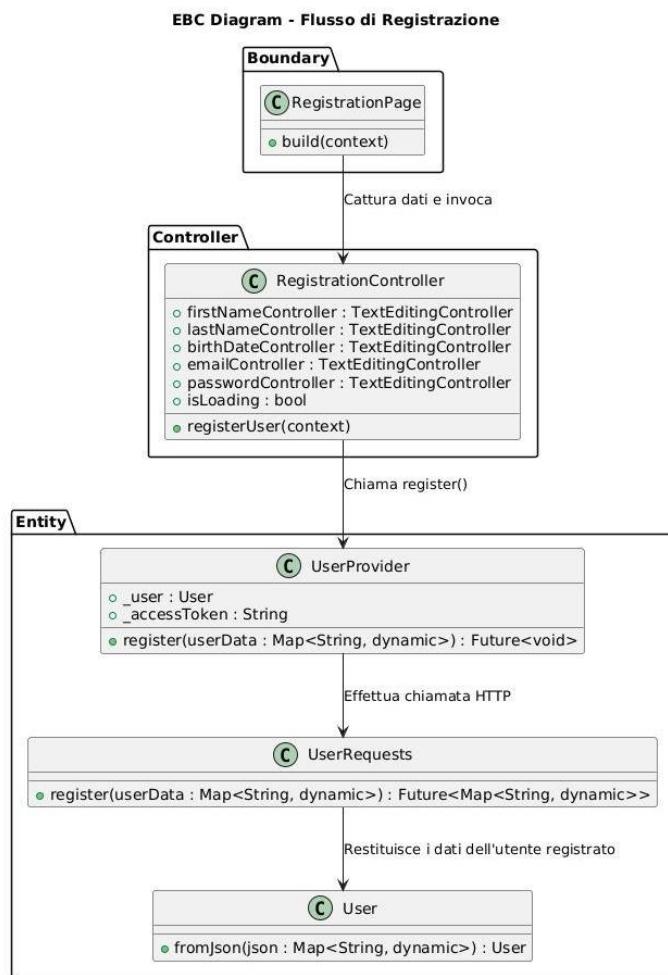


Figura 55: EBC – Registrazione

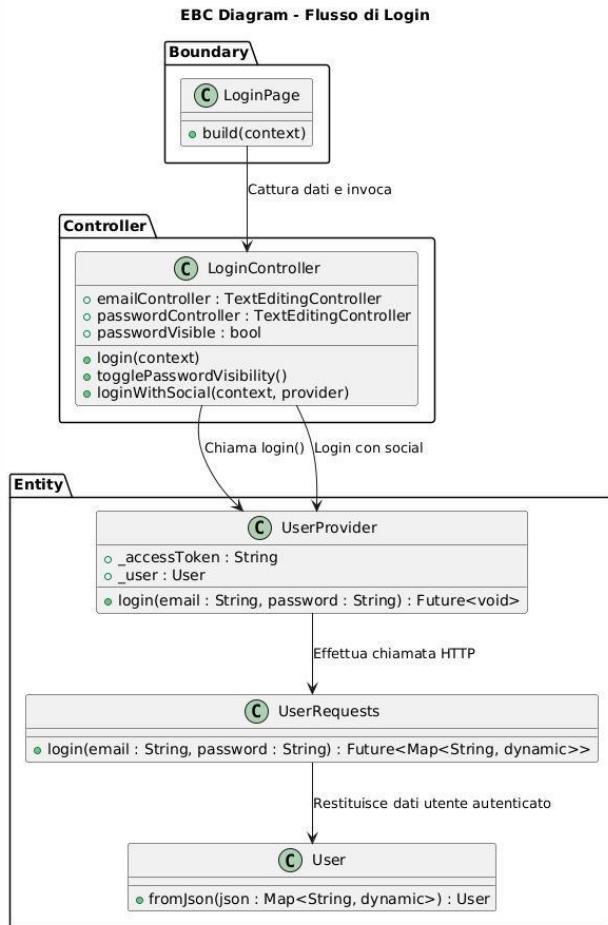


Figura 56: EBC – Login

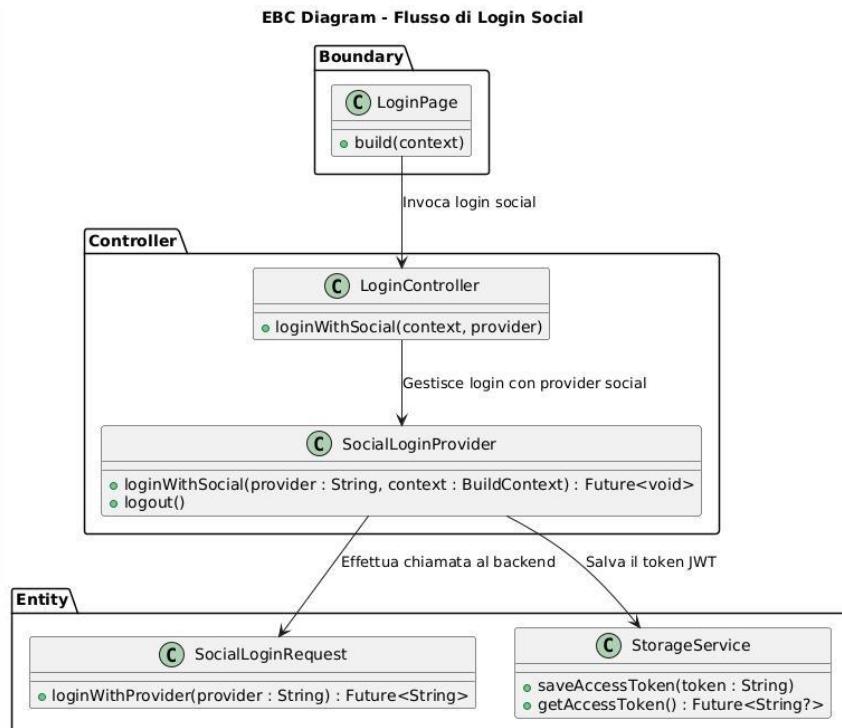


Figura 57: EBC - Login con Social Network

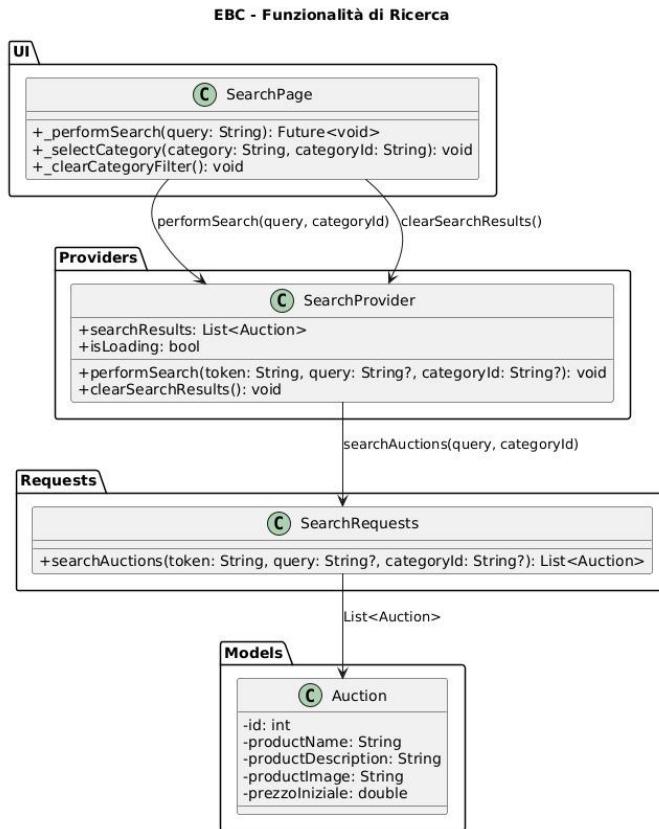


Figura 58: EBC – Ricerca

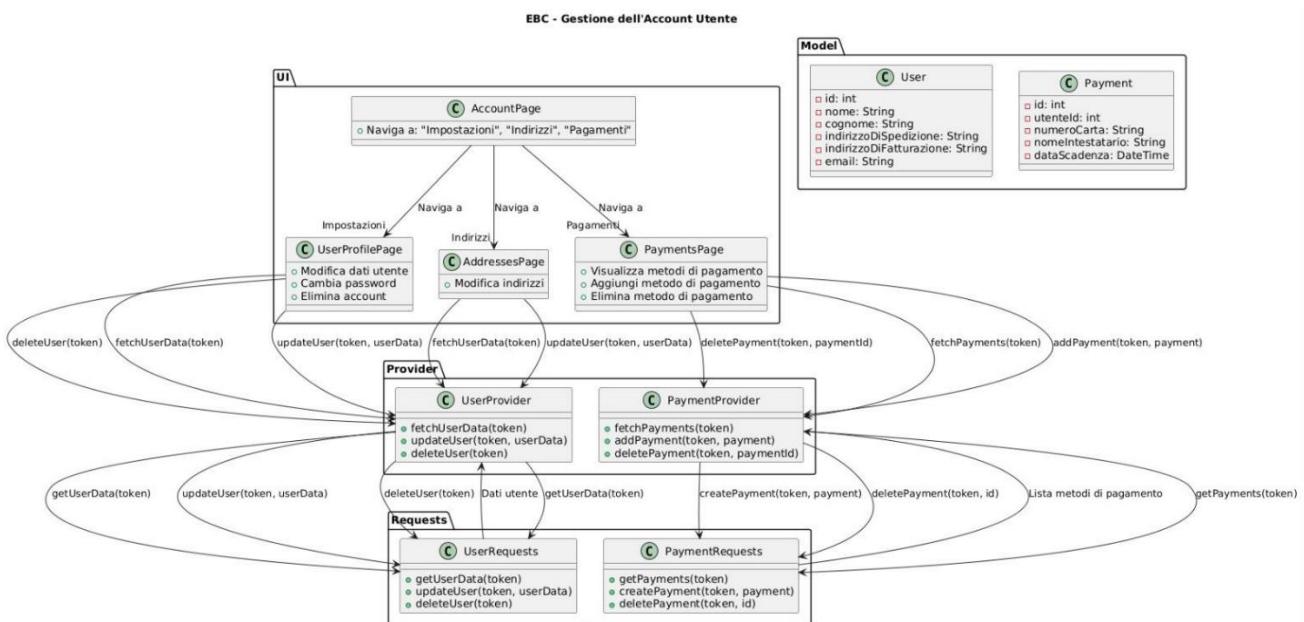


Figura 59: EBC - Gestione Account

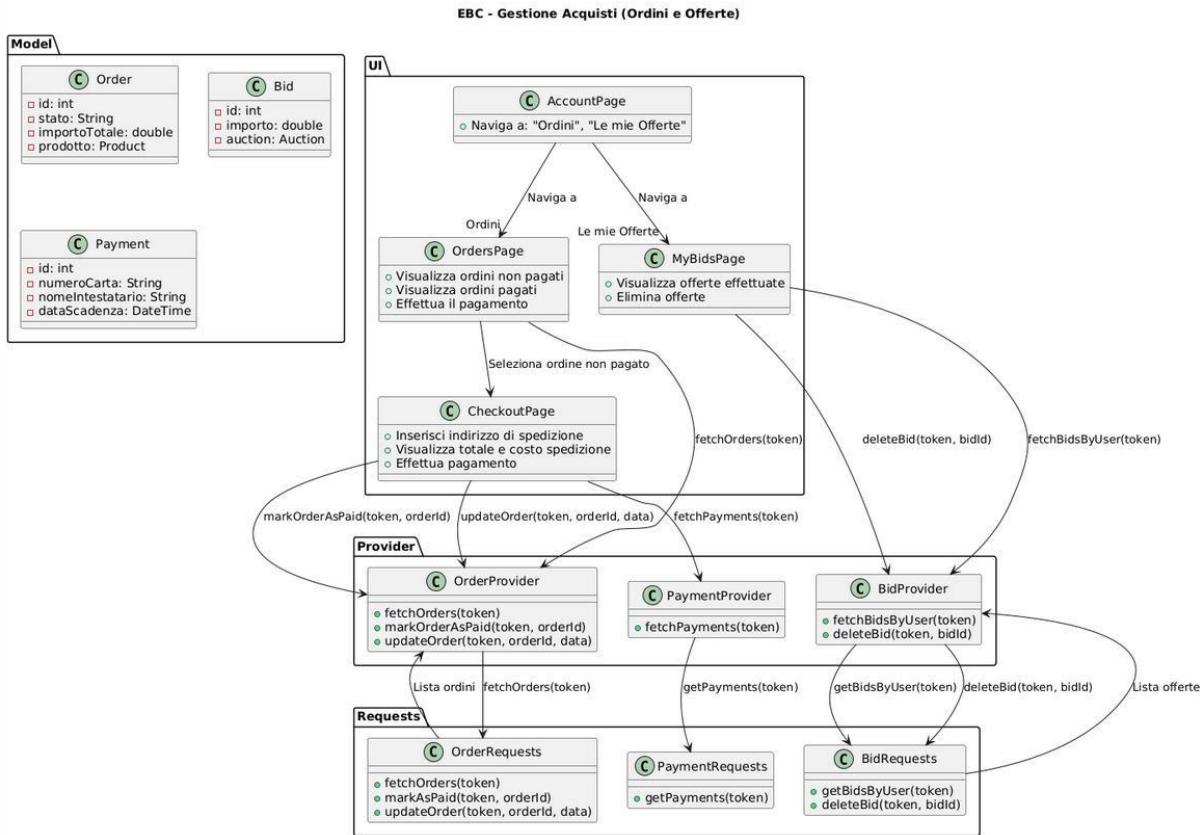


Figura 60: Gestione Acquisti

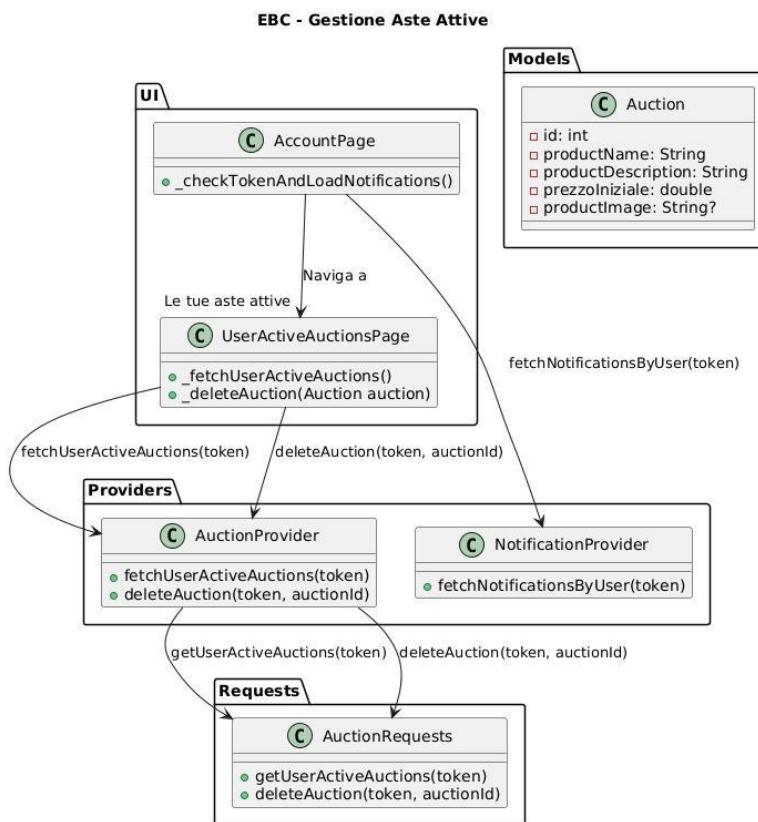


Figura 61: Gestione Aste Attive

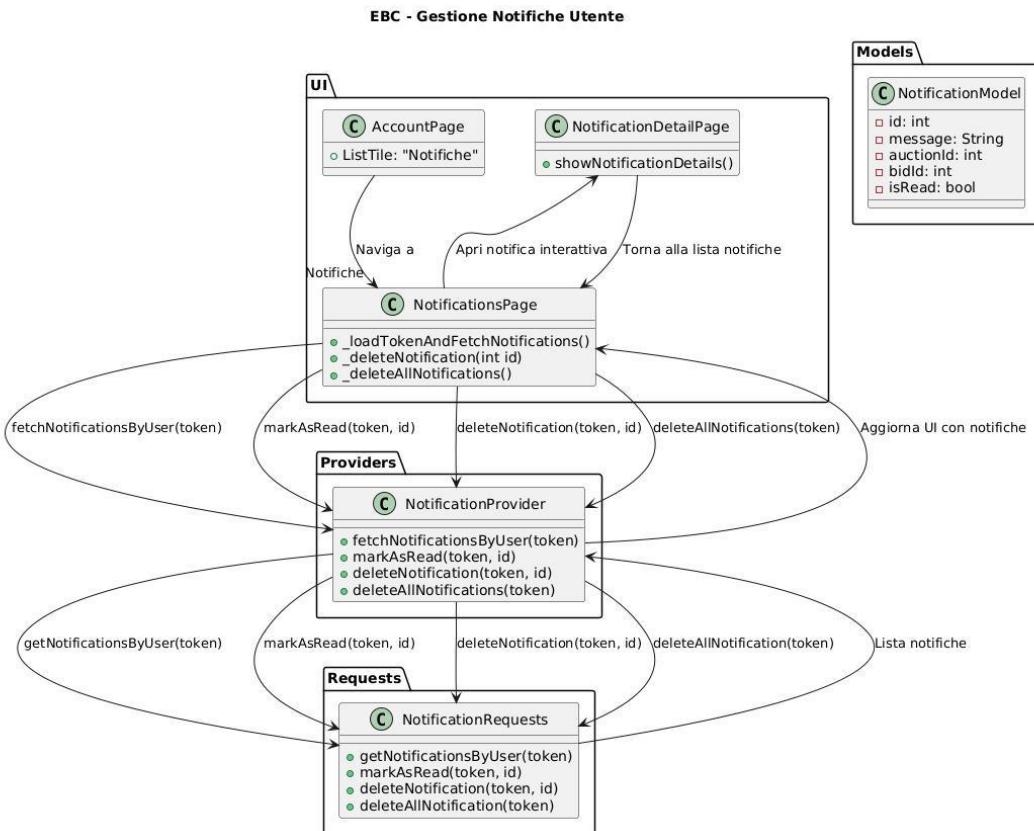


Figura 62: Gestione Notifiche Utente

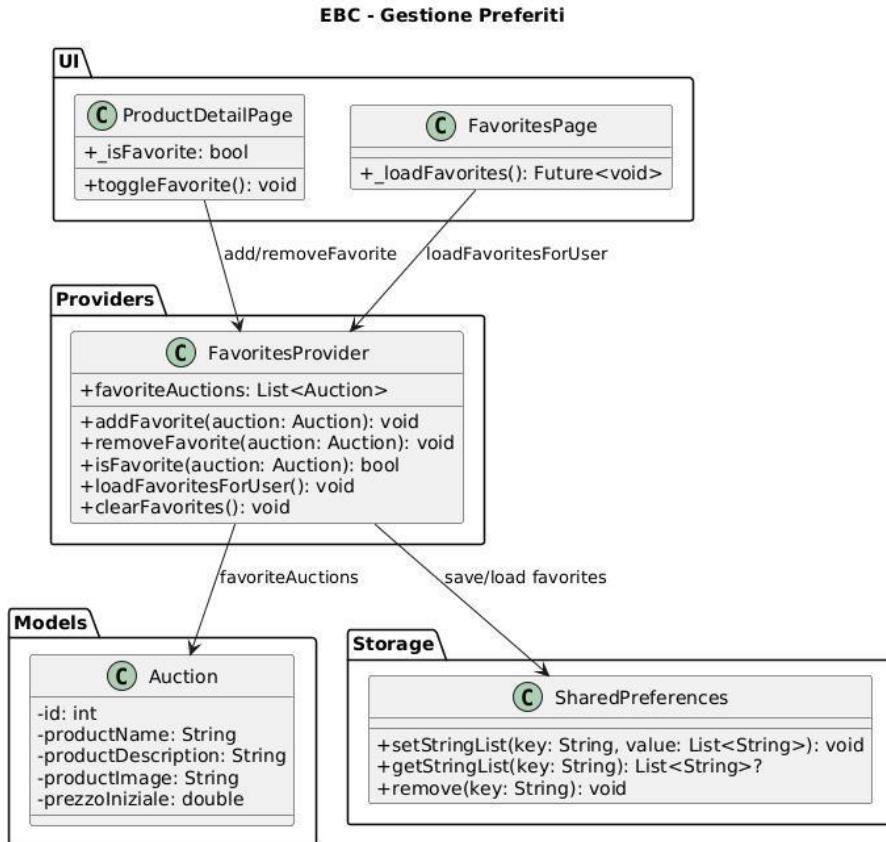


Figura 63: Gestione Preferiti

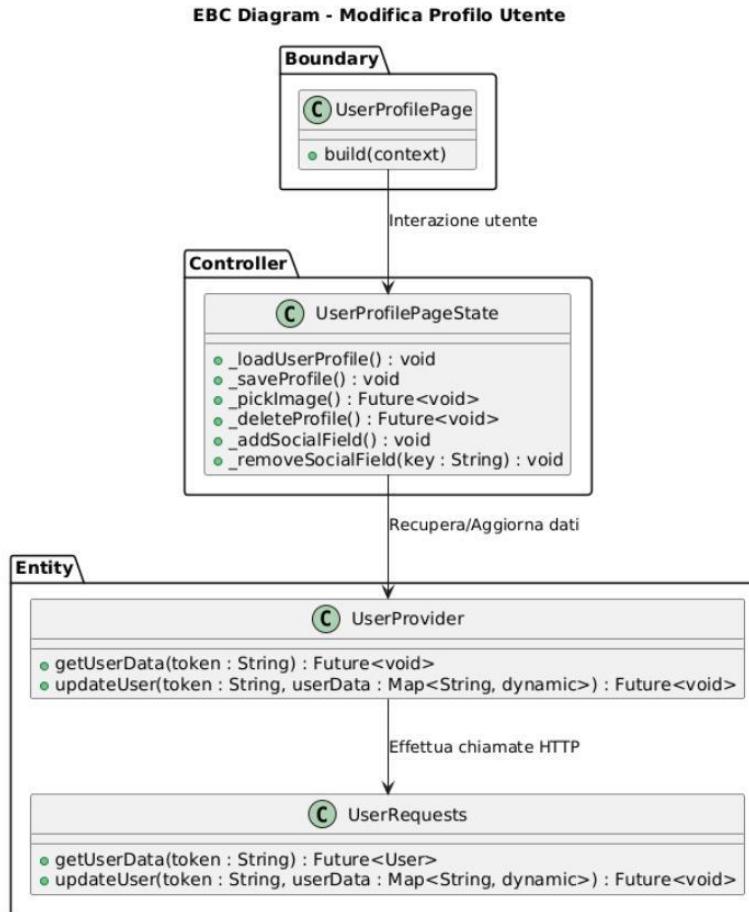


Figura 64: Modifica Profilo Utente

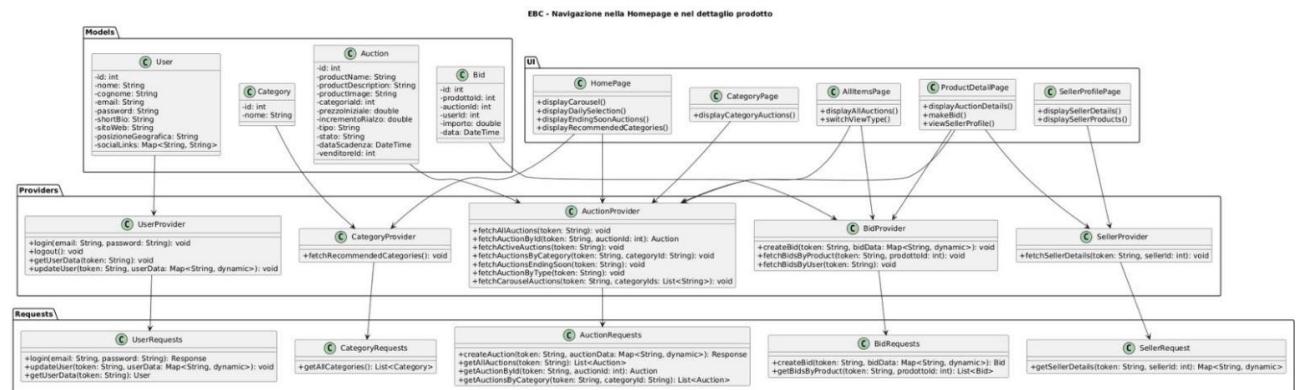


Figura 65: Homepage e Dettaglio Prodotto

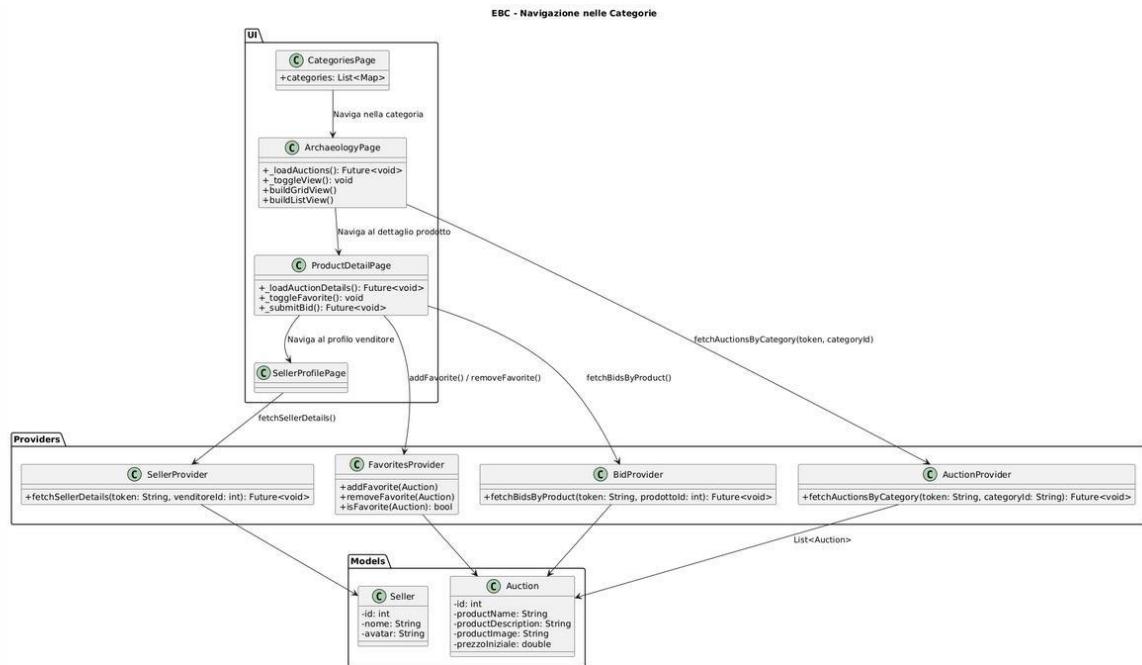


Figura 66: Navigazione e Scelta Categorie

