



GIUGNO 2025

Software Engineering

Indice

1	Introduzione	4
1.1	Struttura della documentazione	4
2	Glossario	6
3	Descrizione dell'idea progettuale	7
3.1	Modello dei Casi d'uso	8
3.2	Tabella dei Casi d'uso:	8
3.3	Descrizione Dettagliata dei casi d'uso:	9
3.4	Use Case Diagrams	11
3.5	Vettoriali Use Case Diagrams.....	13
3.6	Personas	14
3.7	Requisiti non-funzionali.....	19
3.8	Requisiti di dominio.....	20
4	Descrizione Testuali Struturate	20
4.1	Tabella di Cockburn	21
4.2	Mockup.....	25
5	Analisi dell'architettura	48
5.1	Criteri di Design Adottati	48
5.2	Architettura Client	49
5.3	Architettura Server	50
5.4	Descrizione e motivazione delle scelte tecnologiche adottate.....	51
5.5	Descrizione dello schema per la persistenza dati.....	52
5.6	Dizionario delle entità e delle associazioni.....	53
5.7	Definizioni delle tabelle	54
5.8	Definizioni dei vincoli.....	56
5.9	Definizioni dei Trigger e delle Funzioni.....	57
5.10	Schema UML.....	58
5.11	Descrizione e motivazione delle scelte di design dell'interfaccia utente adottate.....	59

5.12	Diagramma delle classi di design.....	60
5.13	Diagramma di sequenza di design per i casi d'uso utilizzati	64
6	Testing e valutazione dell'usabilità.....	69
6.1	Interfaccia Utente dedicata all'esecuzione di ricerche avanzate	70
6.2	Inserimento Annuncio da parte dell'Agente	71
6.3	Richiesta Visita Immobiliare da parte dell'Utente	72
6.4	Validazione Offerta Prezzo Utente.....	73
6.5	Tecniche di valutazione dell'usabilità applicate.....	74
6.6	Descrizione dei soggetti, della procedura sperimentale e delle metriche considerate	76
6.7	Procedura sperimentale	76
6.8	Definizione di un survey somministrato agli utenti dopo l'esperimento	76
6.9	Utenti Selezionati	79
6.10	Presentazione e discussione dei risultati dell'esperimento e del survey.....	84

1 Introduzione

DietiEstates25 è una piattaforma per la gestione di servizi immobiliari. Il sistema permette a più agenzie di pubblicare inserzioni immobiliari. Gli utenti possono quindi visualizzare le inserzioni, prenotare visite e fare offerte per acquistare/affittare un immobile. Il sistema consiste in un'applicazione, attraverso la quale gli utenti possono fruire delle funzionalità.

1.1 Struttura della documentazione

Il seguente documento è suddiviso in tre sezioni:

Documento dei requisiti:

In questa fase iniziale viene effettuato uno studio approfondito delle richieste del cliente, con l'obiettivo di definire con precisione tutti i requisiti necessari per lo sviluppo del software.

Documento di Design del Sistema:

Questa fase illustra e motiva tutte le decisioni tecnologiche adottate per la realizzazione del sistema.

Testing e valutazione dell'usabilità:

Describe come il sistema finale è stato sottoposto a test e analizza il suo utilizzo dal punto di vista dell'esperienza dell'utente.

PARTE I

Documento dei Requisiti

2 Glossario

Use Case Diagram	Diagramma utilizzato per rappresentare graficamente le interazioni tra gli attori (utenti o sistemi) e le funzionalità di un sistema. È composto da attori, casi d'uso e relazioni.
Class Diagram	Un class diagram è un diagramma UML utilizzato per rappresentare la struttura statica di un sistema, mostrando le classi, i loro attributi, metodi e le relazioni tra di esse. È uno strumento fondamentale per progettare il modello concettuale di un'applicazione.
Sequence Diagram	Un sequence diagram è un diagramma UML che rappresenta le interazioni tra gli oggetti in un sistema seguendo una sequenza temporale. Mostra come i messaggi vengono inviati e ricevuti tra gli attori (utenti o sistemi esterni) e i componenti del sistema durante l'esecuzione di un caso d'uso.
State Chart	Un diagramma degli stati rappresenta il ciclo di vita di un oggetto nel sistema. Mostra gli stati possibili in cui l'oggetto può trovarsi e gli eventi che causano il passaggio da uno stato all'altro.
Wireframe	Un wireframe è una rappresentazione schematica e semplificata del layout di una pagina o di un'interfaccia utente. È utilizzato durante la fase iniziale di progettazione per definire la struttura e l'organizzazione degli elementi, senza concentrarsi sugli aspetti visivi o estetici.
Stakeholder	Gli stakeholder sono le persone, i gruppi o le organizzazioni che hanno un interesse diretto o indiretto nel progetto e nel suo esito. Possono influenzare o essere influenzati dagli obiettivi, dallo sviluppo e dal risultato del progetto.
Mockup	Una rappresentazione statica e visiva dell'interfaccia utente, utilizzata per mostrare il layout e il design dell'applicazione. Non è interattivo, ma aiuta a comunicare l'aspetto grafico e le funzionalità.
API	(Application Programming Interface) Un insieme di regole e protocolli che permettono a diversi software di comunicare tra loro. Le API facilitano l'integrazione di funzionalità esterne o la connessione tra il Front end e il Back end.
Norme ISO	Standard internazionali stabiliti dall'International Organization for Standardization (ISO) che garantiscono qualità, sicurezza e affidabilità.
Policy	Insieme di regole, linee guida o procedure definite per garantire un comportamento coerente e conforme agli obiettivi aziendali o normativi.

3 Descrizione dell'idea progettuale

Per assicurare il successo di DietiEstates25, è essenziale acquisire una piena comprensione dei requisiti del sistema. L'analisi dei requisiti sarà focalizzata su:

- Il sistema deve assicurare una chiara separazione dei ruoli tra i diversi attori coinvolti: amministratore, gestore dell'agenzia immobiliare, agente immobiliare e utente finale.
- Descrizione dettagliata delle funzionalità principali:

Amministratore:

Fornisce credenziali predefinite ai gestori delle agenzie immobiliari e crea account specifici per il supporto amministrativo.

Gestore dell'agenzia:

Il gestore dell'agenzia immobiliare crea account dedicati per gli agenti immobiliari.

Agenti immobiliari:

Inseriscono nuovi immobili e li classificano come disponibili per la vendita o per l'affitto.

Utenti:

Gli utenti registrati hanno accesso a tutte le funzionalità della piattaforma: possono prenotare visite, fare offerte per acquistare o affittare un immobile, visualizzare le inserzioni, effettuare ricerche avanzate e applicare filtri per posizione geografica. Gli utenti non registrati, invece, possono solo visualizzare le inserzioni e utilizzare la funzione di ricerca avanzata.

- Definizione del target utente
- Prototipazione dell'applicazione attraverso la creazione di mock-up dell'interfaccia utente. Questo processo di analisi approfondita ci permetterà di sviluppare un sistema solido e perfettamente in linea con le esigenze specifiche degli utenti di DietiEstates25.

3.1 Modello dei Casi d'uso

Lo scopo del modello dei casi d'uso è delineare in modo preciso, ma chiaro e schematico, l'insieme delle funzionalità e degli attori coinvolti nel sistema, grazie all'uso di Use Case Diagrams, i diagrammi risultano facilmente comprensibili.

Un ulteriore obiettivo di questi modelli è definire il confine del sistema in fase di sviluppo, specificando le funzionalità che il sistema dovrà implementare e quelle che invece dovrà escludere.

Nella fase iniziale, è cruciale identificare gli attori coinvolti e tutte le funzionalità che il sistema deve garantire.

Durante questa fase preliminare, è stata realizzata una tabella che riporta gli attori e i rispettivi casi d'uso.

Nella pagina successiva è presentata la tabella dei casi d'uso e dei relativi attori.

3.2 Tabella dei Casi d'uso:

Attore	Casi d'uso
Amministratore	- Fornisce credenziali ai gestori - Crea account di supporto
Gestore dell'agenzia	- Crea Account per Agenti immobiliari
Agente Immobiliare	- Gestisce Visite - Inserisce Immobile - Gestisce offerta - Statistiche sugli immobili
Utente Registrato	- Visualizza Inserzioni - Prenota Visite - Presenta Offerta - Effettua Ricerca Avanzata - Ricerca per Posizione Geografica - Visualizza immobili su mappa interattiva
Utente Non registrato	- Visualizza inserzioni - Ricerca per posizione Geografica - Effettua Ricerca Avanzata - Effettua Registrazione

3.3 Descrizione Dettagliata dei casi d'uso:

Le seguenti descrizioni hanno l'obiettivo di trasformare le richieste generali dell'utente in requisiti di sistema più dettagliati e precisi.

Amministratore:

L'amministratore ricopre un ruolo fondamentale nella gestione del sistema, con diverse responsabilità cruciali per il corretto funzionamento della piattaforma.

- **Fornisce credenziali ai gestori:** L'amministratore fornisce al gestore dell'agenzia immobiliare le credenziali, comprensive di un indirizzo email e una password predefinita dal sistema. Il gestore avrà la possibilità di modificare la password in seguito, al fine di garantire una maggiore sicurezza nell'accesso.
- **Crea account di supporto:** L'amministratore è responsabile della creazione di account di supporto, che consentono di ricevere assistenza da altri amministratori, semplificando così il suo lavoro.

Gestore dell'agenzia:

Il gestore dell'agenzia immobiliare ha un ruolo chiave nella gestione quotidiana del sistema, con diverse responsabilità legate alla supervisione e organizzazione delle operazioni.

- **Crea Account per Agenti immobiliari:** Il gestore dell'agenzia immobiliare ha la possibilità di creare un account dedicato, tramite il quale può assegnare agli agenti immobiliari il compito di gestire l'inserimento di nuovi immobili nel sistema. Gli agenti infatti potranno caricare in autonomia gli immobili disponibili per la vendita o l'affitto, con tutte le informazioni necessarie. Questo processo semplifica e velocizza la gestione delle proprietà, permettendo una gestione più fluida e organizzata dell'intero catalogo immobiliare.

Agente Immobiliare:

Gli agenti immobiliari sono responsabili dell'inserimento e della gestione degli immobili nel sistema, contribuendo in modo fondamentale alla completa operatività della piattaforma.

- **Gestisce Visite:** l'agente immobiliare organizza e monitora le visite degli utenti, mantenendo uno storico dettagliato delle prenotazioni effettuate.
- **Inserisce Immobile:** Il sistema consente di gestire immobili destinati alla vendita o all'affitto, fornendo tutte le informazioni necessarie per aiutare gli utenti nella scelta. Ogni immobile include dettagli come prezzo, dimensioni, numero di stanze, piano, classe energetica, e servizi accessori. Inoltre la posizione geografica esatta è associata a una mappa interattiva, permettendo agli utenti di esplorare la zona e valutare l'ubicazione dell'immobile.
- **Gestisce Offerta:** l'agente immobiliare gestisce le offerte dei clienti, invia nuove proposte, registra manualmente le offerte ricevute al di fuori del sistema e mantiene uno storico dettagliato di tutte le offerte effettuate.
- **Statistiche sugli immobili:** l'agente immobiliare accede a una dashboard con dati statistici

Utente Registrato:

L'utente registrato ha accesso a un ampio set di funzionalità avanzate che migliorano l'esperienza di ricerca, interazione e gestione delle richieste immobiliari.

- **Visualizza Inserzioni:** L'utente può esplorare tutte le inserzioni disponibili, sia per la vendita che per l'affitto. Ogni inserzione è accompagnata da dettagli completi come descrizione, prezzo, caratteristiche tecniche, foto e servizi accessori.
- **Prenota Visite:** L'utente ha la possibilità di prenotare visite agli immobili. È possibile selezionare date e orari in base alla disponibilità del gestore o degli agenti immobiliari.
- **Presenta Offerta:** Per gli immobili in vendita, l'utente può inoltrare un'offerta economica direttamente tramite la piattaforma, indicando eventuali condizioni o richieste particolari.
Per gli immobili in affitto l'utente può proporre un'offerta, specificando durata, modalità di pagamento e altre condizioni che ritiene rilevanti.
- **Effettua Ricerca Avanzata:** Il sistema offre un motore di ricerca avanzato che permette di filtrare gli immobili in base a criteri come prezzo, dimensioni, numero di stanze, classe energetica, piano e servizi accessori.
- **Ricerca Per Posizione Geografica:** Gli utenti possono restringere la ricerca selezionando una specifica area geografica. È possibile filtrare per città, quartiere o vicinanze di un punto d'interesse.
- **Visualizza Immobili Mappa Interattiva:** Gli immobili vengono visualizzati su una mappa interattiva, che consente di esplorare la posizione esatta di ogni proprietà. La mappa include funzionalità come zoom, calcolo delle distanze e visualizzazione dei servizi nelle vicinanze.

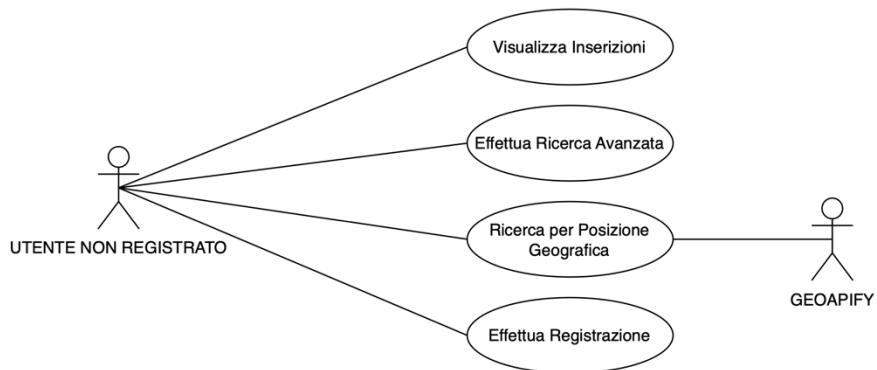
Utente Non Registrato:

Gli utenti non registrati hanno accesso limitato al sistema, ma possono comunque usufruire delle seguenti funzionalità per esplorare e familiarizzare con il servizio.

- **Visualizza Inserzioni:** Gli utenti possono sfogliare le inserzioni disponibili, visualizzando informazioni di base sugli immobili in vendita e in affitto, come descrizioni generali, foto e prezzi indicativi.
- **Effettua Ricerca Avanzata:** È possibile utilizzare il motore di ricerca avanzato per filtrare le inserzioni in base a criteri come prezzo, dimensioni, posizione e altre caratteristiche generali.
- **Ricerca Per Posizione Geografica:** Gli utenti possono restringere la ricerca selezionando una specifica area geografica. È possibile filtrare per città, quartiere o vicinanze di un punto d'interesse.
- **Effettua Registrazione:** Gli utenti non registrati sono incoraggiati a completare la registrazione per sbloccare l'accesso a tutte le funzionalità dedicate agli utenti registrati, tra cui la possibilità di prenotare visite, inviare offerte e comunicare direttamente con gli agenti immobiliari.

3.4 Use Case Diagrams

I diagrammi dei casi d'uso sono strumenti grafici utilizzati per rappresentare le interazioni tra gli attori e il sistema, evidenziando le principali funzionalità che il sistema offre. Questi diagrammi sono fondamentali per comprendere la struttura e il flusso delle operazioni, fornendo una visione chiara e concisa delle funzionalità disponibili per gli utenti del sistema.





3.5 Vettoriali Use Case Diagrams

È possibile visionare l'immagine più definita tramite i vettoriali dei diagrammi, che permettono una visualizzazione scalabile e dettagliata, utile per l'analisi approfondita e la documentazione del progetto.

- I seguenti link sono:

[USES CASE DIAGRAMS - UTENTI NON REGISTRATI](#)

[USES CASE DIAGRAMS - UTENTI REGISTRATI](#)

3.6 Personas

Le Personas sono rappresentazioni dettagliate di utenti, costruite utilizzando dati reali raccolti con ricerche, sondaggi e osservazioni dirette. Attraverso le Personas, si ottiene una comprensione più approfondita di chi utilizzerà l'applicazione, come interagirà con essa e quali sono le sue aspettative. Questo approccio permette di costruire prodotti e servizi che non solo soddisfano i requisiti funzionali, ma che si adattano perfettamente alle necessità degli utenti finali, migliorando la loro esperienza complessiva.

Francesco Moretti:

- È l'amministratore dell'agenzia immobiliare e ha il compito di gestire gli accessi al sistema e assegnare gli account ai gestori. La sua principale responsabilità è assicurarsi che gli agenti abbiano accesso alle funzionalità necessarie per svolgere il loro lavoro, senza compromettere la sicurezza e l'integrità dei dati aziendali. Francesco ha una visione strategica e comprende l'importanza di una gestione centralizzata degli utenti e delle risorse. Inoltre, apprezza l'automazione e l'efficienza, quindi ogni funzionalità che riduca il tempo speso in operazioni manuali è un valore aggiunto.

Marco Rossi:

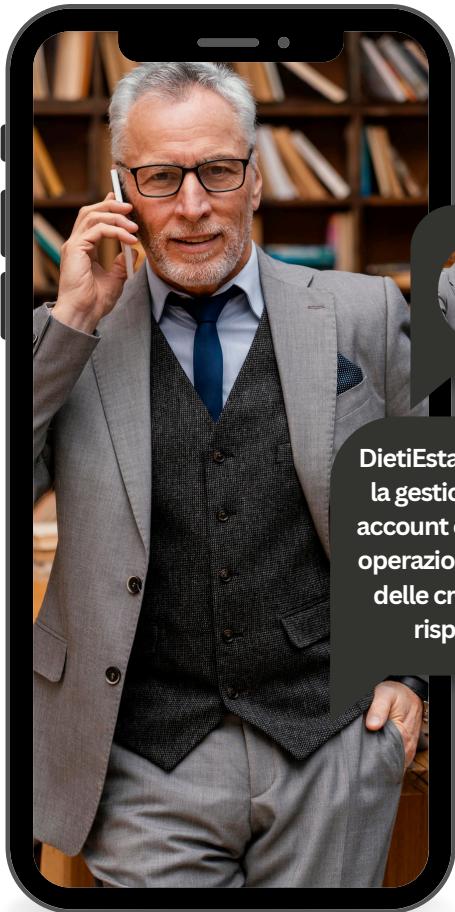
- È il gestore di un'agenzia immobiliare, con oltre 15 anni di esperienza. Sebbene esperto nel settore, è particolarmente interessato a trovare offerte vantaggiose, quindi l'aspetto economico è uno dei suoi principali punti di interesse. Apprezza strumenti che gli consentano di ottimizzare la gestione del tempo, quindi ogni funzionalità che permetta di fare ricerche rapide e precise è fondamentale. Inoltre, essendo molto focalizzato sugli affari, gradirebbe un sistema che metta in evidenza in modo chiaro le opportunità economiche, consentendogli di visualizzare rapidamente offerte e sconti.

Alessandra Bianchi:

- È una nuova utente nell'ambito delle agenzie immobiliari e delle piattaforme online per la gestione di immobili. La sua esperienza è limitata, e ha bisogno di un approccio semplice e didattico per navigare nel sistema. Apprezza un'interfaccia chiara, che non la sovraccarichi di informazioni, permettendole di apprendere in modo progressivo e senza troppa frenesia. La sua esperienza di utente si arricchirebbe se l'app presentasse tutorial e suggerimenti per migliorare la sua comprensione delle operazioni, rendendo la curva di apprendimento meno ripida.

