System Design Document

Componenti Del Gruppo

Giuseppe Di Martino	0512101162
Giuseppe Giordano	0512101798
Luca Diodato De Martino	0512102658
Francesco Napoli	0512101928

Sommario

Revision History

Data	Versione	Descrizione	Autore
1/12/2015	1.0	Prima versione	Tutto il gruppo
2/122015	1.1	Versione senza messaggi	Tutto il gruppo

1.0 Introduction

1.1 Purpose of the system

L'obiettivo del progetto ImmobilUnisa è quello di realizzare un sistema software in grado di gestire un'agenzia immobiliare e i relativi servizi di compravendita offerti. In particolare il sistema catalogherà gli immobili in vendita, e permetterà agli utenti l'apertura di trattative per l'acquisto o la vendita di un immobile. Inoltre il sistema permetterà l'assegnazione di trattative agli agenti e favorirà la comunicazione tra questi e i clienti.

1.2 Design goals

Il sistema ImmobilUnisa deve poter immagazzinare determinate informazioni per ogni singolo immobile. In particolare, tra i criteri di scelta fondamentale per la definizione della struttura del sistema, non può quindi mancare un'attenta e sicura gestione dei dati persistenti, oltre alla possibilità di permettere ad un gran numero di persone di connettersi contemporaneamente. Gli obiettivi di design identificati rispecchiano quattro tipologie di categorie ben distinte: Performance, Dependability, Maintenance ed End User Criteria.

Criteri di Performance

Throughput	Il sistema sarà capace di servire contemporaneamente un numero diverso di utenti dipendente dalle capacità di elaborazione della macchina (server) sulla quale verrà installato. In particolare si prevede un carico medio di 50 utenti giornalieri. Viene fatta una stima approssimata in quanto non è possibile definire il bacino di utenza che deciderà di utilizzarne i servizi attraverso immobilUnisa. Nonostante tutto, se la domanda dei servizi da parte dei clienti aumentasse, sarà possibile gestirla grazie alla struttura altamente scalabile del sistema.
Tempo di risposta	ImmobilUnisa deve garantire tempi di risposta decisamente brevi o quantomeno notificare nel minor tempo possibile eventuali indisponibilità del sistema stesso. Una richiesta di un utente deve essere soddisfatta entro 3 secondi tenendo conto di una velocità media di trasferimento dati di 15-20KBps. Gli accessi in lettura e scrittura alla base di

	dati del sistema saranno comunque eseguiti tanto più rapidamente quando maggiore è la velocità di elaborazione della macchina (server) sulla quale il sistema verrà installato e la comunicazione client/server (la velocità del sistema varierà in base alla tipologia di connessione posseduta 56K – ADSL – Fibra ottica).
Memorizzazione	Per quanto riguarda lo spazio necessario alla memorizzazione dei dati, anche per questo caso non è possibile fare una stima assoluta. Alcuni fattori dipendono strettamente dall'agenzia come: numero di agenti, numero di clienti, numero di immobili (tutte informazioni memorizzate nel sistema). Il sistema dovrà inizialmente essere proposto alla memorizzazione di almeno 200 clienti, 20 agenti, 300 immobili.

Criteri di affidabilità

Sicurezza	L'accesso al sistema sarà controllato da un apposito sistema di autenticazione, che permetterà ad ogni categoria di utente di eseguire il proprio lavoro senza intaccare o modificare quello altrui.
Robustezza	ImmobilUnisa deve gestire eventuali input errati dell'utente senza interrompere il suo completo funzionamento.

Criteri di manutenzione

Estendibilità	Deve essere possibile introdurre nuove funzionalità senza intaccare il design del sistema. A tal proposito il codice dovrà essere di facile comprensione per non creare confusione durante l'aggiunta di nuove funzionalità
Modificabilità	Deve essere possibile intervenire sul codice esistente per correggere eventuali bugs, o implementare nuove funzionalità aggiuntive. Per ottenere questo requisito, anche in questo caso bisogna garantire che il codice sia leggibile per facilitare le modifiche o l'aggiunta di nuove funzionalità
Leggibilità	Il codice deve essere correttamente strutturato e documentato per semplificare eventuali interventi su di esso. I commenti rappresenteranno il 20-30% delle linee codice prodotto in quanto una percentuale più alta minerebbe alla leggibilità del codice stesso. I commenti saranno brevi e non

	banali ed avranno uno stile consistente per tutto il codice prodotto. Sarà utilizzata in modo appropriato l'indentazione sia per il codice sia per i commenti.
Tracciabilità	Le attività di progettazione, definizione e gestione dei requisiti sono alla base di un progetto di successo. Effettuando la tracciabilità dei requisiti sarà possibile ridurre costi e rischi, valutando l'impatto delle modifiche effettuare.

Criteri di End User

Usabilità	Ogni funzione presente ed utilizzabile è resa disponibile tramite interfacce grafiche gradevoli, essenziali e di comprensione immediata, allo scopo di facilitare il compito degli utilizzatori.