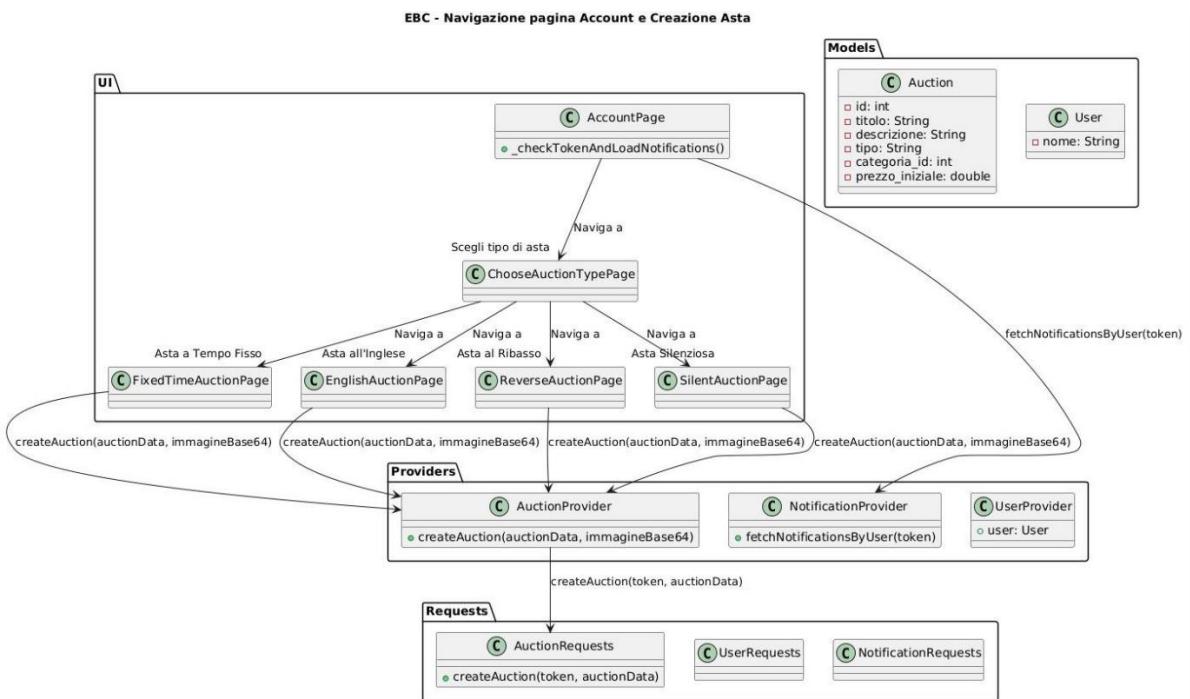


Figura 67: Pagina Account e Creazione Asta

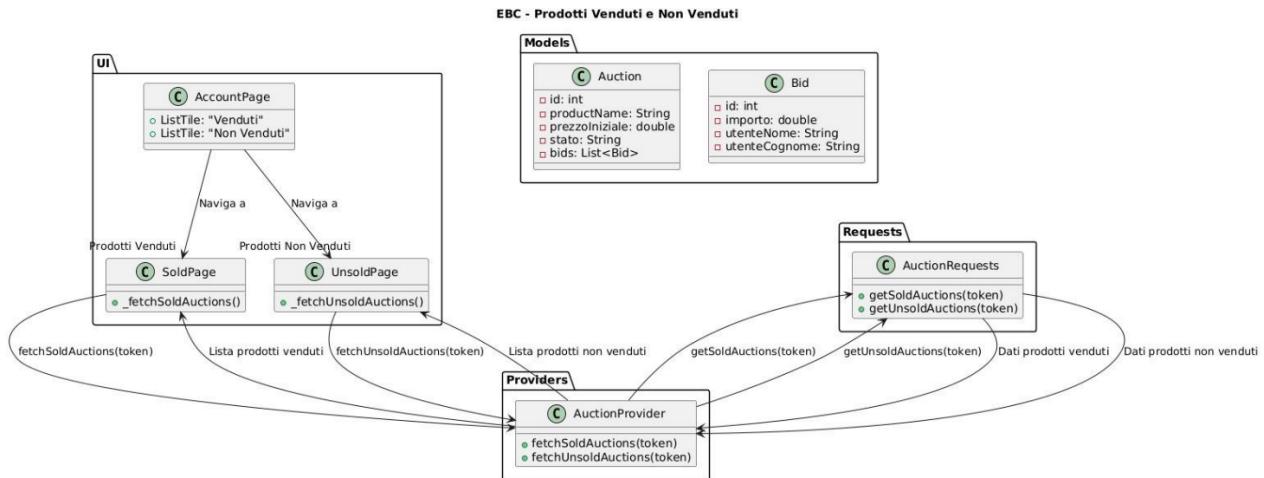


Figura 68: Prodotti Venduti e Non Venduti

EBC Diagram - Recupero Password

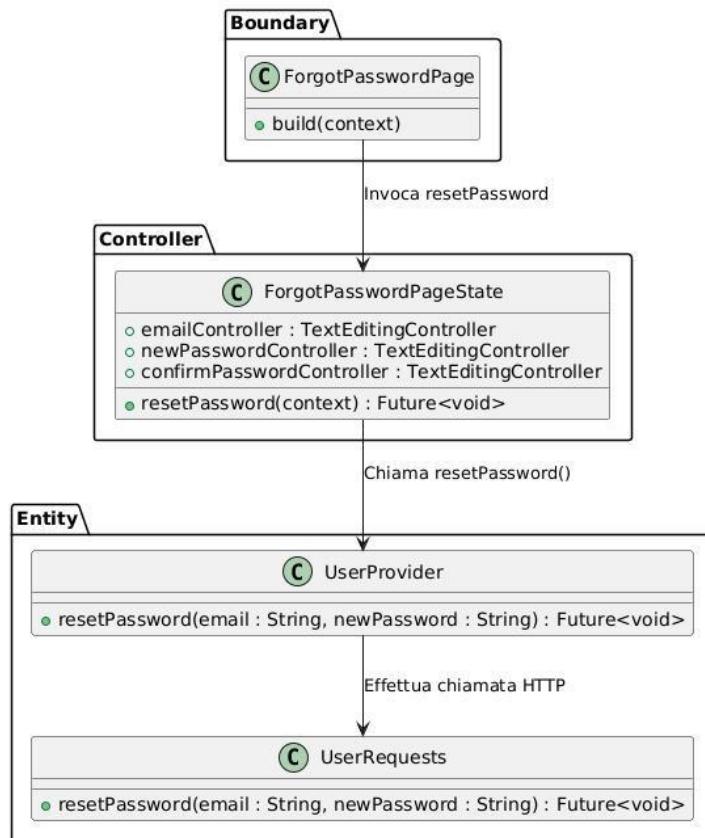


Figura 69: Recupero Password

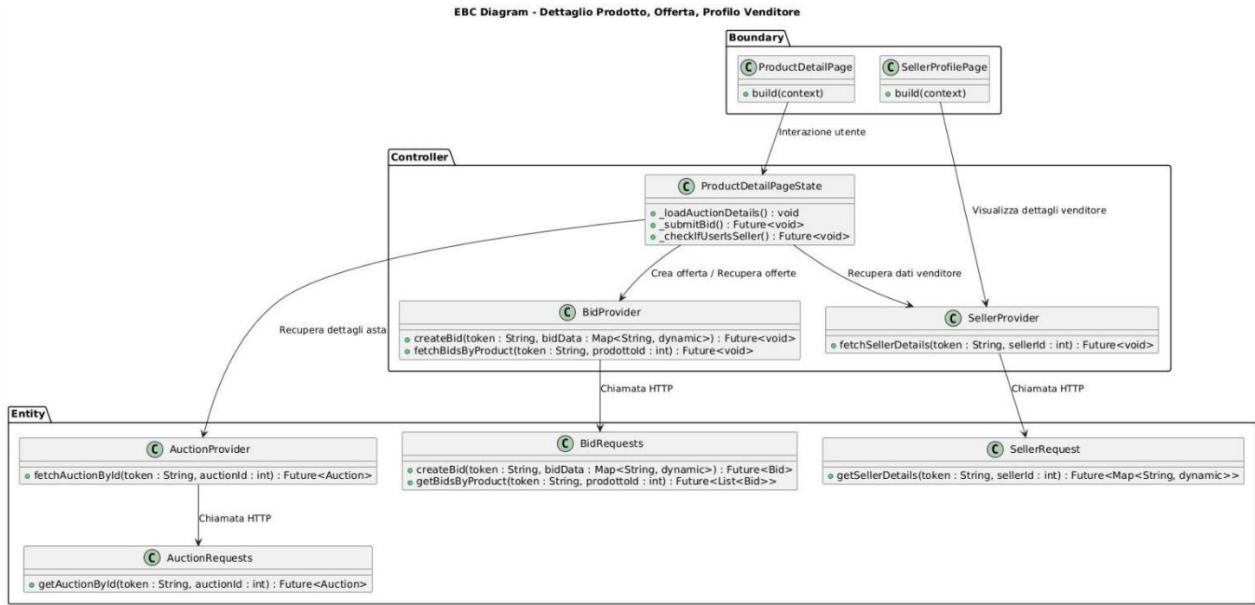


Figura 70: Dettaglio Prodotto, Offerta e Profilo Venditore

Nota: Nel caso in cui le immagini presentate risultino essere poco chiare, è possibile reperirle in alta qualità, ed eventualmente in formato pdf, nella repository pubblica del progetto su GitHub, disponibile al seguente link: [GitHub](#)

DIAGRAMMI DI SEQUENZA DI ANALISI

Per completare la descrizione del sistema di gestione aste online, i diagrammi di sequenza e il modello di dominio forniscono una rappresentazione dettagliata delle interazioni tra i componenti del sistema e della struttura concettuale dell'applicazione.