Giulia Conti:

- È una giovane professionista che lavora nel settore immobiliare, con una mentalità orientata alla crescita e all'innovazione. Utilizza la piattaforma non solo per acquistare immobili, ma anche per promuovere la sua attività e costruire una rete di contatti. La sua propensione all'innovazione richiede che l'app consenta agli utenti come lei di visualizzare lo storico delle offerte e degli immobili trattati, in modo da monitorare l'andamento della propria attività. Inoltre, la possibilità di personalizzare la propria esperienza attraverso funzionalità avanzate e suggerimenti mirati sui trend del mercato potrebbe soddisfare la sua necessità di ottimizzare il suo approccio e prendere decisioni più informate. Un'altra funzione utile potrebbe essere l'integrazione con strumenti di marketing per amplificare la visibilità degli immobili.



FrancescoM ✅
@FranMoretti



DietiEstates25 è un'app che ha migliorato in modo significativo la gestione della nostra agenzia. La creazione e gestione degli account è semplice ed efficace, e l'interfaccia chiara rende ogni operazione intuitiva. Apprezzo molto l'attenzione alla sicurezza delle credenziali e l'organizzazione generale del sistema, che risponde pienamente alle esigenze del nostro lavoro.

Descrizione

Francesco Moretti è un gestore Senior con oltre 30 anni di esperienza nel settore immobiliare e un forte focus sulla gestione operativa e organizzativa. Come amministratore, il suo ruolo principale è quello di fornire gli account agli agenti e assicurarsi che l'accesso all'applicazione avvenga in modo sicuro e controllato. Francesco è abituato a prendere decisioni strategiche e a lavorare in stretta collaborazione con il suo team, pur mantenendo un forte senso di responsabilità e controllo.

Esperienza

Amministratore Agenzia Immobiliare | 2017 - 2025

Imprenditore

- Ha una carriera consolidata nel mondo immobiliare, partendo da un ruolo operativo per poi avanzare alla gestione dell'intera agenzia. Dal 2017 al 2025 ha ricoperto la posizione di amministratore presso la sua agenzia, dove ha ottimizzato l'intero flusso di gestione degli utenti e dei beni immobili.
- Ha acquisito competenze avanzate nella gestione delle piattaforme digitali e nella supervisione di team distribuiti, migliorando continuamente i processi aziendali.
- È esperto nell'utilizzo di software gestionali e ha implementato numerosi miglioramenti per aumentare l'efficienza del sistema, con un particolare focus sulla sicurezza dei dati e sulla gestione centralizzata degli accessi.

Criticità

Pur essendo un professionista esperto, potrebbe trovare delle difficoltà nella gestione di operazioni complesse in cui sono necessarie competenze tecnologiche avanzate, specialmente in ambiti che richiedono una continua evoluzione dei sistemi digitali. La sua preferenza per processi manuali lo rende meno incline ad adottare tecnologie che richiedono una configurazione automatica o un uso intensivo delle funzionalità digitali avanzate. Inoltre, potrebbe avere difficoltà nel gestire un team con diversi livelli di competenza tecnologica, specialmente se alcuni agenti non sono molto esperti nell'uso di piattaforme digitali complesse.

Skill

Sicurezza	★★★★★
Comunicazione	★★★★★
Leadership	★★★★★
Normative	★★★★★

Personalità

Responsabile	★★★★★
Socievole	★★★★★
Affidabile	★★★★★
Diplomatico	★★★★★

Marcos ✓
@MarcoRossi48

Trovo DietiEstates25 un'app eccellente per la gestione quotidiana dell'agenzia. L'interfaccia è intuitiva e facile da usare, il che semplifica il lavoro del team. Le funzionalità offerte sono complete e rispondono perfettamente alle nostre esigenze operative. Sono molto soddisfatto dell'app nel suo complesso.

Esperienza

Gestore Agenzia Immobiliare | 2010 - 2025

Imprenditore

- Gestore di un'agenzia immobiliare con oltre 15 anni di esperienza nel settore. Nel corso della sua carriera, ha gestito la parte amministrativa e strategica dell'agenzia, coordinando le attività di acquisto, vendita e affitto di immobili.
- Si occupa della supervisione del team di agenti immobiliari, della definizione delle strategie di marketing e della gestione delle relazioni con i clienti principali.
- Esperienza nella gestione delle risorse e nell'implementazione di nuove tecnologie per ottimizzare i processi aziendali.

Descrizione

Un gestore pragmatico e molto orientato agli obiettivi, con un forte focus sull'efficienza e sull'ottimizzazione delle operazioni aziendali. È molto organizzato, capace di gestire il team in modo disciplinato, pur mantenendo un ambiente di lavoro collaborativo. La sua visione strategica gli permette di prendere decisioni rapide e informate. È un leader rispettato per la sua capacità di risolvere i problemi e mantenere l'agenzia operativa in modo fluido.

Criticità

Tende a concentrarsi molto sui risultati a breve termine e sulle operazioni quotidiane, a volte tralasciando le opportunità di crescita a lungo termine o l'innovazione. La sua gestione molto pratica può anche ridurre la creatività del team, in quanto preferisce soluzioni rapide e concrete. Inoltre, essendo molto orientato alla produttività, potrebbe dedicare meno attenzione alla formazione continua del personale, che potrebbe limitare lo sviluppo delle competenze del team a lungo termine.

Skill

Leadership	★★★★★
Problem Solving	★★★★★
Marketing	★★★★★
Analisi Finanziaria	★★★★★

Personalità

Determinato	★★★★★
Competitivo	★★★★★
Razionale	★★★★★
Testardo	★★★★★



Esperienza

Agente Immobiliare Senior | 2017 - 2025

Agente Immobiliare

- Ha iniziato la sua carriera come agente immobiliare nel 2017, lavorando con clienti residenziali e specializzandosi nella ricerca di soluzioni per giovani professionisti e famiglie. Nel 2020 ha ottenuto una certificazione in Home Staging, migliorando la presentazione degli immobili e aumentando il successo nelle vendite.
- Nel 2022 è stata promossa ad agente senior, gestendo un portafoglio di immobili di lusso e facendo da mentore ai nuovi agenti. Con competenze avanzate in negoziazione e strumenti digitali, oggi Alessandra è una figura chiave nella sua agenzia, riconosciuta per il suo approccio dinamico e orientato al cliente.

Descrizione

Una professionista dinamica e determinata, con una forte passione per il settore immobiliare. Grazie alla sua capacità di creare connessioni autentiche con i clienti e al suo occhio per i dettagli, si distingue per la sua abilità nel trasformare semplici incontri in collaborazioni di successo.

Empatica e orientata al risultato, Alessandra è sempre pronta a trovare soluzioni innovative per soddisfare le esigenze dei suoi clienti. La sua esperienza e dedizione la rendono un punto di riferimento per chi cerca un agente affidabile, competente e capace di gestire anche le situazioni più complesse.

Criticità

Nonostante la sua esperienza nel settore immobiliare, potrebbe avere difficoltà nell'adattarsi rapidamente a nuovi strumenti tecnologici o gestionali più complessi. Potrebbe anche sentirsi poco sicura nella gestione di trattative molto tecniche o nell'interazione con clienti particolarmente esigenti, soprattutto in ambiti che richiedono competenze specifiche al di fuori della sua formazione, come analisi di mercato avanzate o marketing digitale.

Skill

Gestione del Tempo	★★★★★
Vendita	★★★★★
Tecnologie	★★★★★
Networking	★★★★★

Personalità

Empatica	★★★★★
Energica	★★★★★
Determinata	★★★★★
Comunicativa	★★★★★



Esperienza

Digital Marketing Junior | 2020 - 2025

Marketing Digitale

- Una giovane professionista nel settore del marketing digitale, con una carriera avviata nel 2020. Attualmente lavora come freelance, collaborando con aziende di diverse dimensioni per sviluppare strategie di branding e campagne online. Vive in una grande città e cerca il suo primo appartamento da acquistare, desiderando uno spazio moderno che rifletta il suo stile di vita dinamico e creativo. La sua routine è frenetica, e per questo predilige strumenti che semplifichino il processo di ricerca e acquisto.

Descrizione

Una persona intraprendente e curiosa, sempre alla ricerca di nuove soluzioni per migliorare la propria vita. Ama approfondire ogni dettaglio prima di prendere decisioni importanti e si sente più sicura quando dispone di strumenti intuitivi e informazioni precise. È tecnologicamente competente e abituata a utilizzare app e piattaforme digitali per risolvere problemi quotidiani. La sua determinazione e creatività la rendono capace di affrontare sfide con entusiasmo.

Criticità

Nonostante la sua familiarità con la tecnologia, Giulia tende a essere indecisa quando si trova di fronte a molte opzioni, il che può rallentare il processo decisionale. Inoltre, potrebbe sentirsi sopraffatta se le informazioni non sono presentate in modo chiaro o organizzato. La mancanza di esperienza nel mercato immobiliare la rende insicura nelle trattative, spingendola a cercare supporto esterno per evitare errori.

Skill

Social Network	★★★★★
Carisma	★★★★★
Comunicazione	★★★★★
Ricerca e Valutazione	★★★★★

Personalità

Curiosa	★★★★★
Paziente	★★★★★
Collaborativa	★★★★★
Riflessiva	★★★★★

3.7 Requisiti non-funzionali

I requisiti non funzionali sono caratteristiche qualitative di un sistema software che definiscono come il sistema deve comportarsi, piuttosto che cosa deve fare. Essi stabiliscono vincoli e criteri di qualità che il software deve rispettare per garantire un'esperienza ottimale agli utenti e soddisfare esigenze tecniche specifiche.

Di seguito elenco le specifiche per l'app Dietiestates 2025:

- **Prestazione**
Il cambio di pagina deve avvenire entro 2 secondi.
La ricerca avanzata non deve influire negativamente sulle prestazioni del sistema.
- **Scalabilità**
Il sistema deve essere in grado di gestire fino a 10.000 utenti attivi giornalieri senza degradare le prestazioni.
- **Affidabilità**
I dati devono essere salvati in un database con backup automatici giornalieri, mantenendo almeno 7 giorni di storico.
- **Sicurezza**
Deve essere garantita la protezione contro attacchi comuni come SQL injection, XSS e CSRF.
- **Usabilità**
L'interfaccia deve essere intuitiva e accessibile anche a utenti non esperti.
- **Manutenibilità**
Il codice deve essere documentato e strutturato per consentire interventi di manutenzione per risolvere al meglio i problemi.
Gli aggiornamenti del sistema vengono applicati in tempo reale e, ad ogni ricaricamento della pagina, vengono visualizzate nuove inserzioni o eventuali modifiche agli annunci esistenti.
- **Portabilità**
Il sistema deve essere ottimizzato per dispositivi mobili, garantendo la piena funzionalità su qualsiasi tipo di dispositivo, indipendentemente da dimensioni e sistema operativo.

3.8 Requisiti di dominio

I requisiti di dominio sono vincoli e regole specifiche di un particolare settore o ambito applicativo che il sistema deve rispettare. Sono spesso derivati da normative, regolamenti, best practice o convenzioni proprie del dominio in cui l'applicazione sarà utilizzata.

Di seguito elenco le specifiche per l'app Dietiestates 2025:

- **Gestione degli account**

Il sistema deve permettere al gestore dell'agenzia di creare account per gli agenti immobiliari e gestirne i permessi.

Il sistema deve permettere all'amministratore di poter gestire e d'inviare eventuali credenziali ai gestori delle agenzie.

Il sistema permette agli utenti registrati di poter prenotare visite e fare offerte per le inserzioni visualizzate.

Gli utenti non registrati possono solo visualizzare gli annunci, senza accedere a funzioni avanzate

- **Categorizzazione degli immobili**

Gli immobili devono essere classificati in vendita o affitto.

Ogni immobile deve includere dettagli specifici come: foto, descrizione, prezzo, dimensione, indirizzo, numero di stanze, piano, presenza di ascensore, classe energetica, ulteriori servizi presenti (parcheggio, portineria, giardino)

- **Ricerca avanzata**

Gli utenti devono poter cercare immobili filtrando per posizione geografica, prezzo, dimensione, numero di stanze, piano, presenza di ascensore, classe energetica, ulteriori servizi presenti (parcheggio, portineria, giardino)

- **Prenotazione visite**

Gli utenti devono poter prenotare visite in base alla disponibilità degli agenti immobiliari, che è possibile consultare sull'apposito calendario.

Gli utenti possono cancellare o modificare la loro prenotazione.

- **Gestione offerte**

Gli utenti devono poter inviare offerte per un immobile, con la possibilità di proporre un prezzo inferiore rispetto a quello indicato.

Le offerte saranno accettate o rifiutate dall'agente immobiliare.

- **API Mappe**

Grazie all'utilizzo di un API esterna, è possibile visualizzare gli annunci su una mappa interattiva, deve essere possibile mostrare la distanza dai punti di interesse, come scuole, parchi e trasporti pubblici.

4 Descrizione Testuali Strutturate

Le Descrizioni Testuali Strutturate sono una rappresentazione dettagliata e descrittiva dei casi d'uso, che forniscono una spiegazione precisa e chiara delle funzionalità del sistema. Queste descrizioni sono organizzate in modo da delineare ogni aspetto di un caso d'uso, inclusi gli attori coinvolti, il flusso degli eventi, le condizioni iniziali e finali, le eccezioni e altre informazioni pertinenti.

4.1 Tabella di Cockburn

Questo strumento aiuta a determinare il livello di dettaglio necessario per la documentazione dei casi d'uso, ottimizzando il processo di sviluppo e analisi dei requisiti.

USE CASE #1	Effettua Ricerca Avanzata		
Goal in Context	L'utente desidera individuare un immobile utilizzando filtri di ricerca		
Preconditions	L'utente si trova nella schermata principale		
Succes End Condition	Il sistema registra i filtri applicati ed esegue la ricerca dell'immobile.		
Failed End Condition	Non è stato trovato nessun'immobile		
Primary Actor	Utente		
Trigger	Pressione sul pulsante inizia nuova ricerca		
Main Scenario	Step n.	User	System
	1	Clicca inizia nuova ricerca	
	2		Mostra Tipologia Ricerca
	3	Inserisce nome della città desiderata nel campo ricerca	
	4		Mostra Visualizzazione Ricerca
	5	L'utente seleziona la città desiderata dall'elenco dei suggerimenti	
	6	Preme "Fine".	
	7		Mostra Filtri Avanzati
	8	L'utente seleziona l'opzione "Compra"	
	9	L'utente applica i vari filtri avanzati disponibili (tipologia, prezzo, superficie, bagni, piano, classe energetica, garage, ascensore, portineria, giardino)	
	10	L'utente preme il pulsante "Cerca"	
	11		Mostra Ricerca
Extensions #1	Step n.	User	System
	4.1 <Il sistema non mostra la città inserita nei suggerimenti.>	L'utente modifica l'input o inserisce nuovamente la città correttamente premendo "Fine"	
	4.2		Mostra Visualizzazione Ricerca
	7.1 <L'utente decide di non applicare i filtri avanzati>	L'utente salta la selezione filtri avanzati e preme direttamente "Cerca"	
	9.1 <L'utente applica troppi filtri e la ricerca diventa troppo restrittiva>		Mostra Errore Ricerca
	9.2	L'utente clicca sull'icona "Nuova Ricerca"	
	9.3		Mostra Tipologia Ricerca
	10.1 < L'utente si accorge di aver sbagliato i filtri dopo aver premuto "Cerca" >	L'utente clicca sull'icona filtri nella pagina Ricerca	
	10.2		Mostra Filtri Avanzati
	11.1 <Nessun immobile soddisfa i criteri impostati>		Mostra Errore Ricerca
	11.2	L'utente clicca sull'icona "Nuova Ricerca"	
			Mostra Tipologia Ricerca

USE CASE #2	Inserimento Immobile		
Goal in Context	L'agente desidera inserire l'annuncio di un nuovo immobile		
Preconditions	L'agente ha eseguito l'accesso		
Succes End Condition	Il sistema ha caricato il nuovo immobile.		
Failed End Condition	L'immobile non è stato caricato		
Primary Actor	Agente Immobiliare		
Trigger	Pressione sul pulsante Carica Immobile		
Main Scenario	Step n.	Agente Immobiliare	System
	1	Clicca su "Carica Immobile"	
	2		Mostra Tipologia di Immobile
	3	Clicca su Vendita/Affitto	
	4	Sceglie Tipologia di immobile	
	5	Clicca su "Avanti"	
	6		Mostra Requisiti Necessari
	7	Inserisce foto tramite l'icona "Carica"	
	8	Inserisce Titolo Annuncio	
	9	Inserisce Testo dell'annuncio con ulteriori dettagli dell'immobile	
	10	Inserisce Indirizzo, superficie, Prezzo	
	11	Clicca su "Avanti"	
	12		Mostra Requisiti Opzionali
	13	Inserisce Numero di stanze	
	14	Inserisce Numero Piano	
	15	Inserisce Numero Bagni	
	16	Seleziona una singola casella su Parcheggio	
	17	Inserisce Classe energetica	
	18	Seleziona una o più caselle su Ulteriori Servizi	
	19	Clicca su "Inserisci"	
Extensions #1	Step n.	Agente Immobiliare	Mostra Annuncio Inserito
			System
	7.1 <L'agente non carica immagini>		Mostra Alert – Requisiti Necessari
	10.1 <L'agente non inserisce il prezzo>		Mostra Alert – Requisiti Necessari
	12.1 <L'agente non inserisce nessun dato opzionale>		Mostra Alert – Requisiti Opzionali
	19.1 <Errore durante il caricamento dell'annuncio>		Mostra Alert – Errore Caricamento Annuncio

USE CASE #3	Prenota Visita		
Goal in Context	L'utente vuole prenotare una visita per il giorno e l'orario desiderato.		
Preconditions	L'utente ha effettuato la ricerca		
Success End Condition	Il sistema registra la prenotazione ed è possibile visualizzare nella sezione visite prenotate		
Failed End Condition	Non è stato trovato nessun'immobile.		
Primary Actor	Utente		
Trigger	Pressione sul pulsante Visita		
Main Scenario	Step n.	User	System
	1	L'utente seleziona il bottone "Visita"	
	2		Mostra Richiesta Visita
	3	L'utente seleziona la data e l'ora disponibili per la visita.	
	4	L'utente preme il pulsante "Proseguì"	
	5		Mostra Alert di Conferma
	6	L'utente visualizza il messaggio "Questa non è una prenotazione" e Preme "OK"	
	7		Mostra Riepilogo Visita
	8	L'utente preme il pulsante "Prenota"	
	9		Mostra Visita Inserita
Extensions #1	Step n.	User	System
	2.1 <L'utente decide di annullare la richiesta di visita>	L'utente non vuole più inserire la data e l'ora per richiedere l'appuntamento e preme "X"	
	2.2		Mostra Home Utente
	7.1a <L'utente vuole cambiare la prenotazione>	L'utente vuole modificare l'inserimento della data e dell'ora e preme la freccia indietro	
	7.2a		Mostra Richiesta Visita
	7.3b	L'utente non vuole più inserire la data e l'ora per richiedere l'appuntamento e preme "X"	
	7.4b		Mostra Home Utente

USE CASE #4	Presenta Offerta		
Goal in Context	L'utente vuole presentare un'offerta per un immobile disponibile		
Preconditions	L'utente ha effettuato la ricerca dell'immobile		
Success End Condition	Il sistema registra l'offerta correttamente.		
Failed End Condition	Non è stato trovato nessun'immobile.		
Primary Actor	Utente		
Trigger	Pressione sul pulsante Offerta		
Main Scenario	Step n.	User	System
	1	Clicca "Offerta"	
	2		Mostra Proponi Prezzo
	3	Visualizza il prezzo di partenza e propone il nuovo prezzo.	
	4	Clicca su "Prosegui"	
	5		Mostra Conferma Proposta
	6	Clicca su "Si Confermo"	
	7		Mostra Riepilogo Proposta
	8	L'utente visualizza il riepilogo delle attività e preme "Proponi"	
Extensions #1	9		Mostra Proposta Inserita
	Step n.	User	System
	2.1a <L'utente non vuole più offrire>	L'utente vuole tornare nell'home e clicca sulla freccia "indietro"	
	2.2a		Mostra Ricerca
	2.3b <L'utente decide di eliminare la proposta>	L'utente non vuole più effettuare la proposta e preme "X"	
Extensions #2	2.4b		Mostra Home Utente
	3.1 <L'utente inserisce un prezzo inferiore al 15% rispetto al prezzo richiesto.>		Mostra Alert – Variazione Maggiore
	5.1 <L'utente non vuole più effettuare la proposta>	L'utente vuole modificare o annullare la proposta e clicca su "Rifiuta"	
	5.2		Mostra Proponi Prezzo
	7.1a <L'utente vuole modificare l'offerta>	L'utente vuole modificare la proposta e clicca sulla freccia "indietro"	
	7.2a		Mostra Proponi Prezzo
	7.3b <L'utente non vuole più offrire>	L'utente non vuole più effettuare la proposta e preme "X"	
	7.4b		Mostra Home Utente

4.2 Mockup

Di seguito vengono rappresentati i Mockup dell'applicazione mobile derivati dalle descrizioni testuali (Cockburn) rappresentate nel capitolo precedente.

