

Decisioni di Object oriented design

Responsabilità e classi individuate

Admin: rappresenta gli amministratori che si autenticano nell'applicazione tramite un email specializzata per il ruolo (ad esempio admin@dietiEstates25.it) ed una password fornita dall'amministratore nominante(ovvero quello che l'ha nominato). Tra di essi vi sarà anche il gestore della agenzia immobiliare, che invece non è nominato da un amministratore ma dai responsabili della piattaforma, quindi può non "conoscere" l'amministratore nominante.

1. Conosce la sua password e la sua email personalizzata.
2. Conosce l'amministratore nominante.
3. Crea account per amministratore.
4. Crea account per agente immobiliare.
5. Gestisce un annuncio immobiliare.
6. Modifica la password e l'email di autenticazione.

EstateAgent: rappresenta un agente immobiliare, il cui account viene sempre creato da un amministratore. Anch'esso disporrà di una email personalizzata per il ruolo (ad esempio agent@dietiEstates25.it). Tra le principali mansioni vi sono le interazioni con gli utenti, la gestione di annunci immobiliari e la revisione di offerte. In particolare, la visualizzazione di offerte e di visite avviene esclusivamente in modo correlato ad un immobile, quindi nella pagina di visualizzazione di quest'ultimo.

1. Conosce la sua email personalizzata e la sua password.
2. Conosce l'admin che ha creato il suo account.
3. Pubblica nuovi annunci immobiliari.
4. Valuta offerta(accetta o rifiuta) di un utente per un immobile.
5. Presenta una controproposta all'offerta di un utente per un immobile.
6. Gestisce un annuncio immobiliare(modifica o cancella).
7. Inserisce offerte di utenti esterne al sistema.
8. Visualizza lo storico delle offerte avanzate relativamente ad un immobile.
9. Gestisce le visite richieste da un utente(accetta o rifiuta).

RealEstate: rappresenta un immobile di interesse per il quale è possibile creare un annuncio immobiliare. Quando verrà visualizzato il corrispettivo annuncio, dovranno contestualmente essere recuperate tutte le offerte avanzate per l'immobile e le visite richieste ed eventualmente accordate, visualizzabili sia dall'agente che dall'utente. (idea: potremmo suddividere come concetti immobile da annuncio immobiliare, in modo tale che uno stesso immobile possa essere presente, ad esempio, in un annuncio di affitto e in uno di vendita contemporaneamente)

1. Conosce la foto di pubblicazione.

2. Conosce la descrizione di pubblicazione.
3. Conosce il prezzo.
4. Conosce le dimensioni espresse in metri quadri.
5. Conosce l'indirizzo.
6. Conosce il numero di stanze.
7. Conosce il piano nel condominio(potrebbero esserci anche delle ville come immobili?).
8. Conosce la presenza di un ascensore.
9. Conosce eventuali servizi esterni(portineria...).
10. Conosce la classe energetica.
11. Conosce la categoria, ovvero affitto o vendita, espandibile in futuro.
12. Conosce la posizione geografica esatta individuata in una mappa interattiva.
13. Conosce eventuali "interest spot" come scuole vicine, parchi pubblici oppure fermate dei mezzi.
14. Recupera tutte le offerte avanzate da utenti ed agenti immobiliari dallo storico di offerte.
15. Recupera le visite (storico oppure no?) richieste dagli utenti e accettate o rifiutate dagli agenti.
16. Conosce l'amministratore che può modificare o cancellare l'annuncio.
17. Conosce l'agente che ha creato l'annuncio, e che lo può modificare o cancellare.
18. Conosce le ricerche a cui è associato l'annuncio.

User: rappresenta un utente autenticato, tramite login e registrazione (anche con terze parti) che ha le facoltà di negoziare la compera di un immobile direttamente con un agente immobiliare, di effettuare delle ricerche presentanti dei parametri per il filtraggio degli annunci, che verranno salvate in uno storico personalizzato dell'utente, e di prenotare delle visite per vedere l'immobile.

1. Conosce l'email dell'account e la password
2. Ricerca un annuncio immobiliare, specificando i parametri di interesse.
3. Avanza un offerta per l'acquisto di un immobile all'agente immobiliare che gestisce l'annuncio.
4. Disabilita categorie di notifiche indesiderate.
5. Visualizza lo storico di tutte le offerte presentate ad un immobile (le sue offerte o le offerte di tutti?) nella pagina stessa dell'immobile.
6. Visualizza lo storico delle ricerche che ha effettuato, per rieseguirle facilmente per controllare nuove proprietà in linea con i parametri di ricerca.
7. Richiede una visita dell'immobile all'agente immobiliare competente.

Offer: rappresenta un'offerta formale di acquisto avanzata da un utente ad un agente immobiliare per l'acquisto di un immobile, oppure una contro offerta che invece è presentata dall'agente immobiliare in risposta ad un'offerta non soddisfacente di un utente.

1. Conosce l'importo di acquisto.
2. Salva l'offerta o la contro offerta avanzata nello storico di tutte le offerte di un immobile.
3. Conosce l'agente immobiliare e l'utente coinvolti in un'offerta/controofferta.

