

**Università degli Studi di Salerno**

**Corso di Ingegneria del Software**

**COMMERCIALISTA**

**System Design Document**

**VERSIONE 1.0**

**Data: 21/10/2016**

**Partecipanti:****Nome**

Mazzillo Pasquale

De Luca Danilo

**Matricola**

05121/02562

05121/02242

**Cronologia delle revisioni:**

<b>Data</b>	<b>Versione</b>	<b>Descrizione</b>	<b>Autore</b>
09/09/16	0.1	Stesura introduzione	Pasquale Mazzillo Danilo De Luca
11/09/16	0.2	Stesura decomposizione in sistemi	Pasquale Mazzillo Danilo De Luca
13/09/16	0.3	Stesura mapping hardware/software	Pasquale Mazzillo Danilo De Luca
15/09/16	0.4	Revisione decomposizione in sistemi	Pasquale Mazzillo Danilo De Luca
16/09/16	0.5	Stesura gestione dei dati persistenti	Pasquale Mazzillo Danilo De Luca
17/09/16	0.6	Stesura controllo degli accessi alla sicurezza	Pasquale Mazzillo Danilo De Luca
20/09/16	0.7	Revisione gestione dei dati persistenti	Pasquale Mazzillo Danilo De Luca
23/09/16	0.8	Revisione controllo degli accessi alla sicurezza	Pasquale Mazzillo Danilo De Luca
25/09/16	0.9	Stesura condizioni limite	Pasquale Mazzillo Danilo De Luca
29/09/16	1.0	Revisione finale	Pasquale Mazzillo Danilo De Luca

# INDICE

## **1. Introduzione**

1.1 Descrizione .....	pag. 4
1.2 Obiettivi principali .....	pag. 4
1.3 Overview .....	pag. 6

## **2. Architettura sistema proposto**

2.1 Decomposizione in sistemi .....	pag. 7
2.2 Mapping hardware/software .....	pag. 8
2.3 Gestione dei dati persistenti .....	pag. 16
2.4 Controllo degli accessi alla sicurezza .....	pag. 19
2.5 Le condizioni limite .....	pag. 20

# 1. INTRODUZIONE

Questo sito nasce dalla commissione da parte di uno studio di commercialisti i quali cercano un sito web annesso a un database per ottimizzare il proprio lavoro e tenere i dati degli eventuali clienti in maniera organizzata e reperire questi stessi velocemente ed efficacemente.

Questo sito deve rispondere alle richieste generiche di uno studio di commercialisti. I Clienti devono poter iscriversi al sito, il commercialista o l'amministratore, deve accedere ai rispettivi dati sensibili ed eventualmente cancellare un cliente, inoltre il sito web deve avere una pagina di descrizione, dove ci saranno informazioni sullo studio e sui relativi commercialisti, la posizione dello studio tramite Google Maps, i vari recapiti telefonici. In poche parole questo sistema informatico è uno strumento utile per lo studio dei commercialisti che ci hanno commissionato questo lavoro e per i loro clienti.

## 1.1 Obiettivi progettuali

### CRITERI DI AFFIDABILITA'

Il sistema è strutturato in modo da garantire il corretto svolgimento delle proprie funzionalità. In tal senso, è dotato di un sistema di notifica di errori logici, spesso dovuti a delle negligenze da parte degli utenti, che potranno manifestarsi durante l'utilizzo dell'applicazione. In caso, per esempio, di un errata compilazione di una form, oppure malfunzionamenti della rete, il sistema sarà in grado di segnalare gli errori e gestirne la risoluzione. A tal proposito, il sito rispetta i seguenti requisiti di qualità:

- **Robustezza:** Il sistema offre un buon grado di robustezza agli input errati forniti dagli utenti; non verranno alterati i dati nel database, ma verrà notificato l'errore all'utente.
- **Affidabilità:** Il sistema garantisce le stesse funzionalità su diverse piattaforme, compresi gli smartphone.
- **Disponibilità:** Il sistema deve essere sempre disponibile agli utenti, tranne in caso di eventuali periodi di manutenzione, che saranno opportunamente segnalati.
- **Sicurezza:** Il sistema prevede il controllo degli accessi, che permette ad ogni categoria di utente di eseguire il proprio lavoro senza intaccare o modificare quello di altri utenti. Inoltre, deve essere in grado di assicurare il corretto trattamento di dati personali e la messa in sicurezza dei dati da esaminare.

## **CRITERI DI PERFORMANCE**

Il sistema è scalabile e concorrente, così, in caso di accessi multipli, non viene rallentato. Il sistema dovrà garantire che le varie operazioni disponibili vengano svolte entro un intervallo di tempo accettabile. Il sistema quindi si propone di rispettare i seguenti requisiti di qualità relativi alle prestazioni:

- **Tempo di risposta:** Il sistema deve garantire tempi di risposta brevi o, altrimenti, notificare nel minor tempo possibile eventuali indisponibilità del sistema stesso.
- **Usabilità:** Le funzionalità del sistema devono essere rese disponibili all'utente attraverso un numero minimo di interazioni con l'interfaccia.
- **Throughput:** Il sistema deve massimizzare il numero di operazioni effettuate nel minor tempo possibile, garantendo un grado di affidabilità costante al crescere del numero di utenti connessi
- **Memoria:** Il sistema utilizza un database per memorizzare tutti i dati. Al crescere delle informazioni memorizzate dal sistema verranno utilizzate tecniche di replicazione dei dati (ridondanze) e normalizzazione per garantire migliori prestazioni.
- **Tolleranza:** Il sistema non deve essere soggetto a crash ripetuti.
- **Facilità di apprendimento:** Il sistema grazie ad una semplice ed intuitiva interfaccia, offrirà agli utenti uno strumento di apprendimento facile ed immediato per tutte le funzionalità.

