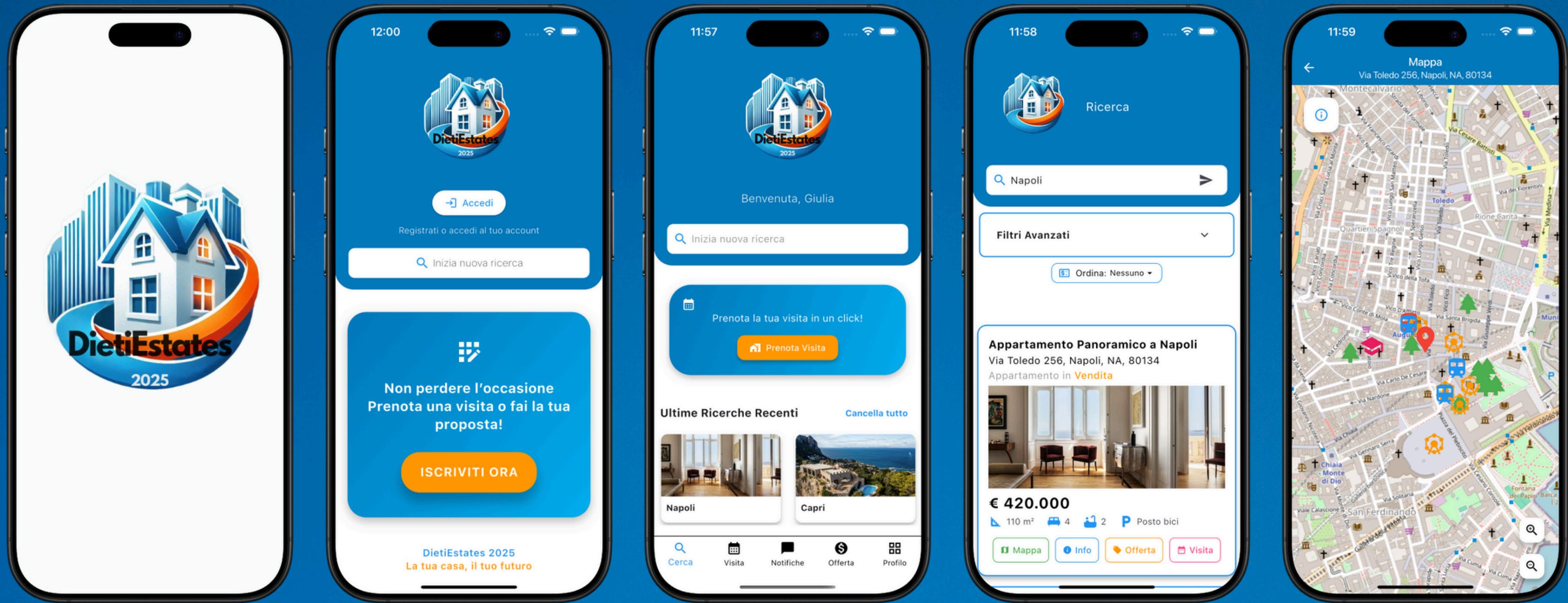
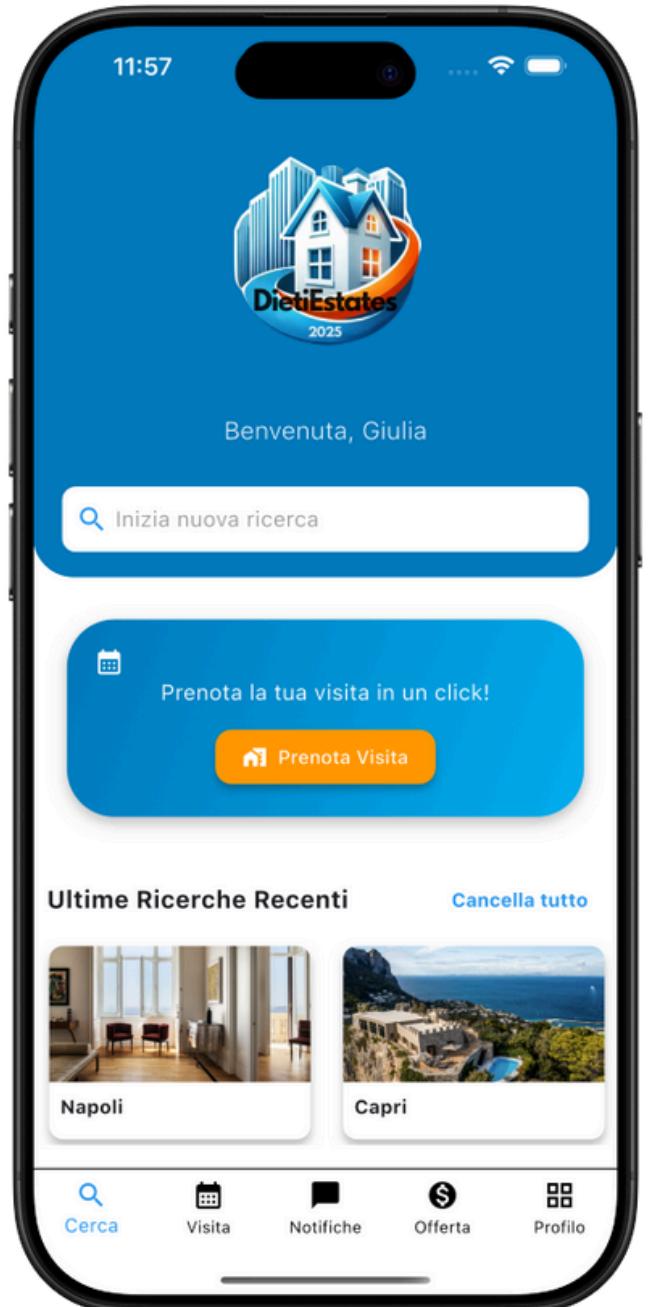