1.3 Overview

Gli elementi del modello di system design sui quali si basa l'architettura software proposta sono:

- Decomposizione del Sistema, nella quale viene descritta la suddivisione del sistema in vari sottosistemi.
- Mapping hardware/software, in cui vengono prese alcune decisioni per quanto riguarda le piattaforme hardware e software su cui il sistema dovrà girare, una volta decise le piattaforme è necessario mappare le componenti su di esse.
- Gestione dei dati Persistenti, in cui sono identificati gli oggetti preesistenti e scelto il tipo strutturato da usare per memorizzarli.
- Controllo degli accessi e sicurezza, che descrive il modello utente del sistema in termini di matrici di accesso.
- Flusso di controllo globale, che descrive come il controllo globale del software è implementato, come le procedure di richiesta sono avviate e come si sincronizzano I sottosistemi.
- Condizioni Limite, in cui sono descritte le condizioni limite del sistema le quali includono errore di connessione al DBMS o errore di connessione al Server.

1.4 Riferimenti

- Bernd Bruegge e Allen H. Dutoit - Object-Oriented Software Engineering– Prentice Hall.
- Sommerville, Software Engineering – Addison Wesley.
- Documentazione di progetto.

2.0 Current software Architecture

Il sistema sarà sviluppato ex-novo, non risulta esserci attualmente un sistema che racchiuda tutte le funzionalità di immobilUnisa.

3.0 Proposed Software Architecture

3.1 Overview

Il sistema ImmobilUnisa è basato su architettura client/server dove il server riceve le richieste da parte dei client e risponde entro i limiti di tempo prestabiliti. La scelta di utilizzo dell'architettura client/server è spiegata nei seguenti punti:

- Portabilità: il sistema funzionerà su dispositivi eterogenei, ad esempio notebook, desktop. Inoltre l'architettura scelta permette al sistema di funzionare su una qualsiasi topologia di rete.

- Semplicità di gestione delle informazioni: qualsiasi operazione di input coinvolge il server, che quindi ha la caratteristica di gestire il trattamento delle informazioni incaricando la componente dedicata ad eseguire tale compito (ad esempio in una stampa delle informazioni di un immobile è incaricato il gestore dell'output e il DBMS per reperire le informazioni dal database)

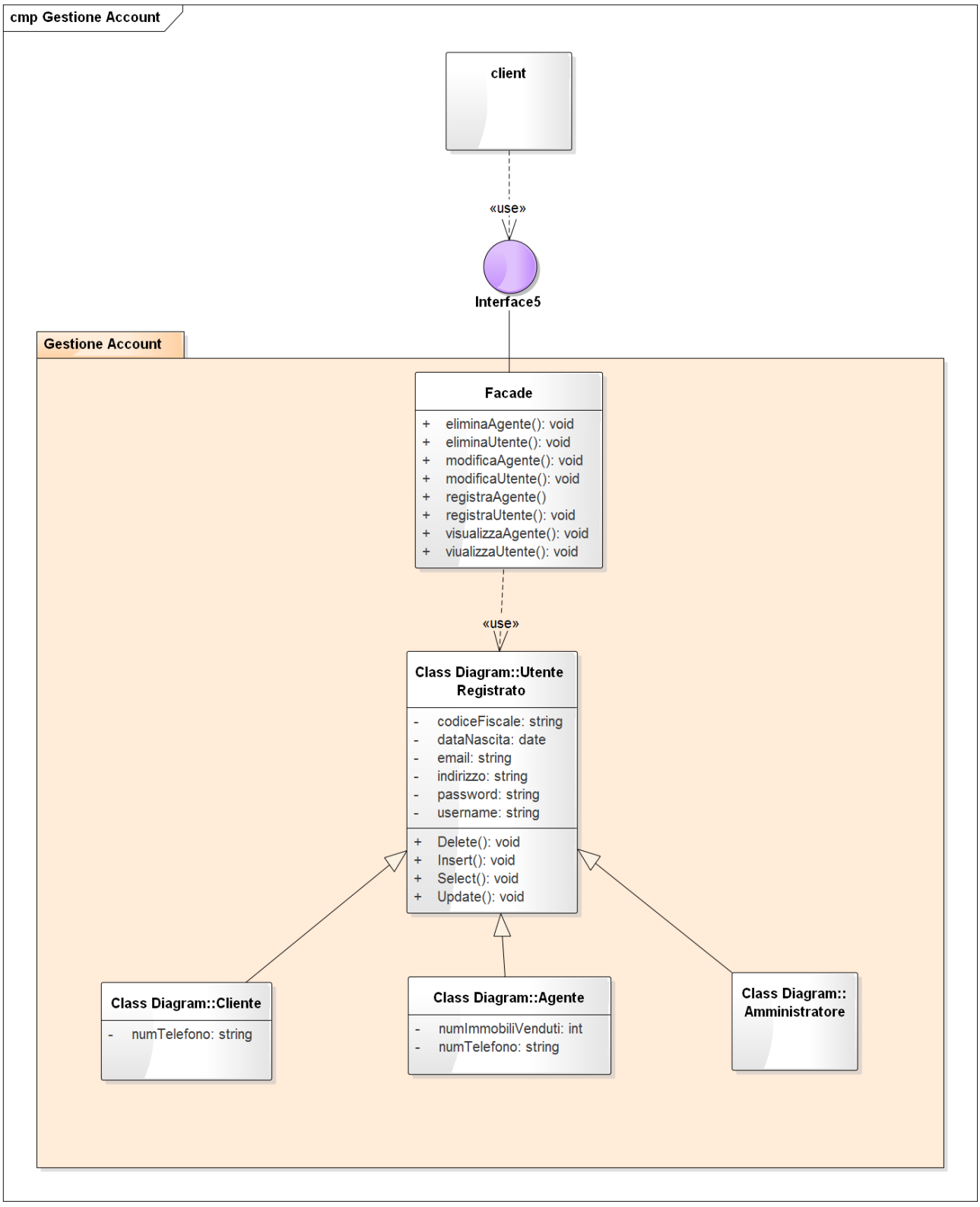
- Flessibilità: I vari client (amministratore, agente, cliente) hanno un'interfaccia grafica dedicata per svolgere le operazioni ad essi riservate.