## **CRITERIO DI MAUTENIBILITA'**

Il sistema deve garantire un alto grado di manutenibilità. Deve rispettare i seguenti requisiti di qualità:

- **Estensibilità:** il sistema è progettato in maniera tale da garantire l'aggiunta di nuove funzionalità in modo semplice, senza dover modificare altre funzionalità.
- **Modificabilità:** il sistema è progettato in modo tale da garantire la modifica di funzionalità già presenti, senza dover intaccare altre funzioni. Il grado di accoppiamento del sistema deve essere mantenuto il più basso possibile.
- **Tracciabilità dei requisiti:** con una buona documentazione del codice è possibile risalire ai corrispondenti requisiti funzionali, a cui fanno riferimento le varie classi e metodi del sistema.

## **CRITERI DEGLI UTENTI FINALI**

Dal punto di vista dell'utente il sistema garantisce i seguenti requisiti di qualità:

- **Utilità:** Dopo aver raccolto i requisiti funzionali, il sistema supporterà in maniera ottimale le esigenze dei vari utenti.

- **Apprendimento:** Il sistema è di facile ed immediata comprensione. I tempi di apprendimento del sistema sono quindi ridotti al minimo.
- **Usabilità:** Il sistema risulta intuitivo, snello e leggero, e, dopo poche interazioni, è in grado di generare nell'utente un grado di esperienza tale da portarlo a compiere le operazioni desiderate in tempi ridotti.

## TRADE-OFF

- **Tempo di rilascio vs Qualità:** Il team ha preferito progettare un prodotto software di qualità tentando di rispettare quanto più possibile le scadenze, che sono un aspetto fondamentale del progetto.
- **Sicurezza vs Efficienza:** La gestione della sicurezza è prevista nell'autenticazione dell'utente. Questa politica di permessi, permette di non appesantire eccessivamente il software ed è un buon compromesso tra sicurezza ed efficienza.
- **Comprensibilità vs Tempo:** Il codice deve essere comprensibile e commentato, così, da poter essere interpretato da altri programmatori che non hanno partecipato al progetto e da chi si occuperà del testing. Questo accrescerà il tempo necessario per completare l'implementazione, ma in una misura accettabile.
- **Design vs usabilità:** L'interfaccia è composta da diverse pagine, pulsanti e varie etichette, associate alle funzioni a cui fanno riferimento, che fanno intendere la loro utilità, rendendo comprensibile ed intuitivo il sistema.

## 1.2 Overview

Il sistema viene suddiviso in sottosistemi secondo due livelli di astrazione:

1) macro-decomposizione, mediante la quale è possibile illustrare le interazioni tra i vari sottosistemi

2) la micro-decomposizione, che mostra, per ogni sottosistema, le interazioni tra le sue componenti.

Dato che parliamo di un sito che avrà bisogno di accedere a grandi quantità di dati il sistema sarà di natura distribuita.

In particolare, verrà utilizzata un architettura three-tier, evoluzione della tipica architettura client-server, dove è inserito un nuovo livello intermedio (middleware) per la gestione del trasferimento dei dati dal livello client al livello archivio dati.

Questo, quindi, rende indipendente il livello client dalle problematiche di accesso al livello archivio dati.

In questo tipo di architettura vengono coinvolti tre tier:

- il livello di presentazione, lato client
- il livello intermedio della logica di business, lato server
- il livello archivio dati

Il presentation layer viene rappresentato dal sito web, quindi si troverà sulla device, e ha il compito di acquisire dati e visualizzare i risultati.

Il logical layer si occupa di inoltrare le richieste dell'utente al livello archivio dati e di trasmettere i relativi risultati, implementando, di fatto, la logica client-server.

Il livello archivio dati rappresenta l'insieme dei dati presenti nel DB, tale livello sarà costituito dal DBMS MySql.

## 2. Architettura del sistema proposto

### 2.1 Decomposizione in sottosistemi

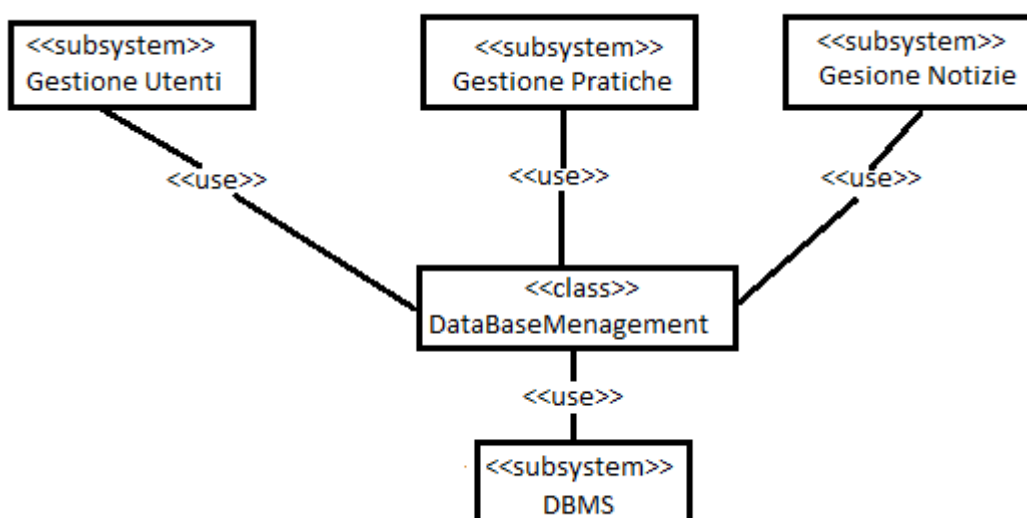
#### Macro-decomposizione:

La macro-decomposizione del sistema prevede la divisione dello stesso in un sottosistemi:

- Gestione utenti: si occuperà della gestione delle informazioni riguardanti gli utenti registrati al sito.
- Gestione notizie: si occuperà della gestione degli avvisi e delle newsletter.
- Gestione pratiche: si occuperà della gestione delle pratiche.

Inoltre verrà creata una classe per schermare l'accesso al DB e alla quale il sottosistema può accedere per comunicare col terzo tier.