I diagrammi di sequenza mostrano come gli attori (utenti o sistemi esterni) interagiscono con il sistema nel contesto dei principali casi d'uso. Viene rappresentato il flusso cronologico delle operazioni, evidenziando:

- **Gli attori** coinvolti nel processo.
- **I messaggi scambiati** tra gli oggetti del sistema e gli attori.
- **L'ordine temporale** delle interazioni.

Questi diagrammi aiutano a comprendere in modo chiaro come il sistema risponde alle richieste degli utenti, mostrando il comportamento dinamico dell'applicazione. Tra i processi rappresentati nei diagrammi di sequenza, troviamo la **registrazione al sistema**, la **creazione di aste**, la **gestione delle offerte** e la **chiusura delle transazioni**.

Il modello di dominio rappresenta la struttura concettuale dell'applicazione, descrivendo le classi principali, i loro attributi e le relazioni tra di esse. Questo modello costituisce la base per lo sviluppo del software, garantendo una corrispondenza tra il dominio del problema e l'implementazione tecnica.

Nel contesto del sistema di gestione aste, il modello di dominio comprende:

- **Classi principali** come *Utente*, *Asta*, *Offerta*, e *Transazione*.
- **Relazioni** tra le classi, ad esempio, la relazione tra un *Utente* (come venditore o acquirente) e le *Aste* create o a cui partecipa.
- **Attributi e metodi** essenziali per rappresentare il comportamento del sistema, come lo stato dell'asta o il valore di un'offerta.

Questa sezione permette di passare dalla fase di analisi a quella di progettazione, assicurando che tutte le funzionalità identificate nei casi d'uso siano correttamente mappate nella struttura e nel comportamento del sistema.

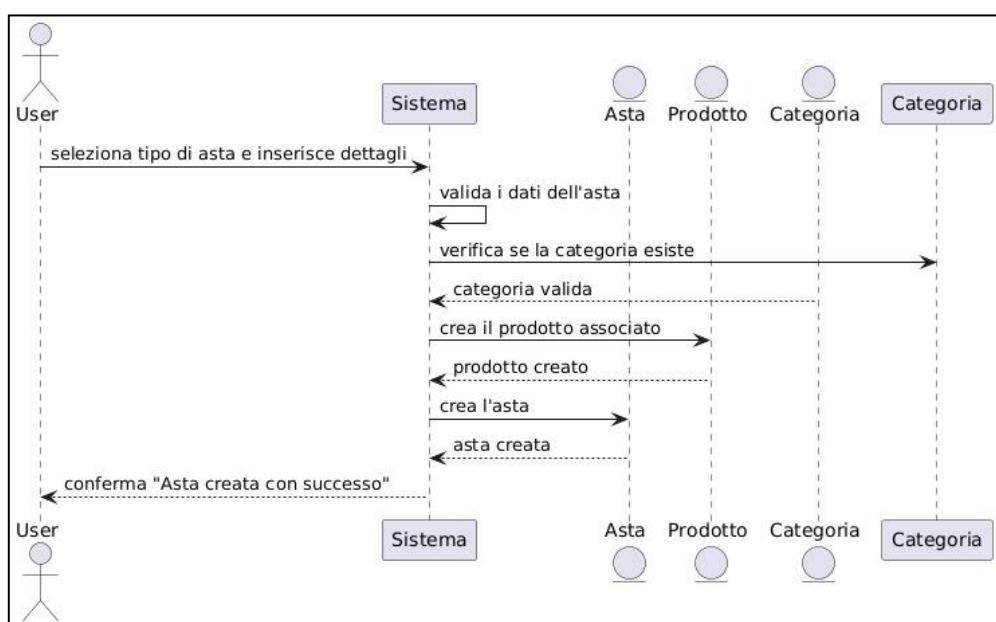


Figure 71: Sequence Diagram di dominio - Creazione asta

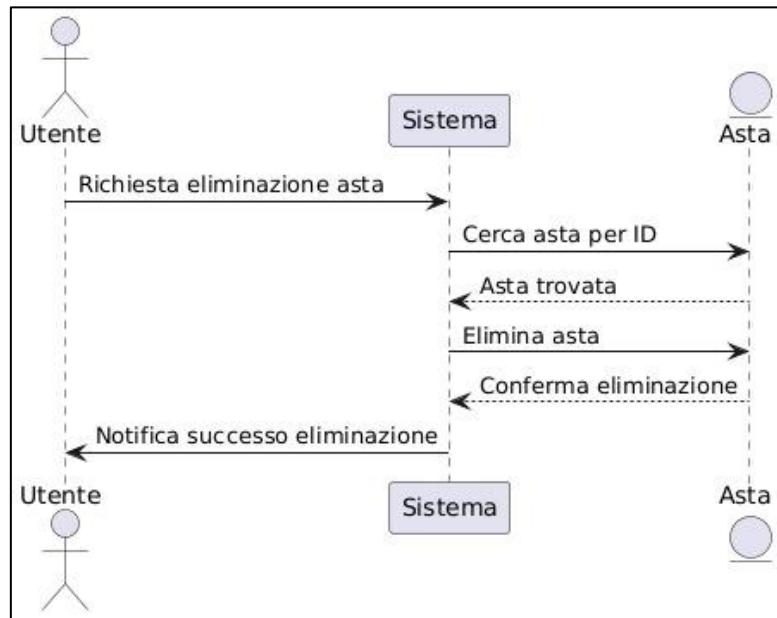


Figure 72: Sequence Diagram di dominio - Richiesta eliminazione asta

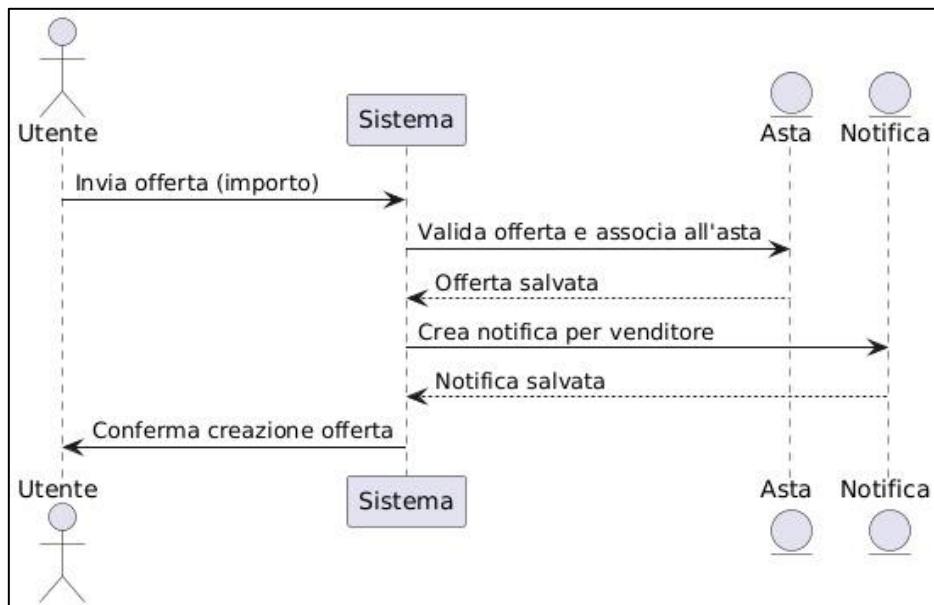


Figure 73: Sequence Diagram di dominio - Invio offerta e ricezione notifica

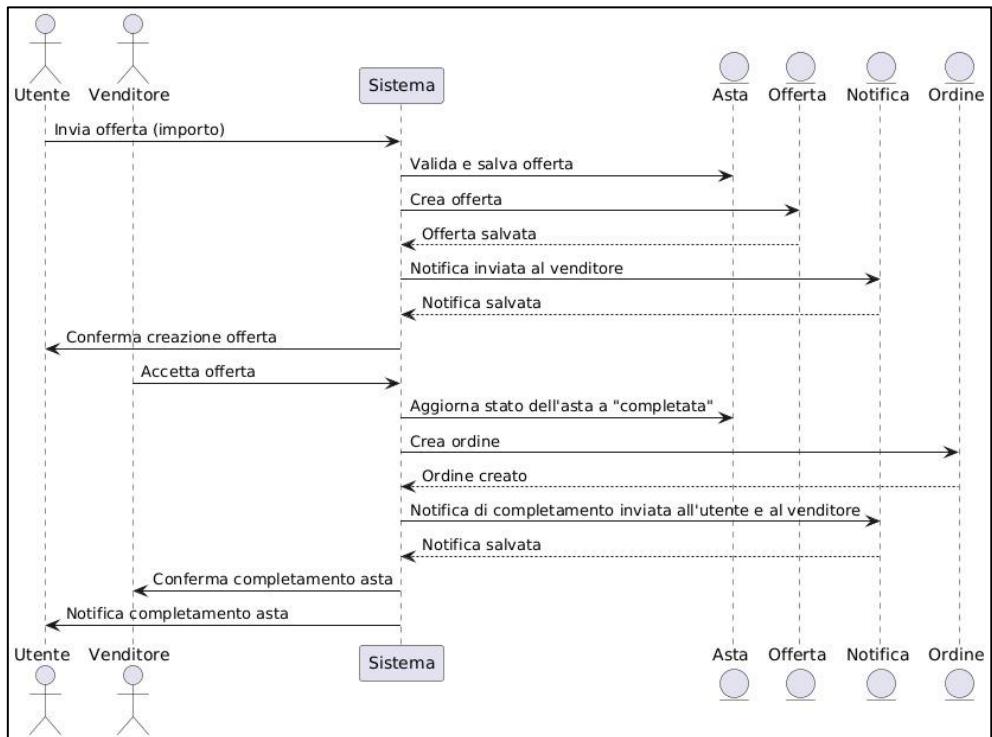


Figure 74: Sequence Diagram di dominio -

PROTOTIPAZIONE FUNZIONALE E PROGETTAZIONE DEGLI EVENT-BASED STATECHART

La prototipazione funzionale è una fase essenziale nello sviluppo di sistemi software complessi, in quanto permette di creare una rappresentazione preliminare e operativa del sistema. Questo processo consente di testarne il comportamento e di verificarne la conformità rispetto ai requisiti definiti, riducendo così il rischio di discrepanze tra progettazione e implementazione finale.

In tale contesto, l'adozione degli **Event-Based Statechart** si rivela particolarmente efficace. Questi strumenti offrono un modello formale e visuale che consente di descrivere, in modo chiaro e strutturato, il ciclo di vita degli stati di un sistema dinamico e le transizioni guidate da eventi.

Questa sezione approfondisce due aspetti centrali:

1. **Prototipazione funzionale**: un approccio iterativo che consente di testare e validare le funzionalità principali, favorendo una progettazione più affidabile e allineata agli obiettivi.
2. **Event-Based Statechart**: un metodo per modellare stati ed eventi che permette di rappresentare graficamente la logica del sistema, facilitando l'individuazione di incongruenze e possibili ottimizzazioni.

L'integrazione di queste metodologie garantisce una rappresentazione chiara e verificabile del comportamento del sistema, agevolando il dialogo tra progettisti, sviluppatori e stakeholder e migliorando l'efficacia complessiva del processo di sviluppo.