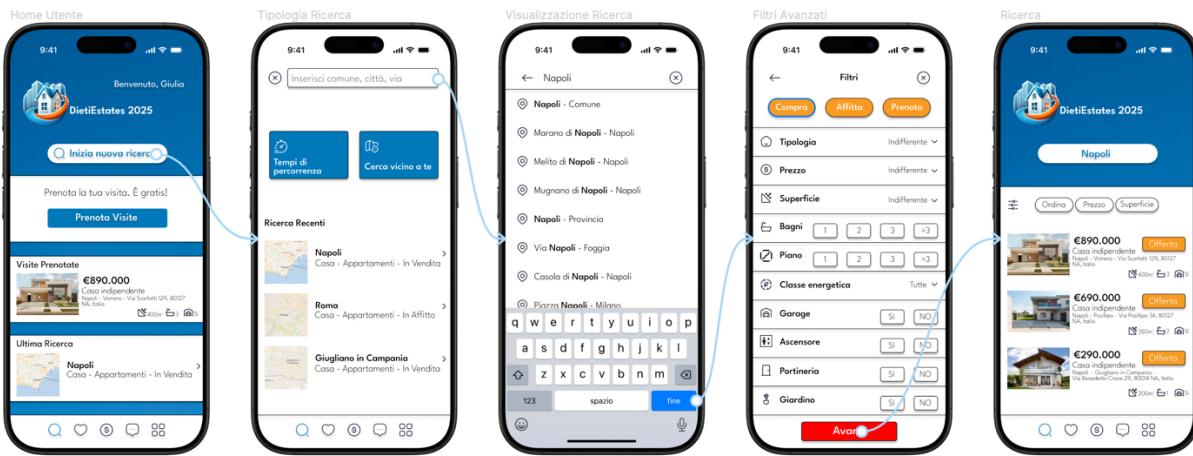
I Mockup presi in considerazioni sono quelli di media fedeltà.

Interfaccia principale con icona dell'applicazione "DietaEstates 2025" visibile sulla schermata home.



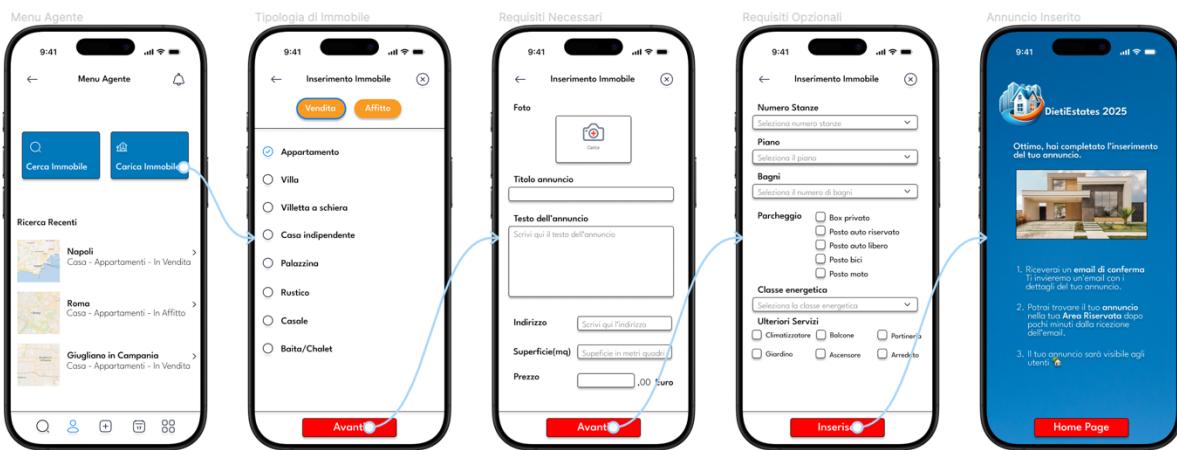
Interfaccia Utente dedicata all'esecuzione di ricerche avanzate.

Con funzionalità di filtraggio dettagliate che permettono agli utenti di affinare i risultati in base a criteri specifici, come tipologia dell'immobile, fascia di prezzo, superficie, numero di bagni, piano, classe energetica, presenza di garage, ascensore, servizio di portineria e giardino, garantendo un'esperienza di ricerca ed efficiente.



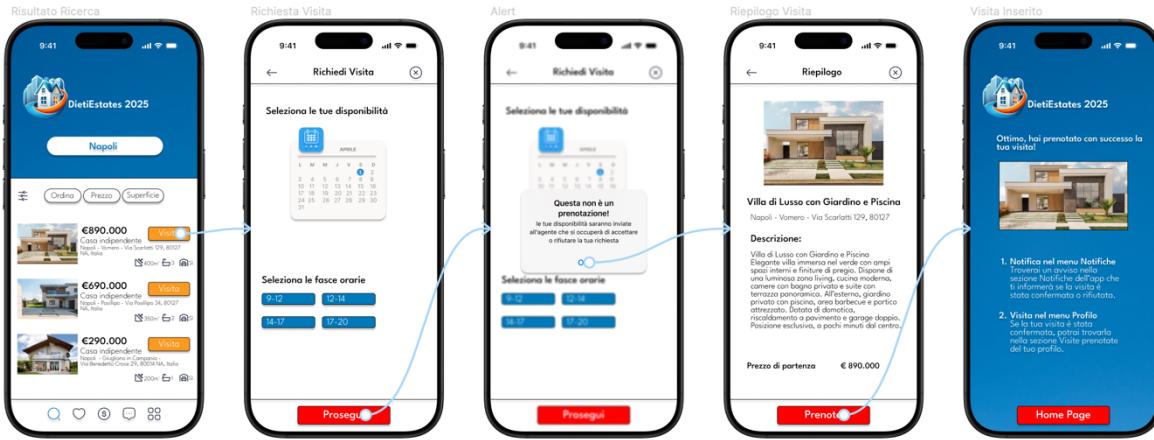
Interfaccia dedicata agli agenti immobiliari per l'inserimento di nuovi annunci.

Permette di selezionare la tipologia di immobile, caricare foto, aggiungere titolo e descrizione, specificare l'indirizzo, la superficie, il prezzo e altri requisiti opzionali. Una volta completata la compilazione, l'agente può visualizzare la pubblicazione dell'annuncio.



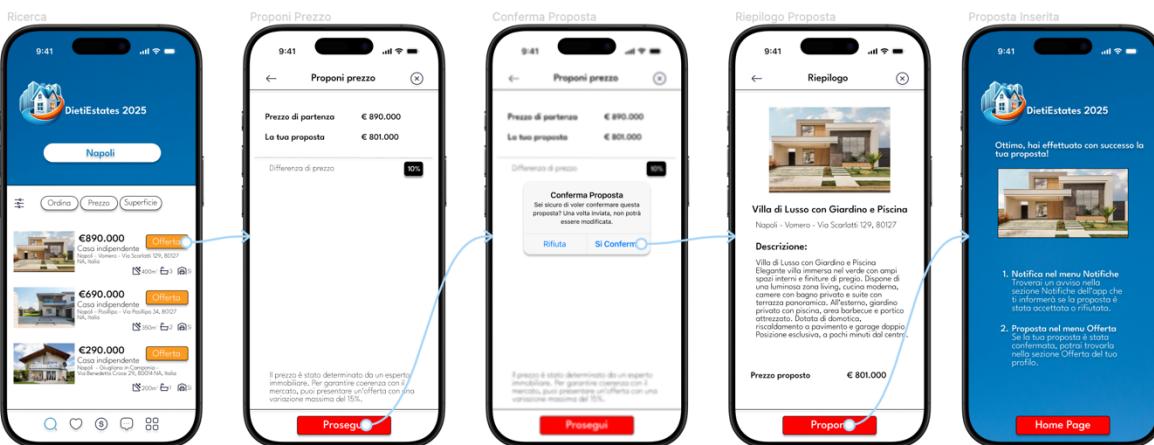
Interfaccia dedicata agli utenti per la richiesta di una visita per un immobile.

Gli utenti possono selezionare l'immobile di interesse, scegliere una data e un orario disponibili per la visita. Una volta inviata, la richiesta viene elaborata dagli agenti responsabili della gestione dell'immobile per pianificare la visita.

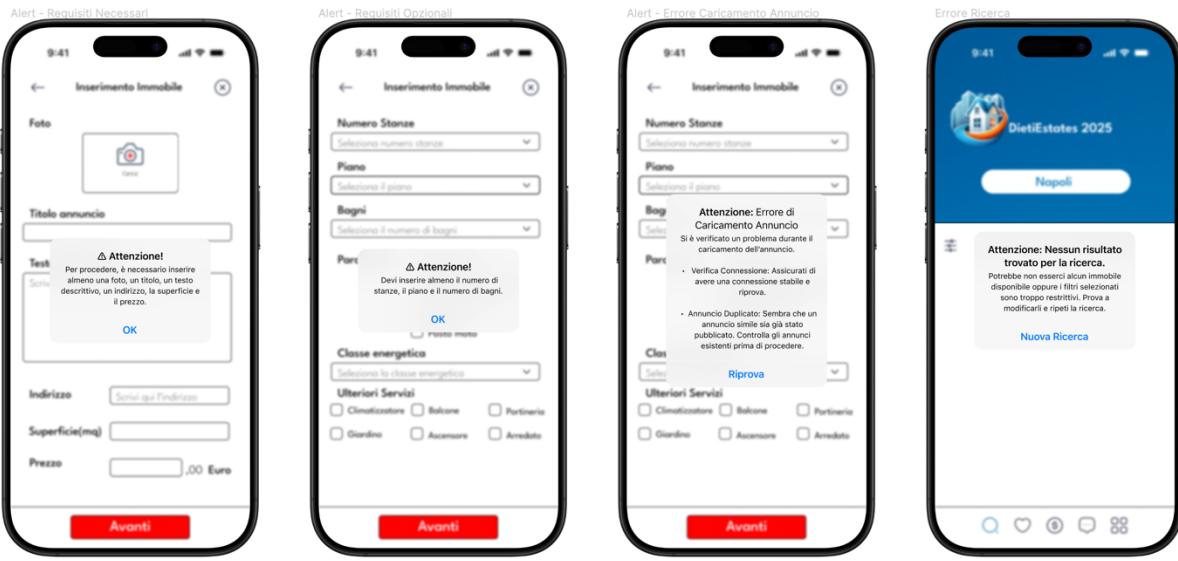


Interfaccia dedicata agli utenti per la proposta di un'offerta.

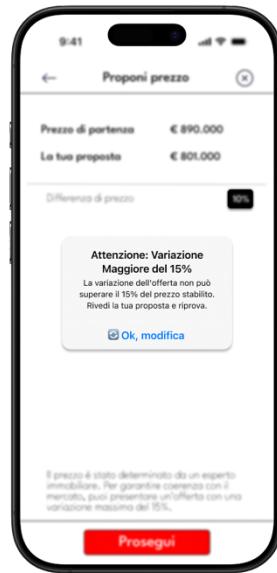
L'interfaccia dedicata agli utenti per la proposta di un'offerta consente agli utenti di inviare una proposta di acquisto per un immobile di loro interesse. Gli utenti possono visualizzare i dettagli dell'immobile, indicare il prezzo o le condizioni proposte. Una volta inviata, la proposta viene inoltrata all'agente immobiliare, che può accettarla, rifiutarla o fare una controproposta.



Interfaccia dedicata agli Alert per l'inserimento di nuovi annunci e per notificare ricerche non riuscite o troppo specifiche.



Interfaccia dedicata agli Alert per l'inserimento della proposta effettuata dall'utente.



Utente

Home Utente



L'interfaccia Utente presenta il pulsante “Inizia nuova ricerca” per poter trovare un immobile in una zona desiderata.

Il pulsante Prenota Visite presente nella Home Utente, anche se non ancora prototipato rimanda ad una schermata dove è possibile effettuare prenotazioni di un immobile attraverso un calendario con le sue disponibilità.

Visite Prenotate contiene uno storico degli immobili da visitare/visitati.

Ultima Ricerca contiene la cronologia di tutti gli immobili ricercati.

Icona “Lente di ingrandimento” ci permette di poter utilizzare la schermata di ricerca.

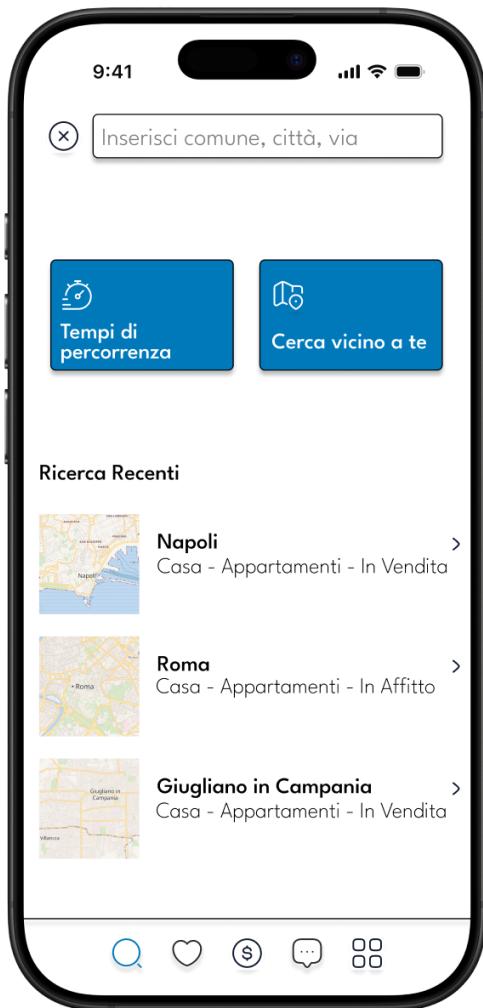
Icona “Cuore” ci permette di poter visualizzare tutti gli immobili messi tra i preferiti.

Icona “Dollaro” ci permette di visualizzare le offerte effettuate/ricevute.

Icona “Messaggio” visualizza le notifiche ricevute.

Icona “Impostazione” gestisce il profilo personale.

Tipologia Ricerca



L'interfaccia Tipologia Ricerca presenta il pulsante "Inserisci comune, città, via per poter trovare un immobile in una zona desiderata.

Il tasto "x" annulla la ricerca.

Il pulsante "Tempi di percorrenza" permette di informare all'utente quanto tempo impiega per arrivare alla posizione desiderata

Il pulsante "Cerca vicino a te" mostra la mappa degli immobili disponibili nelle tue vicinanze

Ricerca Recentí contiene la cronologia di tutti gli immobili ricercati.

Tutti i tasti dedicati all'utente sono disponibili nella parte inferiore delle schermate.

Visualizza Ricerca

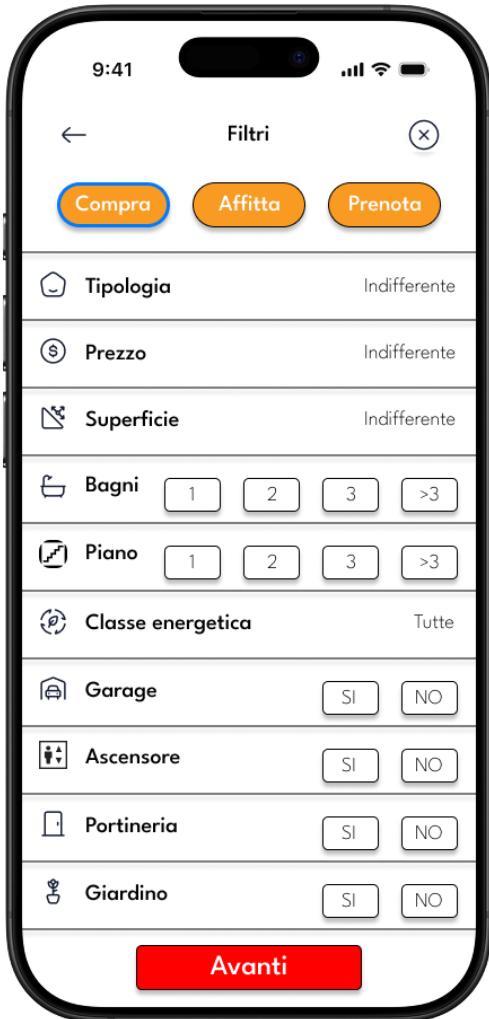


L'interfaccia Visualizza Ricerca ci consente di poter inserire il nome della Città, indirizzo o via desiderata, la ricerca viene facilitata attraverso un elenco di parole suggerite.

Il tasto “indietro” ci permette di tornare all’interfaccia – Tipologia Ricerca.

Il tasto “x” annulla la ricerca.

Filtri Avanzati



L'interfaccia Filtri Avanzati presenta dei pulsanti che ci facilitano a selezionare un immobile da comprare, affittare o prenotare una visita.

