
Software Requirements Specification

for

MusicMatch – Sistema di Gestione delle Prenotazioni per Lezioni di Musica

Prepared by Vassallo Pasquale (A13002577)

Esame di Sistemi Web e Basi Dati

Table of Contents

Table of Contents	0
Assigned Task.....	1
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Product Scope.....	1
2. Overall Description.....	2
2.1 Product Perspective.....	2
2.2 Product Functions.....	2
2.3 User Classes and Characteristics.....	3
2.4 Operating Environment.....	3
2.5 Design and Implementation Constraints	4
2.6 User Documentation.....	4
2.7 Assumptions and Dependencies.....	4
3. External Interface Requirements	4
3.1 User Interfaces	4
3.2 Software Interfaces.....	5
3.3 Communications Interfaces.....	8
4. System Features	9
4.1 System Feature Registrati	10
4.2 System Feature Autenticazione.....	11
4.3 System Feature Aggiungi Lezione	11
4.4 System Feature Modifica Lezione	12
4.5 System Feature Rimuovi Lezione	13
4.6 System Feature Prenota Lezione.....	14
4.7 System Feature Cancella Prenotazione	14
4.8 System Feature Disdici Prenotazione.....	15
4.9 System Feature Visualizza Lezioni Prenotabili	16
4.10 System Feature Visualizza Lezioni Prenotate.....	16
4.11 System Feature Gestione Recensioni	17
4.12 System Feature Visualizza Lista Utenti	18
4.13 System Feature Cancella Utente	18
4.14 System Feature Aggiungi Disciplina	19
5. Other Nonfunctional Requirements.....	20
5.1 Performance Requirements	20
5.2 Security Requirements	20
5.3 Software Quality Attributes	20
5.4 Business Rules	20
6. Database – Organization Idea.....	20
6.1 Modello concettuale: Diagramma ER.....	20
7. Ristrutturazione Logica.....	22
7.1 Tabella dei Volumi.....	22
7.2 Ristrutturazione dello schema concettuale.....	23
7.3 Traduzione dello schema concettuale in schema logico	25
7.4 Glossario dei termini	26
7.5 Vincoli di integrità	27
8. Database implementation.....	28
9. Test-cases	31

Assigned Task

Sviluppare una web app per prenotare lezioni di musica, con gestione degli insegnanti e degli studenti. Il Database deve essere di supporto per gestire lezioni, prenotazioni, insegnanti e studenti.

1. Introduction

1.1 Purpose

Questo documento fornisce una panoramica sul processo di creazione di una web application sviluppata per una scuola di musica (oppure indistintamente, un gruppo di insegnanti indipendenti registrati sulla piattaforma). L'applicazione consente di gestire le lezioni di musica, permettendo la prenotazione delle stesse e facilitando l'amministrazione degli insegnanti e degli studenti.

1.2 Intended Audience and Reading Suggestions

Questo documento, assieme al **Manale di Installazione** allegato, è pensato per essere una guida comprensibile riguardo i requisiti software (funzionali e non), gli obiettivi della web application e le modalità di utilizzo del sistema. È destinato principalmente ai progettisti, sviluppatori e amministratori coinvolti nella gestione del progetto, nello sviluppo del codice e nella manutenzione del sistema.

1.3 Product Scope

La web application è progettata per semplificare la gestione delle prenotazioni delle lezioni di musica e l'amministrazione delle informazioni relative a insegnanti e studenti. L'applicazione offre diverse sezioni, ciascuna dedicata a funzioni specifiche come la registrazione e gestione degli insegnanti e degli studenti, la pianificazione delle lezioni, la gestione delle prenotazioni e delle recensioni.

2. Overall Description

2.1 Product Perspective

Questa web application è stata concepita come una soluzione efficiente per mettere in contatto insegnanti di musica e studenti, facilitando la gestione delle prenotazioni delle lezioni, oltre alla gestione dei profili di insegnanti e studenti. Il suo scopo principale è assistere studenti e insegnanti nelle attività di prenotazione, garantendo un'interfaccia utente intuitiva e facile da navigare.

2.2 Product Functions

- **Gestione Insegnanti**

- Visualizzare l'elenco degli insegnanti registrati
- Aggiungere un nuovo insegnante
- Modificare i dettagli di un insegnante
- Rimuovere un insegnante
- Gestire la disponibilità delle lezioni di un insegnante
- Rispondere ad una recensione ricevuta

- **Gestione Studenti**

- Visualizzare l'elenco degli studenti iscritti
- Aggiungere un nuovo studente
- Modificare i dettagli di uno studente
- Rimuovere uno studente
- Visualizzare lo storico delle lezioni prenotate da uno studente e filtrarlo opportunamente
- Scrivere una recensione per una lezione seguita

- **Gestione Lezioni**

- Visualizzare l'elenco delle lezioni disponibili e di quelle già effettuate e filtrarlo opportunamente
- Creare una nuova lezione
- Modificare i dettagli di una lezione esistente
- Rimuovere una lezione

- **Gestione Prenotazioni**

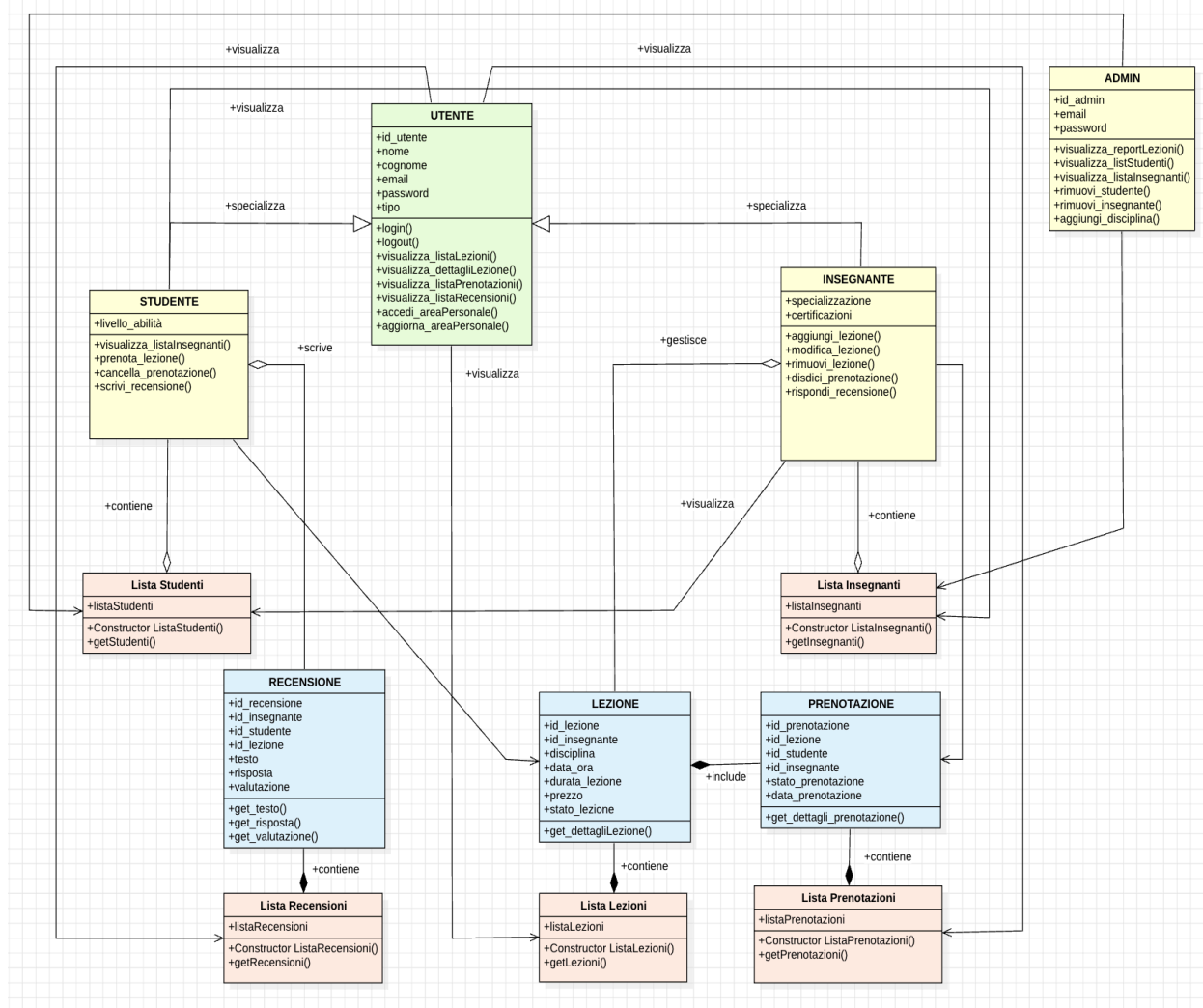
- Visualizzare tutte le prenotazioni effettuate
- Prenotare una lezione per uno studente
- Cancellare (per uno studente) o disdire (per un insegnante) una prenotazione
- Visualizzare le prenotazioni effettuate (per uno studente) o ricevute (per un insegnante) e filtrarle opportunamente

2.3 User Classes and Characteristics

Dal seguente diagramma delle classi si identificano le principali entità del sistema, con i relativi metodi e attributi, oltre alla natura delle relazioni tra le diverse classi. Si notano le classi principali, come **Studiante**, **Insegnante**, **Lezione** e **Prenotazione**, ciascuna dotata di funzionalità specifiche.

Inoltre, la classe **Utente** rappresenta un'entità generica che accorpa sia gli insegnanti sia gli studenti, facilitando la gestione delle diverse operazioni all'interno del sistema, come la visualizzazione delle lezioni e delle recensioni, indipendentemente dal ruolo ricoperto.

E' importante notare anche la classe **Admin**, la quale rappresenta un'entità che amministra il funzionamento generale della piattaforma, visualizzando un report sulle lezioni/prenotazioni effettuate, una lista degli studenti e degli insegnanti iscritti (con la possibilità di rimuoverne il profilo) e inserendo nuove discipline da poter insegnare/studiare.



2.4 Operating Environment

Per la fruizione della web application si richiede l'utilizzo:

- **Web Server:** Apache e PHP
- **Database:** MySQL

2.5 Design and Implementation Constraints

La web application deve essere completamente fruibile tramite i principali browser in uso dagli utenti registrati alla piattaforma, con distinzione fra i tipi di utenti (admin – studenti – insegnanti). Le password degli utenti devono essere criptate nel database per garantire sicurezza.

- Il registro linguistico utilizzato è l'italiano.
- Il sistema deve tracciare tutte le prenotazioni, comprese quelle passate.
- È richiesta una manutenzione periodica del sistema.