1. CREAZIONE ASTA

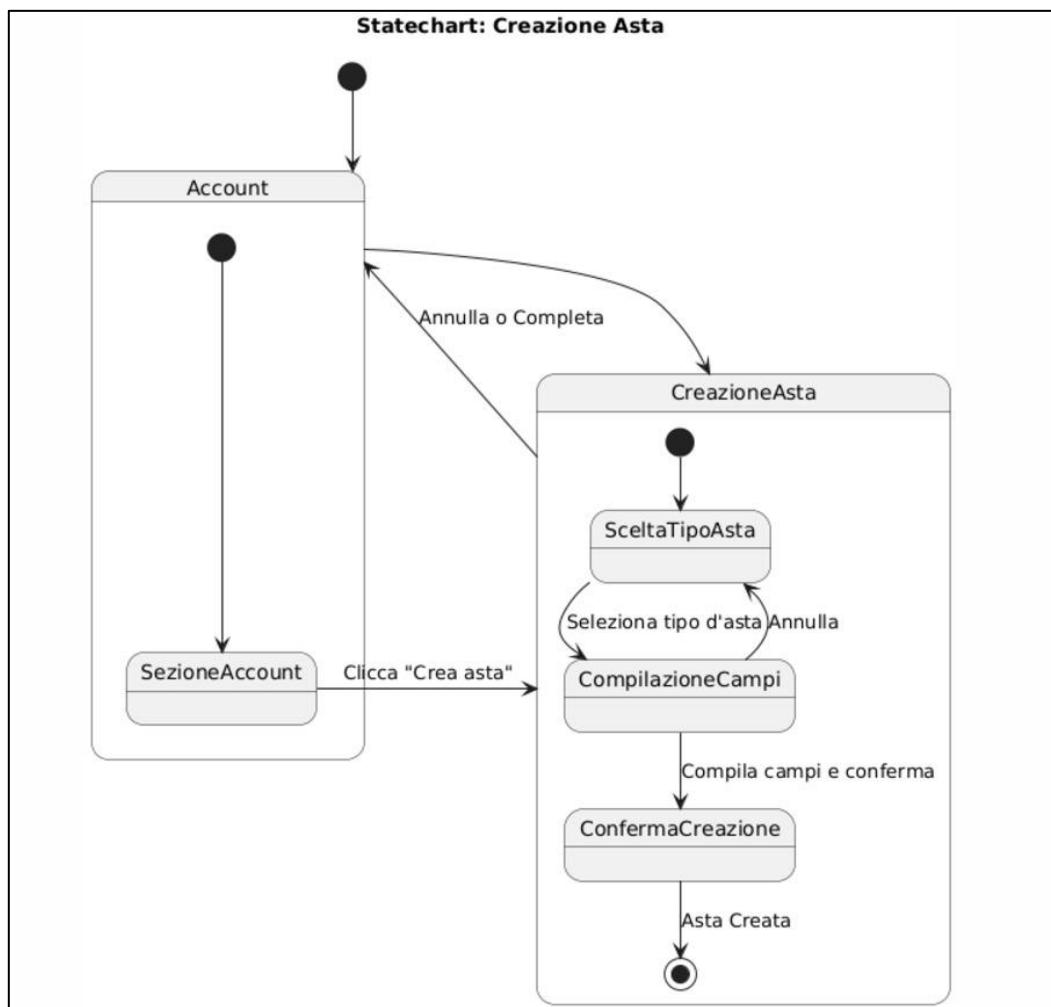


Figure 75: Statechart Diagram - Inserimento Asta

2. CANCELLAZIONE DI UN'ASTA ATTIVA

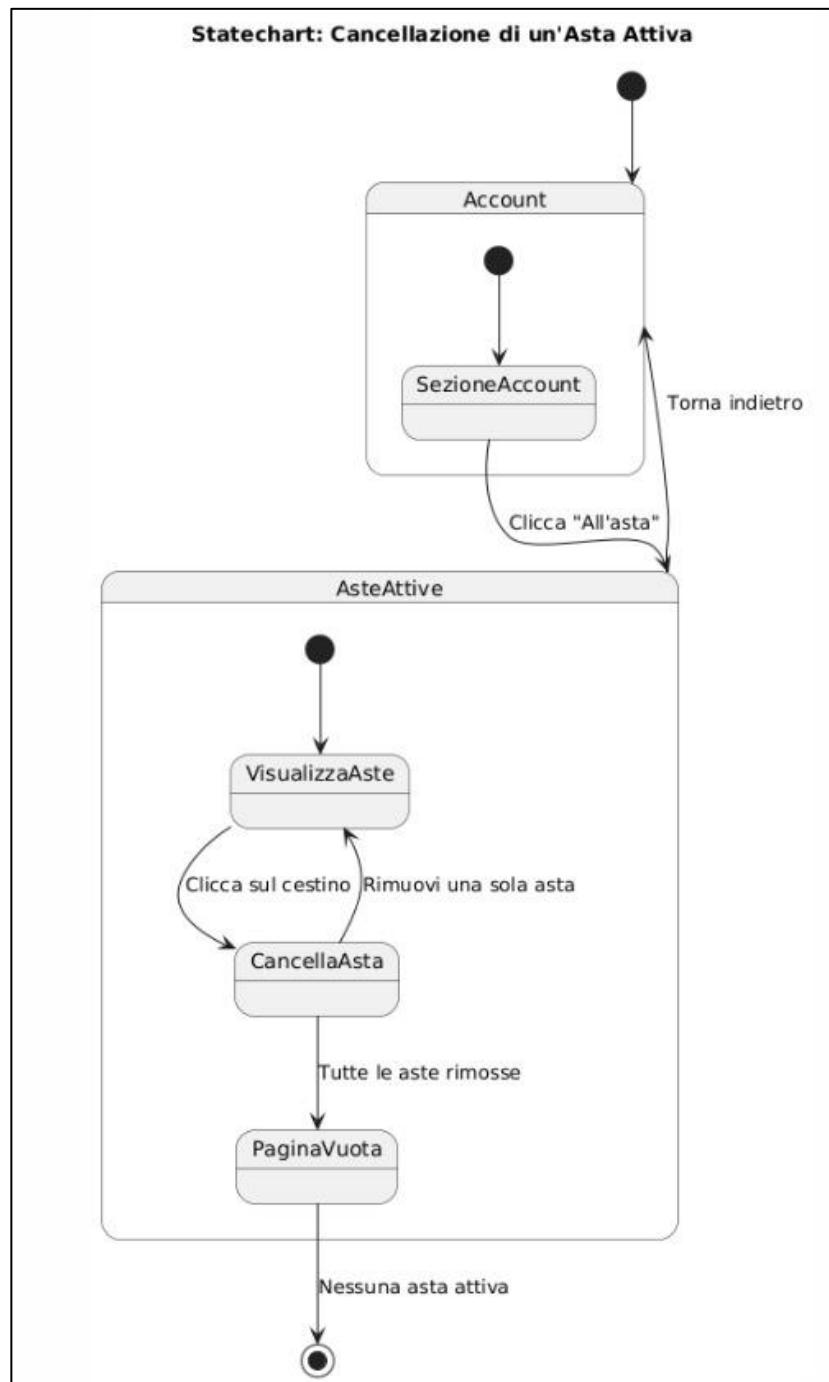


Figure 76: Statechart Diagram - Cancellazione asta attiva

3. INVIO OFFERTA CON RICEZIONE NOTIFICA PER IL VENDITORE

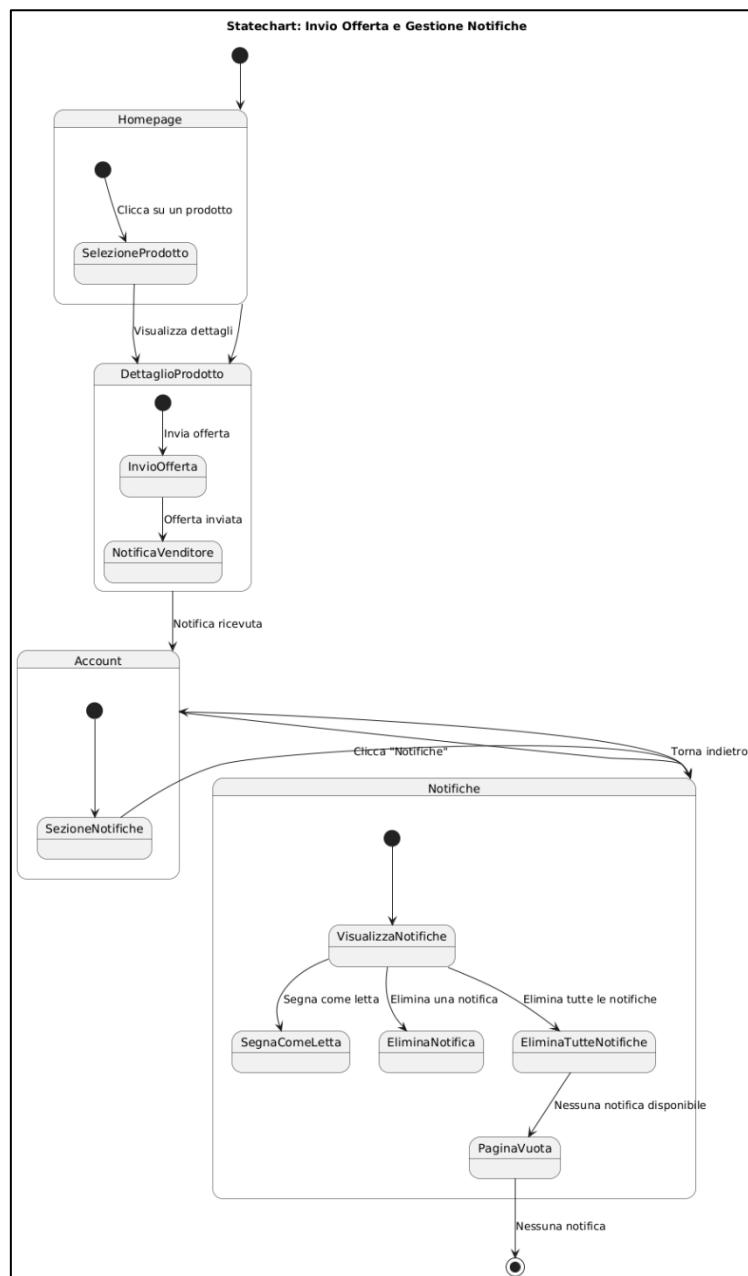


Figure 77: Statechart Diagram - Invio offerta e gestione notifiche

4. ACCETTAZIONE OFFERTA PER ASTA DI TIPO SILENZIOSA

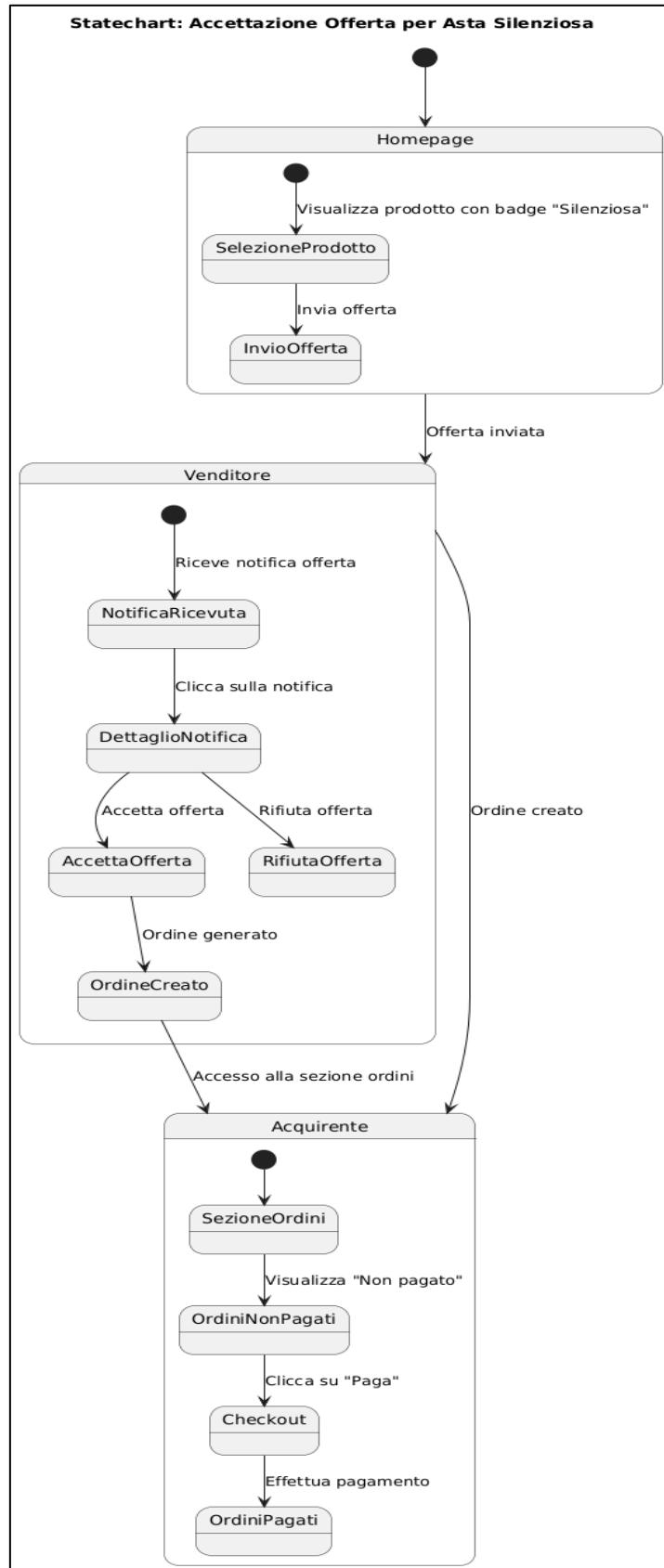


Figure 78: Statechart Diagram - Accettazione offerta per asta silenziosa

Nota: Nel caso in cui le immagini presentate risultino essere poco chiare, è possibile reperirle in alta qualità, ed eventualmente in formato pdf, nella repository pubblica del progetto su GitHub, disponibile al seguente link: [GitHub](#)

DOCUMENTO DI DESIGN DEL SISTEMA

Il **Documento di Design del Sistema** fornisce una visione dettagliata dell'architettura complessiva e delle scelte progettuali del sistema, con l'obiettivo di definire come le varie componenti software interagiranno tra loro per soddisfare i requisiti funzionali e non funzionali. Questa sezione del documento è cruciale per guidare il processo di sviluppo, assicurando che il sistema sia progettato in modo scalabile, manutenibile e facilmente comprensibile per tutti i membri del team.

Il design del sistema si concentra sulla struttura complessiva del software, sulle tecnologie utilizzate e sulle modalità con cui i vari moduli saranno realizzati e integrati. In questa parte, vengono descritti gli aspetti fondamentali che riguardano la scelta dell'architettura, le tecnologie adottate, e la progettazione delle interfacce tra i diversi componenti del sistema.

La sezione è suddivisa nei seguenti sotto-sezioni:

- **Descrizione dell'architettura:** una panoramica dell'architettura complessiva del sistema, incluse le tecnologie e i pattern architettonici scelti.
- **Motivazione delle scelte tecnologiche:** spiegazione delle decisioni relative alle tecnologie utilizzate, come il linguaggio di programmazione, il database, il framework, ecc., e dei vantaggi che queste scelte offrono rispetto ad altre opzioni.
- **Diagramma delle classi di design:** una rappresentazione grafica delle classi e delle loro interazioni, mostrando come le funzionalità vengono distribuite tra i vari componenti del sistema.
- **Diagrammi di sequenza di design:** illustrazione del flusso delle operazioni tra i vari moduli del sistema, evidenziando come le richieste degli utenti vengano elaborate e come i dati vengano trasferiti tra i componenti.

Questo documento funge da guida operativa per il team di sviluppo e contribuisce a mantenere il progetto allineato agli obiettivi, garantendo una progettazione solida e coerente con le esigenze espresse durante l'analisi dei requisiti.

DESCRIZIONE DELL'ARCHITETTURA

L'architettura a tre livelli descritta, utilizza Flutter come front-end, Node.js con Express come back-end e PostgreSQL come database. Qui di seguito, una descrizione dettagliata di ciascun livello.

Front-end

Il front-end dell'architettura è sviluppato utilizzando Flutter, un framework open source creato da Google che permette di realizzare applicazioni mobili multiplattaforma. Basato sul linguaggio di programmazione Dart, Flutter consente di creare interfacce utente moderne, interattive e reattive. Grazie alla sua ampia libreria di widget, è possibile implementare UI consistenti e performanti sia su Android che su iOS, utilizzando un unico codice base. Flutter facilita inoltre un ciclo di sviluppo rapido grazie a funzionalità come l'*hot reload*. La comunicazione con il back-end avviene tramite API RESTful, garantendo un flusso di dati strutturato e sicuro.

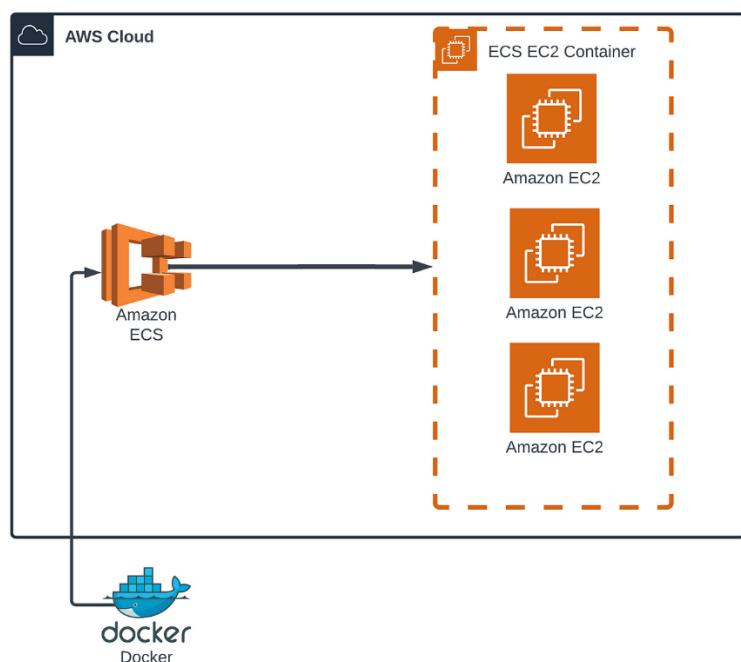


Back-end

Il back-end è implementato utilizzando Node.js ed Express. Node.js è un ambiente runtime JavaScript veloce ed efficiente, ottimizzato per gestire operazioni asincrone, ideale per applicazioni in tempo reale. Express, un framework minimalista per Node.js, semplifica la gestione delle richieste HTTP, il routing e la comunicazione con il database. Il server gestisce le richieste provenienti dal front-end, elabora i dati e restituisce risposte in formato JSON. Inoltre, il back-end sfrutta l'autenticazione tramite Bearer Token per garantire l'accesso sicuro alle API, proteggendo le operazioni riservate degli utenti. Le risorse del server sono containerizzate con Docker e orchestrate su Amazon ECS, assicurando scalabilità e gestione efficiente dei carichi di lavoro.

Database

Il database utilizzato è PostgreSQL, un database relazionale altamente affidabile e performante, scelto per la sua capacità di gestire dati strutturati e semi-strutturati. PostgreSQL supporta funzionalità avanzate, come il trattamento di dati JSON e l'uso di indici per ricerche veloci, rendendolo ideale per applicazioni che richiedono flessibilità e velocità. La comunicazione con il database avviene tramite Sequelize, un ORM (Object-Relational Mapping) specifico per Node.js, che semplifica la gestione delle query e automatizza operazioni come la migrazione e la sincronizzazione delle tabelle.



Nell'architettura a tre livelli, il front-end Flutter invia richieste HTTP al back-end utilizzando API RESTful. Quando un utente interagisce con l'applicazione, ad esempio accedendo alla piattaforma o effettuando operazioni sui dati, il front-end trasmette le informazioni al server tramite chiamate HTTP. Il server elabora tali richieste applicando le regole di business definite, interagisce con PostgreSQL per recuperare o modificare i dati necessari e restituisce una risposta in formato JSON. Il front-end riceve i dati elaborati, li interpreta e aggiorna dinamicamente l'interfaccia utente per garantire un'esperienza fluida e reattiva.

Il back-end funge da nucleo operativo del sistema, gestendo la logica applicativa e garantendo la coerenza e la sicurezza dei dati. PostgreSQL, scelto come database relazionale, si occupa della persistenza dei dati, offrendo elevate prestazioni, scalabilità e supporto per dati strutturati e semi-strutturati. L'interazione con il database è semplificata e ottimizzata tramite Sequelize, che consente di definire modelli e operazioni sui dati in modo efficiente, riducendo la complessità delle query e aumentando la produttività dello sviluppo.

Questo sistema, ben progettato e modulare, permette ai diversi componenti di operare in modo autonomo ma coordinato, consentendo una facile scalabilità per gestire crescenti volumi di utenti e richieste. La separazione chiara tra front-end, back-end e database garantisce inoltre una maggiore flessibilità nello sviluppo e nella manutenzione, adattandosi a nuove esigenze o futuri miglioramenti senza compromettere la stabilità complessiva dell'applicazione.

MOTIVAZIONE DELLE SCELTE TECNOLOGICHE ADOTTATE

Le scelte tecnologiche adottate per l'architettura dell'applicazione sono state guidate dalla necessità di garantire performance, scalabilità e sicurezza, sfruttando strumenti moderni e potenti in ogni livello dell'applicazione.

- **Flutter (Front-end)**

- Permette lo sviluppo multipiattaforma con un unico codice base, riducendo tempi e costi di sviluppo.
- Fornisce un'ampia libreria di widget predefiniti e personalizzabili, ideali per creare interfacce moderne e interattive.
- La funzionalità di *hot reload* accelera il ciclo di sviluppo, consentendo modifiche rapide e test immediati.
- Garantisce prestazioni elevate grazie al rendering nativo, offrendo un'esperienza utente fluida su tutte le piattaforme.

- **Node.js con Express (Back-end)**

- Node.js è leggero e adatto per gestire applicazioni ad alta concorrenza, grazie al modello asincrono basato su eventi.
- Express semplifica lo sviluppo del server, offrendo strumenti per la gestione delle richieste HTTP, il routing e l'integrazione con il database.
- La possibilità di containerizzazione con Docker, orchestrata su Amazon ECS, assicura scalabilità e flessibilità, adattandosi ai picchi di utilizzo.

- **Gestione della sessione e autenticazione con JWT:** L'autenticazione è gestita tramite JWT (JSON Web Token), che garantisce la sicurezza delle comunicazioni tra il client e il server. I token vengono generati al momento del login e inviati al client, che li include nelle richieste future per l'autenticazione, eliminando la necessità di gestire sessioni lato server e migliorando la scalabilità del sistema.
- **PostgreSQL (Database)**
 - Garantisce elevate prestazioni e affidabilità per la gestione dei dati strutturati e semi-strutturati.
 - Supporta funzionalità avanzate, come indici e aggregazioni complesse, per ottimizzare le operazioni sui dati.
 - La natura open source lo rende una scelta conveniente e ben documentata, ideale per ambienti di produzione.
 - L'uso di Sequelize come ORM riduce la complessità delle query SQL, semplificando l'interazione con il database e aumentando la produttività del team.

Queste scelte tecnologiche sono state effettuate per garantire un'architettura moderna, scalabile e performante, capace di rispondere in modo efficiente alle esigenze attuali e future dell'applicazione.

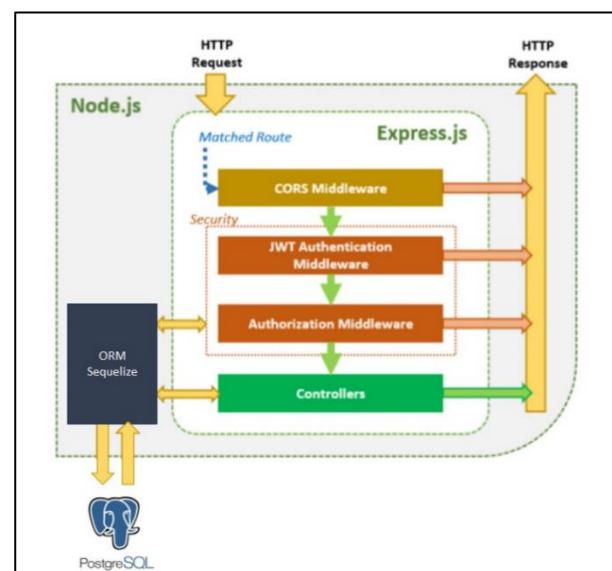
DIAGRAMMA DELLE CLASSI DI DESIGN

La sezione "Diagramma delle Classi di Design" fornisce una rappresentazione dettagliata della struttura interna del sistema, evidenziando le principali classi e gli oggetti che lo compongono, nonché le loro relazioni e interazioni. A differenza dei diagrammi di classi di analisi, che si concentrano sulle entità e le loro caratteristiche, i diagrammi di classi di design esplorano la progettazione del sistema a livello di implementazione, fornendo una visione più dettagliata delle classi, dei metodi, delle proprietà e delle interazioni necessarie per costruire il software.