Software Engineering





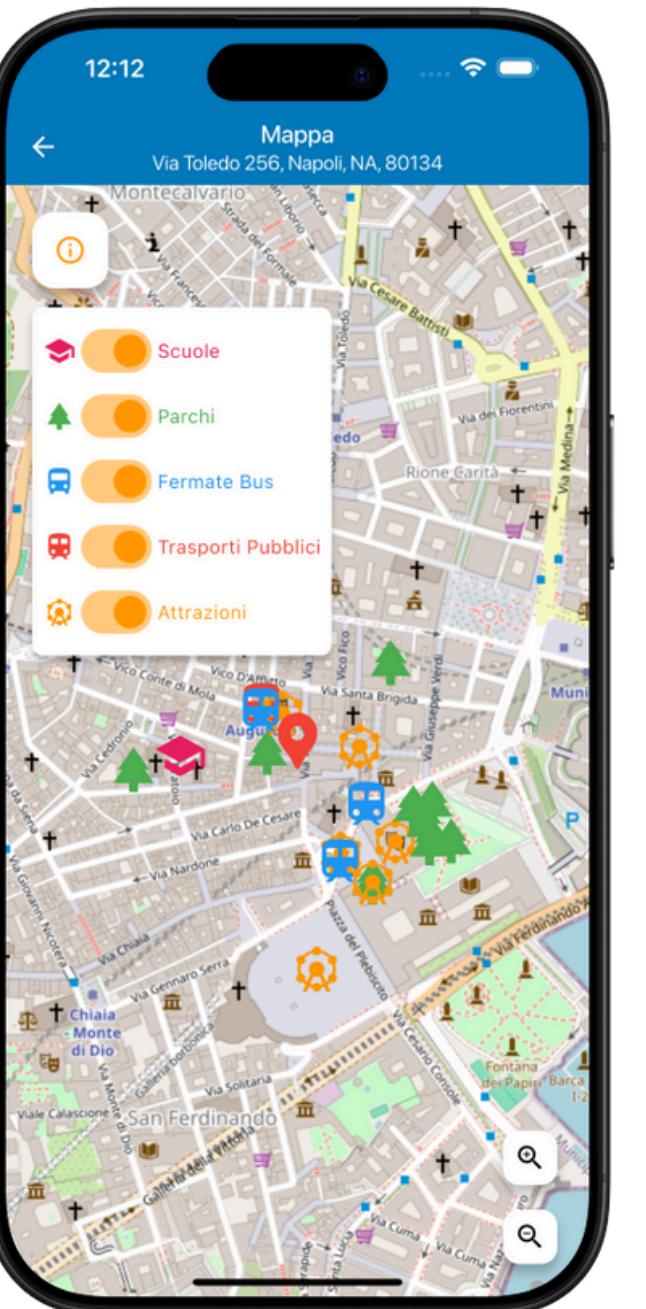
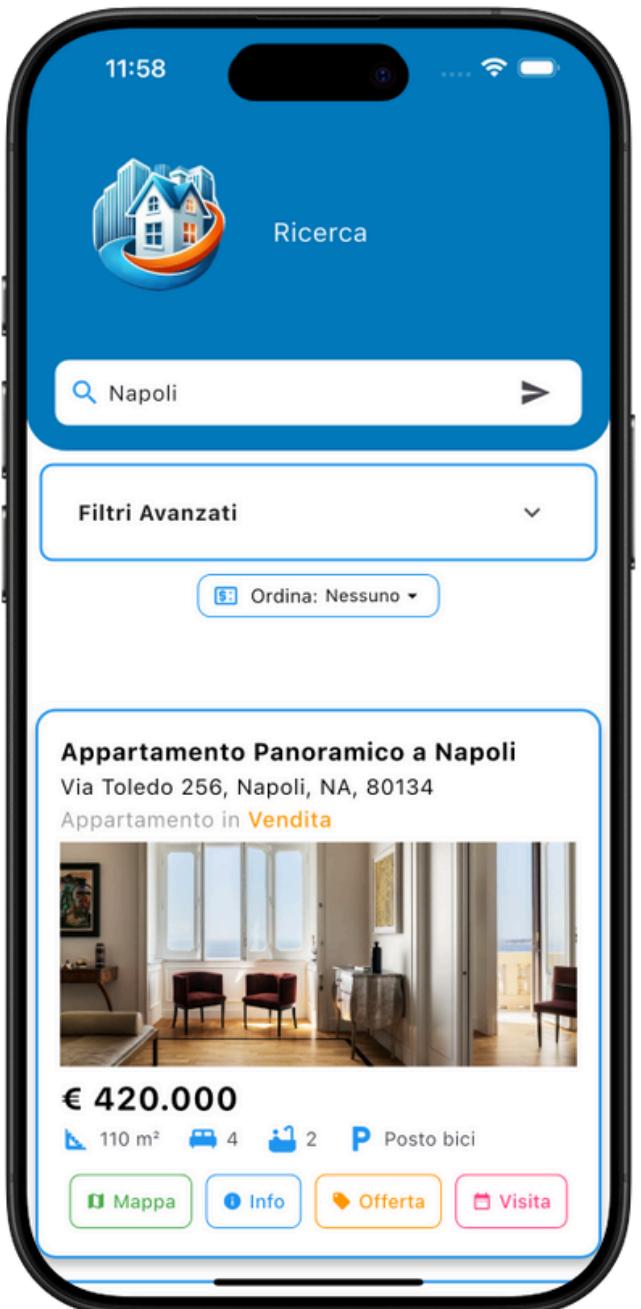
DietiEstates 2025