Di seguito lo schema:



### Micro-decomposizione:

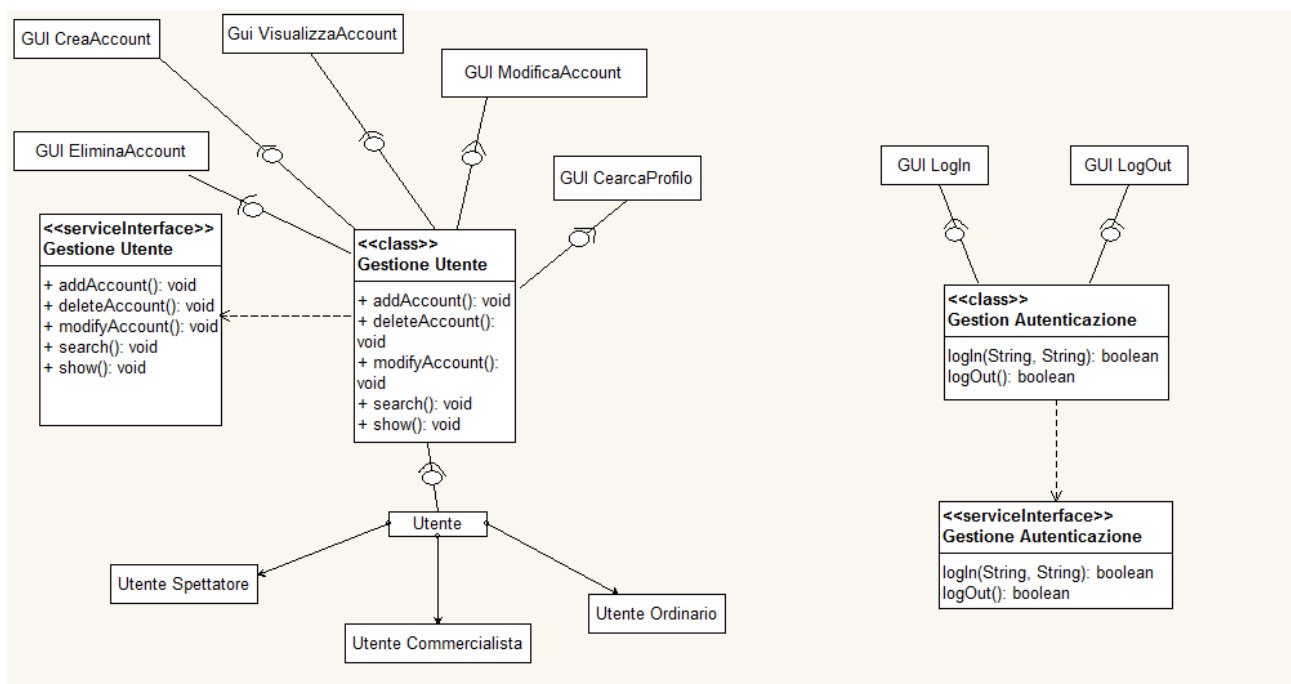
Seguendo i principi della buona programmazione ad oggetti, all'interno di ogni sottosistema vi saranno degli oggetti di controllo che si occuperanno della comunicazione tra i diversi tier.

Questi acquisiranno i dati dall'Interface Application Layer per elaborarli nel Logical Application Layer e richiedere le informazioni necessarie allo Storage Layer. Inoltre, si occuperanno anche della comunicazione in senso opposto, elaborando le informazioni provenienti dall'ultimo tier per fornirle al tier di presentazione.

### Gestione Utente:

È il sottosistema addetto alla gestione degli account, dell'autenticazione e dei profili personali degli utenti. Fornisce all'utente le funzioni per la creazione, ricerca, visualizzazione e modifica degli account. Gestisce le operazioni di login e log-out degli utenti, verificandone l'identità e le credenziali di accesso a tutte le funzionalità del sistema.

Permette di ricercare e visualizzare i profili personali degli utenti.





<b>GUI</b>	<b>Descrizione</b>
LogIn	Tramite questa interfaccia un utente può autenticarsi al sistema.
LogOut	Tramite questa interfaccia un utente può scollegarsi dal sistema.
AggiungiAccount	Questa interfaccia permette al gestoreUtente di aggiungere un account.
EliminaAccount	Questa interfaccia permette al gestoreUtente di eliminare un account.
VisualizzaProfilo	Tramite questa interfaccia è possibile visualizzare il profilo di un utente.
ModificaProfilo	Tramite questa interfaccia è possibile modificare il profilo di un utente.
CercaProfilo	Tramite quest'interfaccia è possibile ricercare un utente.