2.6 User Documentation

La consegna del prodotto è accompagnata da una guida all'utilizzo (**Manuale di Installazione**), sebbene la web app sia concepita per essere user-friendly, con una pagina di presentazione nella homepage.

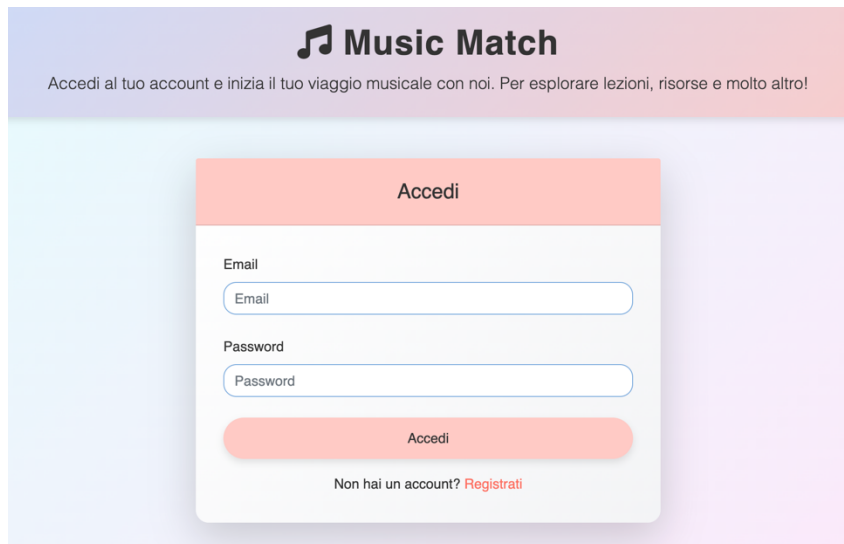
2.7 Assumptions and Dependencies

Si assume l'utilizzo delle versioni più recenti delle librerie PHP, insieme a un web server Apache per la gestione del database.

3. External Interface Requirements

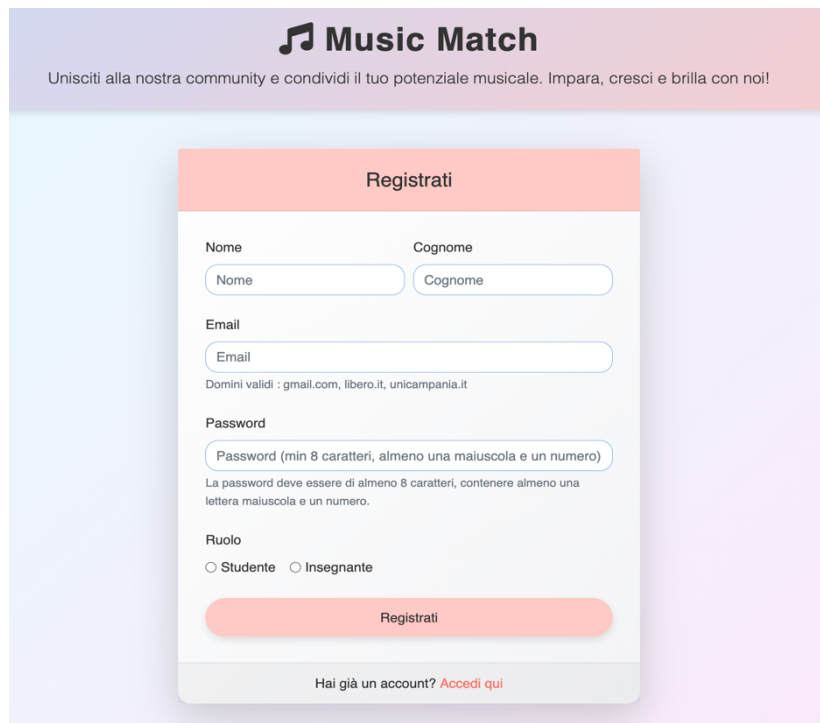
3.1 User Interfaces

- **Login Page:** Dalla Login Page, gli utenti (admin – studenti – insegnanti) possono accedere alla propria homepage. In questa pagina, è quindi possibile effettuare l'accesso oppure, nel caso in cui l'utente non fosse registrato, essere reindirizzato alla Register Page.



The screenshot shows the 'Accedi' (Login) page of the Music Match application. At the top, there is a header with the Music Match logo and a tagline: 'Accedi al tuo account e inizia il tuo viaggio musicale con noi. Per esplorare lezioni, risorse e molto altro!'. Below the header, there is a central form titled 'Accedi'. The form contains two input fields: 'Email' and 'Password'. Below these fields is a red button labeled 'Accedi'. At the bottom of the form, there is a link that says 'Non hai un account? [Registrati](#)'.

- **Register Page:** Dalla Register Page, gli utenti possono registrarsi alla piattaforma inserendo i propri dati personali nel rispetto dei vincoli imposti. In caso di utente già registrato, questo può essere reindirizzato verso la Login Page

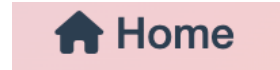


The screenshot shows the 'Registrati' (Register) page of the Music Match application. At the top, there is a header with the Music Match logo and a tagline: 'Unisciti alla nostra community e condividi il tuo potenziale musicale. Impara, cresci e brilla con noi!'. Below the header, there is a central form titled 'Registrati'. The form contains several input fields: 'Nome' (Name), 'Cognome' (Surname), 'Email', and 'Password'. Below the 'Email' field, there is a note: 'Domini validi : gmail.com, libero.it, uncampania.it'. Below the 'Password' field, there is a note: 'La password deve essere di almeno 8 caratteri, contenere almeno una lettera maiuscola e un numero.' Below these fields, there is a section for 'Ruolo' (Role) with two radio buttons: 'Studente' and 'Insegnante'. At the bottom of the form, there is a red button labeled 'Registrati'. At the very bottom of the form, there is a link that says 'Hai già un account? [Accedi qui](#)'.

- **Icona per il Logout:** In ogni pagina della web application sarà sempre disponibile un'icona per il logout. Cliccando su questa icona, l'utente può disconnettersi e tornare alla homepage.

 Logout

- **Icona per tornare alla Homepage:** Questa icona permette all'utente di tornare rapidamente alla homepage da qualsiasi pagina.
- **Icona per accedere alla Area Personale :** Questa icona consentirà all'utente di accedere alla propria personale pagina informativa.



3.2 Software Interfaces

Per lo sviluppo di questa web application è stato deciso di utilizzare i seguenti linguaggi:

- PHP come tecnologia server-side per la gestione dei dati inviati/letti al/dal database.
- HTML per la presentazione dei dati.
- CSS per la formattazione dello stile dell'applicazione.
- Javascript per la visibilità dinamica dei campi di inserimento.

Elenchiamo i codici realizzati che implementano la seguente web application, raggruppati in due macro-directory : **Authentication e Logged**

3.2.1 Authentication

In questa directory sono inseriti i codici utili alla fase di autenticazione.

- **Package Registrazione : *RegisterPage.php, CheckRegistration.php, RegistrationSuccess.php***
 - **RegisterPage.php:** Questa interfaccia consente agli utenti di creare un nuovo account fornendo i dettagli richiesti (ad es. nome, email, password, ruolo). Mostra anche eventuali errori di convalida in tempo reale.
 - **CheckRegistration.php:** Questo script gestisce l'invio del modulo di registrazione. Verifica i dati inseriti dall'utente, controlla che l'email non sia già registrata, che rispetti i vincoli di dominio e, se tutto è valido, aggiunge il nuovo utente al database.
 - **RegistrationSuccess.php:** Dopo che la registrazione è stata completata correttamente, questa interfaccia conferma all'utente che la creazione dell'account è andata a buon fine, fornendo un messaggio di successo e un link per il login.
- **Package Login : *LoginPage.php, CheckLogin.php***
 - **LoginPage.php:** Questa interfaccia permette agli utenti registrati di accedere al proprio account. Include campi per inserire email e password.
 - **CheckLogin.php:** Questo script gestisce il processo di autenticazione. Verifica se le credenziali fornite corrispondono a un account esistente e, in caso positivo, avvia una sessione per l'utente. In caso contrario, mostra errori di accesso con reindirizzamento alla pagina di registrazione.

- **Package Configurazione : *Config.php, Error.php*.**
 - **Config.php:** Questo file contiene le configurazioni globali dell'applicazione, come le credenziali di accesso al database, le impostazioni della sessione e altre variabili di configurazione necessarie per il funzionamento del sistema.
 - **Error.php:** Questa interfaccia viene visualizzata quando si verifica un errore critico nell'applicazione. Fornisce all'utente un messaggio di errore personalizzato e può fornire opzioni di supporto o reindirizzare l'utente a pagine sicure.

3.2.2 Logged

In questa directory sono inseriti i codici utili a fruire delle funzionalità della web application dopo essersi correttamente autenticati ed acceduti alla stessa.

- **Package Admin : *admin_hom.php, view_users.php, add_discipline.php***
 - **admin_home.php:** Questa è la homepage riservata all'amministratore, dove vengono elencati tutti dati amministrativi principali che può gestire, come la visualizzazione degli utenti (e la loro cancellazioni) o l'aggiunta di nuove discipline.
 - **view_users.php:** Questa interfaccia consente all'amministratore di visualizzare una lista di tutti gli utenti registrati nella piattaforma, con la possibilità di filtrarli per ruolo ed eventualmente cancellarli.
 - **add_discipline.php:** Questa pagina permette all'amministratore di aggiungere nuove discipline all'elenco delle discipline disponibili, che possono essere scelte da insegnanti e studenti per le loro lezioni.
- **Package Studente : *student_home.php, lessons_by_discipline.php, lessons_by_teacher.php, confirm_booking.php, remove_booking.php, view_booked_lessons.php, view_completed_lessons.php, write_review.php, personal_area.php, logout.php***
 - **student_home.php:** Questa è la homepage dello studente, da cui può accedere alle lezioni disponibili, visualizzarne i dettagli, gestire le proprie prenotazioni, scrivere le recensioni ed accedere alla propria area personale.
 - **lessons_by_discipline.php:** In questa interfaccia, lo studente può visualizzare tutte le lezioni disponibili per una determinata disciplina, scegliendo quella più adatta in base alle proprie esigenze.
 - **lessons_by_teacher.php:** Questa pagina permette allo studente di cercare le lezioni filtrandole in base all'insegnante, fornendo una panoramica delle lezioni offerte da un determinato docente.