L'applicazione finale è strutturata su tre livelli:
- interface layer
- application logic layer
- storage layer

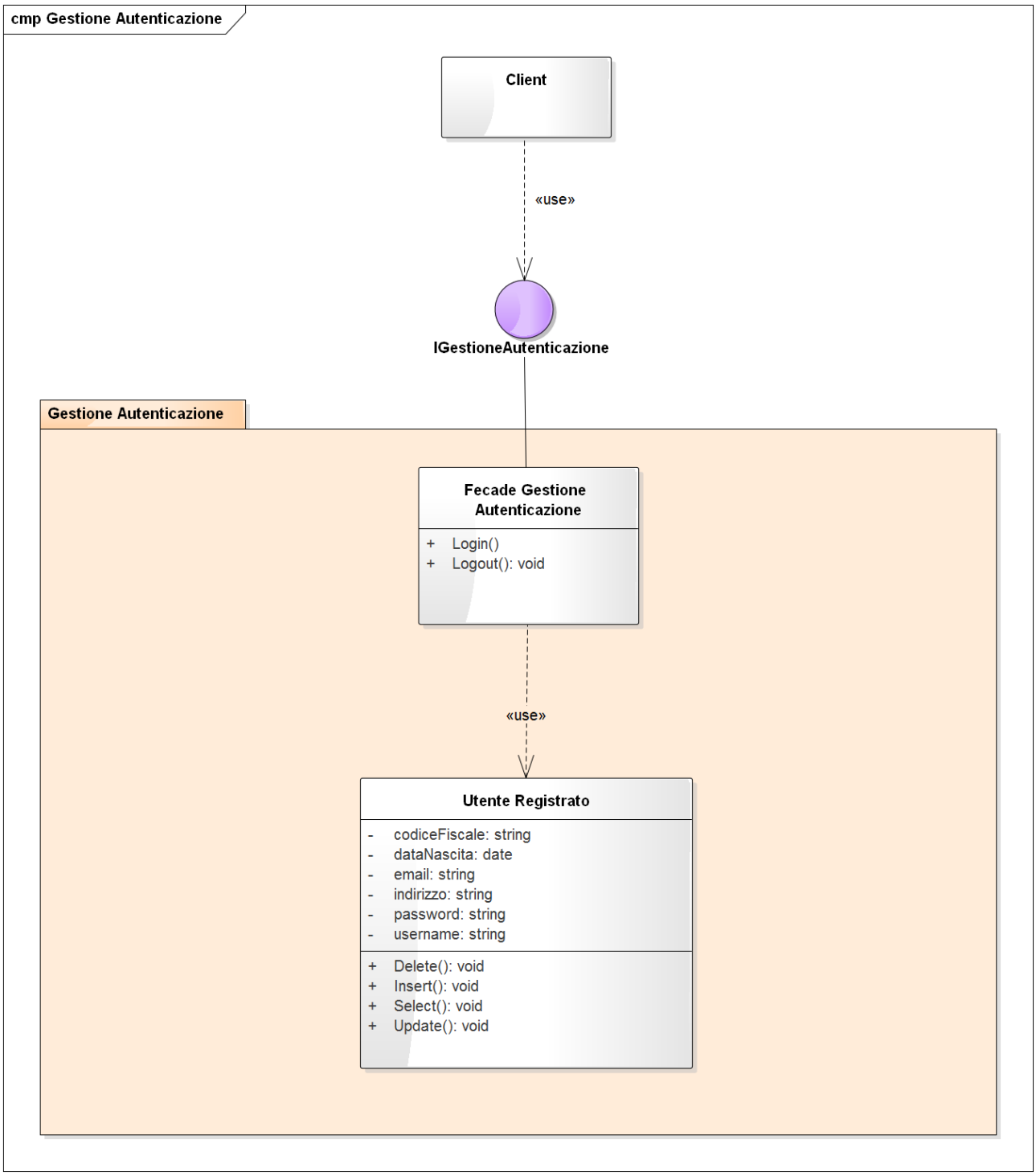
L'applicazione finale lavora quindi su tre strati dove il primo strato si occupa di interfacciarsi con l'utente finale, lo strato intermedio comprende tutto il software applicativo lato client e server che si occupa sia dell'interazione uomo-macchina sia dell'interazione tra software client e software server. Infine il terzo strato si occupa della gestione dei dati sensibili interrogando il DBMS e rispondendo al layer intermedio con i risultati delle query.