Questa sezione del documento ha lo scopo di chiarire come ogni componente del sistema verrà implementato, definendo le responsabilità e le interfacce delle classi in modo che il team di sviluppo possa procedere con la realizzazione del sistema in modo coerente e strutturato.

Il diagramma delle classi di design si concentra su:

- **Definizione delle classi:** specifica le classi necessarie e le loro funzionalità.
- **Relazioni tra le classi:** descrive come le classi interagiscono tra loro attraverso associazioni, ereditarietà, e dipendenze.
- **Metodi e attributi:** fornisce un elenco dei metodi principali e delle variabili necessarie per ogni classe.
- **Interfacce:** definisce le interfacce che permettono alle classi di comunicare tra loro in modo modulare e riutilizzabile.



Questa sezione è fondamentale per garantire che la progettazione del sistema sia coerente con i requisiti stabiliti e che le classi siano strutturate in modo tale da supportare facilmente l'estensibilità e la manutenzione del software.

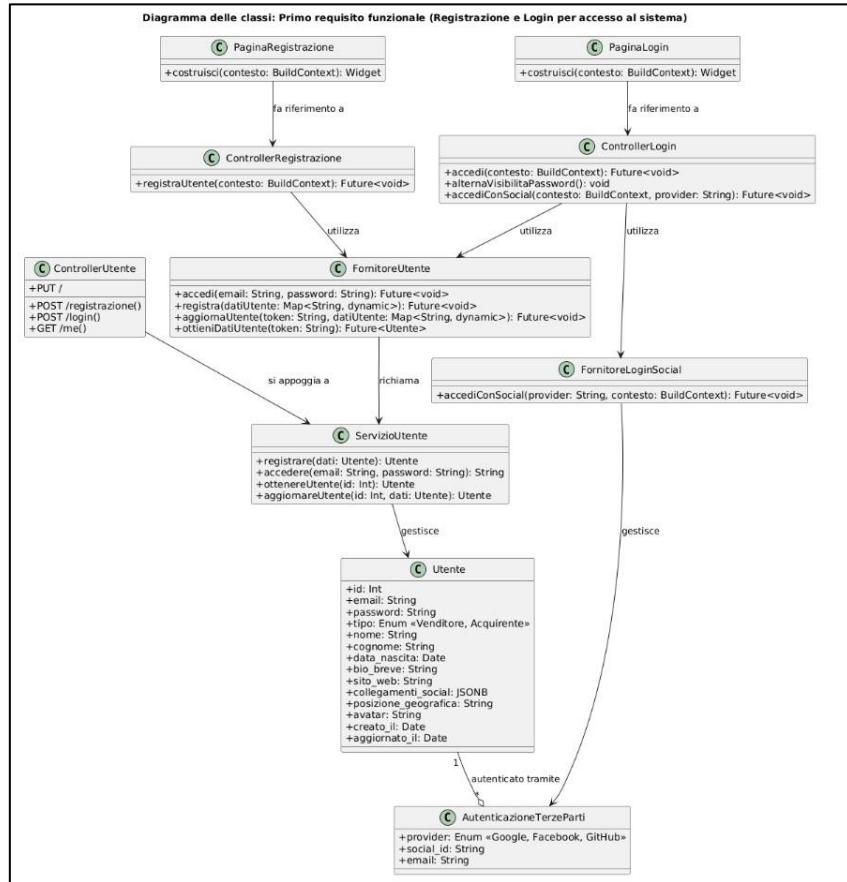


Figure 79: Registrazione e Login per accesso al sistema

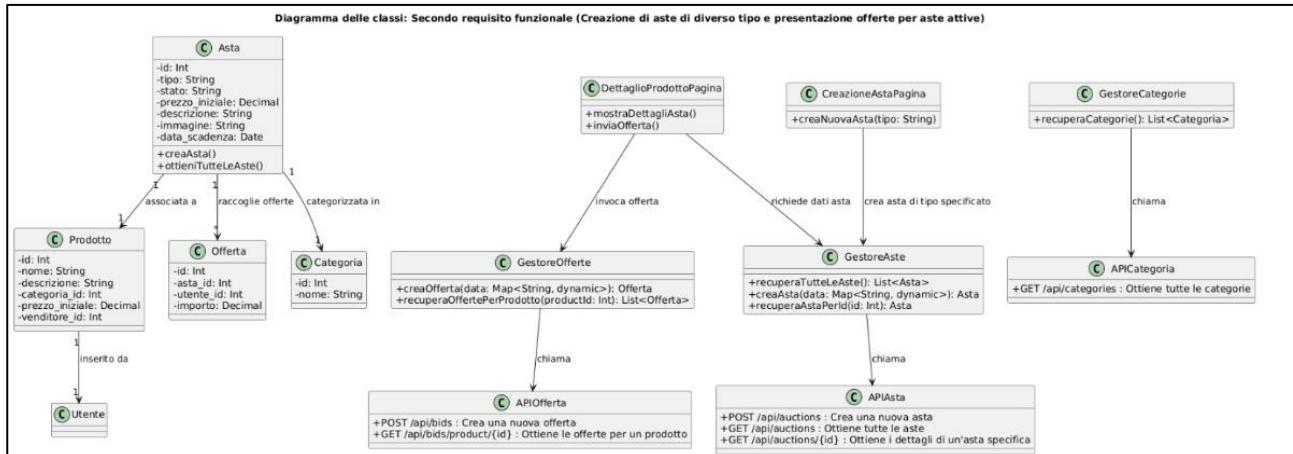


Figure 80: Creazione di un nuovo tipo di asta e presentazione offerta

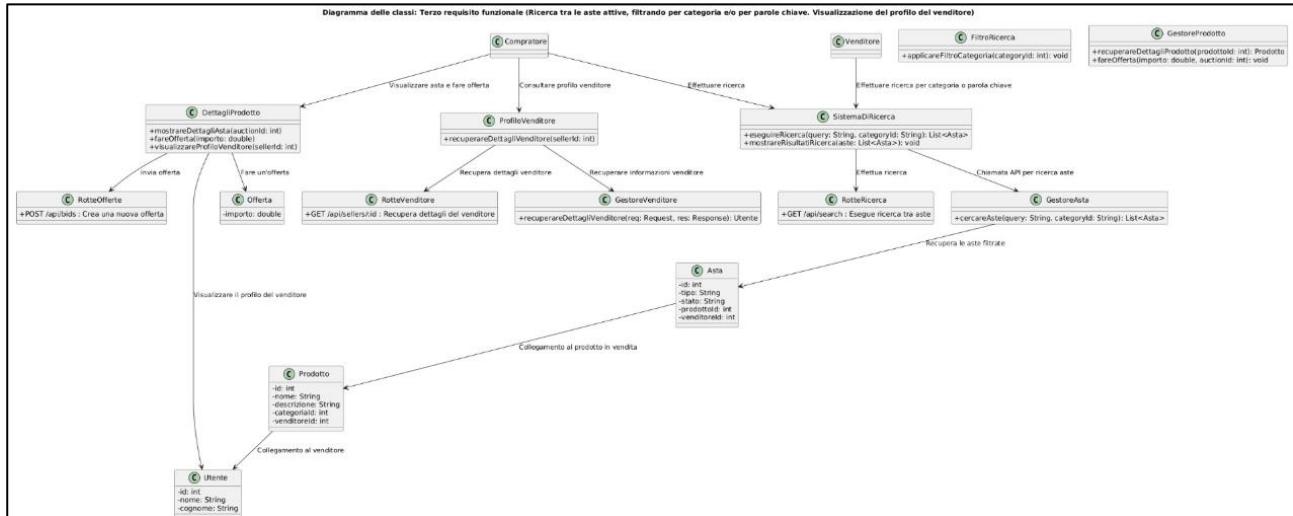


Figure 81: Ricerca di aste attive

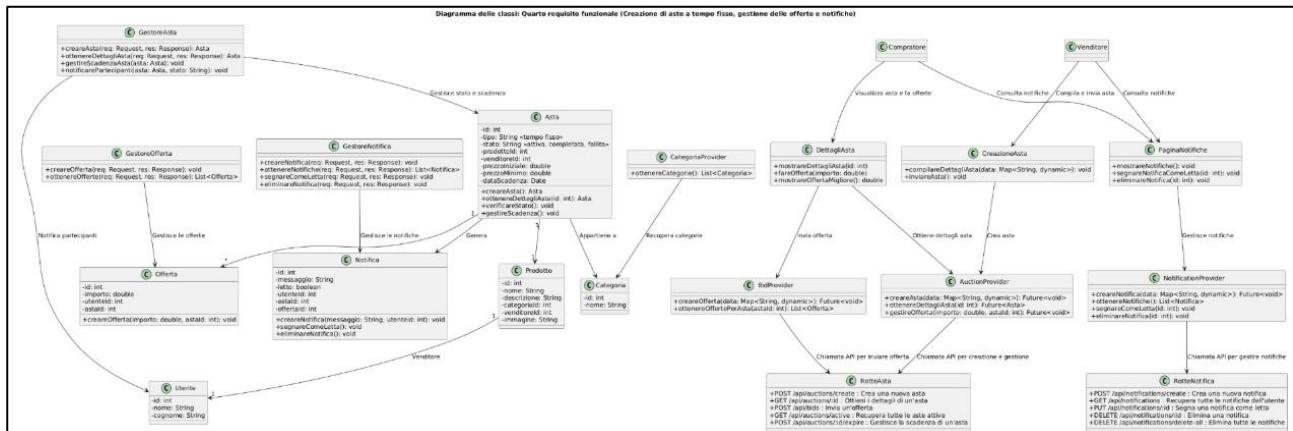


Figure 82: Creazione asta a tempo fisso, gestione offerte e notifiche

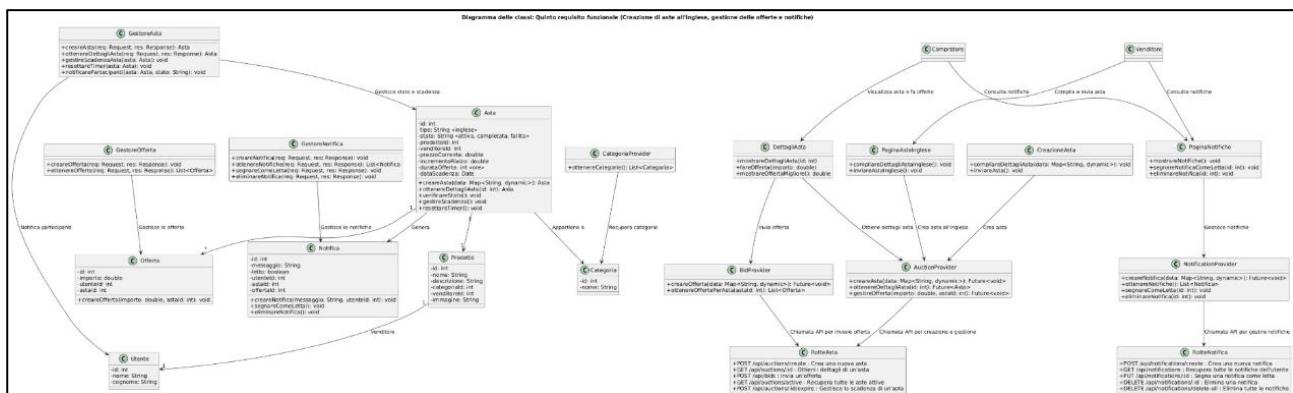


Figure 83: Creazione asta all'inglese, gestione offerte e notifiche

ingsw_dietideals24_app

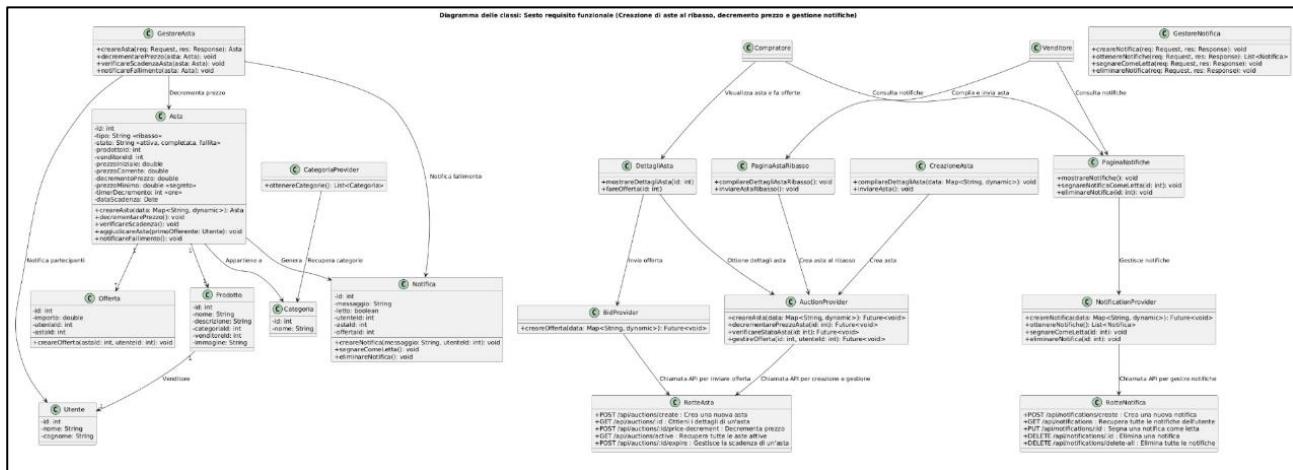


Figure 84: Creazione asta al ribasso, decremento presso e gestione notifiche

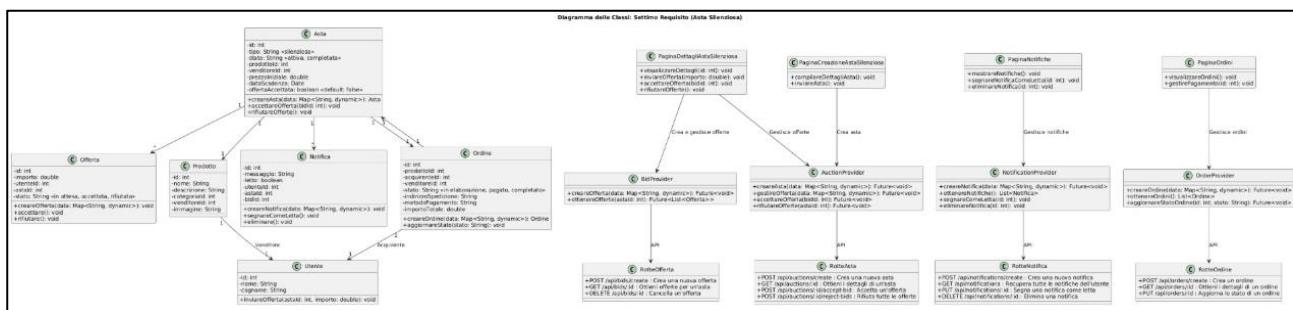


Figure 85: Creazione e gestione asta silenziosa

Nota: Nel caso in cui le immagini presentate risultino essere poco chiare, è possibile reperirle in alta qualità, ed eventualmente in formato pdf, nella repository pubblica del progetto su GitHub, disponibile al seguente link: [GitHub](#)

DIAGRAMMI DI SEQUENZA DI DESIGN

In questa sezione sono presentati i diagrammi di sequenza di design, strumenti fondamentali per rappresentare l'interazione tra gli attori e il sistema durante l'esecuzione delle funzionalità chiave dell'applicazione. A differenza dei diagrammi di sequenza di analisi, che si concentrano su un livello di astrazione più alto, i diagrammi di sequenza di design offrono una visione dettagliata e operativa delle dinamiche del sistema, descrivendo il flusso di messaggi tra i vari componenti e l'ordine in cui le azioni vengono eseguite.

Questi diagrammi sono essenziali per illustrare come il sistema reagirà a determinate azioni degli utenti e come le diverse parti del sistema collaborano per completare le operazioni. Sono particolarmente utili per verificare che il design architettonico e le logiche di interazione siano correttamente implementati, e per garantire che le funzionalità siano sviluppate in modo coerente con i requisiti.

Per questo progetto, sono stati realizzati quattro diagrammi di sequenza che descrivono in dettaglio le interazioni necessarie per implementare le seguenti funzionalità chiave dell'applicazione:

1. Creazione Asta:

Questo diagramma descrive il flusso di eventi a partire dal momento in cui un utente seleziona il tipo di asta che desidera creare, fino alla registrazione dell'asta nel sistema, includendo la definizione del prodotto.

Vengono mostrati i messaggi scambiati tra l'utente, l'interfaccia utente, il sistema di backend e il database, evidenziando i passaggi cruciali come la validazione dei dati, l'inserimento nel database e le eventuali risposte di conferma o errore.

2. Cancellazione Asta Attiva:

In questo diagramma si descrivono le interazioni necessarie per annullare un'asta che è ancora in corso. La sequenza mostra come l'utente (venditore o amministratore) invii una richiesta di cancellazione, come il sistema valida tale richiesta e aggiorna lo stato dell'asta nel database, e come vengono inviate notifiche agli utenti coinvolti per comunicare l'avvenuta cancellazione.

3. Invio Offerta con Ricezione Notifica per il Venditore:

Questo diagramma mostra il flusso di eventi che si verifica quando un offerente invia un'offerta per un prodotto in asta. La sequenza include l'interazione tra il sistema di gestione delle offerte, il backend e il sistema di notifica. Vengono mostrati i dettagli relativi alla registrazione dell'offerta nel sistema e l'invio della notifica al venditore, che riceve un messaggio di avviso in tempo reale sull'offerta ricevuta.