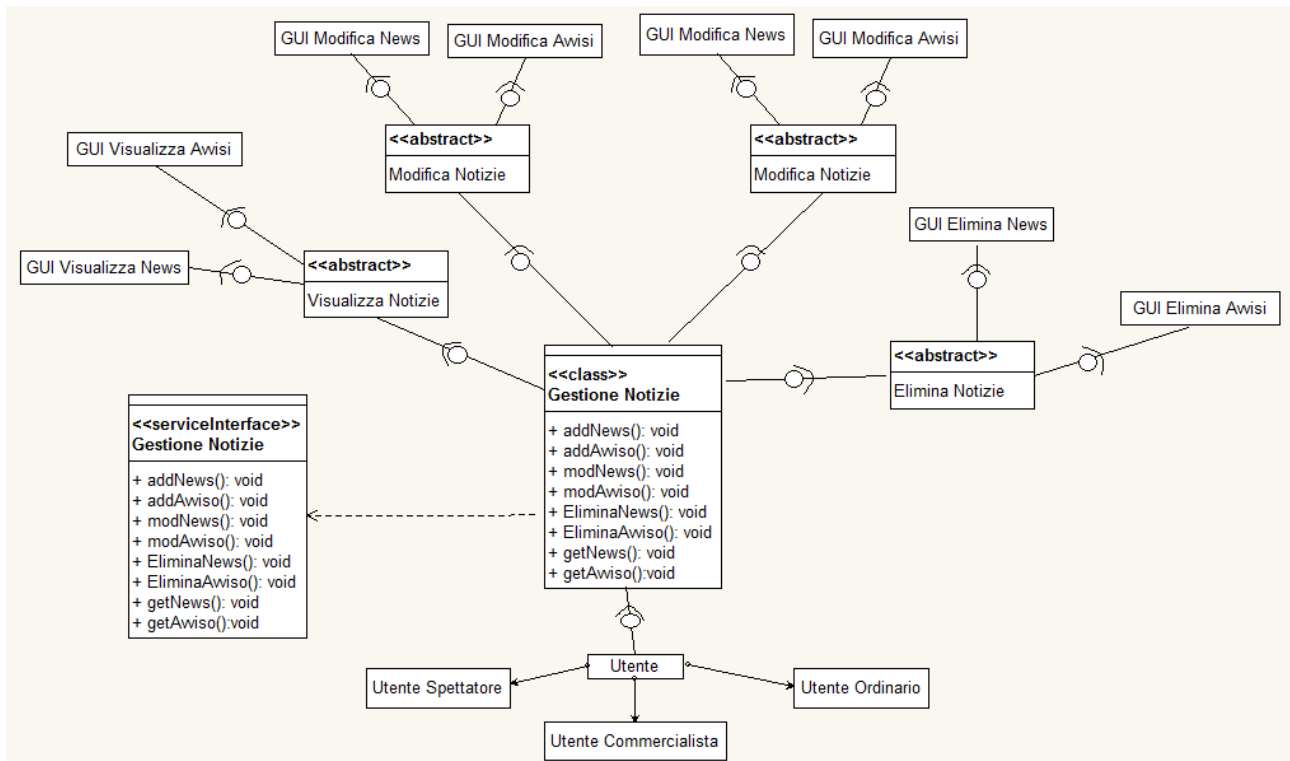
<b>Gestore Utente (gestore)</b>	<b>Descrizione</b>
addAccount()	Si occupa di aggiungere un nuovo account utente al sistema.
deleteAccount()	Si occupa di eliminare un account dal sistema.
modifyAccount()	Si occupa di modificare le informazioni riguardanti un account utente.
search()	Si occupa della ricerca di un account utente.
show()	Si occupa della visualizzazione di un account utente.

<b>Gestore Autenticazione (gestore)</b>	<b>Descrizione</b>
login()	Si occupa della connessione al sistema.
logout()	Si occupa della disconnessione al sistema.

<b>Classi</b>	<b>Descrizione</b>
Utente	Utente generico.
Utente spettatore	Rappresenta un utente che non è registrato alla piattaforma.
Utente ordinario	Può visualizzare pratiche e notizie aggiunte dall'utente commercialista.
Utente Commercialista	Gestisce le pratiche e le notizie e le rende visibili agli utenti ordinari.

## Gestione Notizie:

È il sottosistema addetto alla gestione delle notizie, composte da avvisi e newsletter. Fornisce all'utente le funzioni per la creazione, cancellazione, visualizzazione e modifica delle notizie.



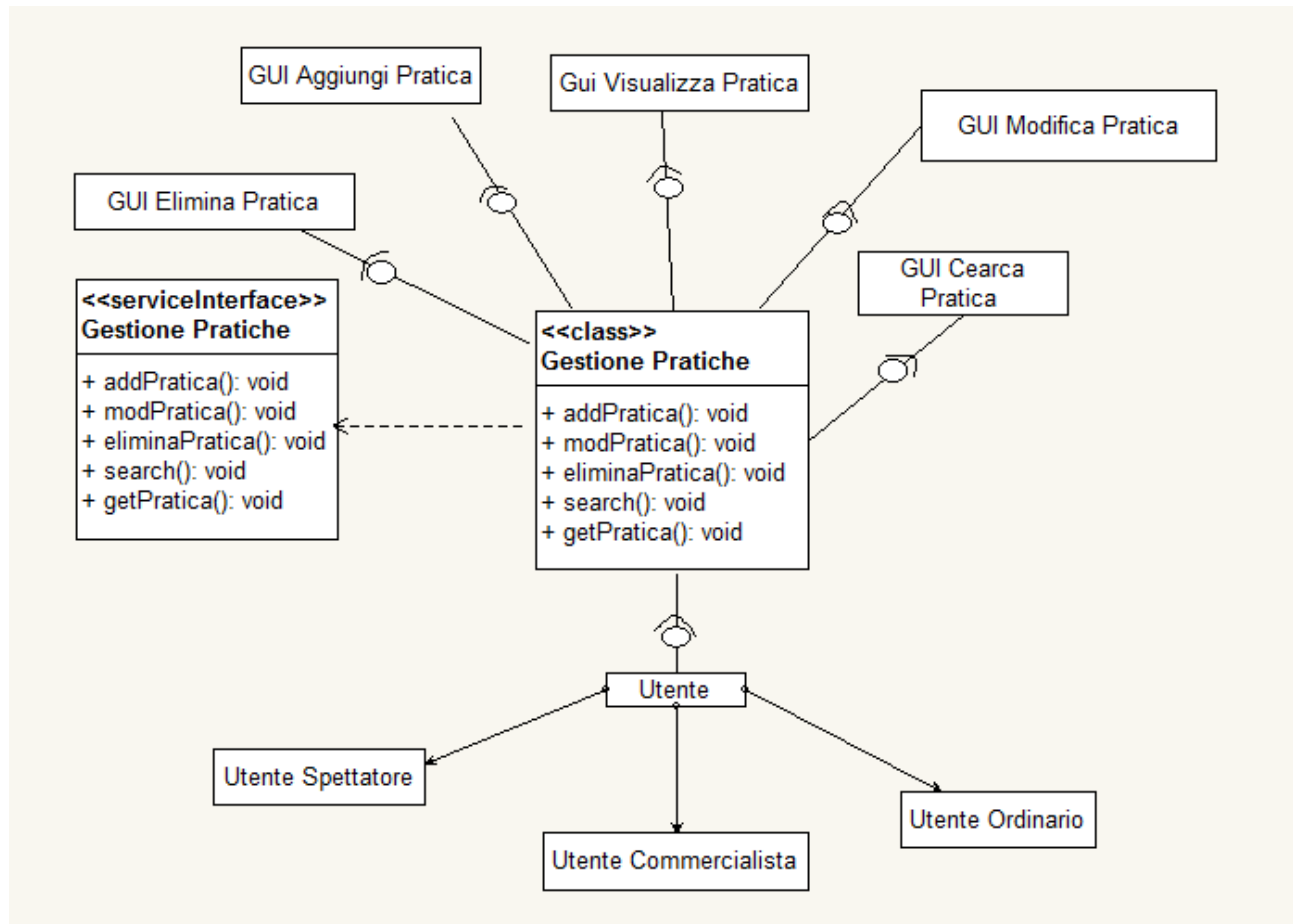
GUI	Descrizione
Visualizza News	Questa interfaccia permette di visualizzare una newsletter.
Visualizza Avvisi	Questa interfaccia permette di visualizzare un avviso.
Modifica News	Questa interfaccia permette di modificare una newsletter.
Modifica Avvisi	Questa interfaccia permette di modificare un avviso.
Elimina News	Questa interfaccia permette di eliminare una newsletter.
Elimina Avvisi	Questa interfaccia permette di eliminare un avviso.