- **confirm_booking.php**: Questa interfaccia consente allo studente di confermare la prenotazione di una lezione scelta. Una volta confermata, la lezione viene salvata nel sistema.
 - **remove_booking.php**: Lo studente può utilizzare questa interfaccia per annullare una prenotazione precedentemente confermata. Quella lezione tornerà nello stato “disponibile”.
 - **view_booked_lessons.php**: Questa pagina permette allo studente di visualizzare tutte le lezioni che ha prenotato e che non sono ancora state effettuate.
 - **view_completed_lessons.php**: In questa interfaccia, lo studente può visualizzare un elenco delle lezioni che ha già effettuato, con la possibilità di scrivere una recensione all'insegnante e visualizzarne l'eventuale risposta.
 - **write_review.php**: Lo studente può utilizzare questa pagina per scrivere una recensione su una lezione completata, valutando l'insegnante e la qualità della lezione.
 - **personal_area.php**: Questa è l'area personale dello studente, dove può gestire i propri dati personali, come nome, email e password e modificare il proprio livello di abilità.
 - **logout.php**: Questo script consente allo studente di disconnettersi dalla propria sessione e tornare alla pagina di login.
- **Package Insegnante : *teacher_home.php, add_lesson.php, edit_lesson.php, remove_lesson.php, view_reservations.php, remove_booking.php, view_reviews.php, personal_area.php, logout.php***
 - **teacher_home.php**: Questa è la homepage riservata agli insegnanti, dove possono accedere a tutte le funzionalità necessarie per gestire le lezioni (aggiungerle, modificarle, rimuoverle), visualizzare le prenotazioni e gestire i propri dati personali.
 - **add_lesson.php**: In questa pagina, l'insegnante può aggiungere nuove lezioni, specificando dettagli come data, orario, disciplina.
 - **edit_lesson.php**: Questa interfaccia permette all'insegnante di modificare una lezione esistente, aggiornando i dettagli relativi alla lezione.
 - **remove_lesson.php**: Qui l'insegnante può rimuovere una lezione precedentemente creata. Questo cancellerà la lezione dal sistema.
 - **view_reservations.php**: Questa interfaccia consente all'insegnante di visualizzare tutte le prenotazioni effettuate dagli studenti per le proprie lezioni e filtrarle per disciplina e livello di abilità dello studente.
 - **remove_booking.php**: L'insegnante può utilizzare questa interfaccia per disdire la prenotazione di uno studente per una lezione, aggiornando lo stato di quella lezione in “disdetta”.
 - **view_reviews.php**: In questa pagina, l'insegnante può visualizzare, ed eventualmente rispondere, alle recensioni lasciate dagli studenti per le lezioni che ha tenuto, con eventuali suggerimenti o feedback.

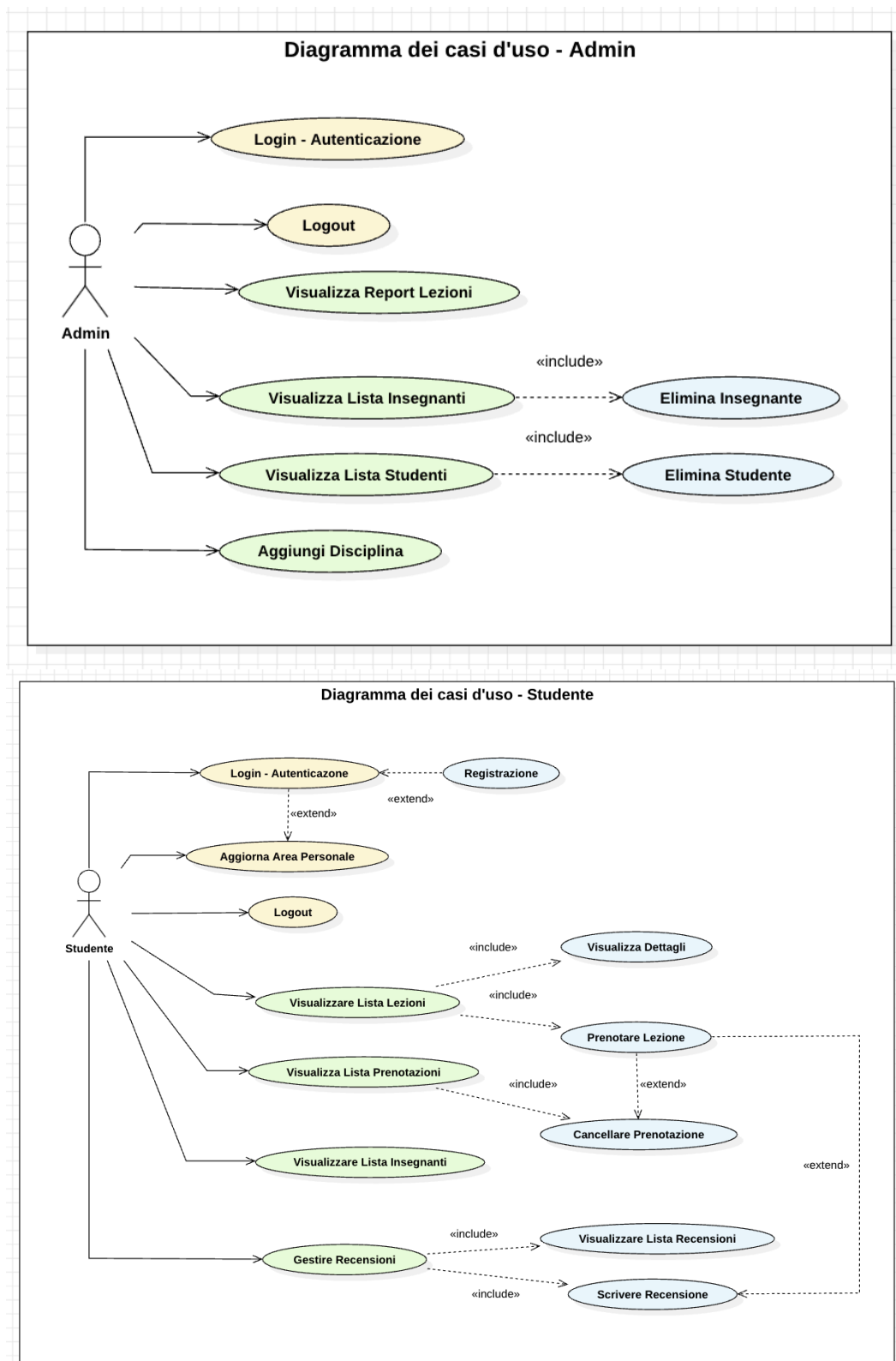
- **personal_area.php:** Questa è l'area personale dell'insegnante, dove può gestire i propri dati, come nome, email, password e aggiungere le certificazioni acquisite e le nuove discipline da insegnare.
- **logout.php:** Questo script consente all'insegnante di disconnettersi dalla propria sessione e tornare alla pagina di login.

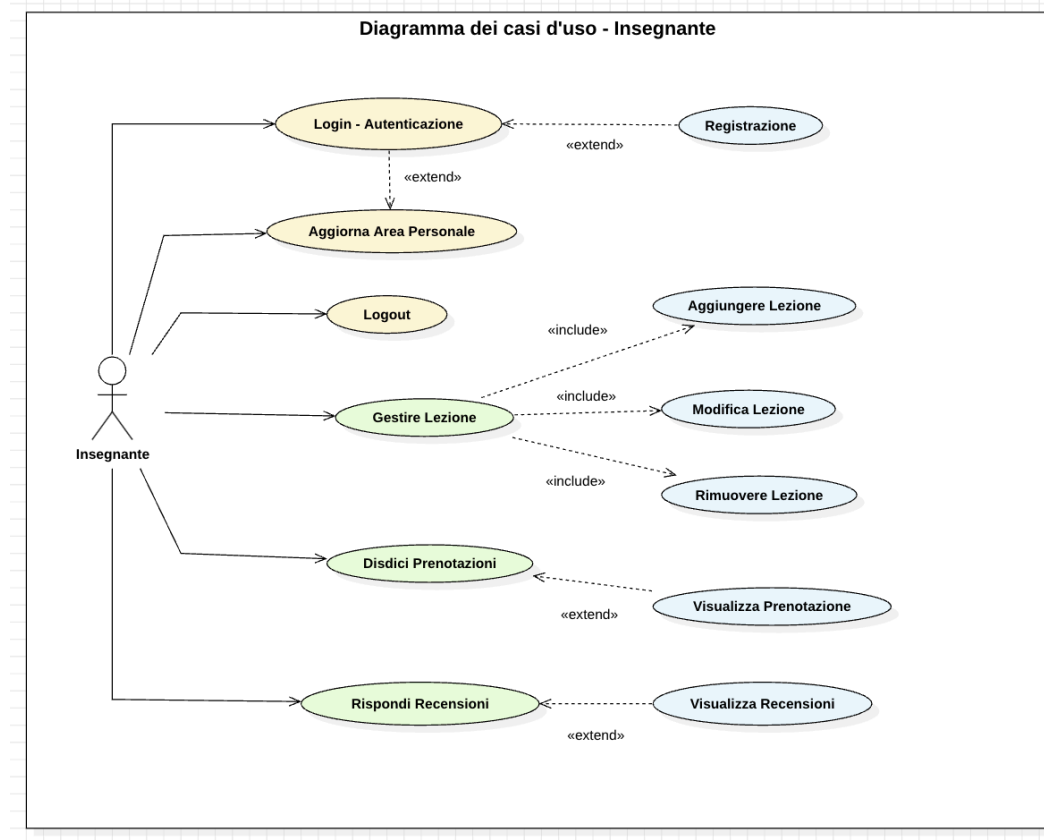
3.3 Communications Interfaces

L'infrastruttura sulla quale poggia il sistema è costituita dal web server Apache che gestisce le richieste del client web (browser) e recupera le risorse richieste dal database. MySql è l'RDBMS che si occupa della gestione del database.

La comunicazione avviene tramite scambio di messaggi HTTP.

4. System Features





4.1 System Feature: Registrati

4.1.1 Description and Priority

Questa feature consente agli utenti (studenti o insegnanti) di registrarsi al sistema compilando un form con i propri dati personali. I campi richiesti includono nome, cognome, email, password, la selezione del ruolo (Studente o Insegnante) e il livello di abilità oppure le discipline insegnate rispettivamente. La priorità è alta poiché rappresenta il punto di accesso iniziale per gli utenti al sistema.

4.1.2 Stimulus/Response Sequences

- **Stimulus:** L'utente accede alla pagina di registrazione, inserisce i dati richiesti (Nome, Cognome, Email, Password) e seleziona il ruolo (Studente o Insegnante), quindi preme il pulsante "Registrati".
- **Response:** Il sistema verifica la correttezza e la validità dei dati inseriti, in particolare controlla che l'email appartenga a uno dei domini accettati e che la password rispetti i criteri di sicurezza. Se i dati sono corretti, l'utente viene registrato. Se ci sono errori, il sistema mostra i messaggi di errore specifici (es. email non valida, password non conforme, ecc.).