3.2 Subsystem decomposition

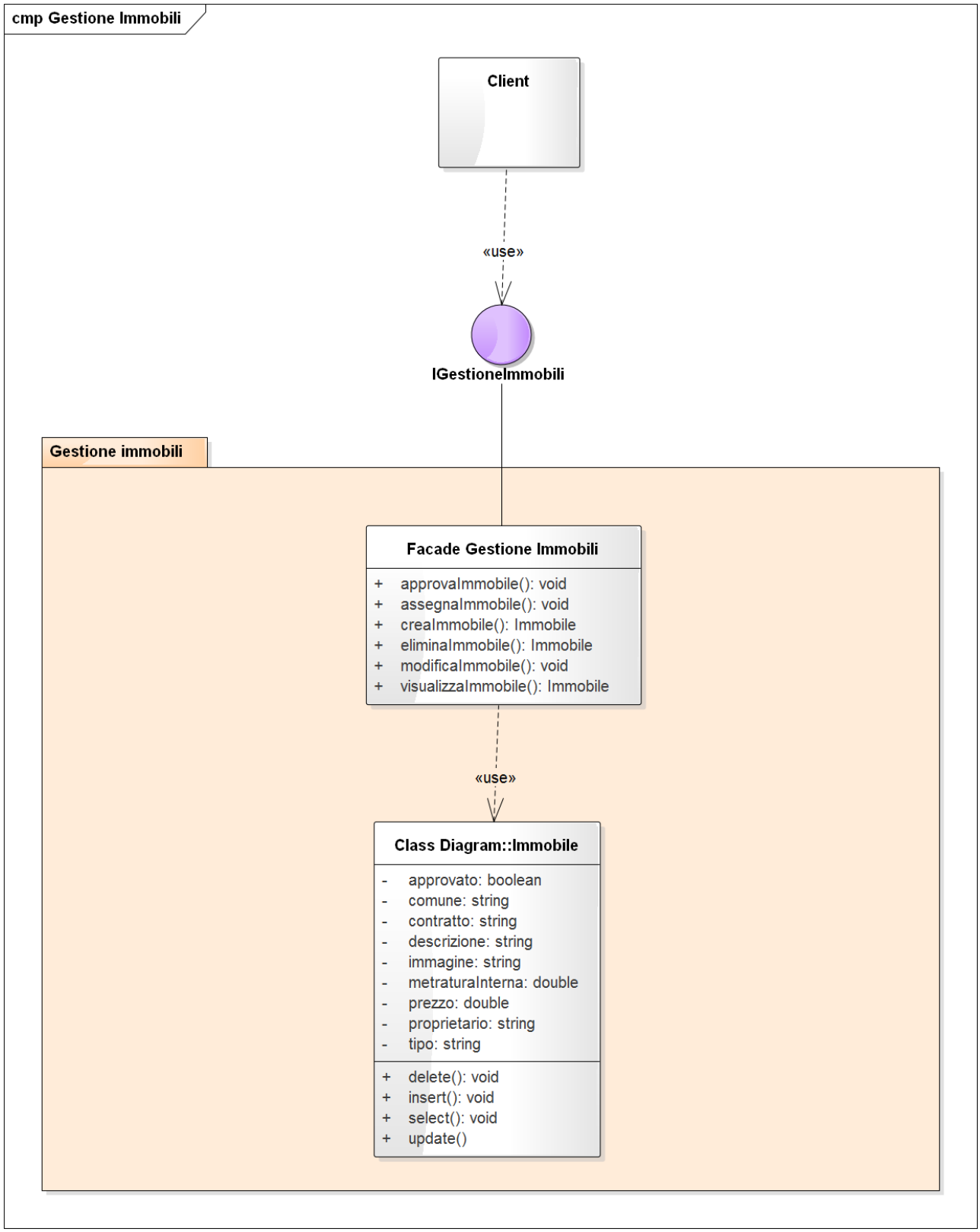
3.2.1 Sottosistema GestioneAccount