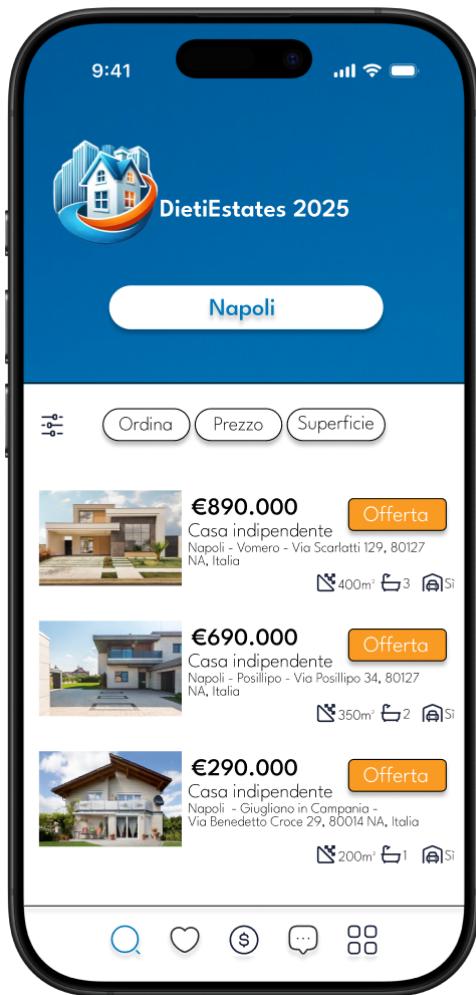
Il tasto “indietro” ci permette di tornare all’interfaccia - Visualizza Ricerca.

Il tasto “x” annulla la ricerca.

La restante dashboard ci permette di filtrare e poter specificare le nostre esigenze.

Il pulsante “Cerca” effettuerà la ricerca in base alle nostre necessità

Tipologia Ricerca



L'interfaccia Ricerca riporta i risultati dei possibili immobili trovati in base alla posizione e filtraggio selezionato.

Cliccando sull'icona dei "filtri" è possibile ritornare alla schermata Filtri Avanzati in modo da poter modificare i parametri da noi scelti.

Cliccando "Ordina" è possibile disporre gli immobili in base alla data di inserimento o ad altri criteri disponibili.

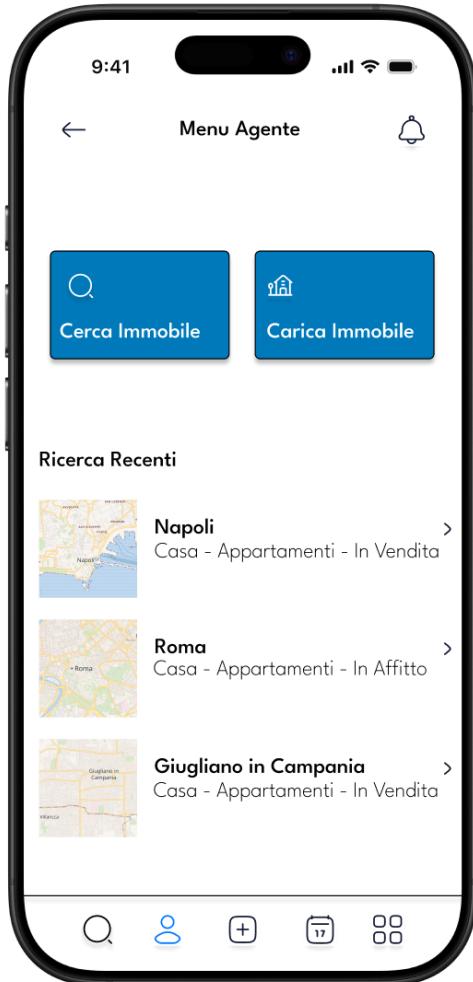
Cliccando "Prezzo" è possibile disporre gli immobili in 2 modi: Prezzo crescente, Prezzo decrescente

Cliccando "Superficie" è possibile disporre gli immobili in base alla dimensione(mq) in ordine crescente o decrescente in base alla nostra esigenza.

Clicca su "Offerta" è possibile effettuare una proposta d'acquisto.

Agente

Menu Agente



L'interfaccia **Menu Agente** è possibile effettuare una ricerca sugli immobili inseriti e caricare un nuovo immobile.

Ricerca Recentí visualizza lo storico ricerca.

Icona “Lente di ingrandimento” ci permette di poter utilizzare la schermata di ricerca.

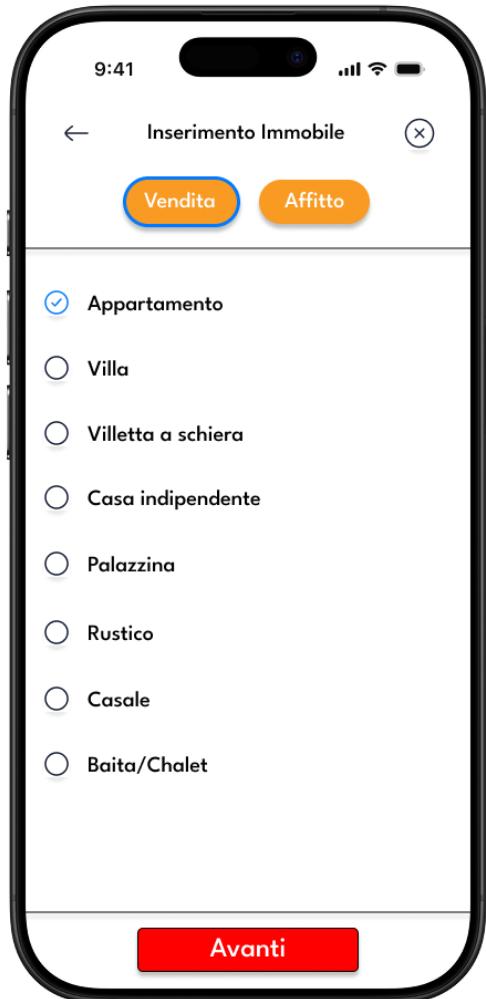
Icona “Profilo Agente” accede al menu agente.

Icona “+” accede alla schermata per creare un nuovo immobile.

Icona “Calendario” si occupa della gestione delle visite.

Icona “Impostazione” gestisce il profilo personale.

Tipologia Immobile



L'interfaccia "Tipologia Immobile" permette di selezionare la categoria dell'immobile e di specificare se è in vendita o in affitto.

Il tasto "indietro" ci permette di tornare all'interfaccia iniziale.

Il tasto "x" annulla l'inserimento.

Il tasto "Avanti" ci permette di proseguire nella schermata - Requisiti Necessari

Tutti i tasti dedicati all'agente sono disponibili nella parte inferiore delle schermate.

Requisiti Necessari



L'interfaccia Requisiti Necessari consente di poter inserire foto dell'immobile, Titolo annuncio, Descrizione, Indirizzo, Superficie e prezzo.

Il tasto “indietro” ci permette di tornare all’interfaccia – Tipologia Immobile.

Il tasto “x” annulla l’inserimento.

Il tasto “Avanti” ci permette di proseguire nella schermata – Requisiti Opzionali.

Tutti i tasti dedicati all’agente sono disponibili nella parte inferiore delle schermate.

Requisiti Opzionali



L'interfaccia Requisiti Opzionali consente di poter inserire foto numero di stanze, piano, bagni disponibili, parcheggio (Se presente), classe energetica, ed ulteriori servizi come climatizzazione, balcone, portineria, giardino, ascensore e se l'immobile è arredato.

Il tasto “Inserisci” ci permette di proseguire nella schermata – Annuncio Inserito.

Tutti i tasti dedicati all’agente sono disponibili nella parte inferiore delle schermate.

Annuncio Inserito



L'interfaccia "Annuncio Inserito" conferma l'inserimento dell'immobile, informando l'agente che riceverà un'email di conferma e potrà visualizzare l'annuncio nell'area riservata. Inoltre, consente di aggiungere un nuovo immobile.

Tutti i tasti dedicati all'agente sono disponibili nella parte inferiore delle schermate.

Utente

Richiesta Visita



L'interfaccia Richiesta Visita permette di impostare le varie informazioni essenziali per organizzare l'appuntamento.

Nella sezione “Selezione le tue disponibilità”, l’utente può interagire direttamente con il calendario cliccando sui numeri corrispondenti ai giorni disponibili per scegliere la data in cui desidera effettuare la visita.

Nella sezione “Selezione le fasce orarie”, l’utente può scegliere l’orario preferito per la visita selezionando una delle fasce disponibili.

Cliccando “Proseguì” l’utente può passare alla pagina successiva.

Cliccando sulla freccia in alto a sinistra l’utente può tornare alla schermata precedente.

Cliccando sull’icona “X” l’utente può annullare la prenotazione e tornare nella Home Utente.

Alert di Conferma



L'interfaccia di alert di conferma avvisa l'utente che la prenotazione che sta effettuando non è ancora definitiva. Questo messaggio serve a chiarire che la richiesta verrà ricevuta, ma è soggetta a una conferma finale o a una disponibilità da parte del' agente prima di essere considerata definitiva.

Cliccando su "OK" l'utente informa di aver compreso il messaggio e continua con la sua richiesta di visita.

Riepilogo Visita



L'interfaccia Riepilogo Visita consente all'utente di visualizzare tutti i dettagli dell'immobile selezionato prima di procedere con la richiesta di prenotazione.

Nella pagina viene mostrato un riepilogo delle informazioni dell'immobile che sono:

- Titolo e descrizione dell'immobile,
- Indirizzo per conoscere la posizione dell'immobile.
- Prezzo di vendita, indicato in evidenza.

Cliccando "Prenota", l'utente può inoltrare la richiesta.

Cliccando sulla freccia in alto a sinistra, l'utente può tornare alla schermata precedente.

Cliccando sull'icona "X", l'utente può annullare la prenotazione e tornare alla Home Utente.

Visita Inserita



L'interfaccia Visita Inserita conferma all'utente che la richiesta di prenotazione per la visita è stata inviata correttamente.

In questa schermata l'utente viene informato che la richiesta sarà valutata dall'agente immobiliare, il quale potrà accettarla o rifiutarla.

L'utente inoltre, può controllare lo stato della richiesta nei seguenti modi:

- Controllando la sezione “Notifiche” per ricevere aggiornamenti sull'esito della richiesta.**
- Accedendo alla sezione “Visite Prenotate” per visualizzare l'elenco degli appuntamenti richiesti e confermati.**

Cliccando “Home Page”, l'utente può tornare alla schermata principale.

Utente

Proponi Prezzo



L'interfaccia Proponi Prezzo permette all'utente di formulare un'offerta per l'immobile selezionato.

Nella sezione “Prezzo di partenza”, viene mostrato il valore stabilito dall'agente immobiliare.

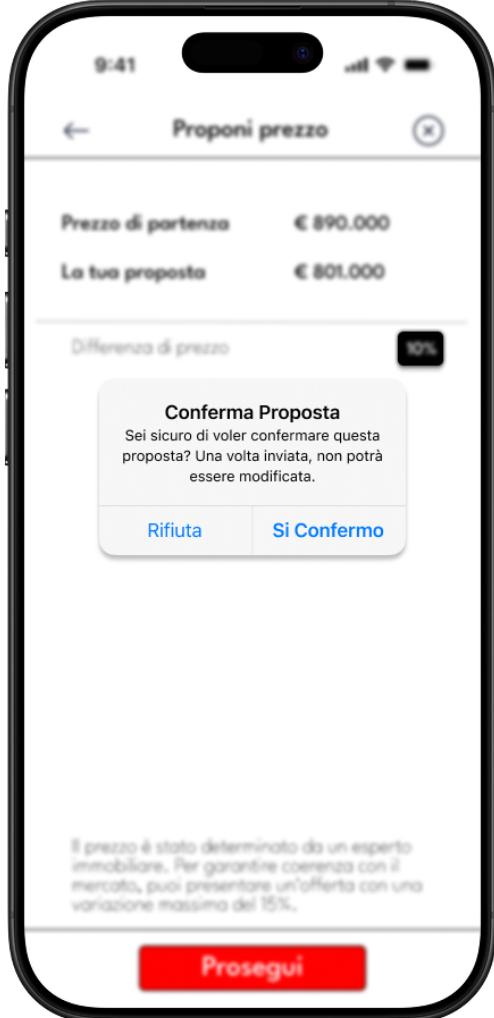
Nella sezione “La tua proposta”, l'utente può inserire un prezzo alternativo, rispettando il limite massimo di variazione del 15% rispetto al prezzo iniziale.

Cliccando “Proseguì”, l'utente può procedere alla schermata di conferma della proposta.

Cliccando sulla freccia in alto a sinistra, l'utente può tornare alla schermata precedente.

Cliccando sull'icona “X”, l'utente può annullare la prenotazione e tornare alla Home Utente.

Conferma Proposta



L'interfaccia Conferma Proposta richiede all'utente di confermare l'offerta inserita prima di inviarla.

Cliccando "Si, Confermo", l'utente invia la proposta e procede alla schermata successiva.

Cliccando "Rifiuta", l'utente annulla la proposta e torna alla schermata precedente.

Riepilogo Proposta



L'interfaccia Riepilogo Proposta consente all'utente di visualizzare tutti i dettagli dell'immobile selezionato prima di procedere con la richiesta di proposta.

Nella pagina viene mostrato un riepilogo delle informazioni dell'immobile che sono:

- Titolo e descrizione dell'immobile,
- Indirizzo per conoscere la posizione dell'immobile.
- Prezzo proposto, indicato in evidenza.

Cliccando "Proponi", l'utente può inoltrare la richiesta.

Cliccando sulla freccia in alto a sinistra, l'utente può tornare alla schermata precedente.

Cliccando sull'icona "X", l'utente può annullare la prenotazione e tornare alla Home Utente.

Proposta Inserita



L'interfaccia Proposta Inserita conferma all'utente che la proposta è stata inviata correttamente.

In questa schermata l'utente viene informato che la proposta sarà valutata dall'agente immobiliare, il quale potrà accettarla, rifiutarla o fare una controproposta.

L'utente inoltre, può controllare lo stato della proposta nei seguenti modi:

- **Controllando la sezione “Notifiche” per ricevere aggiornamenti sull'esito della richiesta.**
 - **Accedendo alla sezione “Offerta” per visualizzare l'elenco delle offerte effettuate dall'utente.**
- Cliccando “Home Page”, l'utente può tornare alla schermata principale.

PARTE II

Documento di design del sistema

5 Analisi dell'architettura

L'architettura dell'applicazione DietiEstates è stata progettata seguendo principi di modularità, scalabilità e manutenibilità. La scelta di un'architettura multi-tier consente di suddividere le responsabilità tra i vari livelli, garantendo una chiara separazione delle preoccupazioni e una maggiore efficienza nello sviluppo e nella gestione dell'applicazione.

5.1 Criteri di Design Adottati

L'applicazione è suddivisa in tre livelli principali:

- **Presentazione (Client) con Flutter/Dart**
Fornisce un'interfaccia utente reattiva e moderna, con supporto multipiattaforma.
- **Logica Applicativa (Server) con Node.js**
Gestisce le API RESTful e la logica di business in modo efficiente
- **Persistenza Dati con PostgreSQL:**
Assicura integrità, sicurezza e scalabilità nella gestione dei dati.

Questa scelta è stata dettata dai seguenti criteri di design:

- **Modularità:** Separare la logica di presentazione, business e dati permette di sviluppare e manutenere ogni livello in modo indipendente.
- **Scalabilità:** Un'architettura a più livelli consente di scalare ogni componente separatamente, distribuendo il carico in base alle necessità.
- **Manutenibilità:** La suddivisione in livelli facilita l'aggiornamento e l'evoluzione del software senza impattare l'intero sistema.
- **Sicurezza:** La separazione dei livelli permette di implementare meccanismi di autenticazione e protezione dei dati in modo più efficace.

5.2 Architettura Client

L'app DietiEstates per dispositivi mobili è stata sviluppata utilizzando **Flutter**, un framework open source di Google, e il linguaggio di programmazione **Dart**.

- **Flutter:** Permette di creare interfacce utente grafiche moderne e performanti, con un'esperienza utente fluida e reattiva. La caratteristica principale di Flutter è la capacità di scrivere codice una sola volta ed eseguirlo su diverse piattaforme (iOS e Android), semplificando lo sviluppo e riducendo i costi.
- **Dart:** È un linguaggio di programmazione orientato agli oggetti, sviluppato da Google, ottimizzato per la creazione di interfacce utente veloci su più piattaforme. Dart è il linguaggio utilizzato per scrivere la logica di business all'interno del framework Flutter.

Caratteristiche chiave:

- **UI Reattiva:** Interfaccia utente dinamica e aggiornamenti in tempo reale.
- **Componenti Riutilizzabili (Widget):** Sviluppo efficiente e manutenzione semplificata.
- **Performance Ottimizzate:** Rendering veloce e animazioni fluide.
- **Multipiattaforma (iOS e Android):** Un'unica base di codice per entrambe le piattaforme.
- **Gestione dello Stato:** Aggiornamenti efficienti dell'interfaccia in base ai dati e alle interazioni.
- **Comunicazione con il Back-end (API RESTful):** Interazione standardizzata per recuperare e inviare dati.
- **User Experience (UX) Intuitiva:** Design incentrato sull'utente per facilitare la navigazione e l'utilizzo.
- **Ricerca Avanzata:** Filtri dettagliati per trovare l'immobile ideale.
- **Mappa Interattiva:** Visualizzazione degli immobili sulla mappa e ricerca per posizione.
- **Gestione Visite e Offerte:** Prenotazione visite e presentazione offerte direttamente dall'app.

5.3 Architettura Server

Il back-end di DietiEstates è stato realizzato con Node.js e PostgreSQL.

- **Node.js:** Un ambiente di runtime JavaScript lato server, leggero e efficiente, basato sull'architettura non-blocking I/O. Node.js è particolarmente adatto per la creazione di API RESTful, che consentono al client (l'app mobile) di comunicare con il server per scambiare dati.
- **PostgreSQL:** Un sistema di gestione di database relazionali (RDBMS) potente e affidabile, noto per la sua robustezza, scalabilità e conformità agli standard SQL. PostgreSQL è utilizzato per archiviare in modo sicuro i dati relativi agli utenti, agli immobili, alle visite e alle transazioni.

Caratteristiche chiave:

Architettura (Node.js):

- Gestione efficiente di un elevato numero di richieste contemporaneamente.
- **Scalabilità:** Capacità di gestire un numero crescente di utenti e dati.
- **API RESTful:** Interfaccia standardizzata per la comunicazione con il client.
- **Sicurezza:** Protezione dei dati degli utenti e prevenzione di accessi non autorizzati.

Database PostgreSQL:

- Affidabilità e Integrità dei Dati (ACID).
- Scalabilità e Performance.
- Supporto per diverse tipologie di dati (testo, numeri, immagini, ecc.).
- Gestione efficiente delle relazioni tra tabelle (immobili, utenti, visite, offerte).
- **Logica di Business:** Gestione delle funzionalità dell'app (ricerca, visite, offerte, statistiche) in base al ruolo dell'utente.
- **Autenticazione e Autorizzazione:** Controllo degli accessi alle diverse funzionalità in base al ruolo dell'utente

5.4 Descrizione e motivazione delle scelte tecnologiche adottate

Flutter/Dart (Client):

- **Base di codice unica per iOS e Android:**
Questa scelta riduce significativamente il tempo e i costi di sviluppo, in quanto consente di mantenere un'unica base di codice per entrambe le piattaforme mobili.
- **UI moderna e fluida:**
Flutter è noto per la sua capacità di creare interfacce utente esteticamente gradevoli e altamente performanti.
- **Gestione dello stato efficiente:**
Flutter supporta vari paradigmi per la gestione dello stato, che permette una comunicazione efficiente tra l'interfaccia e la logica dell'app.