4.1.3 Functional Requirements

- **REQ-1:** Il sistema deve permettere agli utenti di inserire nome, cognome, email e password.

- **REQ-2:** Il sistema deve consentire agli utenti di selezionare il proprio ruolo (Studente o Insegnante).
- **REQ-3:** Il sistema deve verificare che l'email appartenga a uno dei domini accettati (es. gmail.com, libero.it, unicompania.it).
- **REQ-4:** Il sistema deve convalidare che la password inserita abbia almeno 8 caratteri, includa almeno una lettera maiuscola e un numero.
- **REQ-5:** Il sistema deve registrare correttamente l'utente se tutti i campi sono validi.
- **REQ-6:** Il sistema deve mostrare errori specifici se i dati non sono corretti (es. email non valida, password non sicura).
- **REQ-7:** Il sistema deve gestire il reindirizzamento verso la pagina di login una volta completata con successo la registrazione.

4.2 System Feature: Autenticazione

4.2.1 Description and Priority

Questa feature permette l'autenticazione dell'utente tramite l'inserimento di credenziali, consentendo l'accesso alla specifica homepage in base al suo ruolo (studente o insegnante). La priorità di questa funzione è molto alta, poiché una corretta autenticazione è fondamentale per il funzionamento del sistema.

4.2.2 Stimulus/Response Sequences

- **Stimulus:** L'utente inserisce le credenziali (email e password).
- **Response:** Il sistema verifica la correttezza delle credenziali inserite.
- **Stimulus:** In caso di errore, l'utente è reindirizzato alla pagina di login con un messaggio di errore.

4.2.3 Functional Requirements

- **REQ-1:** Il sistema deve presentare un'interfaccia con i campi di inserimento per email e password.
- **REQ-2:** Il sistema deve verificare la correttezza delle credenziali inserite.
- **REQ-3:** Il sistema deve consentire l'accesso all'utente autenticato, mostrando la home page personalizzata (Studente o Insegnante) in caso di autenticazione riuscita.
- **REQ-4:** In caso di credenziali non corrette, il sistema deve mostrare un messaggio di errore e riportare l'utente alla pagina di login.

4.3 System Feature: Aggiungi Lezione

4.3.1 Description and Priority

Questa feature consente agli insegnanti di aggiungere nuove lezioni al sistema, specificando dettagli come disciplina di interesse, data ed orario, durata e prezzo. La priorità è alta, poiché l'aggiunta di lezioni è fondamentale per mantenere aggiornata l'offerta formativa e permettere agli studenti di prenotare ed eseguire lezioni.

4.3.2 Stimulus/Response Sequences

- **Stimulus:** L'insegnante accede alla sezione "Aggiungere Lezione".
- **Response:** Il sistema presenta un modulo con i campi per inserire i dettagli della lezione (es. disciplina, data e orario, durata, prezzo).
- **Stimulus:** L'insegnante compila il modulo e invia i dati.
- **Response:** Il sistema verifica la correttezza dei dati inseriti (in base ad alcuni vincoli) e salva la lezione nel database. Ad esempio l'insegnante non può inserire una lezione in una data precedente a quella attuale oppure per una data successiva di un anno a quella attuale. La durata di una lezione va da un minimo di 30 minuti ad un massimo di 120 minuti. Il prezzo di una lezione va da un minimo di 5€ ad un massimo di 100€
- **Response:** La lezione viene aggiunta alla lista delle lezioni prenotabili per gli studenti.

4.3.3 Functional Requirements

- **REQ-1:** Il sistema deve presentare un'interfaccia per l'inserimento dei dettagli di una nuova lezione (disciplina, data ed orario, durata, prezzo).
- **REQ-2:** Il sistema deve verificare che tutti i campi obbligatori siano compilati correttamente, nel rispetto dei vincoli detti in precedenza.
- **REQ-3:** Il sistema deve salvare i dettagli della lezione nel database.
- **REQ-4:** Il sistema deve aggiornare la lista delle lezioni disponibili dopo l'aggiunta di una nuova lezione.
- **REQ-5:** In caso di errore (es. campi non compilati, oppure non validi), il sistema deve mostrare un messaggio di errore all'insegnante e richiedere la correzione.

4.4 System Feature: Modifica Lezione

4.4.1 Description and Priority

Questa feature permette agli insegnanti di modificare i dettagli delle lezioni esistenti (es. cambiare l'orario o la durata). La priorità è media, in quanto modifica le informazioni delle lezioni già esistenti, ma potrebbe essere essenziale per mantenere la disponibilità delle lezioni aggiornata.

4.4.2 Stimulus/Response Sequences

- **Stimulus:** L'insegnante seleziona una lezione dalla lista delle lezioni create e non ancora prenotate.
- **Response:** Il sistema presenta i dettagli della lezione selezionata con la possibilità di modificarli.
- **Stimulus:** L'insegnante modifica i dettagli (es. orario, durata) e invia i dati.
- **Response:** Il sistema verifica la correttezza dei dati modificati e aggiorna la lezione nel database.
- **Response:** La lezione viene aggiornata nella lista delle lezioni disponibili per gli studenti.

4.4.3 Functional Requirements

- **REQ-1:** Il sistema deve consentire agli insegnanti di selezionare una lezione dalla lista delle lezioni create.
- **REQ-2:** Il sistema deve mostrare i dettagli della lezione selezionata in un'interfaccia modificabile.
- **REQ-3:** Il sistema deve verificare che i nuovi dettagli inseriti siano validi e coerenti.
- **REQ-4:** Il sistema deve aggiornare i dati della lezione nel database e visualizzare la lezione modificata nella lista delle lezioni disponibili.
- **REQ-5:** Il sistema deve notificare eventuali errori di inserimento e richiedere correzioni prima di salvare le modifiche.

4.5 System Feature: Rimuovi Lezione

4.5.1 Description and Priority

Questa feature consente agli insegnanti di cancellare lezioni esistenti dal sistema. Gli insegnanti possono rimuovere lezioni (*non ancora prenotate -> altrimenti in quel caso si entrerebbe nella System Feature : Disdetta Prenotazione*) che non sono più in grado di sostenere. La priorità di questa funzione è alta, poiché la gestione delle lezioni è cruciale per mantenere il calendario aggiornato e garantire che gli studenti non prenotino lezioni non disponibili.

4.5.2 Stimulus/Response Sequences

- **Stimulus:** L'insegnante accede alla sezione delle sue lezioni disponibili e seleziona la lezione da cancellare.
- **Response:** Il sistema presenta i dettagli della lezione selezionata e procede alla cancellazione. Il sistema elimina la lezione dal database e aggiorna la lista delle lezioni disponibili.

4.5.3 Functional Requirements

- **REQ-1:** Il sistema deve presentare un'interfaccia per la selezione e visualizzazione dei dettagli delle lezioni esistenti.
- **REQ-2:** Il sistema deve permettere all'insegnante di selezionare una lezione da cancellare.
- **REQ-3:** Il sistema deve eliminare la lezione selezionata dal database.
- **REQ-4:** Il sistema deve aggiornare la lista delle lezioni disponibili per riflettere la cancellazione.

4.6 System Feature: Prenota Lezione

4.6.1 Description and Priority

Questa feature consente agli studenti di prenotare una lezione nella lista tra quelle disponibili. La priorità di questa funzione è alta, poiché è essenziale per permettere agli studenti di accedere alle lezioni e per gestire il calendario degli insegnanti.

4.6.2 Stimulus/Response Sequences

- **Stimulus:** L'utente (studente) seleziona una lezione alla quale è interessato.
- **Response:** Il sistema visualizza le opzioni di data ed ora disponibili e il prezzo della lezione. Il sistema salva la prenotazione nel database.

4.6.3 Functional Requirements

- **REQ-1:** Il sistema deve mostrare un'interfaccia che permetta all'utente di selezionare una lezione fra quelle visualizzate nella lista ed i relativi dettagli.
- **REQ-2:** Il sistema deve inviare una conferma via email all'utente e all'insegnante con i dettagli della lezione prenotata.
- **REQ-3:** Il sistema deve gestire le situazioni inattese, mostrando un messaggio di errore.

4.7 System Feature: Cancella Prenotazione

4.7.1 Description and Priority

Questa feature consente agli studenti di cancellare una prenotazione effettuata per una lezione. La priorità è alta, in quanto garantisce flessibilità agli studenti e consente ad altri studenti di prenotare quella lezione.

4.7.2 Stimulus/Response Sequences

- **Stimulus:** Lo studente seleziona una prenotazione dalla lista delle prenotazioni effettuate.

- **Response:** Il sistema presenta i dettagli della prenotazione selezionata.
- **Stimulus:** Lo studente clicca su "Cancella Prenotazione".
- **Response:** Il sistema rimuove la prenotazione dal database, e aggiorna lo stato della lezione a "prenotabile" nuovamente.

4.7.3 Functional Requirements

- **REQ-1:** Il sistema deve permettere allo studente di visualizzare la lista delle sue prenotazioni.
- **REQ-2:** Il sistema deve permettere allo studente di selezionare una prenotazione dalla lista e visualizzare i dettagli.
- **REQ-3:** Il sistema deve consentire la cancellazione della prenotazione,.
- **REQ-4:** Il sistema deve aggiornare il database per rimuovere la prenotazione cancellata.

4.8 System Feature: Disdici Prenotazione

4.8.1 Description and Priority

Questa feature consente agli insegnanti di disdire una prenotazione per una lezione effettuata da uno studente. La priorità è alta, poiché la gestione delle disdette è essenziale per mantenere la soddisfazione degli utenti e garantire un servizio corretto e giusto.

4.8.2 Stimulus/Response Sequences

- **Stimulus:** L'insegnante decide di disdire una prenotazione per una lezione programmata.
- **Response:** Il sistema presenta un'interfaccia per la selezione della prenotazione da disdire e richiede conferma della disdetta.
- **Stimulus:** L'insegnante conferma la disdetta della prenotazione.
- **Response:** Il sistema aggiorna lo stato della prenotazione nel database e la marca come "disdetta".

4.8.3 Functional Requirements

- **REQ-1:** Il sistema deve presentare un'interfaccia che consenta all'insegnante di selezionare tra le lezioni prenotate, quella da disdire.
- **REQ-2:** Il sistema deve richiedere conferma all'insegnante prima di procedere con la disdetta della prenotazione.

- **REQ-3:** Il sistema deve aggiornare lo stato della prenotazione nel database come "disdetta" dopo che l'insegnante conferma.