4. Accettazione Offerta per Asta di Tipo Silenziosa:

In questo diagramma, viene descritto il processo attraverso il quale un venditore accetta un'offerta ricevuta in un'asta silenziosa. La sequenza include l'interazione tra il venditore e il sistema, mostrando come l'offerta venga selezionata, accettata e registrata nel sistema. Vengono descritti anche i messaggi di notifica inviati agli offerenti per informare dell'accettazione e gli aggiornamenti nel database, in modo da riflettere correttamente il nuovo stato dell'asta.

Funzione e obiettivi dei diagrammi di sequenza di design:

Questi diagrammi di sequenza di design non solo documentano il comportamento atteso del sistema, ma servono anche come strumento cruciale per il team di sviluppo. Permettono di:

- Verificare la correttezza dell'implementazione:** I diagrammi offrono una rappresentazione dettagliata del comportamento del sistema, utile per confrontare le implementazioni reali con il design e per identificare eventuali discrepanze o errori.

- **Prevenire problematiche nelle interazioni tra componenti:** La visualizzazione del flusso dei messaggi e del coordinamento tra i vari moduli consente di individuare potenziali colli di bottiglia, conflitti di stato o ambiguità nelle interazioni prima che diventino problematiche durante lo sviluppo.
- **Migliorare la comunicazione tra team:** I diagrammi di sequenza di design fungono da punto di riferimento visivo per sviluppatori, analisti, progettisti e stakeholder. Offrono una chiara rappresentazione del funzionamento del sistema, facilitando la comprensione del flusso di lavoro e permettendo ai team di allinearsi sugli aspetti tecnici e funzionali.
- **Rafforzare la documentazione di progetto:** I diagrammi servono come documentazione visiva per il design del sistema, creando un'archiviazione chiara e comprensibile che può essere consultata durante la manutenzione futura del sistema o per eventuali modifiche.

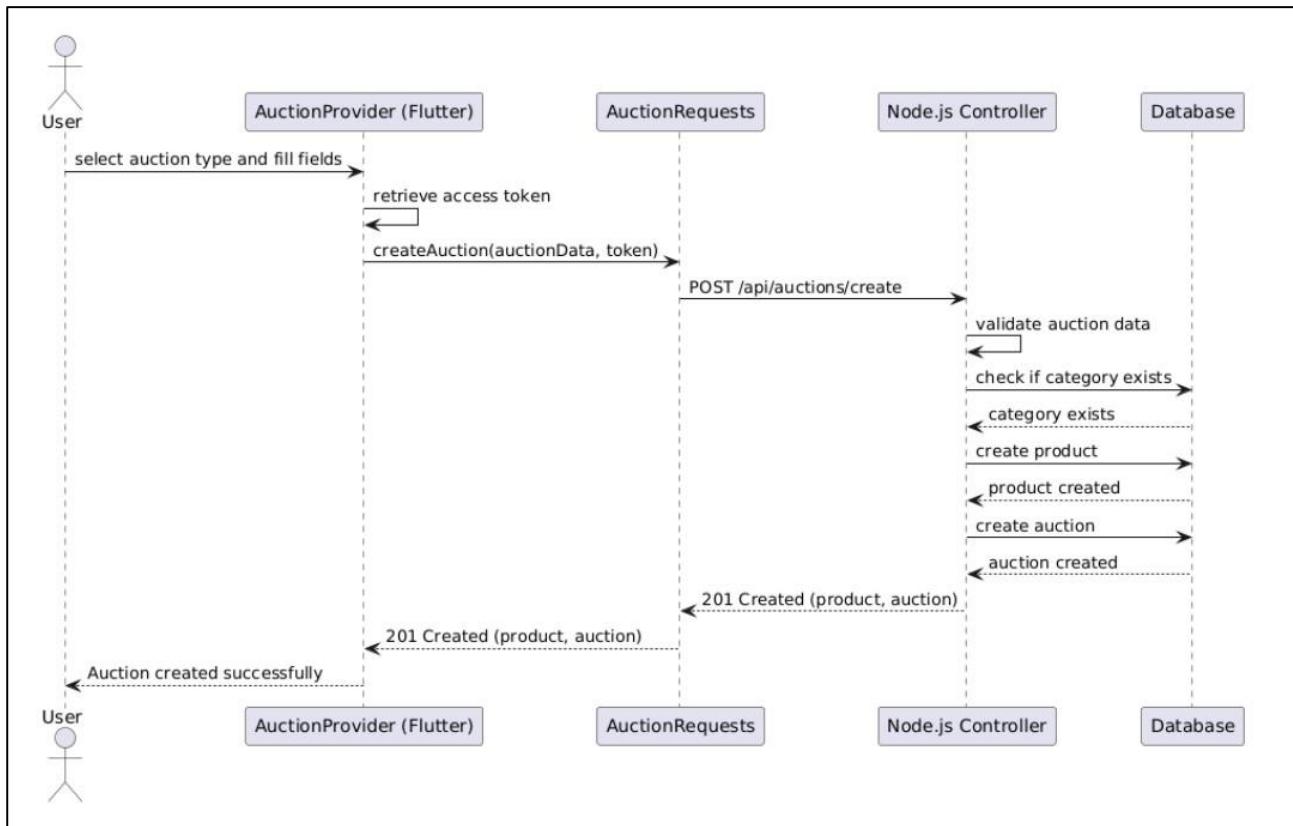


Figure 86: Sequence Diagram per la creazione di un'asta

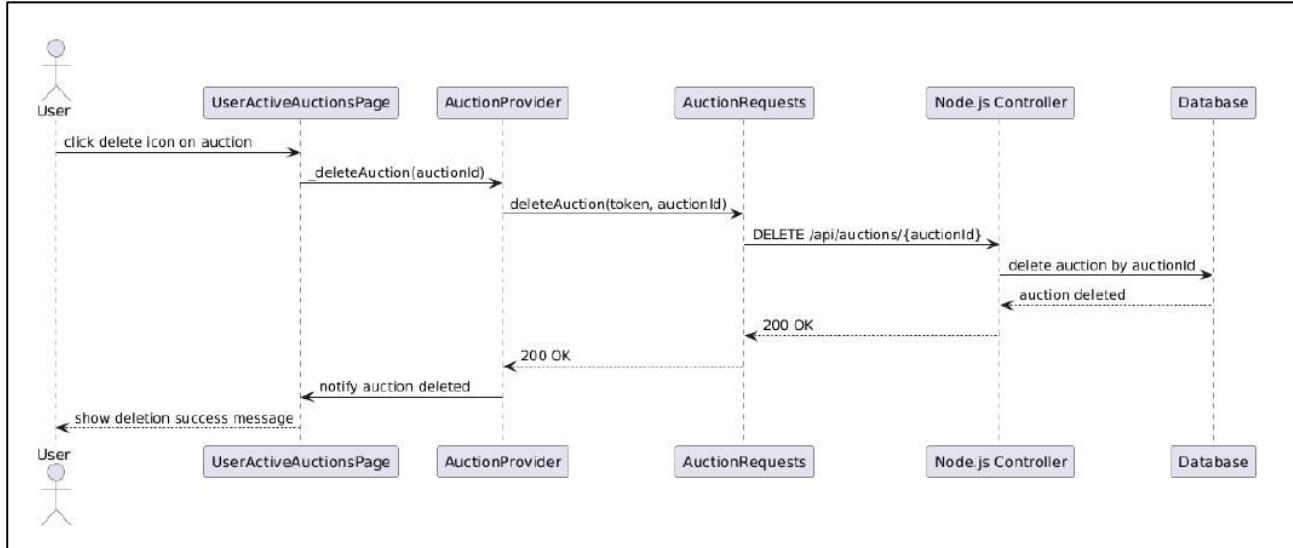


Figure 87: Sequence Diagram per la cancellazione di un'asta attiva

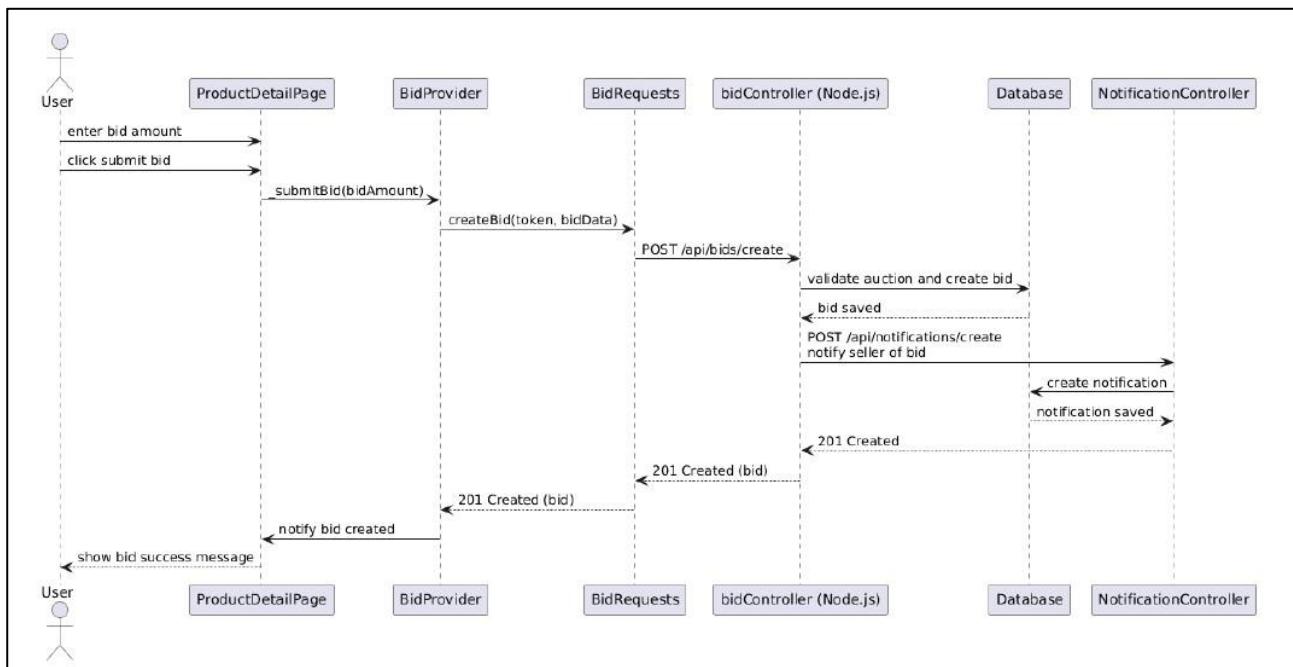


Figure 88: Sequence Diagram per l'invio offerta di un prodotto specifico all'asta con ricezione notifica al venditore

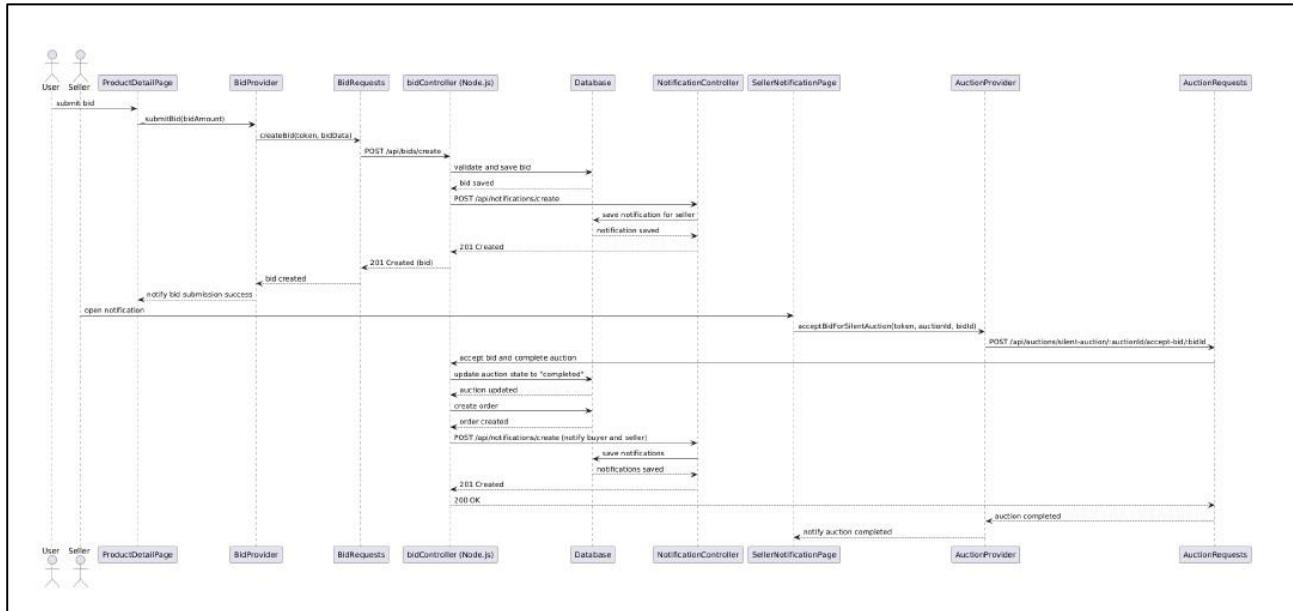


Figure 89: Sequence Diagram per accettare l'offerta di un'asta di tipo silenziosa

Nota: Nel caso in cui le immagini presentate risultino essere poco chiare, è possibile reperirle in alta qualità, ed eventualmente in formato pdf, nella repository pubblica del progetto su GitHub, disponibile al seguente link: [GitHub](#)

STRUMENTI DI VERSIONING

Nel corso dello sviluppo del progetto, è stato utilizzato **GitHub** come unico repository per la gestione del versioning del codice. Questa scelta è stata dettata da diversi fattori chiave, tra cui la facilità di utilizzo, la chiarezza nell'organizzazione dei file e la robustezza offerta dagli strumenti di collaborazione integrati.

Struttura del Repository

Per garantire una gestione ordinata e modulare del progetto, il repository è stato organizzato in quattro principali sottocartelle:

- **Backend:** contiene tutto il codice relativo alla logica server-side, incluse API, configurazioni del database e script di gestione.
- **Frontend:** include il codice della parte client-side, comprendendo interfacce utente, stili e logiche di interazione.
- **Mockup:** questa cartella contiene i file grafici e prototipi di interfaccia utente, utili come riferimento per il design del frontend.
- **Documentazione:** include i file relativi alla descrizione del progetto, ai diagrammi e ad altre informazioni necessarie per il team.

Questa divisione ha permesso di separare chiaramente le responsabilità tra i due ambiti del progetto, facilitando lo sviluppo parallelo e la manutenzione futura.

Author	Commit Message	Date	Time
vinsgiordi	Merge pull request #26 from vinsgiordi/fix/bidhub—update-v1-documenta...	9c73a4e	4 days ago
	Add files via upload		4 days ago
	feature/bidhub--add-sequence-and-mockup-folder: Added ...		last week
	fix/bidhub--various-fix: Made several fixes and last commits		last week
	fix/bidhub--various-fix: Made several fixes and last commits		last week
	.gitignore is now working		3 weeks ago
	Initial commit		5 months ago
	Update README.md		last month

About

An Android application developed with Node.js, Express, and Flutter, designed to facilitate real-time online auctions. Users can create listings, place bids, and complete transactions, with a microservices architecture supported by PostgreSQL and Docker containers. Built for the Software Engineering course at the University of Naples "Federico II."

nodejs, javascript, dart, docker, aws, node, docker-compose, docker-image, projects, flutter, flutter-apps, ingsw

Figura 90: Pagina principale del repository

ingsw_dietideals24_app

Filters Q is:pr is:closed Labels 9 Milestones 0 New pull request

Clear current search query, filters, and sorts

Author	Label	Projects	Milestones	Reviews	Assignee	Sort
0 Open	✓ 26 Closed					
fix/bidhub—update-v1-documentation-docx #26 by 1999-sis was merged 4 days ago						
fix/bidhub--add-title-page-on-report-sonarqube-and-various-fix #25 by vinsgiordi was merged 4 days ago						
fix/bidhub—update-documentation-docx #24 by 1999-sis was merged 4 days ago						
feature/bidhub--add-ebc-and-class-diagram-design-folder #23 by vinsgiordi was merged 5 days ago						
feature/bidhub—add-documentation-file #22 by 1999-sis was merged 5 days ago						
feature/bidhub--add-statechart-folder #21 by vinsgiordi was merged last week						
feature/bidhub--add-sequence-and-mockup-folder #20 by vinsgiordi was merged last week						
fix/bidhub--various-fix #19 by vinsgiordi was merged last week						
fix/bidhub--change-callback-url-social-login #18 by vinsgiordi was merged last week						
feature/bidhub--aws-integration #17 by vinsgiordi was merged 2 weeks ago						
feature/bidhub--upload-documentation-folder #16 by vinsgiordi was merged 2 weeks ago						
fix/bidhub--image-placeholder #15 by vinsgiordi was merged 2 weeks ago						
fix/bidhub--image-integration #14 by vinsgiordi was merged 3 weeks ago						
test/bidhub--implementation-and-testing-structure #13 by vinsgiordi was merged last month						
feature/bidhub--docker-setup #12 by vinsgiordi was merged on Nov 7						
feature/bidhub--adding-and-deleting-auctions #11 by vinsgiordi was merged on Nov 6						
feature/bidhub--adding-and-deleting-auctions #10 by vinsgiordi was merged on Nov 6						
feature/bidhub--control-requirements #9 by vinsgiordi was merged on Nov 6						
feature/bidhub--order-pages #8 by vinsgiordi was merged on Nov 2						
feature/bidhub--creating-page-sold-and-unsold-auctions #7 by vinsgiordi was merged on Oct 28						
feature/bidhub--integration-of-api-for-bids #6 by vinsgiordi was merged on Oct 18						
Feature/bidhub connection of the login with social #5 by vinsgiordi was merged on Oct 14						
Feature/bidhub--linking-backend-and-frontend #4 by vinsgiordi was merged on Oct 11						
feature/bidhub--social-authentication #3 by vinsgiordi was merged on Sep 20						
feature-bidhub--backend-setup #2 by vinsgiordi was merged on Sep 11						

< Previous 1 2 Next >

Figura 91: Pagina delle Pull Request

Branches						New branch
Branch	Updated	Check status	Behind	Ahead	Pull request	
develop	4 days ago		0	0	#25	
fix/bidhub--update-v1-documentation-docx	4 days ago		1	0	#23	
fix/bidhub--add-title-page-on-report-sonarqube-and-various-fix	4 days ago		3	0	#25	
fix/bidhub--update-documentation-docx	5 days ago		5	0	#23	
feature/bidhub--add-ebc-and-class-diagram-design-folder	5 days ago		7	0	#23	
feature/bidhub--add-documentation-file	last week		10	0	#21	
feature/bidhub--add-statechart-folder	last week		12	0	#21	
feature/bidhub--add-sequence-and-mockup-folder	last week		14	0	#20	
fix/bidhub--various-fix	last week		16	0	#19	
fix/bidhub--change-callback-url-social-login	2 weeks ago		19	0	#18	
feature/bidhub--aws-integration	2 weeks ago		21	0	#17	
feature/bidhub--upload-documentation-folder	2 weeks ago		23	0	#16	
fix/bidhub--image-placeholder	2 weeks ago		27	0	#15	
fix/bidhub--image-integration	3 weeks ago		30	0	#14	
feature/bidhub--image-integration	3 weeks ago		33	4	#13	
test/bidhub--implementation-and-testing-structure	last month		34	0	#13	
feature/bidhub--docker-setup	last month		41	0	#12	
feature/bidhub--adding-and-deleting-auctions	last month		45	0	#11	
feature/bidhub--control-requirements	last month		48	0	#9	
feature/bidhub--order-pages	last month		51	0	#8	

Previous [Next >](#)

Figura 92: Pagina dei Branches attivi

Motivazioni della Scelta di GitHub

1. Centralizzazione del Codice

- Utilizzando un unico repository su GitHub, tutti i membri del team hanno avuto un punto di accesso unico al codice sorgente, riducendo i rischi di incongruenze e duplicati.