Gestore Utente (gestore)	Descrizione
addNews()	Si occupa di aggiungere una nuova newsletter al sistema.
addAvviso()	Si occupa di aggiungere un nuovo avviso al sistema.
modNews()	Si occupa di modificare una newsletter.
modAvviso()	Si occupa di modificare un avviso.
eliminaNews()	Si occupa di eliminare una newsletter.
eliminaAvviso()	Si occupa di eliminare un avviso.
getNews()	Si occupa di visualizzare una newsletter.
getAvviso()	Si occupa di visualizzare un avviso.

Classi	Descrizione
Utente Generico	Composto da utente spettatore e utente iscritto al sito
Utente Commercialista	Utente che gestisce avvisi e newsletter
Avvisi	Rappresenta tutti gli avvisi
Newsletter	Rappresenta tutte le newsletter

## Gestione Pratiche:

È il sottosistema addetto alla gestione delle pratiche degli utenti ordinari. Fornisce all'utente le funzioni per la creazione, cancellazione, ricerca, visualizzazione e modifica delle pratiche.



GUI	Descrizione
Aggiungi Pratica	Questa interfaccia permette di aggiungere una pratica.
Elimina Pratica	Questa interfaccia permette di eliminare una pratica.
Visualizza Pratica	Questa interfaccia permette di visualizzare una pratica.
Modifica Pratica	Questa interfaccia permette di modificare una pratica.
Cerca Pratica	Tramite quest'interfaccia è possibile ricercare una

	pratica.
--	----------

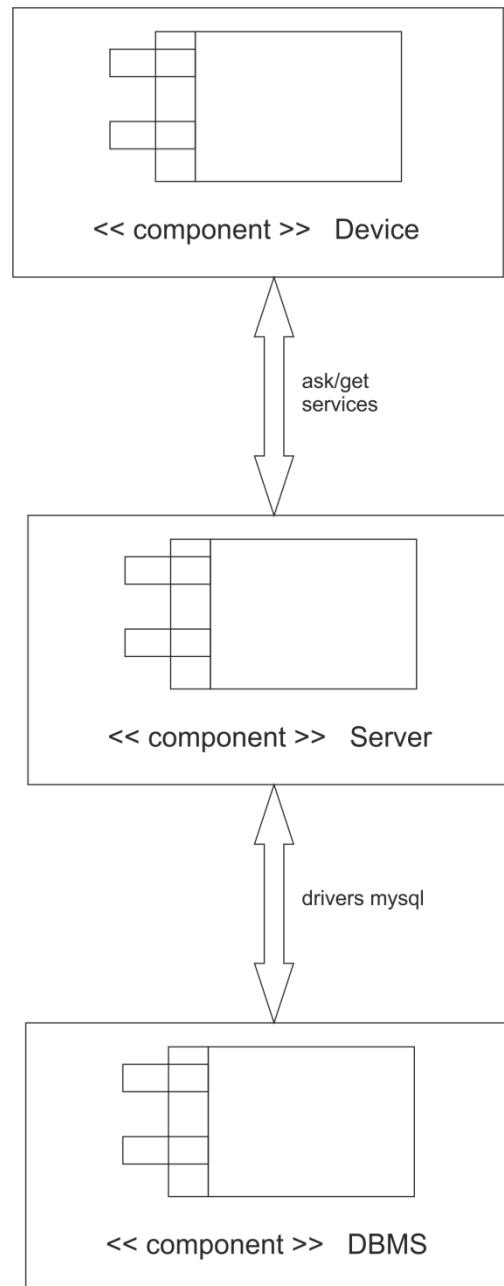
Gestore Utente (gestore)	Descrizione
addPratica()	Si occupa di aggiungere una nuova pratica.
eliminaPratica()	Si occupa di eliminare una pratica.
modPratica()	Si occupa di modificare una pratica.
search()	Si occupa della ricerca di una pratica.
getPratica()	Si occupa della visualizzazione di una pratica.

Classi	Descrizione
Utente Generico	Composto da utente spettatore e utente iscritto al sito
Utente Commercialista	Utente che gestisce avvisi e newsletter
Pratiche	Rappresenta il lavoro del commercialista, ogni pratica è assegnata ad un utente

## 2.2 Mapping Hardware/Software

Il sistema è distribuito su due nodi: client/server

Tali nodi interagiranno tra loro seguendo lo schema sottostante:



**Client:**

- Configurazioni Hardware:

Per la configurazione Hardware del Client è necessaria la presenza di un PC fisso/portatile o dispositivo android e di una connessione ad internet.