4.9 System Feature: Visualizza Lezioni Prenotabili

4.9.1 Description and Priority

Questa feature permette agli studenti di visualizzare una lista delle lezioni disponibili, con la possibilità di filtrarle per disciplina e insegnante, di prenotarle e di accedere ai dettagli di ciascuna lezione. Agli insegnanti permette invece di vedere una lista delle lezioni inserite da esso stesso ancora prenotabili. La priorità è alta, poiché è essenziale e propedeutico per permettere agli studenti di prenotare le lezioni.

4.9.2 Stimulus/Response Sequences

- **Stimulus:** L'utente (studente o insegnante) accede alla sezione "Visualizza Lista Lezioni".
- **Response:** Se l'utente è l'insegnante, il sistema mostra una lista di lezioni con possibilità di modificarle (consulta -> *System Feature 4.4 : Modifica Lezione*) o rimuoverle (consulta -> *System Feature 4.5 : Rimuovi Lezione*). Se l'utente è lo studente, il sistema mostra una lista di lezioni disponibili la possibilità di prenotare una lezione (consulta -> *System Feature 4.7 : Prenota Lezione*) e visualizzarne i relativi dettagli : data ed orario, durata, prezzo, insegnante e relativa valutazione media.

4.9.3 Functional Requirements

- **REQ-1:** Il sistema deve mostrare una lista di lezioni disponibili all'utente.
- **REQ-2:** Il sistema deve permettere agli studenti di selezionare una lezione dalla lista e visualizzare i dettagli completi, tra cui visionare la valutazione media associata all'insegnante sulla base delle recensioni ricevute.
- **REQ-3:** Il sistema deve consentire la prenotazione della lezione.

4.10 System Feature: Visualizza Lezioni Prenotate

4.10.1 Description and Priority

Questa feature permette agli studenti di visualizzare una lista delle lezioni che hanno già prenotato, con la possibilità di filtrarle per disciplina e insegnante, di cancellare le prenotazioni e di accedere ai dettagli di ciascuna lezione prenotata. Per gli insegnanti, la feature mostra un elenco delle proprie lezioni prenotate dagli studenti, visualizzare il livello di abilità di questi ultimi e di disdire le prenotazioni. La priorità è alta poiché consente agli studenti di gestire le loro prenotazioni e agli insegnanti di monitorare le lezioni prenotate.

4.10.2 Stimulus/Response Sequences

- **Stimulus:** L'utente (studente o insegnante) accede alla sezione "Visualizza Lezioni Prenotate".
- **Response:** Se l'utente è uno studente, il sistema mostra una lista delle lezioni prenotate con i relativi dettagli ed un pulsante che determina la cancellazione di quella prenotazione (consulta -> **System Feature 4.7 : Cancellazione Prenotazione**). Se l'utente è un insegnante, il sistema mostra una lista di lezioni che sono state prenotate dagli studenti, con la possibilità di visualizzare le informazioni sul suo livello di abilità. È inoltre presente un pulsante che consente di disdire quella prenotazione (consulta -> **System Feature 4.8 : Disdici Prenotazione**).

4.10.3 Functional Requirements

- **REQ-1:** Il sistema deve mostrare una lista delle lezioni prenotate all'utente (studente o insegnante).
- **REQ-2:** Il sistema deve permettere agli studenti di selezionare (e filtrare) una lezione dalla lista e visualizzare i dettagli completi della lezione prenotata, oltre la possibilità di cancellare (per studenti) o disdire (per insegnanti) la prenotazione.

4.11 System Feature: Gestione Recensioni

4.11.1 Description and Priority

Questa feature consente agli studenti di scrivere recensioni sugli insegnanti e permette agli insegnanti di visualizzare le recensioni ricevute e, opzionalmente, di rispondere. La priorità è media, in quanto le recensioni sono utili per mantenere la qualità del servizio, ma non sono essenziali per il funzionamento di base del sistema.

4.11.2 Stimulus/Response Sequences

- **Stimulus:** Lo studente accede alla lista delle lezioni effettuate e accede alla sezione "Scrivi Recensioni".
- **Response:** Il sistema presenta un modulo per scrivere una recensione (testo e valutazione).
- **Stimulus:** Lo studente compila il modulo e invia la recensione.
- **Response:** Il sistema salva la recensione nel database e la associa all'insegnante.
- **Response:** L'insegnante può visualizzare le recensioni ricevute nella sua sezione dedicata, ed eventualmente rispondervi.

4.11.3 Functional Requirements

- **REQ-1:** Il sistema deve permettere agli studenti di scrivere una recensione per un insegnante dopo aver completato una lezione.

- **REQ-2:** Il sistema deve presentare un modulo per l'inserimento di testo e valutazione.
- **REQ-3:** Il sistema deve salvare le recensioni nel database e associarle all'insegnante corrispondente.
- **REQ-4:** Il sistema deve permettere agli insegnanti di visualizzare le recensioni ricevute in una sezione dedicata ed, eventualmente, di rispondere.
- **REQ-5:** Il sistema deve garantire che la valutazione assegnata contribuisca ad aggiornare la valutazione media per quell'insegnante e renderla visibile a tutti studenti.

4.12 System Feature: Visualizza Lista Utenti

4.12.1 Description and Priority

Questa feature permette all'admin di visualizzare una lista completa degli utenti registrati (studenti e insegnanti). L'admin ha la possibilità di filtrare la lista per tipo di utente (studente o insegnante). La priorità è alta, poiché è una funzionalità essenziale per la gestione degli utenti.

4.12.2 Stimulus/Response Sequences

- **Stimulus:** L'admin accede alla sezione "Visualizza Lista Utenti".
- **Response:** Il sistema mostra una lista di utenti, con opzioni di filtro per tipologia (studente/insegnante). L'admin può selezionare un utente e procedere alla sua cancellazione.

4.12.3 Functional Requirements

- **REQ-1:** Il sistema deve permettere all'admin di filtrare gli utenti in base al tipo/ruolo di utente.
- **REQ-2:** Il sistema deve permettere all'admin di accedere alle azioni per cancellare un utente direttamente dalla lista (e dal database).

4.13 System Feature: Cancella Utente

4.13.1 Description and Priority

Questa feature consente all'admin di eliminare un utente dal sistema. La cancellazione rimuove completamente l'utente dal database, inclusi i dati associati (come lezioni prenotate o insegnate). La priorità è media, in quanto questa azione deve essere disponibile solo in casi specifici quindi non deve influenzare il comportamento generale della web application.

4.13.2 Stimulus/Response Sequences

- **Stimulus:** L'admin seleziona un utente dalla lista e sceglie l'opzione "Cancella Utente".

- **Response:** Il sistema richiede una conferma prima di procedere con la cancellazione. Se confermata, l'utente viene rimosso dal sistema e il database viene aggiornato per riflettere tale cancellazione.

4.13.3 Functional Requirements

- **REQ-1:** Il sistema deve permettere all'admin di cancellare un utente selezionato dalla lista.
- **REQ-2:** Il sistema deve aggiornare automaticamente il database e rimuovere tutti i dati associati all'utente cancellato.
- **REQ-3:** Il sistema deve notificare l'admin che l'utente è stato cancellato con successo.

4.14 System Feature: Aggiungi Disciplina

4.14.1 Description and Priority

Questa feature permette all'admin di aggiungere nuove discipline disponibili per gli insegnanti. Le nuove discipline possono essere selezionate dagli insegnanti per ampliare le loro aree di competenza. La priorità è alta, poiché permette di mantenere aggiornato e flessibile il sistema delle discipline disponibili dando la possibilità a più insegnanti e studenti di registrarsi e tenere lezioni di proprio interesse.

4.14.2 Stimulus/Response Sequences

- **Stimulus:** L'admin accede alla sezione "Aggiungi Disciplina" e inserisce il nome e i dettagli di una nuova disciplina, quali una foto identificativa.
- **Response:** Il sistema verifica che la disciplina non sia già presente e, se la verifica è positiva, aggiunge la nuova disciplina alla lista disponibile per gli insegnanti.

4.14.3 Functional Requirements

- **REQ-1:** Il sistema deve permettere all'admin di aggiungere nuove discipline tramite un'interfaccia dedicata, richiedente nome ed immagine identificativa.
- **REQ-2:** Il sistema deve verificare che la disciplina non sia già presente nel database.
- **REQ-3:** Il sistema deve aggiornare automaticamente la lista delle discipline disponibili per gli insegnanti una volta che la nuova disciplina viene aggiunta.
- **REQ-4:** Il sistema deve notificare l'admin che la disciplina è stata aggiunta con successo.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

I tempi di caricamento delle varie pagine e delle operazioni di scrittura/lettura sul database devono essere minimizzati per offrire una buona esperienza di navigazione all'utente.

5.2 Security Requirements

- **Hashing delle password:** Le password degli utenti non devono mai essere salvate in chiaro. Verranno impiegate tecniche di hashing sicure per trasformare le password in stringhe crittografate prima della memorizzazione nel database.
- **Validazione dei dati:** La validazione sarà eseguita sia lato client che lato server.
 - **Client-side:** Per fornire feedback immediato all'utente e ridurre le richieste errate inviate al server.
 - **Server-side:** Per proteggere il sistema da dati potenzialmente malformati o pericolosi, anche in caso di bypass delle protezioni client-side.
- **Controllo degli accessi:** Solo gli utenti autenticati possono accedere alle pagine e alle funzionalità appropriate in base al loro ruolo (insegnante o studente). Gli utenti non autenticati saranno reindirizzati alla pagina di login.

5.3 Software Quality Attributes

Il codice ha una struttura tale da facilitare la manutenzione, l'aggiornamento e l'aggiunta di nuovi moduli.

5.4 Business Rules

Gli utenti con ruoli diversi (insegnanti e studenti) possono svolgere operazioni differenti e accedere a pagine diverse. Il sistema verifica le credenziali e il tipo di utente durante l'accesso per reindirizzarlo al menu corretto.