DietiEstates2025 nasce come risposta alle esigenze sempre più complesse del settore immobiliare, unendo precisione organizzativa e facilità d'uso. Il nostro approccio si è fondato su una rigorosa applicazione dei principi dell'Ingegneria del Software: dalla raccolta e validazione dei requisiti, alla definizione dell'architettura, fino alla valutazione dell'usabilità.



Scelte di Progettazione

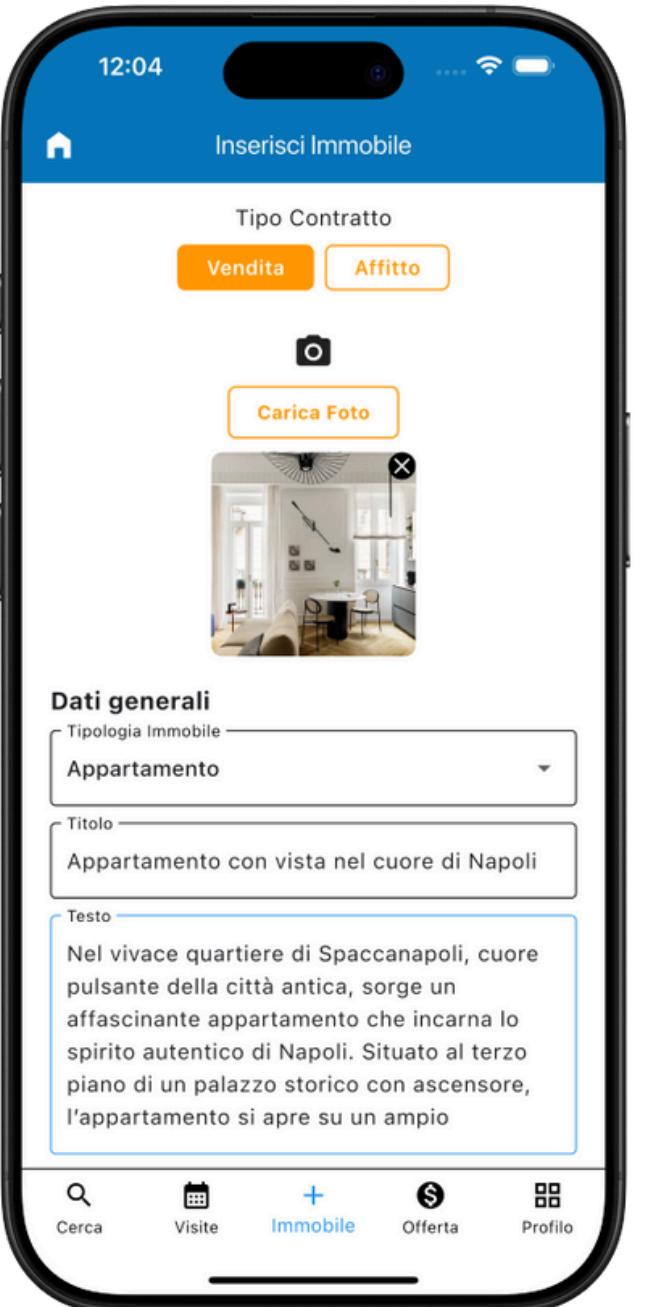


Architettura client-server con Flutter e Node.js + PostgreSQL, orientata a scalabilità e manutenibilità.