Node.js:

- **Gestione di richieste simultanee:**
Node.js è eccellente per la gestione di numerose richieste simultanee grazie alla sua architettura non-blocking I/O. Questo è particolarmente utile per un'applicazione, che richiede una comunicazione costante e veloce tra client e server.
- **API RESTful scalabili:**
Node.js è molto adatto per la creazione di API RESTful che possono scalare facilmente per soddisfare le necessità crescenti dell'applicazione, come il numero crescente di utenti e richieste.

PostgreSQL:

- **Affidabilità e integrità dei dati (ACID):**
PostgreSQL è un database relazionale robusto che garantisce la coerenza e l'affidabilità dei dati, supportando le transazioni ACID (Atomicità, Consistenza, Isolamento, Durabilità), essenziali per mantenere l'integrità dei dati in un'applicazione complessa come questa.
- **Scalabilità e performance:**
PostgreSQL è noto per le sue performance elevate anche con grandi quantità di dati. Questa caratteristica è importante per garantire che l'applicazione possa gestire un numero crescente di immobili, utenti e transazioni.
- **Supporto per tipi di dati complessi:**
PostgreSQL supporta diversi tipi di dati (come immagini, numeri, testo), che è importante per la gestione di contenuti vari come descrizioni di immobili, foto, e transazioni.
- **Gestione delle relazioni tra tabelle:**
PostgreSQL è perfetto per gestire le relazioni tra diverse entità (immobili, utenti, offerte).

5.5 Descrizione dello schema per la persistenza dati.

Lo schema del database è progettato per supportare un'applicazione di gestione immobiliare, in cui utenti e agenti possono interagire con annunci di immobili, proposte di acquisto, visite e ricerche. La struttura si basa su una serie di entità interconnesse che garantiscono un'organizzazione efficiente dei dati e una gestione coerente delle operazioni.

Vincoli di Integrità

Il database è progettato per garantire l'affidabilità dei dati attraverso una serie di vincoli di integrità:

- **Vincoli di chiavi primarie e chiavi esterne** per garantire la coerenza e le relazioni tra le tabelle
- **Vincoli di unicità** per prevenire duplicazioni nei dati critici, come l'email degli utenti.
- **Vincoli di controllo (CHECK constraints)** per garantire che i dati inseriti siano validi, come quelli sulla validità dei prezzi nelle proposte.
- **Vincoli di default** per automatizzare l'assegnazione di valori predefiniti, come il ruolo predefinito per gli utenti.

5.6 Dizionario delle entità e delle associazioni

Entità	Descrizione	Attributi
Utente	Questa tabella memorizza le informazioni sugli utenti registrati nel sistema, tra cui username, password, nome, cognome ed email. Ogni utente è associato a un ruolo specifico (id_ruolo), che determina i suoi privilegi all'interno della piattaforma.	username_utente, password, id_ruolo, nome, cognome, email
Ruolo	Definisce i ruoli disponibili per gli utenti della piattaforma (Amministratore, Gestore, Agente e Utente). Ogni ruolo ha un identificatore univoco e un nome che rispetta i valori predefiniti, garantendo il corretto controllo degli accessi.	id_ruolo, nome_ruolo
Immobile	Gestisce le informazioni relative agli immobili presenti sulla piattaforma, tra cui il tipo di contratto (vendita o affitto), la tipologia, la superficie, il prezzo, l'indirizzo e altre caratteristiche avanzate. Ogni immobile è associato a un agente immobiliare (username_agente) e a un indirizzo (id_indirizzo_immobile). L'integrità dei dati è garantita da vincoli di unicità sul titolo.	id_immobile, data_creazione, ora_creazione, username_agente, tipo_contratto, tipologia_immobile, titolo, testo, superficie, prezzo, id_indirizzo_immobile, id_filtro_avanzato, id_servizio_ulteriore
Indirizzo	Memorizza le informazioni sugli indirizzi degli immobili, inclusi provincia, città, via e CAP. È collegata alla tabella immobile per fornire il riferimento geografico degli annunci.	id_indirizzo, provincia, città, via, cap
Foto	Ogni immobile può avere più immagini associate. Questa tabella memorizza i percorsi dei file delle immagini, il loro ordine e il riferimento all'immobile corrispondente (id_immobile). Il vincolo ON DELETE CASCADE garantisce la rimozione automatica delle immagini quando un immobile viene eliminato.	id_foto, id_immobile, percorso_file, ordine
Modifica	Tiene traccia delle modifiche apportate agli annunci, registrando il valore precedente e quello aggiornato per ogni attributo dell'immobile. Ogni modifica è associata all'utente che l'ha effettuata e alla data/ora della modifica	id_modifica, id_immobile, username_utente, data_modifica, ora_modifica, nuovo_tipo_contratto, nuovo_tipologia_immobile, nuovo_titolo, nuovo_testo, nuova_superficie, nuovo_prezzo, nuovo_id_indirizzo_immobile, nuovo_id_filtro_avanzato, nuovo_id_servizio_ulteriore
Ricerca	Permette agli utenti ricerche personalizzate basate su indirizzi e filtri avanzati. I riferimenti alle tabelle indirizzo e filtro_avanzato consentono di creare criteri di ricerca dettagliati.	id_ricerca, id_indirizzo, id_filtro_avanzato
Filtro_Avanzato	Contiene parametri avanzati di ricerca, come il numero di stanze, il piano, il numero di bagni, il tipo di parcheggio e la classe energetica.	id_filtro_avanzato, tipologia_immobile, stanza, piano, bagno, parcheggio, classe_energetica
Servizio_Ulteriore	Gestisce informazioni sui servizi aggiuntivi degli immobili (ad esempio, climatizzatore, balcone, ascensore). Ogni servizio può essere collegato a un filtro avanzato per affinare le ricerche.	id_servizio_ulteriore, climatizzatore, balcone, portineria, giardino, ascensore, arredato, id_filtro_avanzato
Proposta	Consente agli utenti di fare offerte sugli immobili, specificando il prezzo attuale e il nuovo prezzo proposto. Gli agenti possono rispondere con una controproposta.	id_proposta, vecchio_prezzo, nuovo_prezzo, stato_proposta, data_proposta, ora_proposta, username_utente_proposta, username_agente_controproposta, controproposta, stato_controproposta
Visita	Gestisce la prenotazione e lo stato delle visite agli immobili. Ogni visita è collegata a un immobile e a un utente, mentre l'agente responsabile può approvare o rifiutare la richiesta.	id_visita, id_immobile, username_utente, data_visita, ora_visita, stato_visita, stato_approvazione_agente, username_agente_approvazione

5.7 Definizioni delle tabelle

```

CREATE TABLE ruolo (
    id_ruolo SERIAL PRIMARY KEY,
    nome_ruolo VARCHAR(50) UNIQUE
);

CREATE TABLE utente (
    username_utente VARCHAR(15) NOT NULL PRIMARY KEY, -- Definisce username_utente come chiave primaria
    password VARCHAR(15) NOT NULL,
    id_ruolo INT NOT NULL, -- Colonna che rappresenta il ruolo dell'utente (riferimento alla tabella ruolo_utente)
    nome VARCHAR(20) NOT NULL,
    cognome VARCHAR(20) NOT NULL,
    email VARCHAR(255) NOT NULL,
    FOREIGN KEY (id_ruolo) REFERENCES ruolo(id_ruolo) ON DELETE CASCADE -- Legame con la tabella dei ruoli
);

CREATE TABLE indirizzo (
    id_indirizzo SERIAL PRIMARY KEY,
    città VARCHAR(255),
    provincia VARCHAR(255),
    via VARCHAR(255),
    cap VARCHAR(10)
);

CREATE TABLE filtro_avanzato (
    id_filtro_avanzato SERIAL PRIMARY KEY,
    tipologia_immobile VARCHAR(255),
    stanza INT CHECK (stanza >= 0),
    piano INT CHECK (piano >= 0),
    bagno INT CHECK (bagno >= 0),
    parcheggio VARCHAR(255),
    classe_energetica VARCHAR(10)
);

CREATE TABLE servizio_ulteriore (
    id_servizio_ulteriore SERIAL PRIMARY KEY,
    climatizzatore BOOLEAN,
    balcone BOOLEAN,
    portineria BOOLEAN,
    giardino BOOLEAN,
    ascensore BOOLEAN,
    arredato BOOLEAN,
    id_filtro_avanzato INT,
    FOREIGN KEY (id_filtro_avanzato) REFERENCES filtro_avanzato(id_filtro_avanzato) ON DELETE CASCADE
);

CREATE TABLE immobile (
    id_immobile SERIAL PRIMARY KEY,
    data_creazione DATE DEFAULT CURRENT_DATE,
    ora_creazione TIME DEFAULT CURRENT_TIME,
    username_agente VARCHAR(15), -- Username dell'agente che gestisce l'immobile
    tipo_contratto VARCHAR(20) CHECK (tipo_contratto IN ('Vendita', 'Affitto')),
    tipologia_immobile VARCHAR(255),
    titolo VARCHAR(255),
    testo TEXT,
    superficie DOUBLE PRECISION CHECK (superficie > 0), -- Superficie in metri quadrati
    prezzo DOUBLE PRECISION CHECK (prezzo > 0), -- Prezzo dell'immobile
    id_indirizzo_immobile INT, -- Riferimento alla tabella INDIRIZZO
    id_servizio_ulteriore INT,
    FOREIGN KEY (username_agente) REFERENCES utente(username_utente) ON DELETE CASCADE,
    FOREIGN KEY (id_indirizzo_immobile) REFERENCES indirizzo(id_indirizzo) ON DELETE CASCADE,
    FOREIGN KEY (id_filtro_avanzato) REFERENCES filtro_avanzato(id_filtro_avanzato) ON DELETE CASCADE,
    FOREIGN KEY (id_servizio_ulteriore) REFERENCES servizio_ulteriore(id_servizio_ulteriore) ON DELETE CASCADE
);

CREATE TABLE foto (
    id_foto SERIAL PRIMARY KEY,
    id_immobile INT NOT NULL,
    percorso_file TEXT NOT NULL, -- Salviamo l'URL completo o il percorso
    ordine INT DEFAULT 1, -- Per gestire l'ordine delle foto
    FOREIGN KEY (id_immobile) REFERENCES immobile(id_immobile) ON DELETE CASCADE
);

CREATE TABLE modifica (
    id_modifica SERIAL PRIMARY KEY,
    id_immobile INT NOT NULL, -- ID dell'immobile che è stato modificato
    username_utente VARCHAR(15) NOT NULL, -- Username dell'utente che ha effettuato la modifica
    data_modifica DATE DEFAULT CURRENT_DATE, -- Data della modifica
    ora_modifica TIME DEFAULT CURRENT_TIME, -- Ora della modifica
    nuovo_tipo_contratto VARCHAR(20), -- Nuovo tipo di contratto (vendita/affitto)
    nuova_tipologia_immobile VARCHAR(255), -- Nuova tipologia dell'immobile
    nuovo_titolo VARCHAR(255), -- Nuovo titolo dell'immobile
    nuovo_testo TEXT, -- Nuovo testo descrittivo dell'immobile
    nuova_superficie DOUBLE PRECISION CHECK (nuova_superficie > 0), -- Nuova superficie
    nuovo_prezzo DOUBLE PRECISION CHECK (nuovo_prezzo > 0), -- Nuovo prezzo
    nuovo_id_indirizzo_immobile INT, -- Nuovo indirizzo dell'immobile
    nuovo_id_filtro_avanzato INT, -- Nuovo filtro avanzato applicato
    nuovo_id_servizio_ulteriore INT, -- Nuovo servizio aggiuntivo applicato
    FOREIGN KEY (id_immobile) REFERENCES immobile(id_immobile) ON DELETE CASCADE,
    FOREIGN KEY (username_utente) REFERENCES utente(username_utente) ON DELETE CASCADE,
    FOREIGN KEY (nuovo_id_indirizzo_immobile) REFERENCES indirizzo(id_indirizzo) ON DELETE CASCADE,
    FOREIGN KEY (nuovo_id_filtro_avanzato) REFERENCES filtro_avanzato(id_filtro_avanzato) ON DELETE CASCADE,
    FOREIGN KEY (nuovo_id_servizio_ulteriore) REFERENCES servizio_ulteriore(id_servizio_ulteriore) ON DELETE CASCADE
);

```

```

CREATE TABLE proposta (
    id_proposta SERIAL PRIMARY KEY,
    id_immobile_proposta INT,
    vecchio_prezzo DOUBLE PRECISION,
    nuovo_prezzo DOUBLE PRECISION,
    stato_proposta VARCHAR(50),
    data_proposta DATE,
    ora_proposta TIME,
    username_utente_proposta VARCHAR(15),
    username_agente_controproposta VARCHAR(15),
    controproposta DOUBLE PRECISION,
    stato_controproposta VARCHAR(50),
    FOREIGN KEY (id_immobile_proposta) REFERENCES immobile(id_immobile) ON DELETE CASCADE,
    FOREIGN KEY (username_utente_proposta) REFERENCES utente(username_utente) ON DELETE CASCADE,
    FOREIGN KEY (username_agente_controproposta) REFERENCES utente(username_utente) ON DELETE CASCADE
);

CREATE TABLE visita (
    id_visita SERIAL PRIMARY KEY,
    id_immobile INT,
    username_utente VARCHAR(15),
    data_visita DATE,
    ora_visita TIME,
    stato_visita VARCHAR(255),
    stato_approvazione_agente VARCHAR(50),
    username_agente_approvazione VARCHAR(15),
    FOREIGN KEY (id_immobile) REFERENCES immobile(id_immobile) ON DELETE CASCADE,
    FOREIGN KEY (username_utente) REFERENCES utente(username_utente) ON DELETE CASCADE,
    FOREIGN KEY (username_agente_approvazione) REFERENCES utente(username_utente) ON DELETE CASCADE
);

CREATE TABLE ricerca (
    id_ricerca SERIAL PRIMARY KEY,
    id_indirizzo INT,
    id_filtro_avanzato INT,
    FOREIGN KEY (id_indirizzo) REFERENCES indirizzo(id_indirizzo) ON DELETE CASCADE,
    FOREIGN KEY (id_filtro_avanzato) REFERENCES filtro_avanzato(id_filtro_avanzato) ON DELETE CASCADE
);

```

5.8 Definizioni dei vincoli

```
-- Aggiungi un vincolo di check sulla colonna email nella tabella 'utente'  
ALTER TABLE utente  
ADD CONSTRAINT chk_email_format  
CHECK (email ~* '^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$');  
  
-- Aggiungi un vincolo di unicità sulla colonna 'email' nella tabella 'utente'  
ALTER TABLE utente  
ADD CONSTRAINT uk_email UNIQUE (email);  
  
-- Assicurarti che ogni Utente abbia un Ruolo Predefinito  
ALTER TABLE utente  
ALTER COLUMN id_ruolo SET DEFAULT 4;  
  
-- Aggiungi un vincolo di check sulla colonna 'nome_ruolo' nella tabella 'ruolo_utente'  
-- che accetta solo valori 'Amministratore','Agente Immobiliare','Gestore', 'Utente'  
ALTER TABLE ruolo  
ADD CONSTRAINT chk_ruolo CHECK (nome_ruolo IN ('Amministratore','Gestore','Agente', 'Utente'));  
  
-- Aggiungi un vincolo di unicità sulle colonne 'città', 'provincia', 'via' e 'cap'  
ALTER TABLE indirizzo  
ADD CONSTRAINT uk_indirizzo_unico UNIQUE (città, provincia, via, cap);  
  
-- Aggiungi un vincolo di unicità sulla colonna 'titolo' nella tabella 'immobile'  
ALTER TABLE immobile  
ADD CONSTRAINT uk_titolo UNIQUE (titolo);  
  
-- Aggiungi un vincolo di check sulla colonna 'data_creazione' nella tabella 'immobile'  
-- che assicura che la data di creazione non sia successiva alla data corrente  
ALTER TABLE immobile  
ADD CONSTRAINT chk_data_creazione CHECK (data_creazione <= CURRENT_DATE);  
  
-- Vincolo di check sulla colonna 'parcheggio' nellal tabella 'filtro_avanzato_immoile'  
-- che assicura che parcheggio abbiamo solo i requisiti richiesti  
ALTER TABLE filtro_avanzato  
ADD CONSTRAINT chk_parcheggio  
CHECK (parcheggio IN ('Box privato', 'Posto auto riservato', 'Posto auto libero', 'Posto bici', 'Posto moto'));  
  
-- Aggiungi un vincolo di check sulla colonna 'classe_energetica' nella tabella 'filtro_avanzato_immobile'  
ALTER TABLE filtro_avanzato  
ADD CONSTRAINT chk_classe_energetica  
CHECK (classe_energetica IN ('A', 'B', 'C', 'D', 'E', 'F', 'G'));  
  
-- Aggiungi un vino di check sulla colonna 'prezzo' nella tabella 'immobile'  
ALTER TABLE immobile  
ADD CONSTRAINT chk_prezzo_positive  
CHECK (prezzo IS NULL OR prezzo > 0);  
  
-- Aggiungi un vino di check sulla colonna 'controproposta' nella tabella 'proposta'  
ALTER TABLE proposta  
ADD CONSTRAINT chk_controproposta_positive  
CHECK (controproposta IS NULL OR controproposta > 0);  
  
-- Aggiungi un vincolo di check sulla tabella 'proposta'  
-- che assicura che 'vecchio_prezzo' e 'nuovo_prezzo' siano diversi  
ALTER TABLE proposta  
ADD CONSTRAINT chk_vecchio_nuovo_prezzo_different  
CHECK (vecchio_prezzo <> nuovo_prezzo);  
  
-- Imposta il valore di default 'In attesa' per la colonna 'stato' nella tabella 'proposta'  
ALTER TABLE proposta  
ALTER COLUMN stato_proposta SET DEFAULT 'In attesa';  
  
-- Aggiungi un vincolo di check sulla colonna 'stato_proposta' nella tabella 'proposta'  
-- che accetta solo valori 'Accettata', 'Rifiutata', In attesa'  
ALTER TABLE proposta  
ADD CONSTRAINT chk_proposta CHECK (stato_proposta IN ('Accettata','Rifiutata','In attesa'));  
  
-- Aggiungi un vincolo di check sulla colonna 'stato_visita' nella tabella 'visite'  
ALTER TABLE visita  
ADD CONSTRAINT chk_stato_visita  
CHECK (stato_visita IN ('In attesa', 'Completata', 'Annullata'));  
  
-- Aggiungi un vincolo di check sulla colonna 'stato_approvazione_agente' nella tabella 'visite'  
ALTER TABLE visita  
ADD CONSTRAINT chk_stato_aprovazione_agente  
CHECK (stato_approvazione_agente IN ('Accettata', 'Rifiutata', 'In attesa'));
```