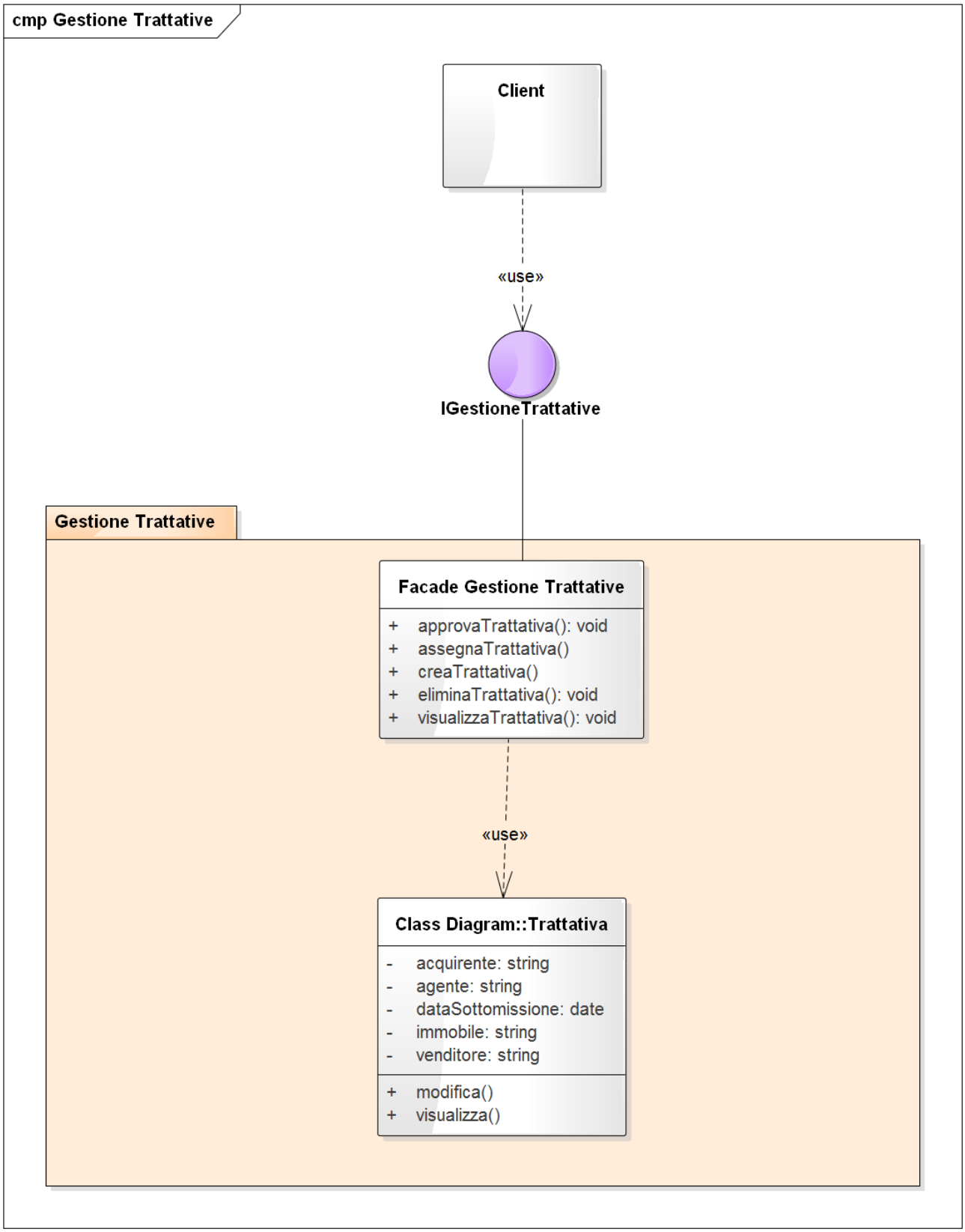
3.2.2 Sottosistema GestioneAutenticazione