- **Client (Flutter):** assicura un'interfaccia coerente su più piattaforme, elevata reattività e un'esperienza utente fluida.
- **Server (Node.js + Express):** ottimizzato per alte prestazioni in scenari concorrenti grazie alla gestione asincrona.
- **Database (PostgreSQL):** schema normalizzato con vincoli, trigger e viste per garantire integrità e coerenza dei dati.
- **Geoapify:** consente visualizzazione e ricerca geolocalizzata in tempo reale, evitando il salvataggio di coordinate statiche.



Approccio dei requisiti



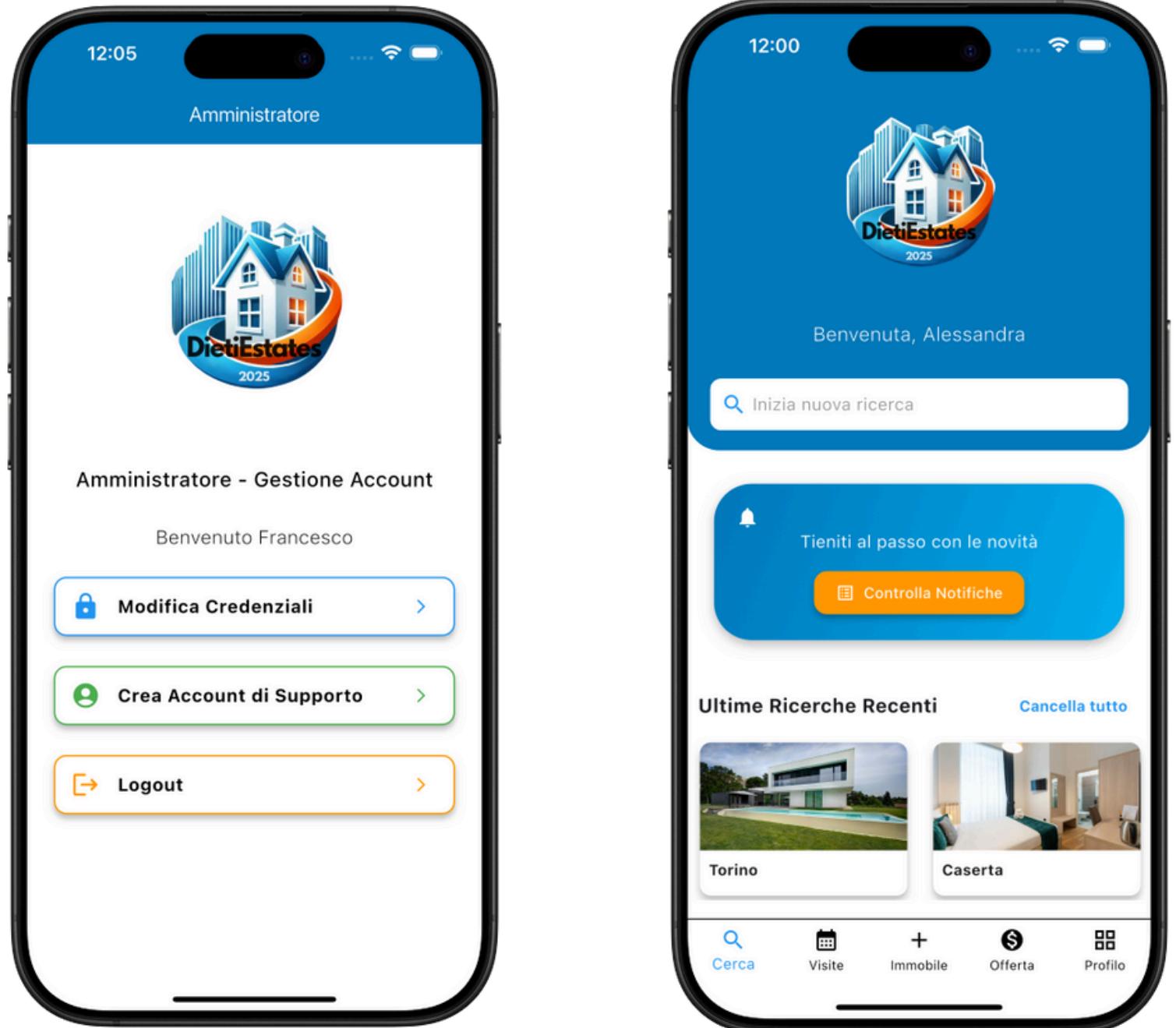
L'intero ciclo di sviluppo è stato guidato da un processo strutturato di raccolta, analisi e validazione dei requisiti. In particolare:

- **Requisiti funzionali**, identificati attraverso interviste simulate con stakeholder, l'utilizzo di prototipi a bassa fedeltà e la somministrazione di questionari agli utenti coinvolti nei test
- **Requisiti non funzionali**, definiti sin dalle prime fasi, con vincoli misurabili in termini di prestazioni e sicurezza
- **Requisiti di dominio**, elaborati per garantire l'aderenza alle prassi e normative del settore immobiliare

L'adozione di personas ha consentito di esplorare i requisiti dal punto di vista dell'utente finale, assicurando coerenza con i reali bisogni e aspettative.



Requisiti non-funzionali



Il sistema è stato progettato per garantire un equilibrio tra prestazioni, sicurezza, usabilità e manutenibilità:

- **Prestazioni:** query ottimizzate e caricamento delle pagine entro 2 secondi per garantire un'esperienza utente reattiva.
- **Scalabilità:** infrastruttura progettata per crescere senza compromettere le prestazioni.
- **Usabilità:** interfaccia unica e coerente, articolata in pagine dedicate alle diverse funzionalità per garantire accessibilità, intuitività e continuità nell'esperienza utente
- **Manutenibilità e portabilità:** codice ben documentato e architettura modulare che facilitano aggiornamenti, riuso e deploy cross-platform.



Creazione dei Mockup



Le tabelle di Cockburn ci hanno permesso di classificare i casi d'uso in base a complessità e frequenza, guidando:

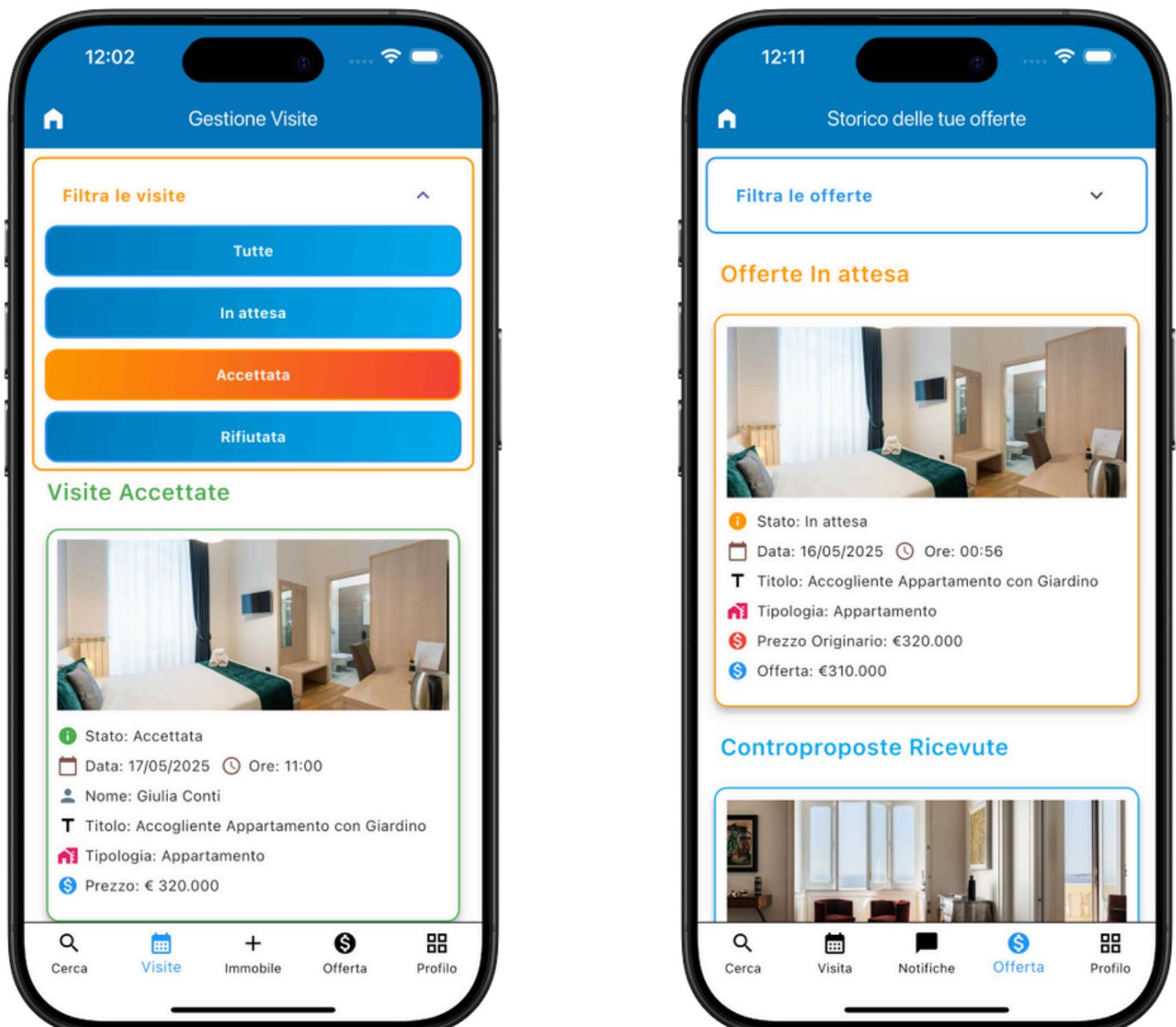
- **la prioritizzazione dello sviluppo,**
- **il livello di dettaglio nella documentazione,**
- **la profondità dei test da applicare.**