- Configurazioni Software:

I nodi Client permettono la comunicazione con il nodo Server tramite collegamenti remoti.

Il protocollo HTTP sarà utilizzato per collegare ogni nodo Client al Server.

### **Server:**

- Configurazioni Hardware

La configurazione Hardware del server comprende:

-Una connessione ad internet.

-Hard Disk capace di immagazzinare grandi quantità di dati.

- Configurazioni Software

Il server necessita inoltre di un software in grado di gestire le connessioni con più client e di una tecnologia che gestisca i dati persistenti. Per questo ci avvaliamo di tecnologie già esistenti, quali Apache2.4, che si occupa della gestione del WebServer (utilizzato per la simulazione in locale) , e MYSQL, un DMBS che gestisce i nostri dati persistenti.

Le configurazioni software dei Server sono:

-Un DMBS (MYSQL con la relativa Base Dati del sistema.

-Un server Web (Apache2.4).

## **2.3 Gestione dei dati persistenti:**

Gestire I dati persistenti è un problema che nasce dalla necessità di rendere permanenti alcune informazioni all'interno del sistema anche quando è inattivo. Il problema è molto più evidente con programmi che gestiscono un grande numero di informazioni che non possono essere reinserite dopo l'arresto del sistema o dopo un blackout. Quindi bisogna trovare il modo per tener traccia dei dati utili anche quando l'applicazione che li ha utilizzati e/o creati ha smesso di funzionare. I dati persistenti vengono cancellati solo se lo si richiede espressamente.

Si è scelto di usare il DBMS open source MySQL dato che permette la creazione di “database relazionali” ossia consente

la conservazione dei dati in tabelle separate anziché in un'unica grande entità-

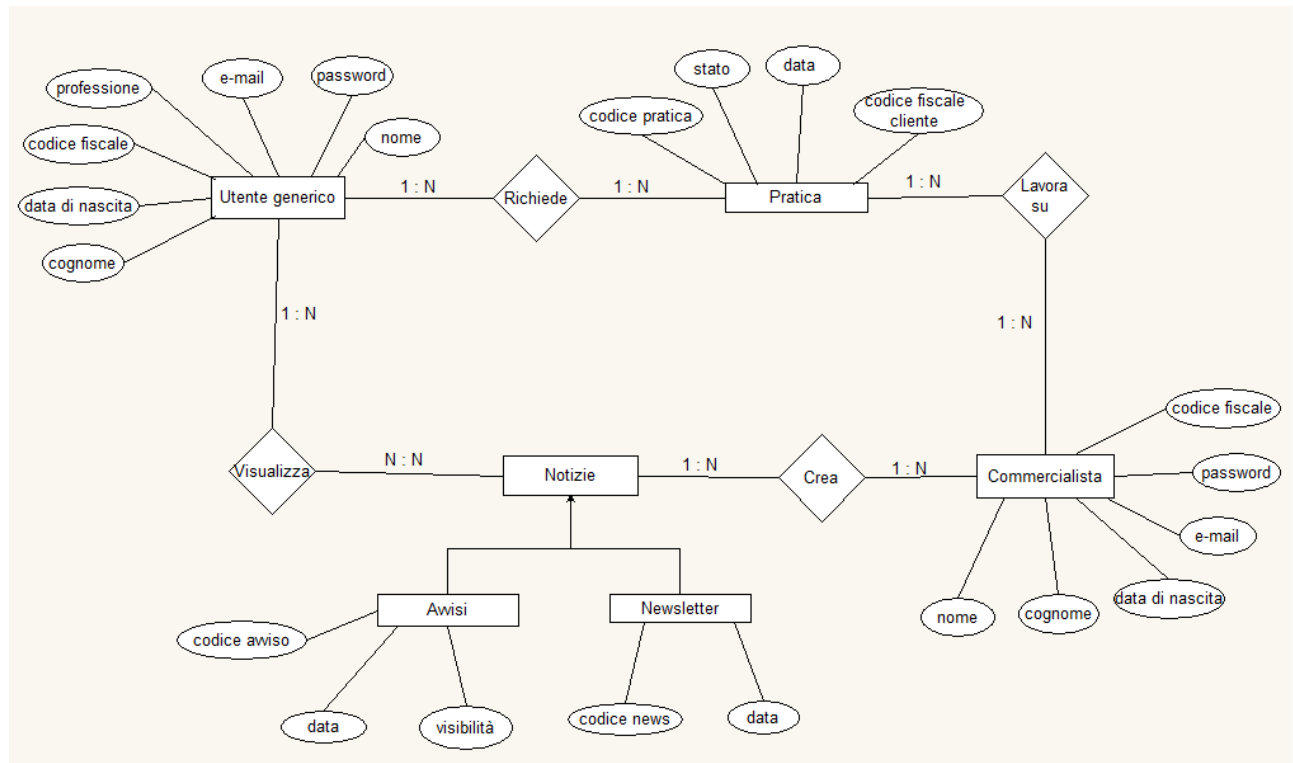
Grazie a questa tecnica si ha una buona flessibilità e velocità di accesso ai dati ed una maggiore modellazione delle basi di dati. MySQL presenta una struttura multi-thread comprendente un thread fisso che controlla la connessione in



ingresso e un thread attivo per ogni connessione cioè fa in modo da impedire che due thread scrivino sulla stessa tabella nello stesso momento evitando così perdita di informazioni.