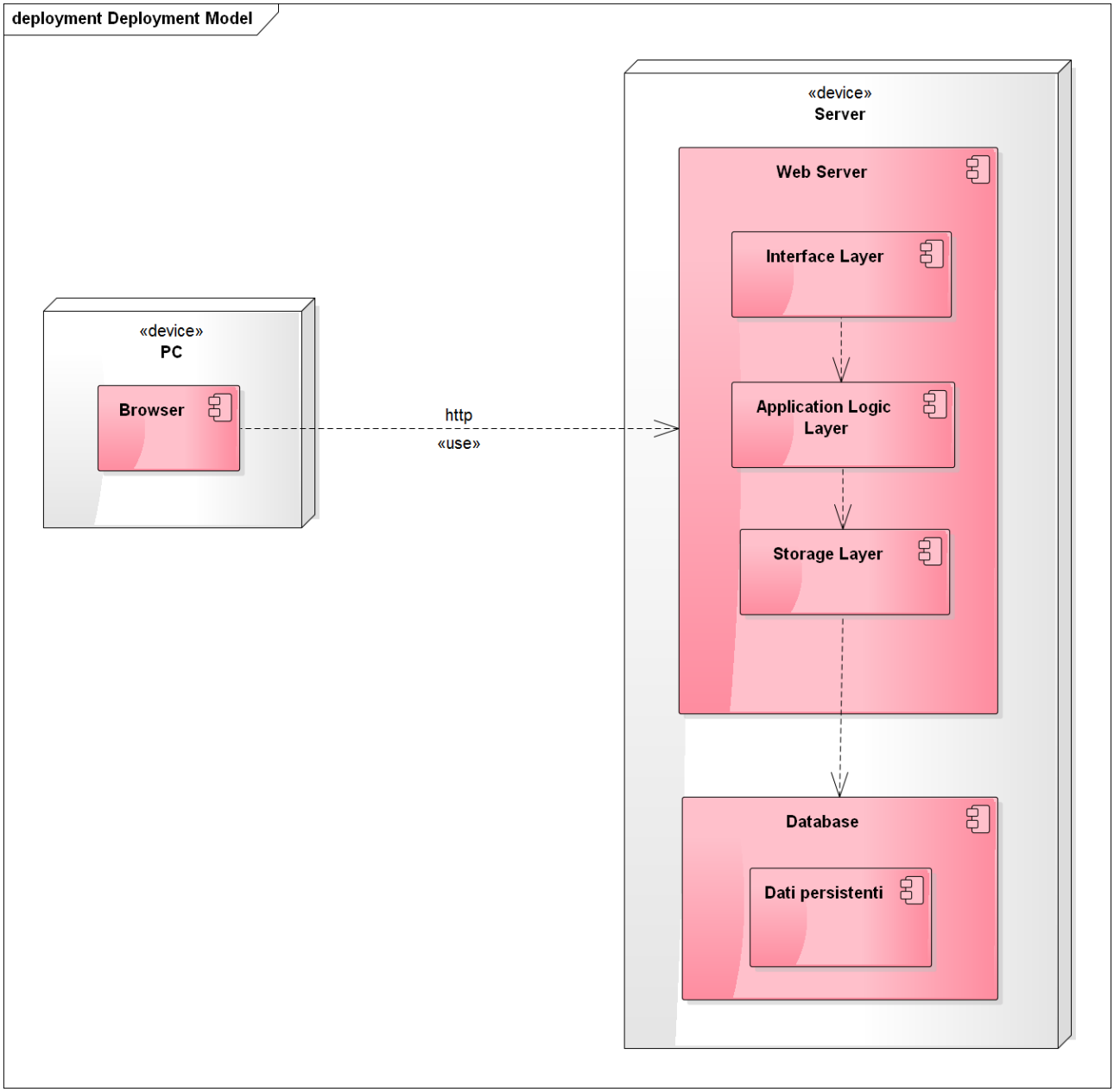
3.2.3 Sottosistema GestioneImmobili



3.2.4 Sottosistema GestioneTrattative



3.3 Hardware/software mapping



Web Server
Il web server utilizzato è Apache/2.4.10 (Unix), compatibile con i più diffusi sistemi operativi
(Windows, Linux e Mac). Nel nostro caso è stato utilizzato Windows 7.

Interface layer
Dipende dal web server, nel nostro caso usiamo il linguaggio HTML.

Application logic layer
Tutto il codice dinamico e le funzionalità sono state implementate in PHP, che non vengono direttamente interpretati dal webserver ma che quindi accorgendosi che si tratta di linguaggio PHP lo manda al “Modulo PHP” che lo interpreta e spedisce al webserver il codice HTML risultante, ed esso lo spedisce al browser del client.

Storage layer
Lo storage layer deve comunicare con il database rispondendo alle richieste dello strato superiore laddove necessitano del trattamento dei dati. Esso si interfaccia tramite le funzioni predefinite in linguaggio PHP.

Database Server
Il DBMS usato è MySQL, che si interfaccia con il web server Apache. Le funzioni MySQL fruibili dagli script PHP permettono tale interazione.

3.4 Persistent data management

Durante la progettazione ci si è accorti che l'utilizzo di dati salvati su disco fisso potevano portare a problemi di consistenza e velocità sia in caso di guasto sia in caso di molteplici operazioni in contemporanea in quanto il disco fisso ha tempi di accesso elevanti per un sistema multiutenza. Più precisamente, nell'eventualità che il sistema avrebbe smesso di funzionare tutti i dati immessi sarebbero andati persi. Si è quindi ripiegato verso l'utilizzo di un DBMS open source, nel nostro caso MYSQL che è di tipo relazionale ossia consente la conservazione dei dati in tabelle separate anziché in un'unica grande entità. MYSQL ci permette di accedere in modo molto veloce al dato richiesto permettendo un accesso simultaneo concorrente in quanto sfrutta un sistema multithread, dove un thread fisso accetta tutte le richieste di connessione ed un thread attivo opera sul database permettendo ad una sola operazione alla volta per tabella in modo da evitare collisione dei dati.
Il nostro database prevederà le seguenti informazioni archiviate per tabelle:

Cliente	
Attributo	**Vincolo di appartenenza**
codiceFiscale	Stringa di lunghezza esatta di 16 caratteri dove il formato è: CCCCCCNNCNNCNNNC Con C carattere ed N numero
dataNascita	Tipo Date
Email	Stringa di formato: stringa@stringa
Indirizzo	Stringa di lunghezza massima 50 caratteri
password	Stringa di lunghezza minima 8 caratteri e lunghezza massima 20 caratteri
username	Stringa di lunghezza minima 5 caratteri e lunghezza massima 20 caratteri
numTelefono	Stringa di lunghezza massima: 20 caratteri

Agente	
Attributo	**Vincolo di appartenenza**
codiceFiscale	Stringa di lunghezza esatta di 16 caratteri dove il formato è: CCCCCCNNCNNCNNNC Con C carattere ed N numero
dataNascita	Tipo Date

Email	Stringa di formato: stringa@stringa
Indirizzo	Stringa di lunghezza massima 50 caratteri
password	Stringa di lunghezza minima 8 caratteri e lunghezza massima 20 caratteri
username	Stringa di lunghezza minima 5 caratteri e lunghezza massima 20 caratteri
numTelefono	Stringa di lunghezza massima: 20 caratteri
numImmobilivenduti	Tipo Intero

Amministratore	
Attributo	**Vincolo di appartenenza**
codiceFiscale	Stringa di lunghezza esatta di 16 caratteri dove il formato è: CCCCCCNNCNNCNNNC Con C carattere ed N numero
dataNascita	Tipo Date
Email	Stringa di formato: stringa@stringa
Indirizzo	Stringa di lunghezza massima 50 caratteri
password	Stringa di lunghezza minima 8 caratteri e lunghezza massima 20 caratteri
username	Stringa di lunghezza minima 5 caratteri e lunghezza massima 20 caratteri

Trattativa	
Attributo	**Vincolo di appartenenza**
acquirente	Stringa di lunghezza minima 5 caratteri e lunghezza massima 20 caratteri
venditore	Stringa di lunghezza minima 5 caratteri e lunghezza massima 20 caratteri
Immobile	Tipo Intero
dataSottomissione	Tipo Date
Agente	Stringa di lunghezza minima 5 caratteri e lunghezza massima 20 caratteri

Immobile	
Attributo	**Vincolo di appartenenza**
approvato	Tipo Boolean
Contratto	Stringa di lunghezza massima 20 caratteri
descrizione	Stringa di lunghezza massima di 500 caratteri
immagine	Stringa di lunghezza minima 2 caratteri e lunghezza massima 200 caratteri
metraturaInterna	Stringa di lunghezza minima 5 caratteri e

	lunghezza massima 20 caratteri
Prezzo	Tipo Double
proprietario	Stringa di lunghezza minima 5 caratteri e lunghezza massima 20 caratteri
Tipo	Stringa di lunghezza massima 30 caratteri

3.5 Access Control and Security

Il sistema ImmobilUnisa permette l'accesso a più tipologie di utente: Amministratore, Cliente e Agente. Ogni tipo di utente accede a funzionalità differenti, risulta quindi necessario definire una politica di controllo degli accessi e di sicurezza. Il sistema presenta quindi un servizio di autenticazione che richiede l'inserimento di un username e di una password. In seguito all'autenticazione di un utente, il sistema reindirizza questo alla home page da cui potrà accedere solo alle funzioni permesse alla tipologia di utente a cui appartiene. Le seguenti matrici degli accessi di tipo CAPABILITY mostrano le operazioni che ogni tipo di attore può effettuare sulle classi del sistema:

Amministratore	
Agente	Crea Modifica Cancella
Immobile	Assegna Visualizza
Trattativa	Assegna Visualizza

Agente	
Cliente	Crea Modifica Cancella
Immobile	Visualizza Crea Approva Elimina
Trattativa	Approva Visualizza Elimina

Cliente	
Immobile	Crea Visualizza
Trattativa	Crea Visualizza

3.6 Global software control

Il controllo globale sul software è basato sul modello centralizzato “event-driven” dove il client/server comunicheranno tramite chiamata generata da un evento, è stato scelto questo modello in quanto dai diagrammi di flusso degli eventi si può notare la struttura lineare dell’intero sistema, dove genericamente le iterazioni sono un susseguirsi di eventi generati dal client, e dalle risposte del server a quegli stessi eventi. Più precisamente quando il client tramite web browser genera un evento, ci sarà un modulo php specifico per ogni determinato evento che sarà usato dal server per generare la risposta da reinviare al client.