I mockup a media fedeltà hanno supportato la progettazione dell'interfaccia permettendo di:

- **definire l'esperienza utente prima della programmazione,**
- **allineare il team e gli stakeholder su una visione comune,**
- **tradurre rapidamente i requisiti in soluzioni visive e funzionali.**



Design dell'interfaccia



Il design dell'interfaccia è stato progettato secondo i seguenti principi chiave:

- **Semplicità e coerenza visiva:** ogni elemento è stato pensato per essere chiaro, essenziale e coerente nell'aspetto e nel comportamento.
- **Identità visiva unificata:** partendo dal logo, abbiamo utilizzato la stessa **paletta cromatica** “blu e arancione” per trasmettere un’immagine coordinata e riconoscibile.
- **Interfaccia minimale e intuitiva:** l’uso armonico dei colori e una disposizione pulita degli elementi favoriscono un’interazione naturale e immediata da parte dell’utente.
- **Esperienza utente migliorata:** la coesione grafica contribuisce a rendere l’applicazione più accessibile, gradevole e professionale.



Strategie di sicurezza

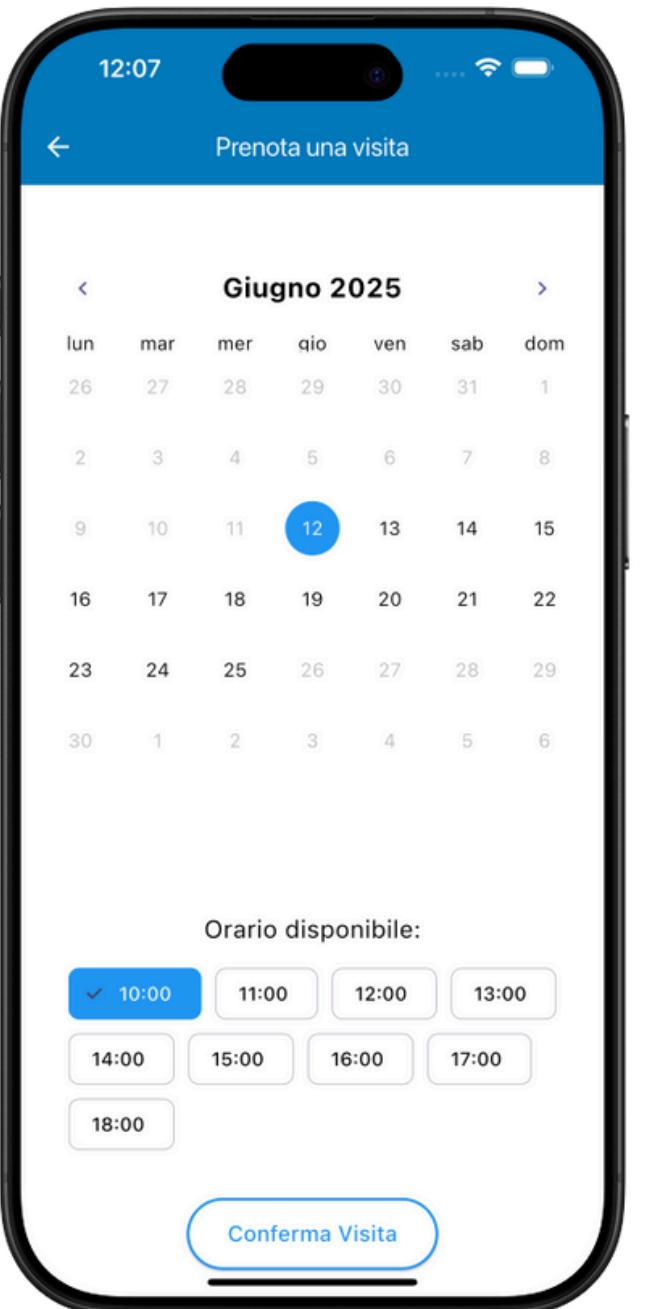
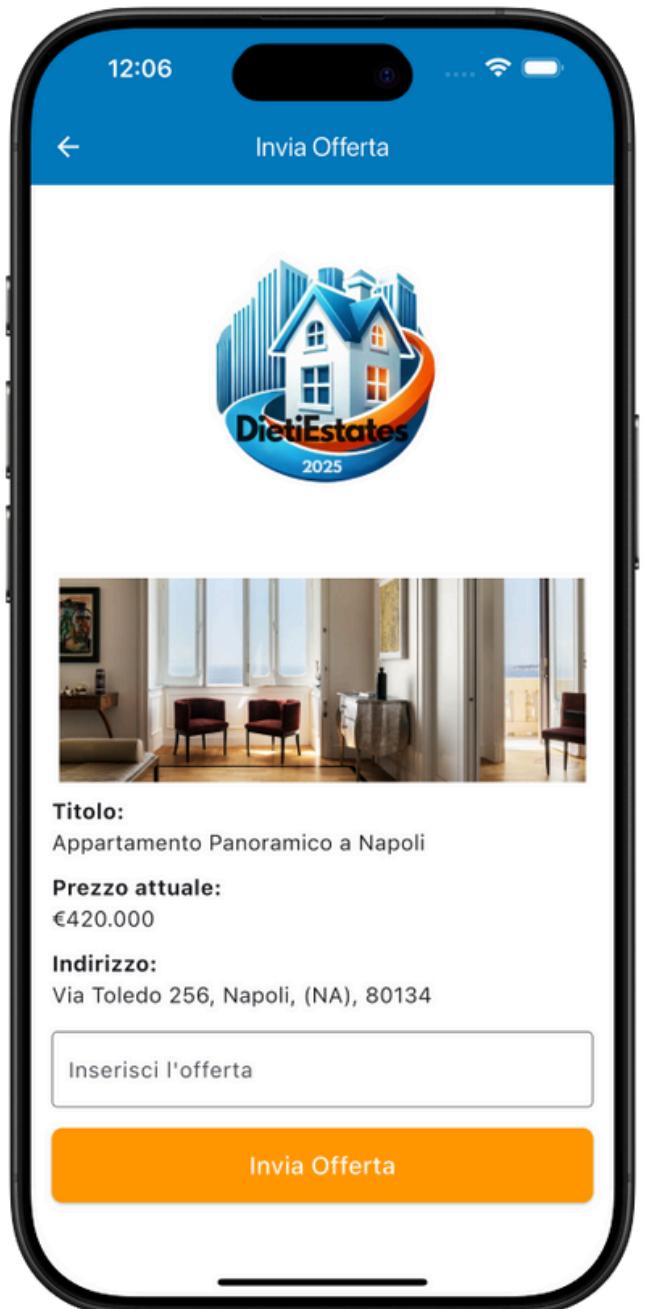


- **Autenticazione con JWT:** per garantire accessi sicuri e stateless.
- **Autorizzazione** con controlli basati sui ruoli per delimitare l'accesso alle risorse.
- **Query parametrizzate** in PostgreSQL per una protezione efficace contro attacchi SQL injection.

Queste misure concorrono a rafforzare la sicurezza, riducendo la superficie d'attacco e garantendo il rispetto delle policy di accesso e integrità dei dati.