6. Database – Organization Idea

6.1 Modello concettuale: diagramma ER

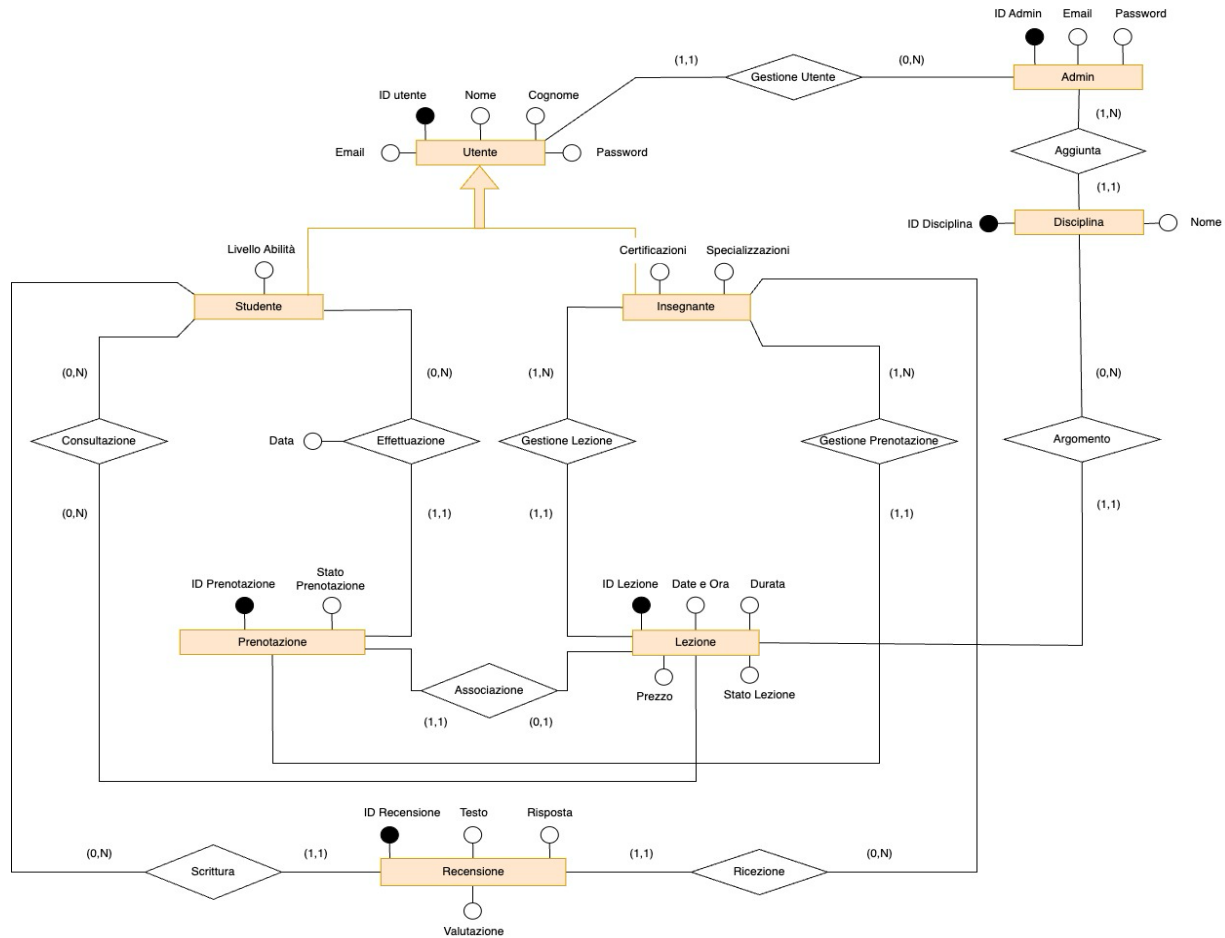
Le entità protagoniste di questa web application risultano essere :

1. **Admin:** [id_admin (PK), email, password];
2. **Utente** : [id_utente (PK), nome, cognome, email, password]
 - **Studente** : [livello_abilità];
 - **Insegnante** : [specializzazioni, certificazioni];

3. **Lezione** : [id_lezione (PK), id_insegnante (FK), disciplina (FK), data_ora, durata, prezzo, stato_lezione];
4. **Prenotazione** : [id_prenotazione (PK), id_insegnante (FK), id_studente (FK), id_lezione (FK), stato_prenotazione, data_prenotazione];
5. **Disciplina** : [id_disciplina (PK), nome];
6. **Recensione** : [id_recensione (PK), id_studente (FK), id_insegnante (FK), id_lezione (FK), valutazione, testo, risposta];

Mentre le relative relazioni, risultano :

1. **Relazione tra Insegnante e Lezione (Uno a Molti)** : Ogni insegnante può gestire molte lezioni nel corso del tempo. Tuttavia, una lezione specifica è gestita da un solo insegnante.
2. **Relazione tra Studente e Lezione (Molti a Molti)** : Ogni studente può gestire (visualizzare, scegliere...) tra più lezioni offerte dagli insegnanti. Ogni lezione può essere gestita da più studenti.
3. **Relazione tra Insegnante e Prenotazione (Uno a Molti)** : La relazione tra insegnante e prenotazione permette a ciascun insegnante di visualizzare tutte le prenotazioni ricevute dagli studenti per le lezioni che ha organizzato e gestirle, potendo sostenerle o disdirle (con conseguente rimborso), in base alla disponibilità o altre condizioni. Ciascuna prenotazione viene gestita solo dal relativo insegnante.
4. **Relazione tra Studente e Prenotazione (Uno a Molti)** : Gli studenti possono prenotare diverse lezioni nel tempo, ma una singola prenotazione riguarda sempre un singolo studente. Le lezioni sono concepite, infatti, come lezioni private.
5. **Relazione tra Lezione e Prenotazione (Uno a Uno)** : Ogni lezione si può associare o meno a una singola prenotazione, poiché si tratta di lezioni individuali. Non ci possono essere più studenti associati/prenotati alla stessa lezione contemporaneamente nella nostra web application.
6. **Relazione tra Studente e Recensione (Uno a Molti)** : Dopo aver partecipato a diverse lezioni, uno studente può scrivere una recensione, ciascuna per le diverse lezioni frequentate. Ogni recensione è unica ed è scritta da un solo studente.
7. **Relazione tra Insegnante e Recensione (Uno a Molti)** : Dopo aver sostenuto diverse lezioni, un insegnante può ricevere una recensione, ciascuna per le diverse lezioni tenute. Ogni recensione è ricevuta da un solo insegnante.
8. **Relazione tra Admin e Utente (Uno a Molti)**: Un admin può gestire più utenti (sia studenti che insegnanti), ma un utente non può essere gestito da più di un admin in questa applicazione.
9. **Relazione tra Admin e Disciplina (Uno a Molti)**: Ogni admin ha la possibilità di aggiungere, più discipline, mentre ciascuna disciplina è gestita da un solo admin.
10. **Relazione tra Disciplina e Lezione (Uno a Molti)**: Ogni lezione è associata a una sola disciplina, ma una disciplina può avere più lezioni nel corso del tempo.



7 Ristrutturazione Logica

7.1 Tabella dei Volumi

La tabella dei volumi è uno strumento essenziale per pianificare e ottimizzare la struttura e le performance di una piattaforma. Durante la fase di progettazione e successiva ristrutturazione, aiuta a stimare il carico di dati che ciascuna entità e relazione gestirà. Prevedere i volumi iniziali consente di dimensionare correttamente il sistema, ottimizzare le query e organizzare lo spazio di archiviazione. Man mano che la piattaforma cresce, questa tabella può essere aggiornata per riflettere cambiamenti come l'incremento di utenti, lezioni, prenotazioni.

Entità/Relazione	Tipo	Volume	Spiegazione
Studente	ENTITÀ	10	Stima iniziale di 10 studenti iscritti alla piattaforma.
Insegnante	ENTITÀ	10	Si ipotizza che la piattaforma abbia inizialmente 10 insegnanti registrati.

Lezione	ENTITÀ	50	Si prevede che ci siano 50 lezioni totali, create dagli insegnanti. Ogni insegnante potrebbe tenere più lezioni
Prenotazione	ENTITÀ	30	Stima iniziale di 30 prenotazioni, con ogni studente che potrebbe prenotare più lezioni. Ad esempio, una media di 3 prenotazioni per studente.
Recensione	ENTITÀ	15	Si ipotizza che circa 15 recensioni vengano lasciate dagli studenti, con la maggior parte degli studenti che lascia recensioni per alcune delle lezioni prenotate.
Disciplina	ENTITÀ	10	Si prevede che ci siano 10 discipline disponibili inizialmente, aggiunte dall'admin per coprire le materie delle lezioni.
Consultazione	RELAZIONE	100	Ipotizziamo che gli studenti visualizzino diverse lezioni prima di effettuare una prenotazione. Si stima che ci siano circa 100 consultazioni di lezioni (ogni studente consulta una media di 10 lezioni prima di prenotarne una).
Effettuazione	RELAZIONE	30 *	Questo numero coincide con il numero di prenotazioni, poiché ogni prenotazione è fatta da un singolo studente.
Gestione Lezione	RELAZIONE	50 *	Ogni insegnante gestisce una o più lezioni. Il numero di relazioni coincide con il numero totale di lezioni (50).
Gestione Prenotazione	RELAZIONE	30 *	Ogni insegnante gestisce le prenotazioni relative alle proprie lezioni. Poiché ogni prenotazione è legata a una lezione, il numero di relazioni coincide con il totale delle prenotazioni (30).
Scrittura	RELAZIONE	15 *	Questo numero coincide con il numero totale delle recensioni lasciate dagli studenti.
Ricezione	RELAZIONE	15 *	Gli insegnanti ricevono recensioni per le loro lezioni. Il numero di relazioni coincide con il totale delle recensioni (15).
Aggiunta	RELAZIONE	10 *	Si stima che inizialmente ci siano circa 10 discipline aggiunte dall'admin per coprire le materie delle lezioni.