3.7 Boundary Condition

3.7.1 Startup del Sistema

Allo startup del client, il sistema non dovrà caricare nessun dato dal database. Infatti verrà mostrata un’interfaccia di login in cui l’utente dovrà inserire i propri dati di accesso. In seguito alla sottomissione dei dati, il sistema verificherà la validità di questi, e nel caso in cui i dati sono corretti, l’utente verrà reindirizzato alla sua homepage.

3.7.2 Fallimento del Sistema

Nel caso si verifichi un crash nel sistema, si cercherà di effettuare il riavvio di esso o un ripristino ad una configurazione precedente. I crash del sistema non comportano il rischio di perdita dei dati, in quanto questi vengono gestiti dal DBMS. Tuttavia, per evitare che guasti al supporto di memorizzazione del database server comportino la perdita dei dati, verranno create periodicamente copie di backup del database.

In caso di crash del sistema possiamo distinguere tre casi:

- Crash del client: se il crash si verifica nel client, il sistema potrà comunque continuare a funzionare.
- Crash del server: il sistema risulta inutilizzabile in quanto non può più comunicare con il database.
- Crash del database server: il sistema risulta inutilizzabile in quanto non è più possibile accedere ai dati.

In ogni caso il guasto verrà segnalato in maniera opportuna.

3.7.3 Terminazione del Sistema

La terminazione avviene solo quando vengono terminati tutti i sottosistemi. Per non incorrere in problemi, prima di disattivare il server e il database server, è consigliabile disattivare prima tutti i client. Ogni sottosistema potrà essere disattivato tramite la funzione di logout presente in ogni area utente del sistema.

4.0 Subsystem Services

4.1Gestione Account

Descrizione:	Sottosistema che gestisce le operazioni relative alle tre tipologie di account ognuna con i rispettivi diritti.

Servizi	
Servizio	**Descrizione**
Crea Account	Permette di creare un nuovo account all'interno di un database.
Modifica Account	Permette di modificare i dati relativi ad un account nel database.
Elimina Account	Permette di eliminare un account dal database.
Visualizza Account	Permette di visualizzare i dati relativi ad un account.

4.2 Gestione Immobile

Descrizione:	Sottosistema che gestisce le operazioni relative agli immobili.

Servizi	
Servizio	**Descrizione**
Crea Immobile	Permette di creare un nuovo immobile all'interno di un database.
Modifica Immobile	Permette di modificare i dati relativi ad un immobile nel database.
Elimina Immobile	Permette di eliminare un immobile dal database.
Visualizza Immobile	Permette di visualizzare i dati relativi ad un immobile.
Assegna Immobile	Permette all'amministratore di assegnare un immobile ad un agente per permetterne l'approvazione.
Approva Immobile	Permette all'agente di approvare un immobile a lui assegnato.

4.3 Gestione Trattative

Descrizione:	Sottosistema che gestisce le operazioni

	relative alle trattative.

Servizi	
Servizio	**Descrizione**
Crea Trattativa	Permette di creare una nuova trattativa relativa ad un immobile.
Assegna Trattativa	Permette ad un amministratore di assegnare una trattativa ad un agente per permetterne l'approvazione.
Elimina Trattativa	Permette di eliminare una trattativa dal database.
Visualizza Trattativa	Permette di visualizzare i dati relativi ad una trattativa.
Approva Trattativa	Permette ad un agente di approvare una trattativa che gli è stata assegnata.

Glossario

ImmobilUnisa: Nome del prodotto software che verrà sviluppato.

GUI: Acronimo di “Graphics User Interface”, ovvero un interfaccia grafica per l’utente utilizzatore.

Amministratore: Il termine identifica la persona che amministra il sistema, cioè la persona che ha privilegi totali sul sistema e che può essere identificato come titolare dell’azienda.

Agente: Identifica la persona che si occupa di approvare le registrazioni dei clienti verificando la coerenza dei dati. Inoltre deve approvare l’inserimento dei dati relativi ad un immobile inserito da un cliente, e con quest’ultimo se è venditore, deve interloquire per la gestione degli appuntamenti per la vendita.

Cliente: Esso può essere Acquirente o venditore (oppure entrambi).

Cliente venditore: persona registrata al sistema che può inserire nuovi immobili e verificare le trattative attive sui propri immobili approvati.

Cliente acquirente: persona che può sfogliare una lista di immobili inseriti nel sistema, e se interessato, può decidere di acquistare tramite apertura di trattativa solo se registrato alsistema.

Trattativa: è un messaggio di conferma che indica che un cliente acquirente è deciso ad acquistare un immobile, la trattative è assegnata ad un solo Agente il quale deve portarla avanti e concluderla se possibile in modo da eliminarla non appena l’immobile è stato venduto.

Immobile: è un locale abitativo o commerciale che viene messo in vendita sul sistema da un cliente loggato. Esso può essere approvato se coerente con i dati che lo descrivono oppure non approvato e quindi eliminato dal sistema.

Login: Operazione di verifica di identità che viene effettuata online, cioè quando un utente immette le proprie credenziali generate durante la fase di registrazione.