2. Facilità di Gestione

- Con GitHub, è stato possibile monitorare facilmente i progressi del progetto grazie alla cronologia dei commit e alla possibilità di confrontare versioni differenti del codice.

3. Branching e Workflow

- L'uso di branch separati per ogni funzionalità o correzione ha garantito un workflow organizzato. La ramificazione ha permesso di:

ingsw_dietideals24_app

- Lavorare su nuove funzionalità senza compromettere il codice stabile.
- Integrare i cambiamenti con pull request, garantendo un processo di revisione chiaro e collaborativo.

4. Chiarezza e Storico delle Modifiche

- I **commit** regolari e descrittivi hanno documentato in modo chiaro le modifiche apportate nel tempo. Questo approccio ha facilitato la revisione del codice e la risoluzione di eventuali problemi.

5. Collaborazione

- GitHub offre strumenti come il codice condiviso, le pull request e i commenti inline, che migliorano significativamente la collaborazione, anche in caso di team distribuiti.

6. Integrazione e Automazione

- Grazie al supporto per le pipeline CI/CD e altre integrazioni, il repository su GitHub si è dimostrato un ottimo punto di partenza per automatizzare i processi di testing e deployment.

Nomenclatura dei Branch e Gestione del Workflow

Durante lo sviluppo del progetto, è stata adottata una convenzione chiara e strutturata per la nomenclatura dei **branch**, al fine di garantire un workflow organizzato e sicuro. I **branch** sono stati denominati secondo il seguente schema:

- **feature/bidhub--nome-task**: utilizzato per la creazione di nuove funzionalità.
- **fix/bidhub--nome-task**: dedicato alla risoluzione di bug o alla modifica di elementi già esistenti.
- **test/bidhub--nome-task**: impiegato per lo sviluppo e la gestione dei test.

Questa convenzione ha avuto un duplice obiettivo: favorire la chiarezza nel distinguere il tipo di attività svolta e mantenere un ordine logico all'interno del repository. Ogni modifica è stata sviluppata in un branch dedicato, senza intaccare direttamente il branch principale di sviluppo, denominato **develop**.

Il branch **develop** ha agito come base stabile per l'integrazione delle funzionalità, garantendo un ambiente sicuro per lo sviluppo parallelo. Solo dopo una revisione accurata e il completamento delle verifiche necessarie, le modifiche sviluppate nei branch secondari sono state integrate, preservando la coerenza e la qualità del codice nel repository.

Benefici Riscontrati

- **Trasparenza**: Ogni modifica è stata tracciata e documentata, garantendo piena visibilità sulle attività.
- **Efficienza**: L'organizzazione in branch ha evitato conflitti e sovrascritture di codice, accelerando i tempi di sviluppo.
- **Scalabilità**: La struttura del repository permette una facile espansione futura, poiché nuove funzionalità possono essere sviluppate in branch separati senza intaccare la stabilità del progetto.

Conclusione

L'adozione di **GitHub** come strumento di versioning è stata fondamentale per il successo del progetto. Grazie alla sua interfaccia intuitiva e alla potenza degli strumenti di collaborazione offerti, è stato possibile garantire un workflow fluido, una gestione chiara del codice e una documentazione accurata delle modifiche. La struttura organizzata del repository ha inoltre posto solide basi per una manutenzione efficace e lo sviluppo futuro del progetto.

Nota: Per maggiore chiarezza sui branch attivi, i commit effettuati, e la struttura generale del progetto, è possibile fare riferimento alla repository ufficiale disponibile al seguente link: [GitHub](#)

TESTING E VALUTAZIONE SUL CAMPO DELL'USABILITÀ

UNIT TESTING

I test descritti e documentati sono stati progettati seguendo due approcci : **White Box** e **Black Box**, concentrandosi sulle funzionalità visibili all'utente finale e verificando che il comportamento del sistema sia conforme ai requisiti specificati. Questo tipo di test è particolarmente importante per garantire che l'applicazione soddisfi le aspettative degli utenti e funzioni correttamente in scenari realistici. Di seguito, viene spiegata l'importanza dei test chiave effettuati:

Login

Il login rappresenta il punto di accesso principale all'applicazione per gli utenti autenticati. Testare questa funzionalità è cruciale per garantire:

- **Sicurezza:** Solo gli utenti con credenziali valide possono accedere.
- **Affidabilità:** L'utente deve essere in grado di accedere senza errori, evitando blocchi o malfunzionamenti.
- **Esperienza utente:** Feedback chiari (ad esempio, in caso di credenziali errate) migliorano l'interazione con l'app.

Un sistema di login robusto è essenziale per proteggere i dati sensibili degli utenti e mantenere la fiducia degli stessi nell'applicazione.

Registrazione

La registrazione consente a nuovi utenti di entrare nell'ecosistema dell'applicazione. Testare questa funzionalità garantisce che:

- **Dati validi:** Vengano accettati solo dati corretti e completi (es. email valide, password sufficientemente robuste).
- **Gestione degli errori:** Messaggi chiari informano l'utente di eventuali problemi (ad esempio, account già registrato o dati mancanti).
- **Accesso immediato:** Dopo la registrazione, l'utente dovrebbe essere in grado di accedere senza difficoltà.

Una registrazione ben progettata è fondamentale per attrarre nuovi utenti e offrire loro una prima impressione positiva.

Reset Password

La funzionalità di reset della password è un elemento chiave per la gestione degli account, poiché offre agli utenti un modo sicuro per recuperare l'accesso al proprio profilo in caso di smarrimento delle credenziali. I test su questa funzione garantiscono:

- **Sicurezza:** Assicurarsi che solo il proprietario dell'account possa reimpostare la password.
- **Usabilità:** L'utente deve essere in grado di completare il processo facilmente, con istruzioni chiare.
- **Funzionalità:** Verifica che le nuove credenziali siano immediatamente utilizzabili.

Questa funzione migliora la fiducia degli utenti e riduce il rischio di perdita di accesso.

Creazione di un'Asta

La creazione di un'asta rappresenta il nucleo centrale dell'applicazione. Testare questa funzionalità è essenziale per:

- **Integrità del sistema:** Garantire che i dati relativi all'asta siano salvati correttamente e riflettano le specifiche dell'utente.
- **Esperienza utente:** Assicurarsi che il processo sia intuitivo e privo di ostacoli, con feedback chiari.
- **Core Business:** Essendo una delle funzionalità principali, eventuali malfunzionamenti avrebbero un impatto diretto sulla percezione del valore dell'applicazione. La corretta implementazione di questa funzione è critica per il successo complessivo del progetto.

```
void main() {
  WidgetsFlutterBinding.ensureInitialized();

  // Registrazione
  group('User Registration', () {
    // Test di registrazione con dati validi (email dinamica)
    test('Registrazione utente con dati validi', () async {
      final userProvider = UserProvider();
      final email = 'testuser_${DateTime.now().millisecondsSinceEpoch}@example.com';

      // Crea un mock di SharedPreferences
      final mockPrefs = MockSharedPreferences();

      // Definisci cosa fa il mock quando viene chiamato getString
      when(mockPrefs.getString('accessToken')).thenReturn('dummyAccessToken');

      // Imposta il mock nelle SharedPreferences
      SharedPreferences.setMockInitialValues({});

      // Esegui la registrazione con dati validi
      await userProvider.register({
        'nome': 'Test',
        'cognome': 'User',
        'data_nascita': '1990-01-01',
        'email': email,
        'password': 'password123'
      });

      // Verifica che l'email restituita sia quella generata dinamicamente
      expect(userProvider.user?.email, email);

      // Recupera l'access token da SharedPreferences
      final accessToken = mockPrefs.getString('accessToken');

      // Assicurati che l'access token non sia null
      expect(accessToken, 'dummyAccessToken');
    });
  });
}
```

Figure 93: Registrazione utente con dati validi (email dinamica)

```
// Test di registrazione con dati validi (email statica)
test('Registrazione utente con dati validi', () async {
  final userProvider = UserProvider();
  const email = 'testuser@example.com';

  // Crea un mock di SharedPreferences
  final mockPrefs = MockSharedPreferences();

  // Definisci cosa fa il mock quando viene chiamato getString
  when(mockPrefs.getString('accessToken')).thenReturn('dummyAccessToken');

  // Imposta il mock nelle SharedPreferences
  SharedPreferences.setMockInitialValues({});

  // Esegui la registrazione con dati validi
  await userProvider.register({
    'nome': 'Test',
    'cognome': 'User',
    'data_nascita': '1990-01-01',
    'email': email,
    'password': 'password123'
  });

  // Verifica che l'email restituita sia quella generata dinamicamente
  expect(userProvider.user?.email, email);

  // Recupera l'access token da SharedPreferences
  final accessToken = mockPrefs.getString('accessToken');

  // Assicurati che l'access token non sia null
  expect(accessToken, 'dummyAccessToken');
});
```

Figure 94: Registrazione utente con dati validi (email statica)

ingsw_dietideals24_app

```
// Test di registrazione con dati validi per creazione dell'asta
test('Registrazione utente con dati validi per creazione dell\'asta', () => {
  final userProvider = UserProvider();
  const email = 'test_auction@example.com';

  // Crea un mock di SharedPreferences
  final mockPrefs = MockSharedPreferences();

  // Definisci cosa fa il mock quando viene chiamato getString
  when(mockPrefs.getString('accessToken')).thenReturn('dummyAccessToken');

  // Imposta il mock nelle SharedPreferences
  SharedPreferences.setMockInitialValues({});

  // Esegui la registrazione con dati validi
  await userProvider.register({
    'nome': 'Test',
    'cognome': 'User',
    'data_nascita': '1990-01-01',
    'email': email,
    'password': 'password123'
  });

  // Verifica che l'email restituita sia quella generata dinamicamente
  expect(userProvider.user?.email, email);

  // Recupera l'access token da SharedPreferences
  final accessToken = mockPrefs.getString('accessToken');

  // Assicurati che l'access token non sia null
  expect(accessToken, 'dummyAccessToken');
});
```

Figure 95: Registrazione utente con dati validi per creazione dell'asta

ingsw_dietideals24_app

```
// Test di registrazione con dati mancanti
test('Registrazione con dati incompleti', () async {
  final userProvider = UserProvider();

  // Esegui la registrazione con dati incompleti
  try {
    await userProvider.register({
      'nome': 'Test',
      'cognome': 'User',
      // 'data_nascita' mancante
      'email': 'testuser@example.com',
      'password': 'password123'
    });
    fail('La registrazione con dati incompleti dovrebbe generare un\'eccezione.');
  } catch (e) {
    // Verifica che l'errore contenga la stringa corretta
    expect(e.toString(), contains('Per favore inserisci tutte le informazioni richieste!'));
  }
});
```

Figure 96: Registrazione con dati incompleti

```
// Test di registrazione senza dati
test('Registrazione senza dati', () async {
  final userProvider = UserProvider();

  try {
    await userProvider.register({
      'nome': '',
      'cognome': '',
      'data_nascita': '',
      'email': '',
      'password': ''
    });

    fail('La registrazione senza dati dovrebbe generare un\'eccezione.');
  } catch (e) {
    expect(e.toString(), contains('Per favore inserisci tutte le informazioni richieste!'));
  }
});
```

Figure 97: Registrazione senza dati

```
// Login
group('User Login', () {
    // Test di login con credenziali corrette
    test('Login con credenziali corrette', () async {
        final userProvider = UserProvider();
        const email = 'testuser@example.com';
        const password = 'password123';

        // Mock di SharedPreferences
        final mockPrefs = MockSharedPreferences();
        when(mockPrefs.getString('accessToken')).thenReturn('dummyAccessToken');
        SharedPreferences.setMockInitialValues({});

        // Esegui login con credenziali corrette
        await userProvider.login(email, password);

        // Verifica che l'utente sia stato impostato correttamente nel provider
        expect(userProvider.user?.email, email);
        expect(userProvider.isLoading, false);

        // Verifica che l'access token sia stato salvato correttamente
        final accessToken = mockPrefs.getString('accessToken');
        expect(accessToken, 'dummyAccessToken');
    });

    // Test di login con email vuota
    test('Login con email vuota', () async {
        final userProvider = UserProvider();
        const email = '';
        const password = 'password123';

        // Simuliamo un errore nel login
        try {
            await userProvider.login(email, password);
            fail('Il login con un\'email vuota dovrebbe generare un\'eccezione.');
        } catch (e) {
            // Verifica che l'errore contenga la stringa corretta
            expect(e.toString(), contains('Login failed'));
        }
    });
});
```

Figure 98: Login con credenziali corrette

ingsw_dietideals24_app

```
// Test di login con email errata
test('Login con email errata', () async {
  final userProvider = UserProvider();
  const email = 'wrongemail@example.com';
  const password = 'password123';

  // Simuliamo un errore nel login
  try {
    await userProvider.login(email, password);
    fail('Il login con una email errata dovrebbe generare un\'eccezione.');
  } catch (e) {
    // Verifica che l'errore contenga la stringa corretta
    expect(e.toString(), contains('Login failed'));
  }
});
```

Figure 99: Login con email errata

```
// Test di login con password errata
test('Login con password errata', () async {
  final userProvider = UserProvider();
  const email = 'testuser@example.com';
  const password = 'wrongpassword';

  // Simuliamo un errore nel login
  try {
    await userProvider.login(email, password);
    fail('Il login con una password errata dovrebbe generare un\'eccezione.');
  } catch (e) {
    // Verifica che l'errore contenga la stringa corretta
    expect(e.toString(), contains('Login failed'));
  }
});

// Test di login con password vuota
test('Login con password vuota', () async {
  final userProvider = UserProvider();
  const email = 'testuser@example.com';
  const password = '';

  // Simuliamo un errore nel login
  try {
    await userProvider.login(email, password);
    fail('Il login con una password vuota dovrebbe generare un\'eccezione.');
  } catch (e) {
    // Verifica che l'errore contenga la stringa corretta
    expect(e.toString(), contains('Login failed'));
  }
});
```

Figure 100: Login con password errata

```
// Reset Password
group('Password Reset', () {
    // Test di reset della password con email valida
    test('Reset password con email valida', () async {
        final userProvider = UserProvider();
        const email = 'testuser@example.com';
        const newPassword = 'newpassword123';

        // Crea un mock di SharedPreferences
        final mockPrefs = MockSharedPreferences();

        // Imposta il mock nelle SharedPreferences
        SharedPreferences.setMockInitialValues({});

        try {
            await userProvider.resetPassword(email, newPassword);
        } catch (e) {
            fail('Il reset della password con email valida dovrebbe passare.');
        }
    });

    // Test di reset della password con email non valida
    test('Reset password con email non valida', () async {
        final userProvider = UserProvider();
        const email = 'wrongemail@example.com'; // Email non esistente
        const newPassword = 'newpassword123';

        try {
            await userProvider.resetPassword(email, newPassword);
            fail('Il reset della password con email non valida dovrebbe fallire.');
        } catch (e) {
            // Verifica che l'errore contenga la stringa corretta per email non valida
            expect(e.toString(), contains("L'email fornita non corrisponde a nessun account!"));
        }
    });
});
```

Figure 101: Reset password con email valida

ingsw_dietideals24_app

```
// Test di errore generico durante il reset della password
test('Errore generico durante il reset della password', () async {
  final userProvider = UserProvider();
  const email = 'testuser@example.com';
  const newPassword = 'newpassword123';

  // Simula un errore nel backend (es. fallimento della richiesta HTTP)
  try {
    // Simula un errore nella risposta HTTP
    throw Exception('Errore generico durante il reset della password');
    await userProvider.resetPassword(email, newPassword);
    fail('Il reset della password con errore generico dovrebbe fallire.');
  } catch (e) {
    // Verifica che l'errore contenga la stringa corretta
    expect(e.toString(), contains('Errore generico durante il reset della password'));
  }
});
```