All'interno del nostro database relazionale verranno custodite un gran numero di informazioni:

## DIAGRAMMA ER



Entità	Descrizione	Attributi	Chiave
Utente generico	Contiene tutti i dati relativi agli utenti ordinari.	nome cognome data_di_nascita professione email password codice_fiscale	codice_fiscale
Avvisi	Contiene tutti i dati relativi agli avvisi	codice_avviso data visibilit�	codice_avviso
Newsletter	Contiene tutti i dati relativi alle newsletter	codice_news data	codice_news
Pratica	Contiene tutti i dati	codice_pratica	codice_pratica

	relativi alle pratiche	stato data codice_cliente	
Commercialista	Contiene tutti i dati relativi all'utente commercialista	nome cognome data_di_nascita email password codice_fiscale	codice_fiscale

## UTENTE

Nome	Tipo	Null	Key	Descrizione
codice_fiscale	varchar(20)	No	Primary	Identificativo dell'utente
nome	varchar(20)	No		Nome dell'utente
cognome	varchar(20)	No		Cognome dell'utente
data_di_nascita	varchar(20)	No		Data di nascita dell'utente
professione	varchar(20)	No		Professione dell'utente
email	varchar(20)	No		Email per accedere al sito
password	varchar(20)	No		Password per accedere al sito

## Avvisi

Nome	Tipo	Null	Key	Descrizione
codice_avviso	varchar(20)	No	Primary	Identificativo dell'avviso
data	varchar(20)	No		Nome dell'utente
visibilita	varchar(20)	No		Rende visibile l'avviso a tutti gli utenti oppure solamente agli iscritti al sito

## Newsletter

Nome	Tipo	Null	Key	Descrizione
codice_news	varchar(20)	No	Primary	Identificativo della newsletter
data	varchar(20)	No		Data della newsletter

## Pratica

Nome	Tipo	Null	Key	Descrizione
codice_pratica	varchar(20)	No	Primary	Identificativo della pratica
stato	varchar(20)	No		Descrive se la pratica è conclusa o è in corso
data	varchar(20)	No		Data della pratica
codice_cliente	varchar(20)	No		Codice fiscale del cliente proprietario della pratica

## Commercialista

Nome	Tipo	Null	Key	Descrizione
codice_fiscale	varchar(20)	No	Primary	Identificativo del commercialista
nome	varchar(20)	No		Nome del commercialista
cognome	varchar(20)	No		Cognome del commercialista
data_di_nascita	varchar(20)	No		Data di nascita del commercialista
email	varchar(20)	No		Email per accedere al sito
password	varchar(20)	No		Password per accedere al sito

## 2.4 Controllo degli accessi e sicurezza

Per tutelare gli utenti e per fornire a ciascuno di essi esattamente le funzioni di cui ha bisogno, sono state previste all'interno del sistema politiche di controllo degli accessi. La verifica dell'identità dell'utente e l'individuazione dei componenti di cui avrà bisogno

vengono effettuate al login. Il sistema identifica il tipo di account associato all'utente e rende disponibili tutte e sole le funzioni a cui può accedere. Le funzioni di autenticazioni sono rese disponibili a tutti gli utenti dotati di account.

Tutti gli utenti sprovvisti di account possono solamente visualizzare notifiche, avvisi, newsletters presenti sul sito.

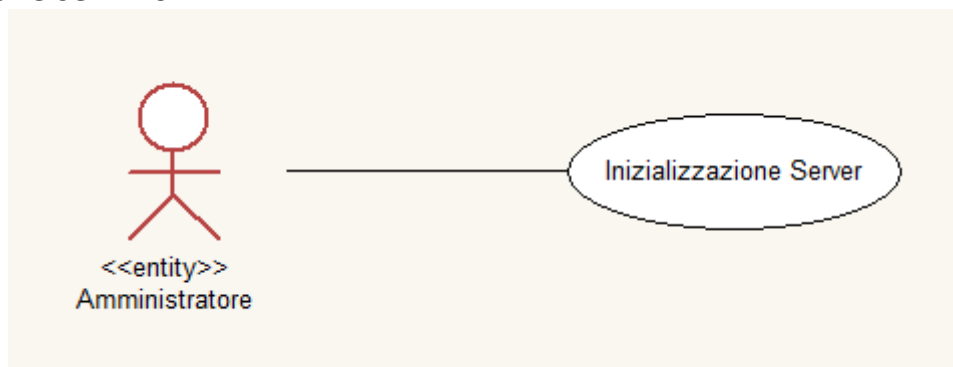
Un utente registrato ha a disposizione le operazioni di gestione del proprio account, come la modifica dei dati personali, visualizzare le pratiche effettuate e quelle in

corso da parte del commercialista, può visualizzare ogni notifica avviso newsLetters pubblicata dal commercialista.

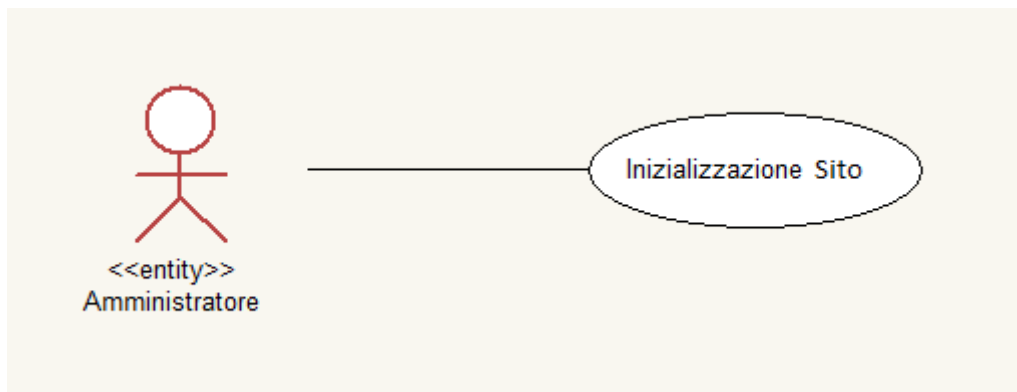
Le politiche di accesso vengono illustrate nelle seguenti tabelle