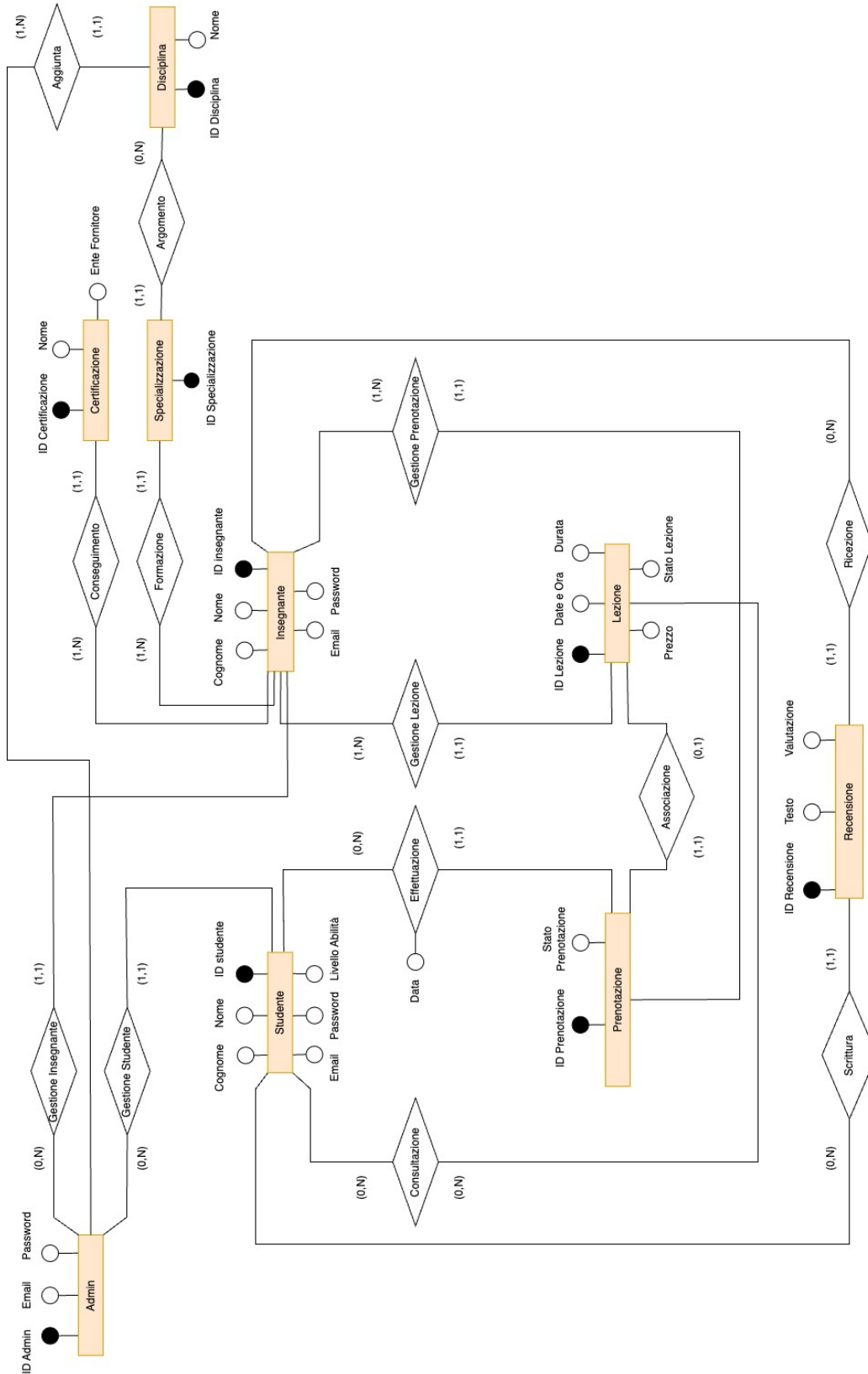
7.2 Ristrutturazione dello schema concettuale

In fase di ristrutturazione sono stati eliminati gli attributi multi-valore **Certificazioni** e **Specializzazioni** relativi all'entità Insegnante, introducendo per ciascuno di essi una nuova omonima Entità, legata da una associazione **uno a molti** (rispettivamente Conseguitamento e Formazione) all'Entità originale.

Dal momento in cui ci troviamo in presenza di una **generalizzazione totale ed esclusiva** (l'entità Utente generalizza le entità figlie Studente e Insegnante), si attua un processo di eliminazione delle gerarchie mediante **un collasso verso il basso** (che implica trasporto degli attributi del padre sulle entità figlie).

Sono state inoltre effettuate le traduzioni di relazioni a seconda della loro cardinalità.

7.2.1 Diagramma ER ristrutturato



7.3 Traduzione dello schema concettuale in schema logico

Dopo aver ristrutturato il diagramma ER, procediamo col processo di traduzione dello schema concettuale in schema logico, che costituisce un passaggio fondamentale nella progettazione di basi di dati, poiché offre diversi vantaggi chiave quali : chiarezza, ottimizzazione delle prestazioni, rispetto dei vincoli di integrità, supporto per la implementazione della base dati e della documentazione associata.

Ecco il nostro schema logico, con elementi simili a prima ...

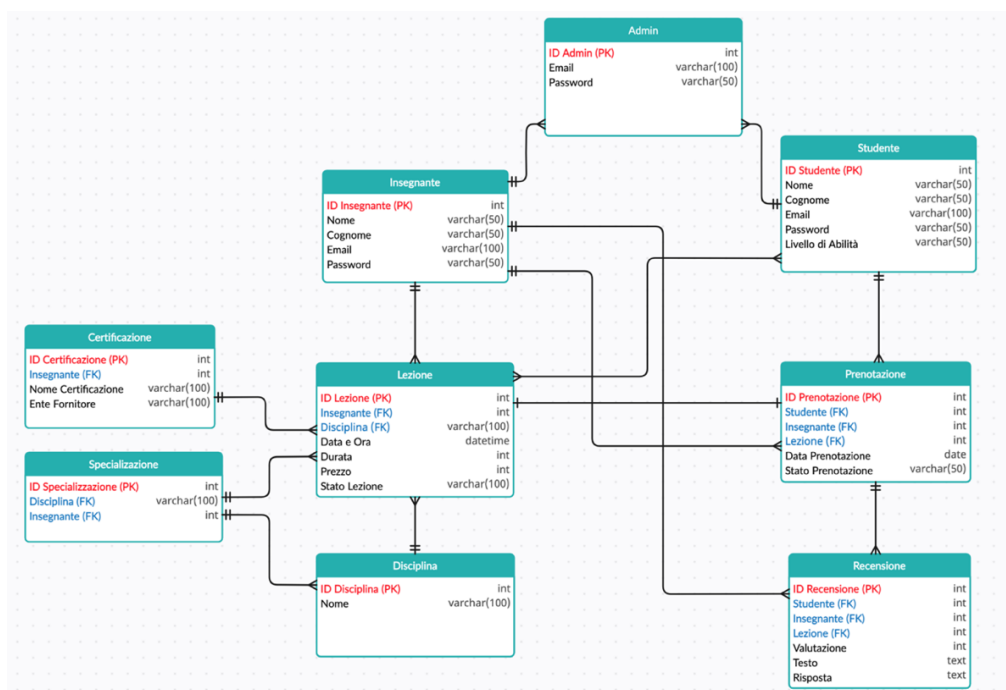
- Admin (**ID_admin**, Email, Passowrd);
- Studente (**ID_studente**, Nome, Cognome, Email, Password, Livello Abilità);
- Insegnante(**ID_insegnante**, Nome, Cognome, Email, Password);
- Lezione (**ID_lezione**, Data e ora, Durata, Prezzo, Stato_Lezione, **Insegnante**, **Disciplina**);
- Prenotazione(**ID_prenotazione**, Stato Prenotazione, Data Prenotazione, **Studente**, **Insegnante**, **Lezione**);
- Disciplina (**ID_disciplina**, Nome);
- Recensione (**ID_recensione**, Valutazione, Testo, Risposta, **Studente**, **Insegnante**, **Lezione**);

... ed alcuni appena introdotti data la ristrutturazione eseguita.

- Certificazione (**ID_certificazione**, Nome Certificazione, Ente Fornitore, **Insegnante**);
- Specializzazione (**ID_specializzazione**, **Disciplina**, **Insegnante**);
- Consultazione (**Studente***, **Lezione**);

Si intende precisare che nell'implementazione del database non è stata presa in considerazione la tabella **Consultazione** poiché non necessaria al raggiungimento degli obiettivi della nostra web application.

Quest'ultima nasce dalla traduzione della relazione Molti a Molti tra Studente e Lezione.



7.4 Glossario dei Termini

Entità	Descrizione	Attributi	Identificatore Primario
Admin	Utente – Amministratore del sistema	ID_admin, Email, Password	ID_admin
Studente	Utente – Utilizzatore del sistema	ID_studente, Nome, Cognome, Email, Password, Livello Abilità	ID_studente
Insegnante	Utente – Utilizzatore del sistema	ID_insegnante, Nome, Cognome, Email, Password	ID_insegnante
Lezione	Lezione messa a disposizione dagli insegnanti agli studenti autenticati alla web application	ID_lezione, Data e ora, Durata, Prezzo, Stato_Lezione, Insegnante, Disciplina	ID_lezione
Prenotazione	Attività consentita agli studenti se interessati ad una particolare lezione	ID_prenotazione, Stato Prenotazione, Data Prenotazione, Studente, Insegnante, Lezione	ID_prenotazione
Recensione	Strumento importante a disposizione degli studenti per valutare la qualità delle lezioni seguite e destinate all'insegnante responsabile di quella lezione.	ID_recensione, Valutazione, Testo, Risposta, Studente, Insegnante	ID_recensione
Certificazione	Attestati conseguiti dall'insegnante nel corso della sua formazione	ID_certificazione, Nome Certificazione, Ente Fornitore, Insegnante	ID_certificazione
Specializzazione	Discipline in cui l'insegnante è specializzata e per le quali offre lezioni.	ID_specializzazione, Disciplina, Insegnante	ID_specializzazione
Disciplina	Elenco delle discipline disponibili nella web application.	ID_disciplina, Nome	ID_disciplina

Relazione	Descrizione	Componenti	Attributi
Consultazione	Questa feature permette agli studenti di visualizzare una lista delle lezioni disponibili	Studente, Lezione	
Effettuazione	Questa feature consente agli studenti di prenotare una lezione con un insegnante disponibile	Studente, Prenotazione	Data Prenotazione
Gestione Lezione	Questa feature consente agli insegnanti di aggiungere, modificare o eliminare una lezione.	Insegnante, Lezione	

Gestione Prenotazione	Questa feature consente agli insegnanti di disdire la prenotazione per una lezione.	Insegnante, Prenotazione	
Gestione Insegnante	Questa feature consente all'admin di eliminare il profilo di un insegnante.	Admin, Insegnante	
Gestione Studente	Questa feature consente all'admin di eliminare il profilo di uno studente.	Admin, Studente	
Conseguimento	Legame tra gli insegnanti e le certificazioni conseguite nel corso della loro formazione	Insegnante, Certificazione	Data
Formazione	Legame tra gli insegnanti e le materie in cui si sono specializzati	Insegnante, Specializzazione	
Aggiunta	Legame tra l'admin e le discipline che aggiunge nella piattaforma affinché possano essere insegnate	Admin, Disciplina	
Argomento	Legame tra le materie in cui un insegnante è specializzato e l'insieme delle discipline aggiunte dall'admin	Specializzazione, Disciplina	
Associazione	Legame tra una lezione e la prenotazione per quella lezione	Lezione, Prenotazione	
Scrittura	Questa feature consente agli studenti di scrivere recensioni sugli insegnanti e sulle lezioni seguite	Studente, Recensione	
Ricezione	Questa feature permette agli insegnanti di visualizzare le recensioni ricevute ed eventualmente rispondere.	Insegnante, Recensione	

7.5 Vincoli di integrità

In questa sezione vengono riportati alcuni dei vincoli di integrità che la nostra web application deve rispettare. Per ulteriori dettagli su questi ed altri vincoli, si rimanda al **Manuale di Installazione** dell'applicazione.