Figure 102: Errore generico durante il reset della password

```
// Creazione asta all'inglese con login valido
test('Creazione asta all\'inglese', () async {
  final userProvider = UserProvider();
  final auctionProvider = AuctionProvider();

  const email = 'test_auction@example.com';
  const password = 'password123';

  // Simula il login per ottenere il token
  final mockPrefs = MockSharedPreferences();
  when(mockPrefs.getString('accessToken')).thenReturn('dummyAccessToken');
  SharedPreferences.setMockInitialValues({});

  // Login
  await userProvider.login(email, password);

  // Verifica che il login sia stato effettuato correttamente e che il token sia stato ottenuto
  expect(userProvider.user?.email, email);
  expect(userProvider.isLoading, false);

  // Dati dell'asta
  final auctionData = {
    'tipo': 'inglese',
    'data_scadenza': '2024-12-31T23:59:59Z',
    'prezzo_minimo': 100.0,
    'incremento_rialzo': 10.0,
    'prezzo_iniziale': 50.0,
    'stato': 'attiva',
    'titolo': 'Asta di prova all\'inglese',
    'descrizione': 'Descrizione dell\'asta di prova all\'inglese',
    'categoria_id': 1,
    'immagine_principale': null,
  };

  // Crea l'asta con il token ottenuto dal login
  await auctionProvider.createAuction(auctionData, null);

  // Verifica che l'asta sia stata creata con successo
  expect(auctionProvider.isLoading, false);
});
```

Figure 103: Creazione dell'asta all'inglese

ingsw_dietideals24_app

```
// Creazione asta a tempo fisso con login valido
test('Creazione asta a tempo fisso', () async {
  final userProvider = UserProvider();
  final auctionProvider = AuctionProvider();

  const email = 'test_auction@example.com';
  const password = 'password123';

  // Simula il login per ottenere il token
  final mockPrefs = MockSharedPreferences();
  when(mockPrefs.getString('accessToken')).thenReturn('dummyAccessToken');
  SharedPreferences.setMockInitialValues({});

  // Login
  await userProvider.login(email, password);

  // Verifica che il login sia stato effettuato correttamente e che il token sia stato ottenuto
  expect(userProvider.user?.email, email);
  expect(userProvider.isLoading, false);

  // Dati dell'asta
  final auctionData = {
    'tipo': 'tempo fisso',
    'data_scadenza': '2024-12-31T23:59:59Z',
    'prezzo_minimo': 100.0,
    'prezzo_iniziale': 400.0,
    'stato': 'attiva',
    'titolo': 'Asta di prova a tempo fisso',
    'descrizione': 'Descrizione dell\'asta di prova a tempo fisso',
    'categoria_id': 2,
    'immagine_principale': null,
  };

  // Crea l'asta con il token ottenuto dal login
  await auctionProvider.createAuction(auctionData, null);

  // Verifica che l'asta sia stata creata con successo
  expect(auctionProvider.isLoading, false);
});
```

Figure 104: Creazione dell'asta a tempo fisso

ingsw_dietideals24_app

```
// Creazione asta al ribasso con login valido
test('Creazione asta al ribasso', () async {
  final userProvider = UserProvider();
  final auctionProvider = AuctionProvider();

  const email = 'test_auction@example.com';
  const password = 'password123';

  // Simula il login per ottenere il token
  final mockPrefs = MockSharedPreferences();
  when(mockPrefs.getString('accessToken')).thenReturn('dummyAccessToken');
  SharedPreferences.setMockInitialValues({});

  // Login
  await userProvider.login(email, password);

  // Verifica che il login sia stato effettuato correttamente e che il token sia stato ottenuto
  expect(userProvider.user?.email, email);
  expect(userProvider.isLoading, false);

  // Dati dell'asta
  final auctionData = {
    'tipo': 'ribasso',
    'prezzo_iniziale': 1000.0,
    'decremento_prezzo': 50.0,
    'prezzo_minimo': 500.0,
    'timer_decremento': 1,
    'stato': 'attiva',
    'titolo': 'Asta di prova al ribasso',
    'descrizione': 'Descrizione dell\'asta di prova al ribasso',
    'categoria_id': 3,
    'immagine_principale': null,
  };

  // Crea l'asta con il token ottenuto dal login
  await auctionProvider.createAuction(auctionData, null);

  // Verifica che l'asta sia stata creata con successo
  expect(auctionProvider.isLoading, false);
});
```

Figure 105: Creazione dell'asta al ribasso

ingsw_dietideals24_app

```
// Creazione asta silenziosa con login valido
test('Creazione asta silenziosa', () async {
  final userProvider = UserProvider();
  final auctionProvider = AuctionProvider();

  const email = 'test_auction@example.com';
  const password = 'password123';

  // Simula il login per ottenere il token
  final mockPrefs = MockSharedPreferences();
  when(mockPrefs.getString('accessToken')).thenReturn('dummyAccessToken');
  SharedPreferences.setMockInitialValues({});

  // Login
  await userProvider.login(email, password);

  // Verifica che il login sia stato effettuato correttamente e che il token sia stato ottenuto
  expect(userProvider.user?.email, email);
  expect(userProvider.isLoading, false);

  // Dati dell'asta
  final auctionData = {
    'tipo': 'silenziosa',
    'data_scadenza': '2024-12-31T23:59:59Z',
    'prezzo_iniziale': 50.0,
    'stato': 'attiva',
    'titolo': 'Asta di prova silenziosa',
    'descrizione': 'Descrizione dell\'asta di prova silenziosa',
    'categoria_id': 4,
    'immagine_principale': null,
  };

  // Crea l'asta con il token ottenuto dal login
  await auctionProvider.createAuction(auctionData, null);

  // Verifica che l'asta sia stata creata con successo
  expect(auctionProvider.isLoading, false);
});
```

Figure 106: Creazione dell'asta silenziosa

ingsw_dietideals24_app

```
// Creazione asta con parametri mancanti
test('Creazione asta con parametri mancanti', () async {
  final userProvider = UserProvider();
  final auctionProvider = AuctionProvider();

  const email = 'test_auction@example.com';
  const password = 'password123';

  // Simula il login per ottenere il token
  final mockPrefs = MockSharedPreferences();
  when(mockPrefs.getString('accessToken')).thenReturn('dummyAccessToken');
  SharedPreferences.setMockInitialValues({});

  // Login
  await userProvider.login(email, password);

  // Verifica che il login sia stato effettuato correttamente e che il token sia stato ottenuto
  expect(userProvider.user?.email, email);
  expect(userProvider.isLoading, false);

  // Dati dell'asta
  final auctionData = {
    'tipo': 'inglese',
  };

  // Crea l'asta con il token ottenuto dal login
  await auctionProvider.createAuction(auctionData, null);

  // Verifica che l'asta sia stata creata con successo
  expect(auctionProvider.isLoading, false);
});

});
```

Figure 107: Creazione dell'asta con parametri mancanti

Testing Whitebox con SonarQube

Nel processo di sviluppo di BidHub, è stato implementato un rigoroso approccio di testing whitebox per garantire la qualità del codice e la stabilità del sistema. Questo tipo di test si basa sull'analisi interna del codice sorgente, con l'obiettivo di identificare ed eliminare eventuali difetti strutturali, vulnerabilità di sicurezza e violazioni di standard di programmazione.

Per eseguire questa attività, è stato utilizzato **SonarQube**, una piattaforma leader per l'analisi statica del codice. Grazie a SonarQube, sono stati analizzati i seguenti aspetti del codice:

- **Qualità del Codice:** verifica della conformità agli standard di programmazione e identificazione di code smell.
- **Sicurezza:** rilevamento di potenziali vulnerabilità e problemi di sicurezza.
- **Copertura del Codice:** monitoraggio del livello di test unitari rispetto al codice scritto.
- **Manutenibilità:** valutazione della complessità e del livello di duplicazione del codice.

I report generati da SonarQube hanno guidato il team di sviluppo nell'adozione di misure correttive rapide, migliorando la solidità e la manutenibilità del software. L'approccio whitebox, combinato con l'utilizzo di strumenti avanzati come SonarQube, ha permesso di costruire una base di codice pulita, efficiente e sicura per BidHub.

Questo processo ha rappresentato un elemento fondamentale per garantire un'esperienza utente affidabile e un'elevata qualità del prodotto finale.

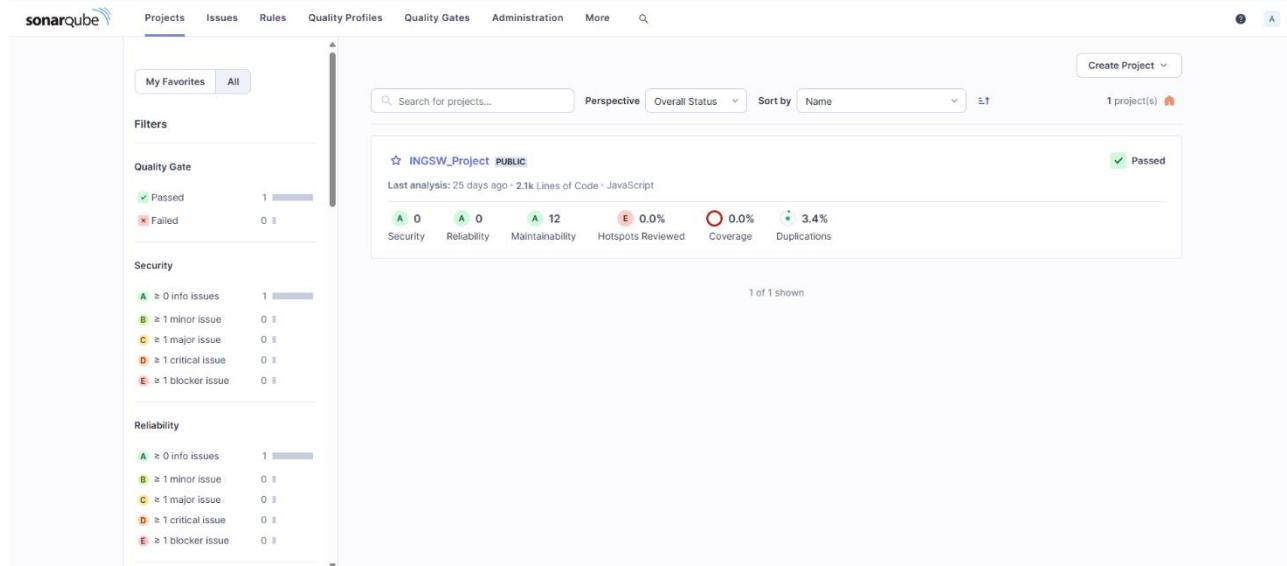


Figura 108: Report SonarQube

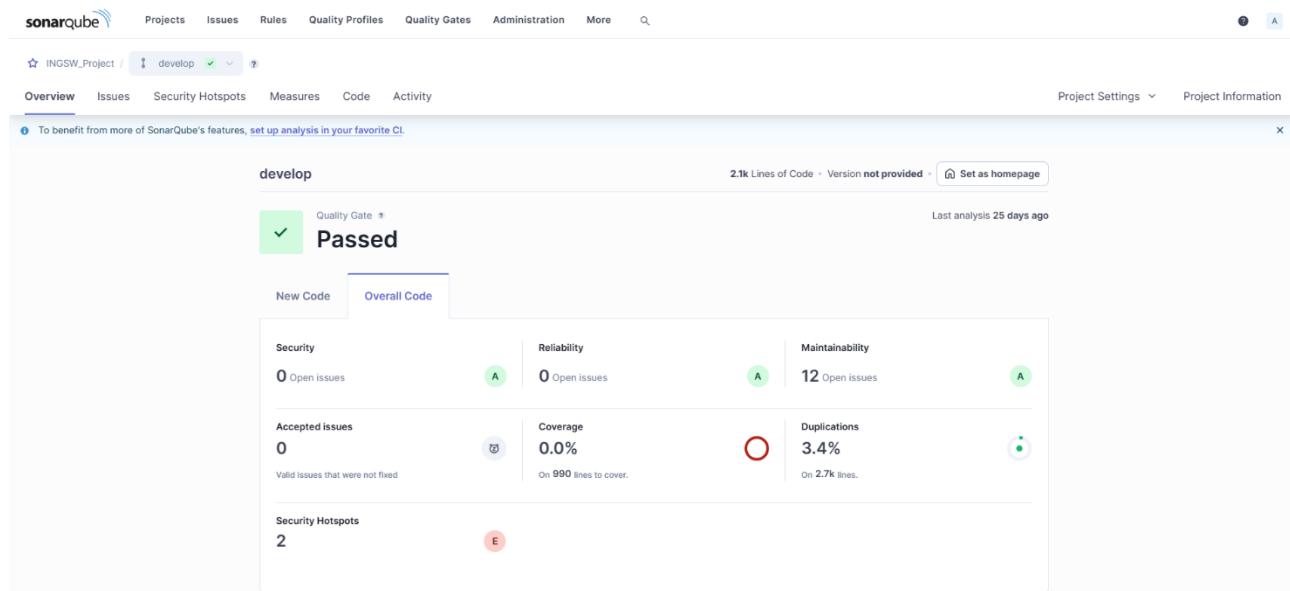


Figura 109 : Report SonarQube

VALUTAZIONE DELL'USABILITÀ SUL CAMPO

Questa sezione descrive le attività di valutazione dell'usabilità e di analisi dei file di log per il sistema di gestione delle aste sviluppato. L'obiettivo di queste attività è quello di valutare l'efficacia, l'efficienza e la soddisfazione dell'utente nell'utilizzo dell'applicazione, individuando possibili aree di miglioramento per ottimizzare l'esperienza dell'utente.

OBIETTIVO DELLA VALUTAZIONE

La valutazione dell'usabilità sul campo ha l'obiettivo di analizzare come gli utenti reali interagiscono con l'applicazione in un ambiente controllato e ottenere un feedback diretto sull'esperienza dell'utente.

METODOLOGIA

Sono stati selezionati 5 utenti rappresentativi del target del sistema per completare i task principali dell'applicazione. I task assegnati agli utenti sono i seguenti:

1. Creazione di un'asta ;
2. Visualizzare l'elenco delle proprie aste attive ;
3. Cancellare un'asta attiva ;

Durante l'interazione, sono stati registrati tempi di completamento, difficoltà riscontrate e suggerimenti di miglioramento.

RISULTATI DELLA VALUTAZIONE

TASK	Tempo Medio di Completamento	Difficoltà Riscontrate	Suggerimenti di Miglioramento
Creazione di un'asta	1 minuto	Due utenti su 5 hanno avuto difficoltà a trovare il pulsante la pagina dedicata alla creazione dell'asta (essa presente nella pagina account, basta cliccare sull'omino presente nella navbar)	Inserire linee guida all'avvio dell'applicazione in modo da potersi orientare meglio all'interno dell'applicazione
Visualizzare l'elenco delle proprie aste attive	30 secondi	Nessuna difficoltà riscontrata	Nessuna modifica suggerita
Cancellare un'asta attiva	40 secondi	Un utente ha cancellato un'asta per errore, credendo di modificarne i dettagli	Aggiungere un avviso di conferma per ridurre errori involontari

CONCLUSIONI DELLA VALUTAZIONE

L'osservazione ha rivelato che gli utenti sono stati generalmente in grado di completare i task assegnati, ma è emerso un problema di visibilità per la sezione della creazione dell'asta e la necessità di aggiungere un messaggio di conferma per la cancellazione di un'asta. Questi suggerimenti di miglioramento sono stati inclusi per ottimizzare l'esperienza utente.

ANALISI DEI FILE DI LOG

OBIETTIVO DELL'ANALISI

L'analisi dei file di log permette di ottenere informazioni oggettive sull'uso effettivo dell'applicazione e individuare i pattern d'uso, i task più frequenti e i possibili errori.

CONFIGURAZIONE E RACCOLTA DEI LOG

Per tracciare le interazioni degli utenti, è stato implementato un sistema di logging nel backend Node.js utilizzando il modulo Winston. Gli eventi principali registrati includono:

- Creazione di un'asta ;
- Visualizzare l'elenco delle proprie aste attive ;
- Cancellare un'asta attiva ;

Esempio di configurazione del logging nel backend (Node.js) :

```
const winston = require('winston');
const logger = winston.createLogger({
  level: 'info',
  format: winston.format.json(),
  transports: [
    new winston.transports.File({ filename: 'app.log' }),
    new winston.transports.File({ filename: 'errors.log', level: 'error' })
  ],
});
```

Figure 110: Snippet di codice relativo alla configurazione del logging nel backend

ESTRATTO DEL FILE DI LOG

Di seguito un estratto del file « **app.log** » con le interazioni utente :

```
{
  "timestamp": "2024-11-01T08:15:22Z",
  "level": "info",
  "message": "Asta creata dall'utente: 79",
  "endpoint": "/api/auctions/create",
  "params": { "titolo": "Fumetto Spiderman", "prezzo_iniziale": 500, "data_scadenza": "2024-11-15T10:00:00Z" }
}
{
  "timestamp": "2024-11-01T09:05:47Z",
  "level": "info",
  "message": "Visualizzazione aste attive richiesta dall'utente: 79",
  "endpoint": "/api/auctions/user/active"
}
{
  "timestamp": "2024-11-01T11:22:05Z",
  "level": "info",
  "message": "Asta 5678 cancellata dall'utente: 79",
  "endpoint": "/api/auctions/5678"
}
```

Figure 111: Estratto del file di log

ANALISI DEI DATI DI LOG

Azione	Frequenza	Note
Creazione asta	10 volte	Gli utenti hanno completato l'operazione senza problemi
Visualizzazione aste attive	15 volte	L'azione di visualizzazione è risultata la più utilizzata.
Cancellazione asta	2 volte	Un utente ha riportato di una cancellazione accidentale

CONCLUSIONI DELL'ANALISI

L'analisi dei file di log ha confermato che la funzionalità di **visualizzazione delle aste** è la più utilizzata. Sono emerse alcune cancellazioni accidentali, suggerendo la necessità di un ulteriore controllo di conferma prima dell'eliminazione di un'asta.

ALLEGATI

- **File di log** : Viene allegato una porzione del file **app.log** contenente tutti gli eventi di utilizzo per un periodo di test di 15 giorni ;

CONCLUSIONI GENERALI

La valutazione dell'usabilità e l'analisi dei file di log hanno evidenziato alcuni aspetti migliorabili dell'applicazione. Implementare i suggerimenti derivati da queste analisi aiuterà a:

- Migliorare l'accessibilità dei pulsanti principali.
- Ridurre gli errori accidentali con conferme per le operazioni di cancellazione.

Questi miglioramenti renderanno l'interfaccia più intuitiva e ridurranno le frustrazioni degli utenti, garantendo una migliore esperienza d'uso.