5.9 Definizioni dei Trigger e delle Funzioni

```

-- Registrare automaticamente le modifiche sugli immobili mantenendo lo storico modifiche
CREATE OR REPLACE FUNCTION log_modifica()
RETURNS TRIGGER AS $$
BEGIN
    -- Inserisce una nuova riga nella tabella 'modifica' per mantenere lo storico
    INSERT INTO modifica (
        id_immobile, username_utente, data_modifica, ora_modifica,
        nuovo_tipo_contratto, nuova_tipologia_immobile, nuovo_titolo, nuovo_testo,
        nuova_superficie, nuovo_prezzo, nuovo_id_indirizzo_immobile,
        nuovo_id_filtro_avanzato, nuovo_id_servizio_ulteriore
    )
    VALUES (
        OLD.id_immobile, NEW.username_agente, CURRENT_DATE, CURRENT_TIME,
        NEW.tipo_contratto, NEW.tipologia_immobile, NEW.titolo, NEW.testo,
        NEW.superficie, NEW.prezzo, NEW.id_indirizzo_immobile,
        NEW.id_filtro_avanzato, NEW.id_servizio_ulteriore
    );
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Creazione del trigger
CREATE TRIGGER trg_log_modifica
AFTER UPDATE ON immobile
FOR EACH ROW
EXECUTE FUNCTION log_modifica();

-- Impedisce l'eliminazione di un agente se ha immobili attivi
CREATE OR REPLACE FUNCTION verifica_immobili_agente()
RETURNS TRIGGER AS $$
BEGIN
    -- Verifica se l'agente ha immobili
    IF EXISTS (SELECT i FROM immobile WHERE username_agente = OLD.username_utente) THEN
        RAISE EXCEPTION 'Non puoi eliminare un agente con immobili attivi.';
    END IF;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_blocca_cancellazione_agente
BEFORE DELETE ON utente
FOR EACH ROW
EXECUTE FUNCTION verifica_immobili_agente();

-- Cancellazione per provare
DELETE FROM utente WHERE username_utente = 'alessandrabi';

-- Impedisce la modifica del prezzo di un immobile se una proposta è già stata fatta su di esso
CREATE OR REPLACE FUNCTION impedisce_modifica_prezzo_con_proposta()
RETURNS TRIGGER AS $$
BEGIN
    -- Verifica se esistono proposte per l'immobile
    IF EXISTS (SELECT 1 FROM proposta WHERE id_immobile_proposta = OLD.id_immobile) THEN
        RAISE EXCEPTION 'Non puoi modificare il prezzo dell''immobile perché ci sono proposte in corso.';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_impedisci_modifica_prezzo
BEFORE UPDATE ON immobile
FOR EACH ROW
WHEN (OLD.prezzo <> NEW.prezzo)
EXECUTE FUNCTION impedisce_modifica_prezzo_con_proposta();

-- Non permette la cancellazione di un immobile se è stato proposto
CREATE OR REPLACE FUNCTION impedisce_cancellazione_immobile_con_proposta()
RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (SELECT 1 FROM proposta WHERE id_immobile_proposta = OLD.id_immobile) THEN
        RAISE EXCEPTION 'Non puoi cancellare un immobile che ha proposte in corso.';
    END IF;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

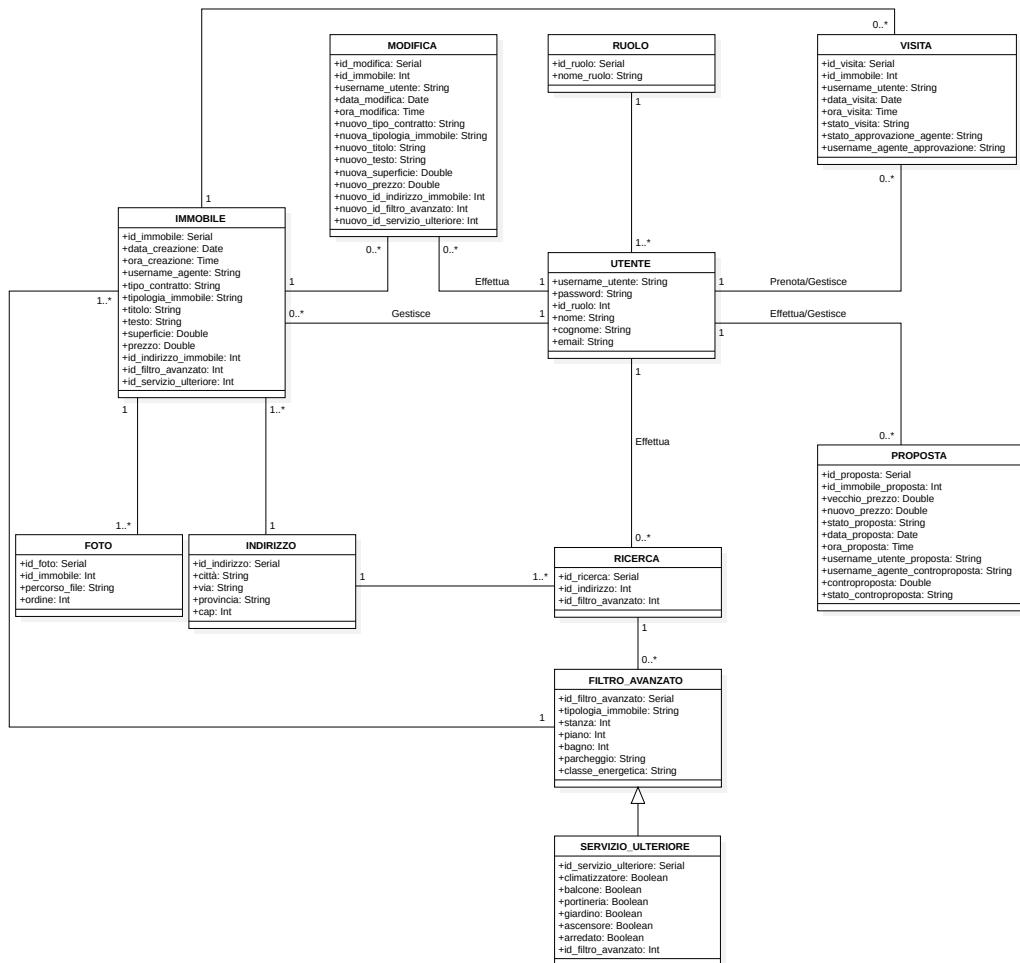
-- Registrazione automatica di una modifica quando un agente modifica un immobile
CREATE OR REPLACE FUNCTION registra_modifica_immobile_agente()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO modifica (
        id_immobile, username_utente, data_modifica, ora_modifica,
        nuovo_tipo_contratto, nuova_tipologia_immobile, nuovo_titolo, nuovo_testo,
        nuova_superficie, nuovo_prezzo, nuovo_id_indirizzo_immobile,
        nuovo_id_filtro_avanzato, nuovo_id_servizio_ulteriore
    )
    VALUES (
        OLD.id_immobile, NEW.username_agente, CURRENT_DATE, CURRENT_TIME,
        NEW.tipo_contratto, NEW.tipologia_immobile, NEW.titolo, NEW.testo,
        NEW.superficie, NEW.prezzo, NEW.id_indirizzo_immobile,
        NEW.id_filtro_avanzato, NEW.id_servizio_ulteriore
    );
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_registra_modifica_immobile_agente
AFTER UPDATE ON immobile
FOR EACH ROW
WHEN (NEW.username_agente IS DISTINCT FROM OLD.username_agente)
EXECUTE FUNCTION registra_modifica_immobile_agente();

```

5.10 Schema UML

Model::Main



È possibile visionare l'immagine più definita, che permettono una visualizzazione scalabile e dettagliata

Per Visionare al meglio Seguire il link
[UML](#)

5.11 Descrizione e motivazione delle scelte di design dell'interfaccia utente adottate

Utente:

L'interfaccia utente dell'applicazione DietiEstates 2025 è stata progettata con un design moderno, intuitivo e accessibile, per offrire un'esperienza fluida agli utenti nella ricerca e gestione degli immobili.

Coerenza visiva e identità grafica:

- L'app segue una linea grafica coerente, con l'utilizzo di colori aziendali predominanti (blu e arancione) per garantire riconoscibilità e professionalità.
- Il logo ben visibile nella schermata di avvio rafforza l'identità del brand.

Navigazione semplice e intuitiva:

- La barra di navigazione inferiore fornisce accesso rapido alle sezioni principali: ricerca, preferiti, notifiche e profilo.
- L'uso di icone chiare e testi esplicativi facilita l'interazione anche per utenti meno esperti.

Ricerca avanzata e filtraggio immediato:

- La barra di ricerca permette di inserire direttamente città, comune o via, con suggerimenti automatici per velocizzare la selezione
- Un sistema di filtri dettagliati consente agli utenti di personalizzare la ricerca in base a parametri specifici come prezzo, superficie, numero di bagni, piano, presenza di garage, giardino e altri criteri.
- Il pulsante "Cerca" evidenziato in rosso garantisce un richiamo visivo forte, incentivando l'azione.

Visualizzazione chiara degli annunci:

- Ogni immobile è presentato con un'anteprima che include immagine, prezzo e tipologia.
- Un'opzione di ordinamento consente di riordinare i risultati in base a prezzo, superficie o altre preferenze.

Gestione degli errori e feedback all'utente:

Se una ricerca non produce risultati, l'app mostra un messaggio chiaro con suggerimenti per migliorare la ricerca e un pulsante per riprovare, riducendo la frustrazione dell'utente.

5.12 Diagramma delle classi di design

Il diagramma delle classi è uno degli strumenti fondamentali della progettazione orientata agli oggetti, usato principalmente nel contesto dell'UML (Unified Modeling Language).

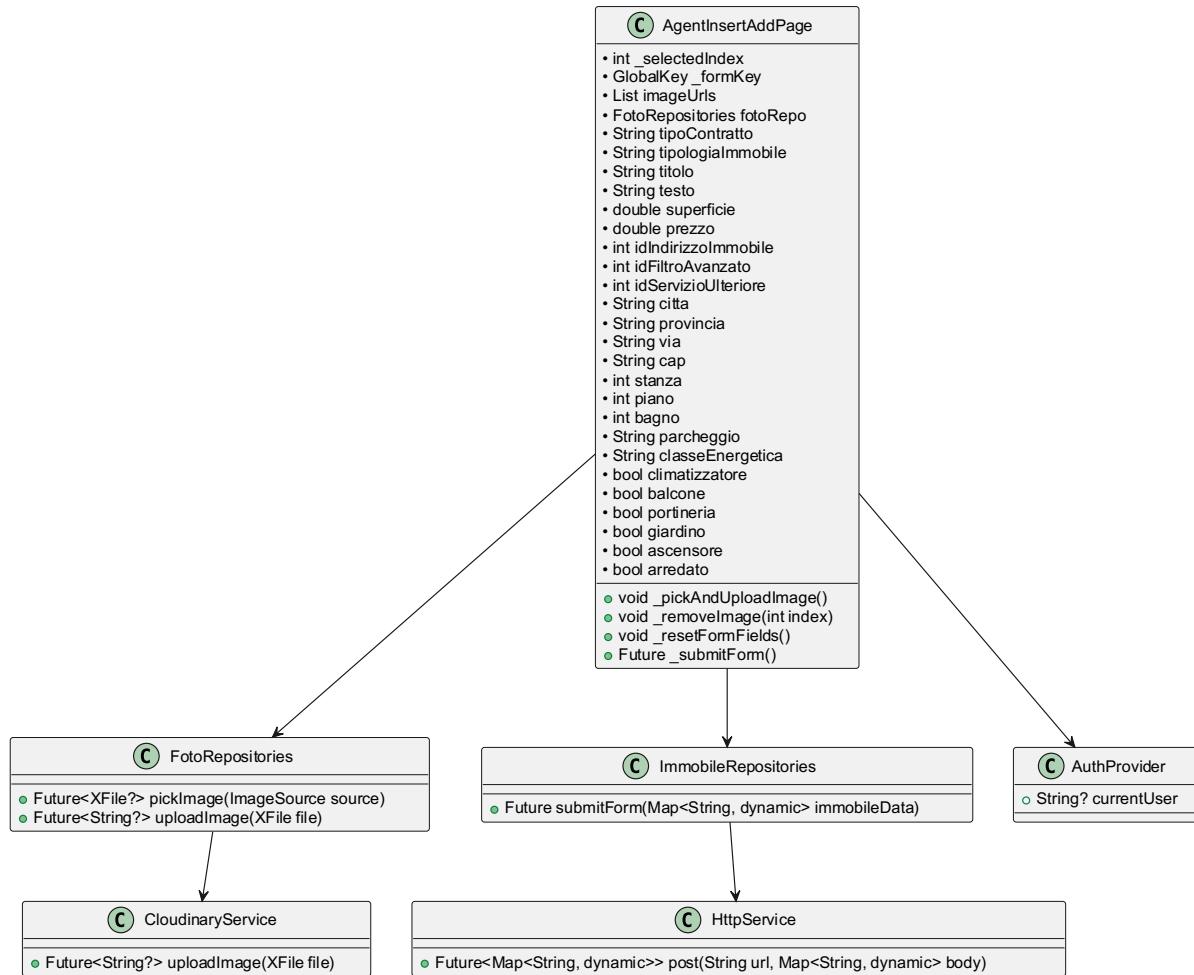
Diagramma dedicato all'esecuzione di ricerche avanzate



Per Visionare al meglio Seguire il link

[Class Diagram](#)

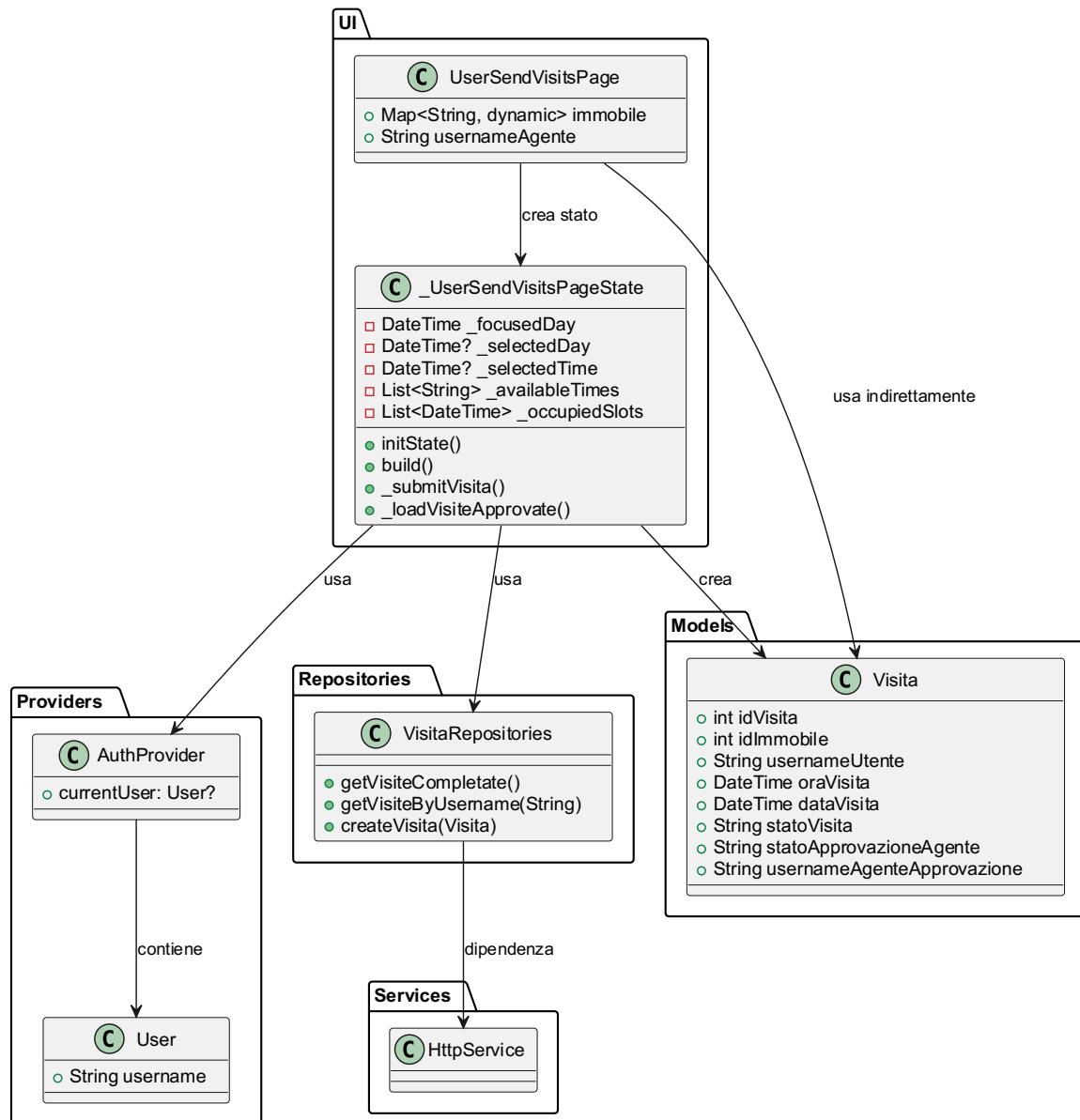
Diagramma dedicato agli agenti immobiliari per l'inserimento di nuovi annunci



Per Visionare al meglio Seguire il link

[Class Diagram](#)

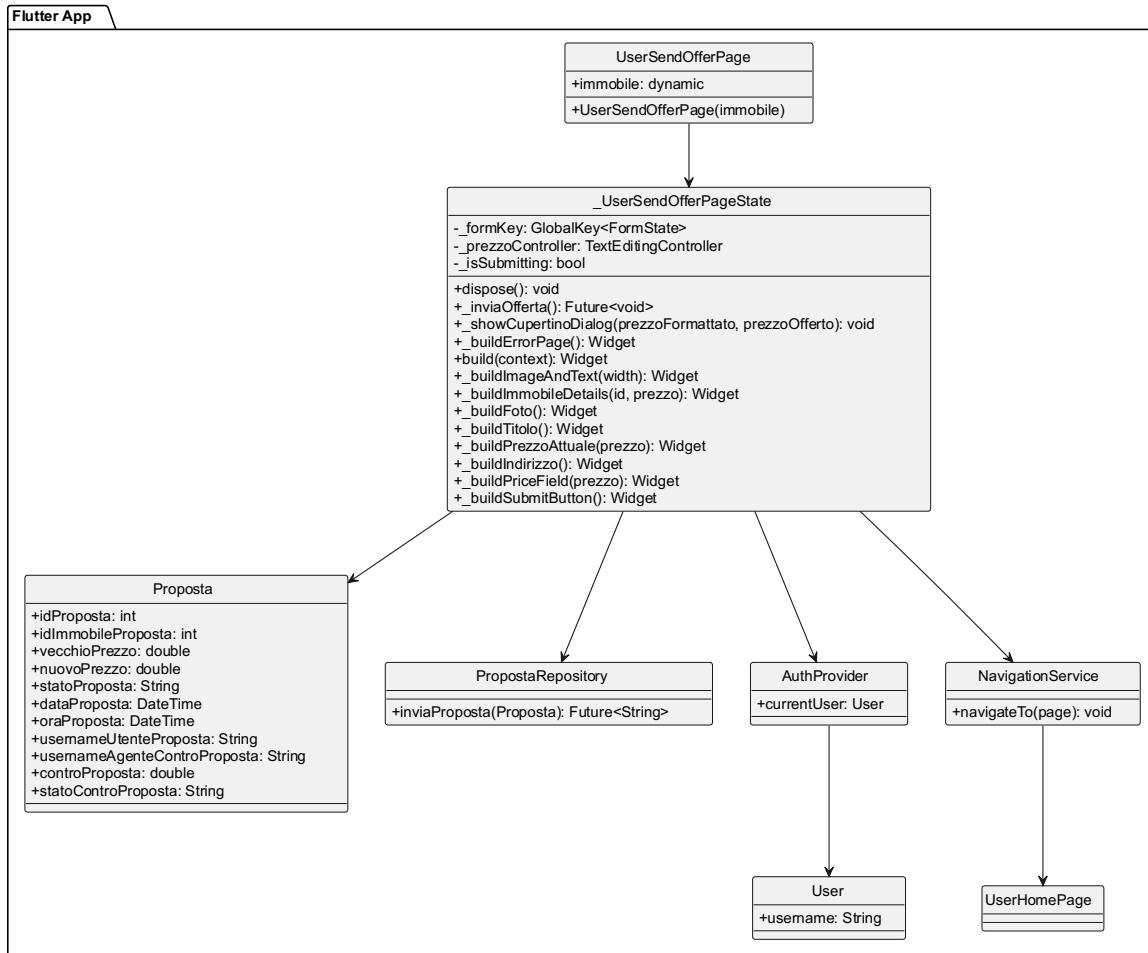
Diagramma dedicato agli utenti per la richiesta di una visita per un immobile