## 2.5 Le condizioni limite

Inizializzazione servizio

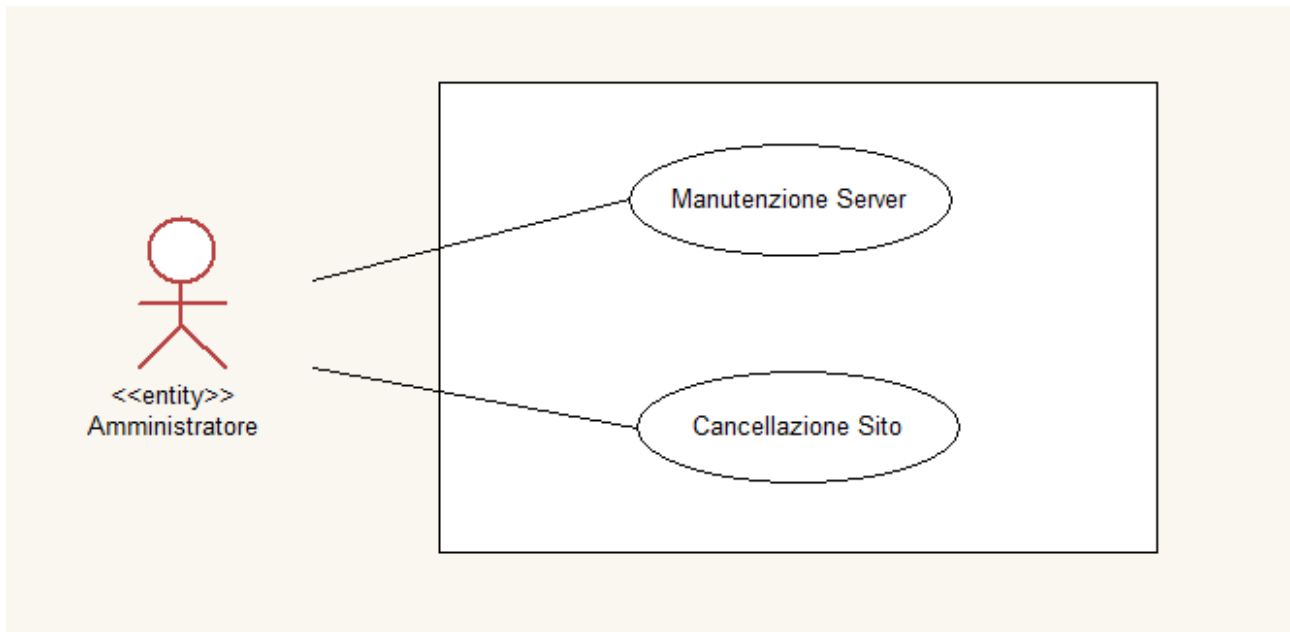


Casi d'uso	Inizializzazione server	
Attori	amministratore	
Condizioni di entrata	Possesso infrastruttura server.	
Flusso degli eventi	Attore	Sistema
	1. L'amministratore inizializza il server	
	2. L'amministratore installa il software necessario sul server	
	3. L'amministratore imposta il software per il corretto funzionamento	
Condizioni di uscita	Il sistema de essere funzionante e raggiungibile 24h/24h	



Casi d'uso	Inizializzazione sito	
Attori	Utente spettatore, utente ordinario, utente commercialista	
Condizioni di entrata	L'utente deve essere connesso ad internet.	
Flusso degli eventi	Attore	Sistema
	1. L'utente digita l'indirizzo del sito	
		2. Il sistema mostra la schermata principale
	3. L'utente si registra alla piattaforma	
		4. Il sistema verifica la correttezza dei dati
	5. L'utente accede al sito con le sue credenziali	
Condizioni di uscita	Il sistema risulta funzionante	

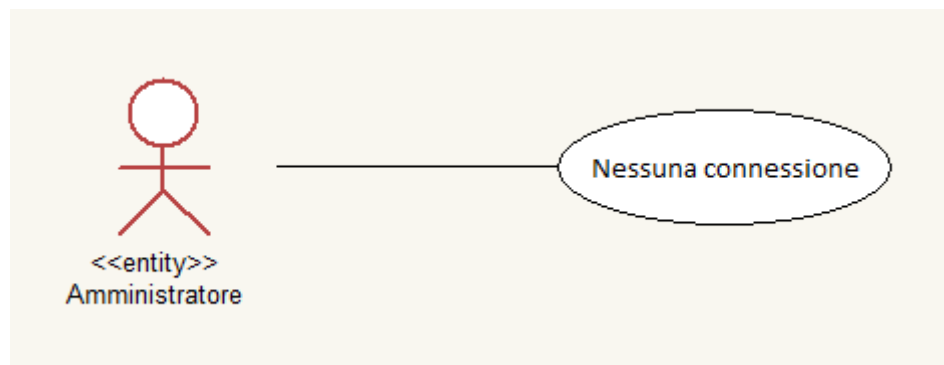
## Termine servizi



Casi d'uso	Manutenzione server	
Attori	amministratore	
Condizioni di entrata	Il server ha bisogno di manutenzione	
Flusso degli eventi	Attore	Sistema
	1. L'amministratore blocca tutti (o parte) dei servizi del server.	
	2. L'amministratore effettua la manutenzione.	
	3. L'amministratore riattiva tutti i servizi sospesi.	
Condizioni di uscita	Il server riprende il suo normale funzionamento	

Casi d'uso	Cancellazione sito	
Attori	amministratore	
Condizioni di entrata	Non sussistono più motivazioni per mantenere il sito attivo	
Flusso degli eventi	Attore	Sistema
	1. L'amministratore elimina il sito	
	2. L'amministratore dismette il server	
Condizioni di uscita	Il sito web non esiste più	

## Errori



Casi d'uso	Nessuna connessione a internet	
Attori	Utente spettatore, utente ordinario, utente commercialista	
Condizioni di entrata	L'utente ha una connessione a internet attiva	
Flusso degli eventi	Attore	Sistema
		1. Il sistema rileva la mancanza e/o l'impossibilità di connettersi al server e/o altri servizi.
		2. Il sistema notifica questa perdita di connessione.
Condizioni di uscita	Il sistema riprende il suo normale funzionamento	