Search: rappresenta una ricerca parametrizzata che l'utente esegue per ricercare un immobile di interesse nel sistema, e ogni volta che viene eseguita una nuova ricerca quest'ultima viene salvata nello storico che contiene tutte le ricerche eseguite dall'utente, in modo che possano facilmente essere rieseguite successivamente.

1. Conosce la categoria dell'annuncio immobiliare, ovvero vendita o affitto.
2. Conosce il prezzo minimo dell'immobile.
3. Conosce il prezzo massimo dell'immobile.
4. Conosce il numero di stanze dell'immobile.
5. Conosce potenzialmente tutti i campi di un immobile (da decidere).
6. Conosce la classe energetica di un immobile.
7. Conosce la città relativa ad un annuncio immobiliare.
8. Conosce la municipalità relativa ad un annuncio immobiliare.
9. Conosce un raggio abitativo selezionato su una mappa interattiva.
10. Seleziona gli immobili che soddisfano la ricerca.
11. Salva la ricerca nello storico delle ricerche del corrispettivo utente.
12. Conosce l'utente che ha effettuato la ricerca.

Notification: rappresenta un messaggio che viene recapitato all'utente (potrebbe essere che una notifica viene inviata ad esempio anche all'agente immobiliare quando è disponibile una nuova offerta oppure una richiesta di una visita? Sarebbe una funzionalità in più, ma si deve adattare il sistema a questo probabile cambiamento futuro) ogni qual volta che accade un evento di interesse, come la pubblicazione di un nuovo annuncio, messaggi promozionali, oppure altro

1. Conosce il testo da notificare.
2. Conosce la categoria della notifica.
3. Conosce l'utente destinatario di una notifica

OfferHistory: rappresenta lo storico dove vengono salvate e registrate permanentemente tutte le offerte e le contro proposte avanzate sia dagli agenti che dagli utenti per un immobile. La richiesta di presentare una lista di tutte offerte e controproposte viene avanzata da un annuncio immobiliare, che quindi consulterà lo storico.

1. Conosce le offerte avanzate da utenti e agenti immobiliari.

2. Conosce l'annuncio immobiliare per cui deve creare una lista di offerte.
3. Invia tutte le offerte registrate all'annuncio immobiliare che le richiede per la visualizzazione.

SearchHistory: rappresenta lo storico all'interno del quale vengono salvate tutte le ricerche eseguite di un utente, con i corrispettivi parametri (da decidere quali parametri effettivamente presenta al fine di riutilizzare quelli dell'immobile), che a loro volta individueranno una lista di immobili (o annunci immobiliari) che rispettano i parametri specificati dall'utente.

1. Conosce l'utente che ha effettuato la ricerca.
2. Conosce le ricerche effettuate dall'utente.
3. Invia tutte le ricerche registrate all'utente che le richiede.

Appointment (DA RIFINIRE): rappresenta una richiesta di visita che può avanzare un utente nei confronti di un agente immobiliare, al fine di visitare il corrispettivo immobile nell'annuncio. Gli appuntamenti sono prenotabili direttamente nella pagina di visualizzazione di un immobile, e non ci sarà uno storico di tutte le visite effettuate, ma verrà visualizzato solo il prossimo appuntamento preso.

1. Conosce la data dell'appuntamento
2. Invia tutte le visite registrate (no, non è uno storico).
3. Conosce l'utente che ha richiesto la prenotazione.
4. Conosce l'agente immobiliare che deve valutare la prenotazione.

Da decidere.

1. Formato email e password predefinita di agente immobiliare e amministratore.
2. Suddividere un annuncio immobiliare da un immobile, in modo tale che possano esistere più annunci immobiliari che si riferiscano allo stesso immobile, ad esempio per la vendita e per l'affitto. Inoltre, questo favorirebbe una suddivisione degli attributi di RealEstate in modo più gestibile.
3. Introduzione di tipologie differenti di immobili, ad esempio ville che non richiedono un parametro/attributo di ricerca "floor".
4. Presenza anche di uno storico delle visite richieste ed approvate per un immobile (orientativamente no, funzionalità in più).
5. Un utente è in grado di consultare tutta la storia delle offerte e delle controproposte degli agenti riguardo un immobile, quindi sarà in grado di vedere anche quelle avanzate da altri utenti intenzionati alla compra, oppure visualizza lo storico solo delle proprie offerte, senza quelle altrui.
6. Stabilire la similitudine dei campi di un annuncio immobiliare con i parametri di una ricerca eseguita dall'utente.
7. Stabilire se una notifica viene inviata solo ad un utente al verificarsi di relativi eventi, oppure (funzionalità in più) viene inviata anche, ad esempio, agli agenti immobiliare quando

è presente una nuova offerta/visita. Almeno si deve predisporre il sistema per l'aggiunta di questo meccanismo in futuro.

8. Definire le dinamiche della richiesta di una visita e stabilire se una visita è relegata ad un annuncio immobiliare oppure tramite un meccanismo diverso da quello delle offerte.