Per Visionare al meglio Seguire il link

[Class Diagram](#)

Diagramma dedicato agli utenti per la proposta di un'offerta



Per Visionare al meglio Seguire il link

[Class Diagram](#)

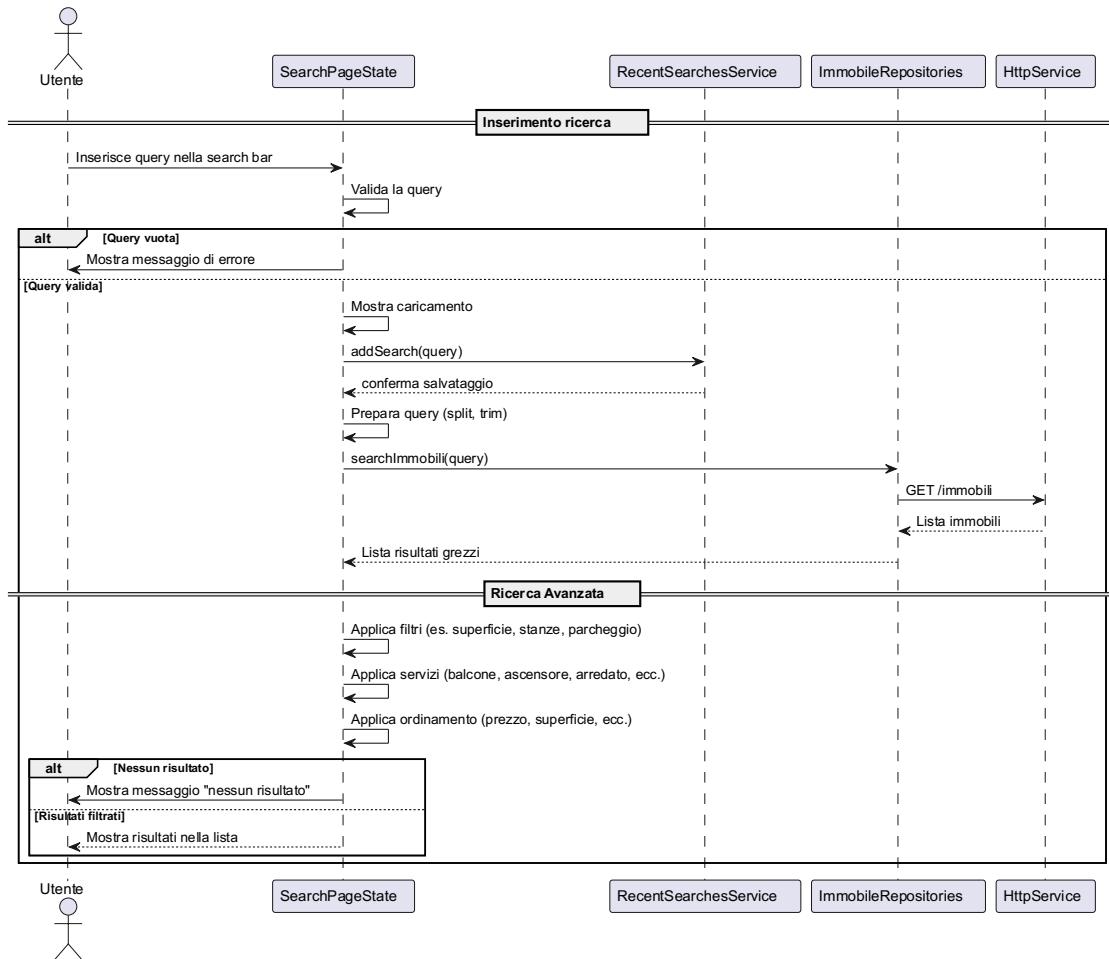
È possibile visionare il diagramma delle classi completo attraverso questo link:

[UML](#)

5.13 Diagramma di sequenza di design per i casi d'uso utilizzati

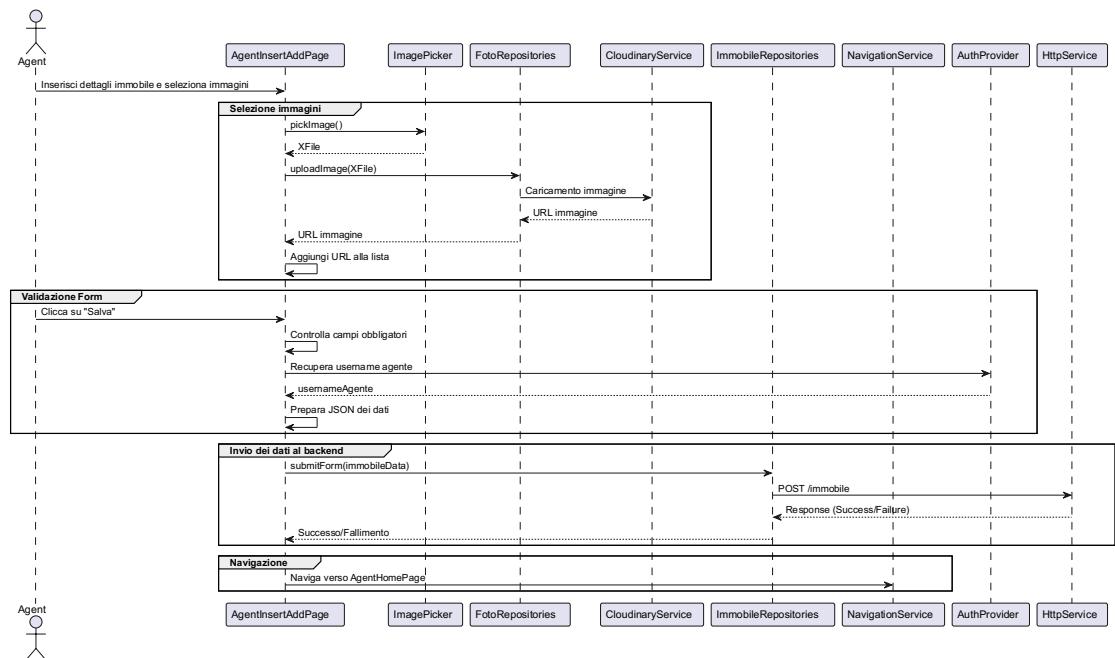
Un diagramma di sequenza (Sequence Diagram) è uno strumento dell'UML (Unified Modeling Language) utilizzato per rappresentare graficamente l'interazione tra attori e oggetti in un sistema, evidenziando l'ordine temporale dei messaggi scambiati. È particolarmente utile per descrivere scenari specifici di un caso d'uso, mostrando come le entità coinvolte collaborano nel tempo per realizzare una funzionalità.

Diagramma di sequenza dedicato all'esecuzione di ricerche avanzate



Per Visionare al meglio Seguire il link
[Sequence Diagram](#)

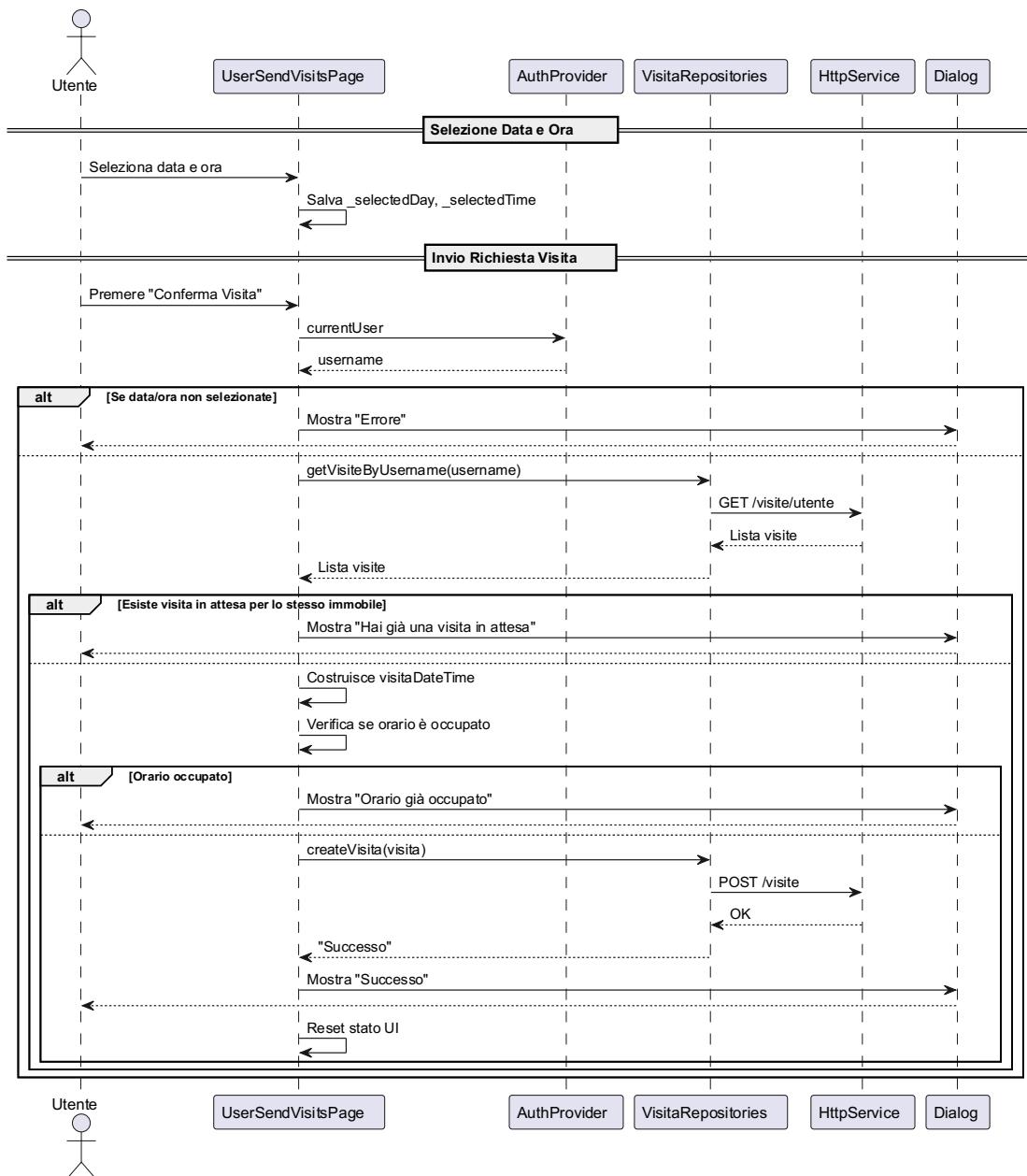
Diagramma di sequenza dedicato agli agenti immobiliari per l'inserimento di nuovi annunci



Per Visionare al meglio Seguire il link

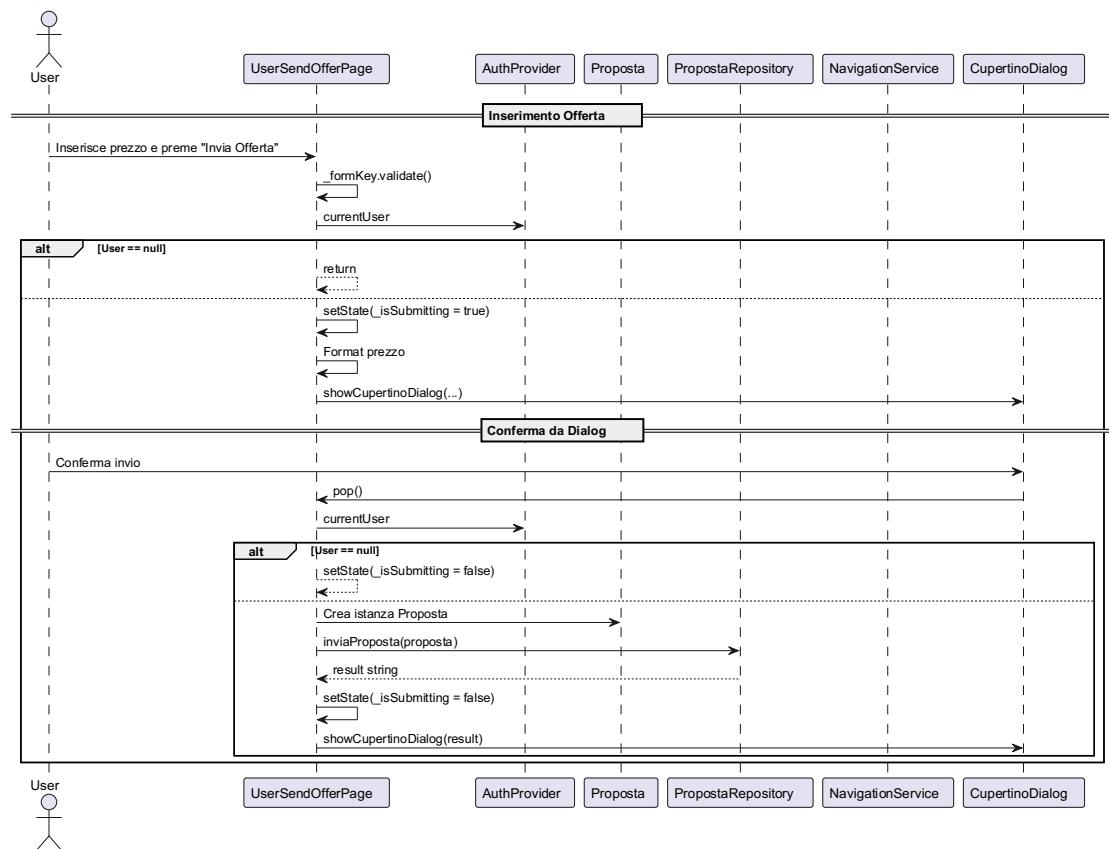
[Sequence Diagram](#)

Diagramma di sequenza dedicato agli utenti per la richiesta di una visita per un immobile



Per Visionare al meglio Seguire il link
[Sequence Diagram](#)

Diagramma di sequenza dedicato agli utenti per la proposta di un'offerta



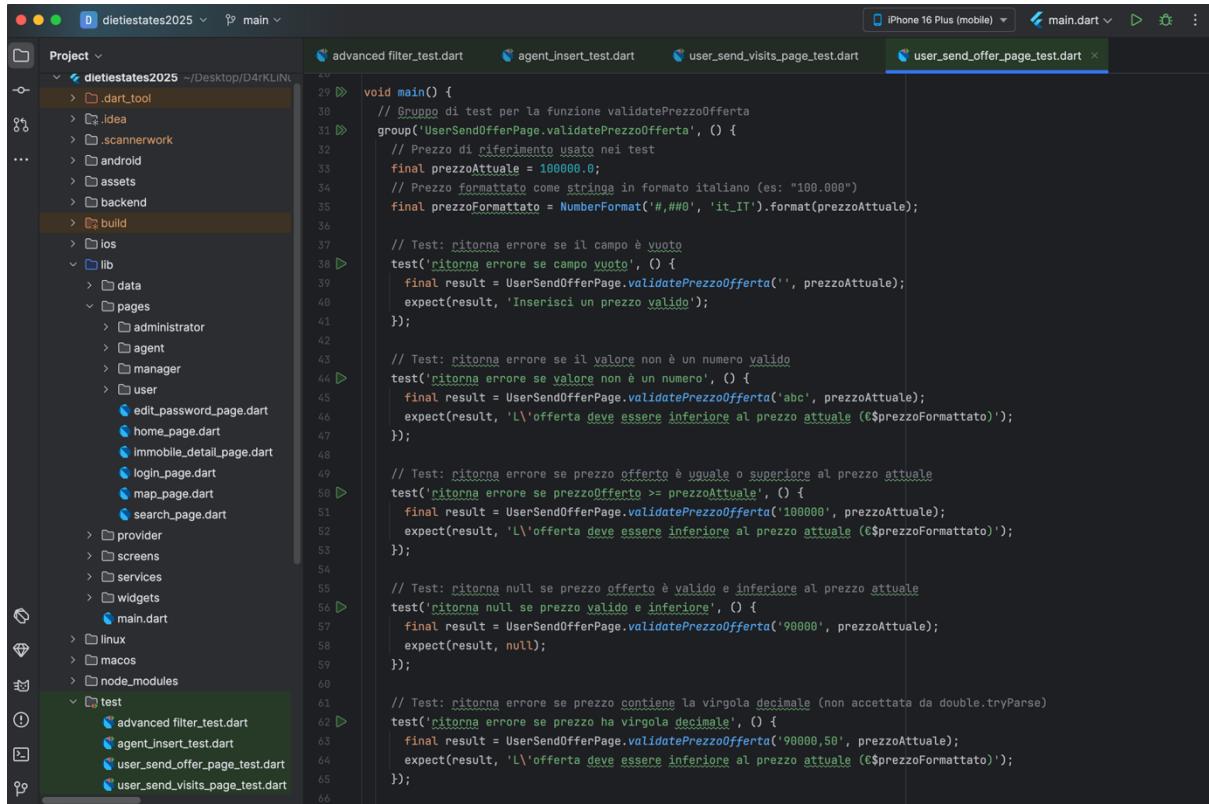
Per Visionare al meglio Seguire il link
[Sequence Diagram](#)

PARTE IV

Testing e valutazione dell'usabilità

6 Testing e valutazione dell'usabilità

Per garantire l'affidabilità e la correttezza delle funzionalità più complesse del sistema, sono stati sviluppati test automatici utilizzando il framework **xUnit**, specificamente per quattro metodi considerati non banali per logica o struttura.



The screenshot shows a Dart code editor interface with the following details:

- Project Structure:** The left sidebar displays the project structure for "dietestates2025". It includes folders for .dart_tool, .idea, .scannerwork, android, assets, backend, build, lib (containing data, pages, administrator, agent, manager, user), tests (containing advanced_filter_test.dart, agent_insert_test.dart, user_send_visits_page_test.dart, user_send_offer_page_test.dart), and main.dart.
- Code Editor:** The main area shows the content of the "user_send_offer_page_test.dart" file. The code is written in Dart and uses the xUnit testing framework to validate the "UserSendOfferPage.validatePrezzoOfferta" function. The code includes several test cases, each checking for specific validation errors based on input values like empty strings, non-numeric strings, or invalid decimal separators.
- Toolbars and Status:** At the top, there are tabs for "iPhone 16 Plus (mobile)" and "main.dart". On the right side, there are various icons for file operations like save, close, and refresh.