Vincoli di integrità sui dati

- (1) Una lezione deve essere associata a un insegnante esistente.
- (2) Uno studente non può avere più di una prenotazione attiva per la stessa lezione.
- (3) Una lezione non può essere prenotata da più studenti.

- (4) Una recensione deve essere associata a una prenotazione esistente e completata dallo studente.
- (5) Uno studente non può lasciare una recensione se non ha completato la lezione.
- (6) La valutazione di una recensione deve essere compresa tra 1 e 5.
- (7) Un insegnante non può prenotare una lezione.
- (8) Uno studente può cancellare una prenotazione solo se la lezione non è ancora iniziata.
- (9) Un insegnante può disdire una prenotazione solo se la lezione non è ancora iniziata.
- (10) Uno insegnante può modificare o cancellare una lezione solo se questa non è ancora iniziata.
- (11) Ogni lezione deve avere una data e un orario futuri (non oltre un anno), non possono essere programmate per il passato.
- (12) La durata di una lezione non può essere inferiore a un valore minimo (ad esempio, 30 minuti) e superiore a un valore massimo (ad esempio, 120 minuti).
- (13) Il costo di una lezione non può essere inferiore a un valore minimo (ad esempio, 5€) e superiore a un valore massimo (ad esempio, 100€).
- (14) Ogni studente deve avere un livello di abilità registrato per la disciplina a cui partecipa.

8 Database Implementation

In questa fase, procederemo con l'implementazione del database, creando le tabelle necessarie per gestire gli utenti (admin, studenti e insegnanti), le lezioni, le prenotazioni e le recensioni all'interno del sistema. Ogni tabella riflette le entità principali e le loro relazioni, già esplicitate nel modello ER.

Utilizzeremo **chiavi primarie (PK)** per identificare in modo univoco ogni record e **chiavi esterne (FK)** per stabilire relazioni tra tabelle come studenti, insegnanti, lezioni e prenotazioni. Verranno anche definiti i vincoli di integrità referenziale, come l'eliminazione a cascata (ON DELETE CASCADE), per garantire che i dati correlati siano mantenuti coerenti.

```
CREATE TABLE Admin (  
    id_admin INT NOT NULL AUTO_INCREMENT,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    password VARCHAR(50) NOT NULL,  
    PRIMARY KEY (id_admin)  
);
```

```
CREATE TABLE Studente (  
    id_studente INT NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(50) NOT NULL,  
    cognome VARCHAR(50) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    password VARCHAR(50) NOT NULL,  
    livello_abilita VARCHAR(50),  
    PRIMARY KEY (id_studente)
```

);

```
CREATE TABLE Insegnante (  
  id_insegnante INT NOT NULL AUTO_INCREMENT,  
  nome VARCHAR(50) NOT NULL,  
  cognome VARCHAR(50) NOT NULL,  
  email VARCHAR(100) UNIQUE NOT NULL,  
  password VARCHAR(50) NOT NULL,  
  PRIMARY KEY (id_insegnante)  
);
```

```
CREATE TABLE Lezione (  
  id_lezione INT NOT NULL AUTO_INCREMENT,  
  data_ora DATETIME NOT NULL,  
  durata INT NOT NULL  
  prezzo INT NOT NULL  
  stato_lezione VARCHAR(100) NOT NULL,  
  disciplina VARCHAR(100) NOT NULL,  
  id_insegnante INT NOT NULL,  
  PRIMARY KEY (id_lezione),  
  
  FOREIGN KEY (id_insegnante)  
    REFERENCES Insegnante(id_insegnante) ON DELETE CASCADE  
  
  FOREIGN KEY (disciplina)  
    REFERENCES Specializzazione(disciplina) ON DELETE CASCADE  
);
```

```
CREATE TABLE Prenotazione (  
  id_prenotazione INT NOT NULL AUTO_INCREMENT,  
  stato_prenotazione VARCHAR(50) NOT NULL,  
  data_prenotazione DATE NOT NULL,  
  id_studente INT NOT NULL,  
  id_insegnante INT NOT NULL,  
  id_lezione INT NOT NULL,  
  PRIMARY KEY (id_prenotazione),  
  
  FOREIGN KEY (id_studente)  
    REFERENCES Studente(id_studente) ON DELETE CASCADE,  
  
  FOREIGN KEY (id_insegnante)  
    REFERENCES Insegnante(id_insegnante) ON DELETE CASCADE,  
  
  FOREIGN KEY (id_lezione)  
    REFERENCES Lezione(id_lezione) ON DELETE CASCADE  
);
```



```
CREATE TABLE Recensione (  
    id_recensione INT NOT NULL AUTO_INCREMENT,  
    testo TEXT,  
    risposta TEXT,  
    valutazione INT NOT NULL,  
    id_studente INT NOT NULL,  
    id_insegnante INT NOT NULL,  
    id_lezione INT NOT NULL,  
    PRIMARY KEY (id_recensione),  
  
    FOREIGN KEY (id_studente) REFERENCES Studente(id_studente),  
    FOREIGN KEY (id_insegnante) REFERENCES Insegnante(id_insegnante),  
    FOREIGN KEY (id_lezione) REFERENCES Lezione(id_lezione)  
);
```

```
CREATE TABLE Certificazione (  
    id_certificazione INT NOT NULL AUTO_INCREMENT,  
    ente_fornitore VARCHAR(100),  
    nome_certificazione VARCHAR(100) NOT NULL,  
    id_insegnante INT NOT NULL,  
    PRIMARY KEY (id_certificazione),  
  
    FOREIGN KEY (id_insegnante)  
        REFERENCES Insegnante(id_insegnante) ON DELETE CASCADE  
);
```

```
CREATE TABLE Specializzazione (  
    id_specializzazione INT NOT NULL AUTO_INCREMENT,  
    disciplina VARCHAR(100) NOT NULL,  
    id_insegnante INT NOT NULL,  
    PRIMARY KEY (id_specializzazione),  
  
    FOREIGN KEY (id_insegnante)  
        REFERENCES Insegnante(id_insegnante) ON DELETE CASCADE  
  
    FOREIGN KEY (nome_specializzazione)  
        REFERENCES Disciplina(nome)  
);
```

```
CREATE TABLE Disciplina (  
    id_disciplina INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL UNIQUE  
    PRIMARY KEY (id_disciplina),  
);
```

```
CREATE TABLE Consultazione (  
    id_consultazione INT NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(100) NOT NULL,  
    PRIMARY KEY (id_consultazione),  
);
```

```

id_studente INT NOT NULL,
id_lezione INT NOT NULL,
PRIMARY KEY (id_studente, id_lezione),

FOREIGN KEY (id_studente) REFERENCES Studente(id_studente),
FOREIGN KEY (id_lezione) REFERENCES Lezione(id_lezione)
);

```

Si intende precisare nuovamente che nell'implementazione del database non è stata presa in considerazione la tabella **Consultazione** poiché non necessaria al raggiungimento degli obiettivi della nostra web application.

9. Test-cases

Per dettagli più approfonditi, e visualizzazione grafica dei comportamenti attesi, consultare la sezione **TEST FUNZIONALI** del **Manuale di Installazione** (pagina 13).

AZIONE	COMPORTAMENTO ATTESO	OSSERVAZIONE
Inserimento di credenziali corrette (email e password)	Il sistema verifica le credenziali.	Assicurarsi che l'email e la password siano nel formato corretto.
Inserimento di credenziali errate	Il sistema mostra un messaggio di errore e reindirizza alla pagina di login.	Verificare che il messaggio di errore sia chiaro e informativo.
Inserimento di un'email presente nel database	Il sistema consente l'accesso alla homepage dell'utente.	Controllare che l'email sia effettivamente nel database.
Inserimento di un'email non presente nel database	Il sistema mostra un messaggio di errore e una richiesta di re-inserzione delle credenziali.	Testare con varie email per garantire la corretta gestione delle eccezioni.
Inserimento dei dettagli della lezione (data, orario, durata, prezzo)	Il sistema verifica la correttezza dei dati e aggiunge la lezione al database.	Testare con dati mancanti o non validi.
Dati non completi o errati	Il sistema mostra un messaggio di errore e richiede correzioni.	Verificare la gestione degli errori per ciascun campo.
Modifica dei dettagli della lezione (es. orario, durata)	Il sistema verifica la correttezza dei nuovi dettagli e aggiorna la lezione nel database.	Testare con modifiche non valide (es. orari passati o futuri di un anno).

Selezione di una lezione dalla lista con eventuale Prenotazione o Cancellazione	Il sistema mostra i dettagli della lezione selezionata e consente una prenotazione o cancellazione.	Testare che i dettagli visualizzati siano esatti e che la lezione sia prenotata oppure non sia più visibile agli studenti dopo la cancellazione.
Accesso alla sezione "Visualizza Lista Lezioni" come Studente	Il sistema mostra una lista di lezioni disponibili con dettagli.	Verificare la corretta visualizzazione delle lezioni e dei relativi dettagli
Accesso alla sezione "Visualizza Lista Lezioni" come insegnante	Il sistema mostra una lista delle lezioni fissate dall'insegnante	Controllare la distinzione corretta tra lezioni disponibili e prenotate.
Conferma della cancellazione	Il sistema rimuove la prenotazione dal database e invia una conferma della cancellazione.	Assicurarsi che il posto liberato sia nuovamente disponibile per altri studenti.
Selezione di una prenotazione da disdire	Il sistema mostra i dettagli della prenotazione e richiede conferma della disdetta.	Testare il processo di disdetta per prenotazioni pagate e non pagate.
Conferma della disdetta	Il sistema aggiorna lo stato della prenotazione nel database come "disdetta".	Verificare che lo stato della prenotazione sia aggiornato e che la lezione non sia più disponibile per essere nuovamente prenotata.
Scrittura di una recensione da parte di uno studente	Il sistema consente di scrivere una sola recensione per ciascuna lezione eseguita e salva la recensione nel database e la associa all'insegnante.	Testare la possibilità di lasciare un'unica recensione solo dopo aver completato una lezione.
Visualizzazione delle recensioni da parte dell'insegnante	L'insegnante può visualizzare tutte le recensioni ricevute nella sua sezione dedicata, e rispondere opzionalmente.	Verificare che le valutazioni (medie) siano visibili anche agli altri studenti, e che ci sia un campo per l'inserzione di una eventuale risposta.