Testing e gestione dei bug

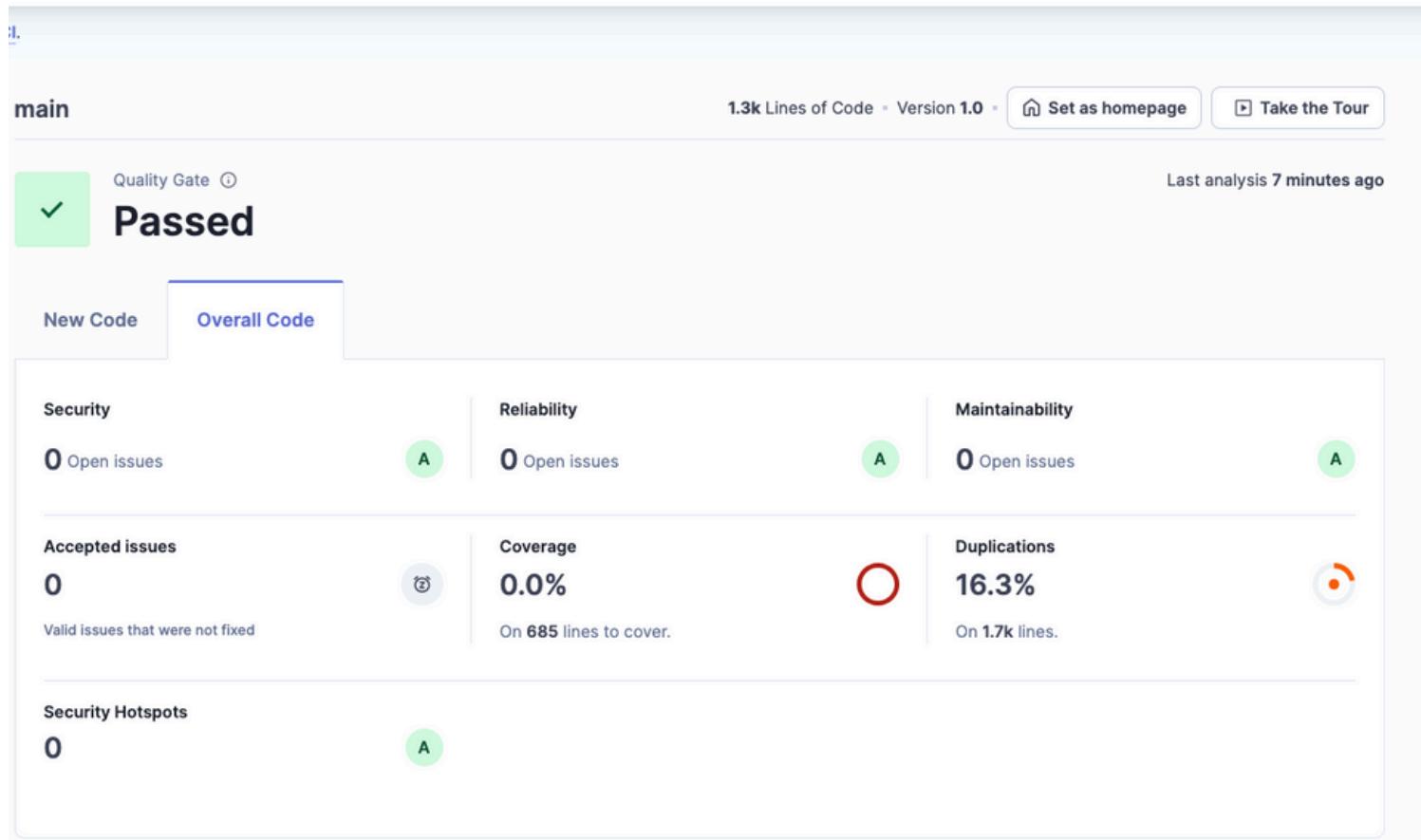


Abbiamo eseguito test manuali e automatici per individuare e correggere eventuali bug, concentrando su:

- **Test funzionali delle principali operazioni:** inserimento dati, filtro avanzato, prenotazione di visite e gestione delle offerte
- **Analisi di comportamenti anomali** e verifica della corretta gestione degli errori, con particolare attenzione alla chiarezza dei messaggi di supporto all'utente



Controllo con SonarQube



Abbiamo integrato **SonarQube** nel processo di sviluppo per effettuare analisi statica continua del codice, rilevando automaticamente **code smells**, **errori sintattici** e **vulnerabilità di sicurezza**. Questo ha contribuito a mantenere elevati standard qualitativi, facilitando il refactoring, riducendo il debito tecnico e permettendo un intervento tempestivo su potenziali criticità prima che si manifestassero in fase di esecuzione o rilascio.



Manutenzione



Il sistema è stato progettato per essere facilmente **manutenibile** ed estendibile grazie a:

- **Codice strutturato per livelli logici** (presentazione, logica, persistenza) e ben commentato
- **Configurazioni esterne e costanti** per una maggiore flessibilità
- **Struttura modulare**, che facilita l'aggiunta di filtri o ruoli



Futuro evolutivo



DietiEstates25 è disponibile come app mobile per **iOS** e **Android**. Prossimamente sarà accessibile anche da desktop (**Windows** e **macOS**) e via **web**, per garantire la massima flessibilità.

Sono previsti ulteriori sviluppi, tra cui:

- **Pannello di gestione per amministratori e gestori**
- **Espansione del database per nuove funzionalità**

Il database è già strutturato per:

- **Modifica e gestione degli immobili**
- **Estensione a nuove tipologie di utenti e immobili (es. case vacanza, affitti brevi)**

L'architettura modulare consente un'evoluzione agile e non invasiva del sistema.



Conclusione



Il progetto **DietiEstates25** dimostra l'applicazione concreta dei fondamenti dell'Ingegneria del Software: dalla raccolta dei requisiti alla verifica sperimentale della qualità. Le scelte architetturali, tecnologiche e di design sono state documentate, motivate e validate per garantire una piattaforma sostenibile, efficiente e realmente usabile.



Grazie per l'attenzione