6.1 Interfaccia Utente dedicata all'esecuzione di ricerche avanzate

1. Classi di Equivalenza

Per ogni criterio di filtro, sono state identificate **classi di valori equivalenti**, ognuna rappresentata da almeno un test. L'idea è che, se una classe è ben testata, tutti i suoi elementi si comportano allo stesso modo.

Esempi:

- **tipoContratto:**
 - Valori equivalenti: "Vendita", "Affitto", "Tutti"
 - Test specifici coprono ciascuno di questi casi.
- **superficie:**
 - Range ben definiti: "Da 0 a 50 m²", "Da 51 a 100 m²", ..., "Da 501 a 1000 m²", "Tutti"
 - Coperti con almeno un caso rappresentativo (es. 80 m² e 100 m²).
- **stanze:**
 - Classi: "0-5", "6-10", ..., "30+", "Tutti"
 - Test effettuati con "0-5" e "Tutti".
- **Booleani** (es. climatizzatore, balcone, ecc.):
 - Classi: true (attiva il filtro), false (non applica il filtro)
 - Test eseguiti in entrambi i casi (es. portineria = true, nessun immobile lo ha → output vuoto).
- **Edge case per valori nulli o estremi:**
 - Immobili con tutti i valori null → copertura di casi problematici.
 - Verifica che 100 m² sia incluso nel range corretto (Da 51 a 100 m²) → test di bordo.

2. Criteri di copertura strutturale

Sono stati applicati diversi criteri di **copertura del codice**:

- **Copertura delle Condizioni (Condition Coverage)**

Ogni condizione booleana nei filtri viene testata per valori true e false:

Esempio:

- matchesClimatizzatore != climatizzatore || result['climatizzatore'] == true
- Testato quando climatizzatore == true e result['climatizzatore'] == true
- Testato anche quando climatizzatore == false (filtro ignorato)

- **Copertura delle Diramazioni (Branch Coverage)**

Ogni switch-case per i range (superficie, stanze, bagni, piano) è almeno parzialmente testato. Non tutti i case sono coperti, ma quelli principali e quelli ai bordi (es. 100 m²) sono presenti.

- **Copertura delle Decisioni Multiple (Multiple Condition Coverage)**

Nel test Filtra per più parametri attivi vengono attivati **tutti i filtri contemporaneamente**, verificando che un immobile soddisfi **tutte le condizioni combinate**. Questo permette di verificare che le condizioni siano congiunte in modo corretto (AND tra tutti i match).

- **Copertura dei Valori Nulli**

Un test specifico verifica come il sistema gestisce gli immobili con valori null, garantendo robustezza in scenari reali.

6.2 Inserimento Annuncio da parte dell'Agente

1. Classi di Equivalenza

Per verificare il corretto comportamento della funzione buildImmobileData, sono state identificate classi di equivalenza per ogni tipo di campo, in modo da testare input rappresentativi e garantire la coerenza dei dati generati.

Esempi:

- **Campi obbligatori**
 - Verificati i campi richiesti e facoltativi nei test: Tutti i campi sono forniti almeno una volta
- **Numerici**
 - superficie, prezzo, stanza, bagno, piano testati in vari range, inclusi casi estremi:
 - superficie: 0.0, 20.0, 100.5, 300.0
 - prezzo: 0.0, 500.0, 250000.0, 600000.0
 - piano: -1, 0, 1, 5
- **Booleani**
 - Tutti i campi booleani (climatizzatore, balcone, portineria, giardino, ascensore, arredato) testati sia con true che false per assicurare la corretta mappatura.
- **Date/Ora**
 - Campi data_creazione e ora_creazione generati automaticamente → test con espressioni regolari per verificarne il formato ISO.
- **Collezioni**
 - percorso_file: testato sia con lista vuota che con più URL, mantenendo l'ordine.
 - Test specifico verifica che i valori degli URL siano mantenuti nell'ordine originale.

2. Criteri di copertura strutturale

Sono stati applicati diversi criteri di **copertura del codice**:

- **Copertura delle Condizioni (Condition Coverage)**
 - Ogni campo booleano è stato verificato individualmente.
 - Test come Tutti i campi booleani vengono mappati correttamente coprono true/false.
- **Copertura delle Diramazioni (Branch Coverage)**
 - Anche se non presenti rami logici complessi, i test includono tutte le possibili configurazioni di campi, coprendo tutte le variazioni di input.
 - Coperto anche il campo ordine (fisso a 0) per verificare la corretta inizializzazione.
- **Copertura delle Decisioni Multiple (Multiple Condition Coverage)**
 - In ciascun test vengono forniti più campi contemporaneamente → si verifica la composizione completa della mappa dati generata.
- **Copertura dei Valori Nulli o Vuoti**
 - Non sono testati valori null, perché la funzione richiede tutti i campi required. Tuttavia:
 - Verificati campi stringa con valori vuoti e default.

6.3 Richiesta Visita Immobiliare da parte dell'Utente

1. Classi di Equivalenza

Questa funzione effettua la validazione preliminare dell'invio di una richiesta di visita per un immobile, eseguendo i seguenti controlli:

Esempi:

Input nulli o mancanti

- Coperti nei test per ciascun campo obbligatorio:
 - username = null
 - selectedDay = null
 - selectedTime = null

Visite già presenti

- Classe: Visita in attesa per lo stesso immobile
 - stato_visita == "In attesa" e id_immobile corrispondente → blocco visita
- Classe: Visita presente ma con stato diverso (Accettata, Rifiutata, Completata) → permesso invio
- Classe: Visita "in attesa" ma per immobile diverso → permesso invio

Slot orario occupato

- Classe: slot identico già presente in occupiedSlots → errore
- Classe: orario vicino ma non identico → successo

Combinazioni multiple

- Presenza di più visite per lo stesso immobile → sempre errore se almeno una è "In attesa"
- Altre visite con stato diverso → successo

2. Criteri di copertura strutturale

Sono stati applicati diversi criteri di **copertura del codice**:

Copertura delle Condizioni (Condition Coverage)

- Ogni condizione booleana (== null, == "In attesa") è testata separatamente nei singoli test.

Copertura delle Diramazioni (Branch Coverage)

- Tutti i rami della funzione (ritorno precoce per errore, continuazione e successo) sono coperti:
 - if per null
 - if per visitaInAttesa
 - if per occupiedSlots.contains()

Copertura delle Decisioni Multiple (Multiple Condition Coverage)

- In ogni test vengono combinati più campi (utente, data, orario, visite, slot), coprendo anche interazioni non banali.

Copertura dei percorsi (Path Coverage)

- Ogni possibile percorso logico del flusso di esecuzione è stato coperto da almeno un test.

6.4 Validazione Offerta Prezzo Utente

1. Classi di Equivalenza

Questa funzione valida il prezzo offerto da un utente nella pagina di invio offerta. I controlli effettuati sono:

Esempi:

Input non valido

- Campo vuoto → errore: “Inserisci un prezzo valido”
- Campo null → errore: “Inserisci un prezzo valido”
- Input non numerico (abc, ecc.) → errore con messaggio contenente prezzo attuale
- Input con virgola decimale (90000,50) → errore

Valori numerici validi

- Valore inferiore a prezzoAttuale → accettato (ritorna null)
- Valore con decimali ma valido (99999.5) → accettato
- Valore negativo → accettato (è tecnicamente inferiore)
- Valore appena sotto al prezzo attuale (99999.99) → accettato

Valori non validi ma numerici

- Valore uguale o superiore al prezzoAttuale → errore

2. Criteri di copertura strutturale

Sono stati applicati diversi criteri di **copertura del codice**:

Copertura delle Condizioni (Condition Coverage)

- Verifica per `value == null || value.isEmpty`
- Verifica per `double.tryParse(value)` fallita o riuscita
- Verifica per `prezzoOfferto >= prezzoAttuale`

Copertura delle Diramazioni (Branch Coverage)

- Tutti i rami principali esercitati:
 - Campo null o vuoto
 - Parse fallito
 - Parse riuscito ma offerta \geq prezzo attuale
 - Parse riuscito e offerta $<$ prezzo attuale

Copertura dei percorsi (Path Coverage)

- Ogni percorso di esecuzione nel metodo è coperto:
 - Errore per campo vuoto
 - Errore per parse fallita
 - Errore per offerta \geq prezzo attuale
 - Successo per offerta valida

6.5 Tecniche di valutazione dell'usabilità applicate

Per garantire che l'applicazione DietiEstates25 risponda a criteri di usabilità, avere un'interfaccia utente efficace, accessibile e coerente con i principi di user-centered design, sono state applicate tecniche di valutazione dell'usabilità sia **empiriche** (test e simulazioni con casi d'uso) sia **ispettive**, tra cui la **expert review** (o ispezione euristica), che rappresenta il focus di questa sezione.

Descrizione del progetto

La piattaforma è stata realizzata utilizzando **Flutter** e **Dart**, garantendo così il supporto attuale per dispositivi mobili (Android/iOS), e ponendo le basi per una futura espansione verso applicazioni **desktop** (Windows, macOS, Linux) e **web** da un'unica base di codice.

Architettura e Componenti Principali

Il sistema è composto da:

- Un'applicazione mobile multiplattforma sviluppata in Flutter.
- Un backend sicuro con autenticazione gestita da **JWT** e protezione delle credenziali sensibili tramite file **.env**, escluso dal versionamento.
- Un database relazionale strutturato per supportare diverse funzionalità avanzate.

Nel database è stata introdotta la **classe "Modifica"**, che consente di registrare eventuali aggiornamenti o cancellazioni sugli immobili. Questo rende il sistema altamente **estendibile** e pronto per la gestione futura di funzionalità.

Inoltre, grazie alla struttura del campo **tipologiaContratto**, è possibile in futuro includere **immobili destinati ad affitti brevi, case vacanze** e altre formule contrattuali, rendendo la piattaforma estremamente flessibile.

Interfaccia Utente e Accessibilità

La UI è stata progettata per garantire uniformità e usabilità su tutta l'app. Le principali pagine Dart – tra cui LoginPage, SearchPage, MapPage, EditPasswordPage, ImmobileDetailPage – sono riutilizzate in modo coerente tra utenti con ruoli diversi (ad esempio agente e utente), riducendo la duplicazione e migliorando la manutenzione del codice.

Anche pagine apparentemente simili, come VisitsPage e OfferPage, condividono **l'interfaccia grafica** ma offrono logiche differenti secondo il contesto, offrendo un'esperienza utente coerente ma specifica.

I **colori** dell'interfaccia seguono una linea guida visiva coerente con il **logo**: arancione e blu sono utilizzati per mantenere una forte identità visiva in tutte le sezioni dell'applicazione.

Sicurezza e Gestione Utenti

L'**AuthProvider** centralizza la gestione dell'autenticazione per tutti i ruoli utente. Dopo il login, l'utente viene reindirizzato dinamicamente verso l'interfaccia corrispondente al proprio ruolo (Amministratore, gestore, agente, utente).

Tutti i dati sensibili, come le chiavi API (ad esempio per Geoapify, utilizzata per le mappe) e le credenziali di connessione al database, sono **nascosti** nel file **.env**, non incluso nel repository e quindi al sicuro.

Expert Review / Inspection: checklist per revisione sistematica

Obiettivo

L'obiettivo di questa attività è stato valutare il prodotto software da un punto di vista esperto, verificando che rispondesse ai principi fondamentali dell'usabilità, come chiarezza, efficienza, coerenza e accessibilità.

Tale revisione è stata effettuata dagli sviluppatori: **Alessandro Martiniello, Cristina Marano e Luigi Di Giovanni** durante tutto il ciclo di sviluppo del progetto.

Ad ogni nuova implementazione o rilascio parziale è stata aggiornata e riapplicata una **checklist sistematica**, al fine di garantire la qualità e l'usabilità del prodotto prima del rilascio finale.

Checklist adottata

È stata sviluppata una checklist articolata in diverse categorie fondamentali. Per ogni voce sono stati effettuati test qualitativi su dispositivi Ios/Android simulando l'utilizzo da parte dei diversi ruoli (Amministratore, gestore, agente, utente).

Categoria: Criteri e Verifiche

- Coerenza visiva: uso uniforme di colori, font, icone (verificato)
- Navigazione: intuitività dell'interazione (verificato)
- Feedback: messaggi chiari in caso di errore o successo (verificato)
- Controllo dell'utente: azioni reversibili, conferme (verificato)
- Accessibilità: testi leggibili, icone comprensibili (verificato)
- Efficienza: riduzione passaggi per compiti frequenti (verificato)
- Flessibilità: pagine Dart riutilizzate per ruoli diversi (verificato)
- Estendibilità: struttura pronta per nuove funzionalità (verificato)
- Portabilità: predisposizione multiplattaforma (verificato)
- Sicurezza: protezione dati tramite JWT e file .env (verificato)

Applicazione pratica

La checklist è stata applicata alle principali interfacce e ruoli. Sono stati testati i casi più frequenti, tra cui ricerca, prenotazione visite, offerte e inserimento immobili.

Risultati

Punti di forza: coerenza grafica, riuso componenti, preparazione futura, sicurezza.

Conclusione

La valutazione ha confermato l'alta usabilità del prodotto, supportata da una progettazione centrata sull'utente, interfaccia coerente, e architettura pronta a supportare scenari futuri

6.6 Descrizione dei soggetti, della procedura sperimentale e delle metriche considerate

Soggetti reclutati

La selezione dei soggetti che hanno testato il software non è stata semplice, dato che il potenziale modello di utenti è molto ampio e diversificato. È stato però fondamentale scegliere persone con profili differenti, così da raccogliere feedback eterogenei e significativi, utili per valutare l'efficacia del sistema e apportare eventuali miglioramenti.

Metriche considerate

- Tempo di completamento
- Errori commessi
- Numero di tentativi
- Fluidità e autonomia
- Soddisfazione soggettiva

6.7 Procedura sperimentale

Ogni soggetto ha ricevuto un account con permessi coerenti al proprio ruolo e ha completato una sequenza di task su dispositivi mobile. Le attività includevano la ricerca di immobili, prenotazione visite, invio offerte, gestione degli immobili e controllo amministrativo della piattaforma.

6.8 Definizione di un survey somministrato agli utenti dopo l'esperimento

Ai partecipanti è stato somministrato un questionario post-esperimento con scala da 1 a 5 per valutare l'usabilità del sistema.

Questionario:

- L'interfaccia grafica era chiara e intuitiva?
- Le funzionalità erano facilmente accessibili e ben organizzate?
- Ho trovato semplice cercare un immobile con i filtri disponibili?
- Il processo di prenotazione di una visita è stato fluido?
- Inviare una proposta economica è stato semplice e comprensibile?
- Il mio ruolo nel sistema era ben rappresentato e adeguatamente guidato?
- Non ho riscontrato errori o rallentamenti durante l'uso?
- Mi sento soddisfatto dell'esperienza complessiva?
- Commenti o suggerimenti (facoltativo)

Studio

Il grafico seguente rappresenta, in percentuale, la distribuzione per età e genere degli utenti maggiormente attivi nell'utilizzo della piattaforma immobiliare, in particolare per quanto riguarda la consultazione e l'inserimento degli annunci, la prenotazione delle visite e l'invio di proposte economiche.

Dall'analisi emergono due elementi chiave: una partecipazione più significativa da parte del pubblico femminile rispetto a quello maschile e una maggiore attività concentrata nella fascia d'età compresa tra i 30 e i 55 anni. I dati raccolti evidenziano un trend crescente negli ultimi anni: le donne tra i 35 e i 50 anni si dimostrano sempre più coinvolte in maniera attiva nel processo di ricerca immobiliare, dalla selezione dell'immobile alla gestione diretta delle trattative.

Queste osservazioni ci portano a identificare il target prioritario nella popolazione over 30, soprattutto femminile, da tenere in considerazione nello sviluppo dell'interfaccia utente e nella progettazione delle funzionalità principali del sistema, con particolare attenzione alla semplicità d'uso e alla chiarezza informativa.

GENERE



Grafico Genere
Uomo/Donna

ETA'

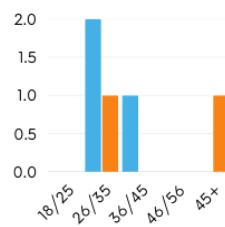


Grafico età compresa tra i
18/45+

6.9 Utenti Selezionati



Elena Caruso 
@ElyCaruso

“L'interfaccia mi è sembrata molto chiara.
Ottimo il sistema di sicurezza e la struttura ruoli.
Molto apprezzata anche la gestione della sicurezza: sapere che
le credenziali e le API sono protette con JWT e file .env dà
fiducia, soprattutto in contesti amministrativi.”

Account Utilizzato
AMMINISTRATORE

Esperienza

Manager presso una catena di ristorante

Competenze informatiche

Buone

Valutazione 1-5

Interfaccia: 5

Funzionalità accessibili: 4

Assenza di errori: 5

Soddisfazione: 5



Esperienza

capo turno in un magazzino logistico

Competenze informatiche

Discrete

Valutazione 1-5

Interfaccia: 4
Funzionalità accessibili: 4
Assenza di errori: 5
Soddisfazione: 4



Esperienza

Assistente di volo

Competenze informatiche

Basse

Account Utilizzato

AGENTE

Valutazione 1-5

Interfaccia: 4

Funzionalità accessibili: 3

Assenza di errori: 5

Ricerca immobile: 4

Accettazione/Rifiuto Visita: 4

Accettazione/Rifiuto Offerta: 4

Rappresentazione del ruolo: 4

Soddisfazione: 4



Esperienza

Educatore in una scuola dell'infanzia

Competenze informatiche

Discrete

Valutazione 1-5

Interfaccia: 5
Funzionalità accessibili: 5
Assenza di errori: 5
Ricerca immobile: 5
Prenotazione Visita: 4
Inserimento Offerta: 4
Rappresentazione del ruolo: 4
Soddisfazione: 5



Esperienza

Tacnino audio in uno studio di registrazione

Competenze informatiche

Ottime

Valutazione 1-5

Interfaccia: 5
Funzionalità accessibili: 5
Assenza di errori: 5
Ricerca immobile: 4
Prenotazione Visita: 4
Inserimento Offerta: 4
Rappresentazione del ruolo: 4
Soddisfazione: 4

6.10 Presentazione e discussione dei risultati dell'esperimento e del survey

Tutti i soggetti hanno completato i compiti assegnati senza errori né difficoltà, dimostrando che la piattaforma è solida, intuitiva e priva di problemi tecnici. Anche utenti con competenze digitali di base hanno portato a termine le attività senza richiedere assistenza.

Inoltre, l'interfaccia utente è risultata chiara e ben strutturata, facilitando la comprensione delle funzionalità disponibili. La navigazione è stata fluida e coerente in tutte le sezioni della piattaforma, contribuendo a un'esperienza complessivamente positiva. I partecipanti hanno evidenziato una percezione di controllo e sicurezza durante l'utilizzo, confermando l'efficacia delle scelte progettuali sia a livello tecnico che comunicativo. Questi risultati indicano che la piattaforma è adatta a un pubblico ampio e variegato, e può essere introdotta su scala più estesa senza necessità di modifiche sostanziali.

**Contattaci
per ulteriori
domande**

