

Elaborato per il corso di Basi di dati A.A. 2024/2025

Progetto di una base di dati per la gestione di un applicazione di consegne di cibo a domicilio

Alessandro Valmori
alessandro.valmori2@studio.unibo.it
0001089308

Indice

Capitolo 1 – Analisi dei requisiti

- 1.1 Intervista
- 1.2 Estrazione dei concetti principali

Capitolo 2 - Progettazione concettuale

- 2.1 Consumatori
 - 2.1.1 Progettazione dello schema E/R
 - 2.1.2 Raffinamenti proposti
 - 2.1.3 Schema concettuale parziale
- 2.2 Fattorini
 - 2.2.1 Progettazione dello schema E/R
 - 2.2.2 Raffinamenti proposti
 - 2.2.3 Schema concettuale parziale
- 2.3 Amministratore
 - 2.3.1 Note sull'Amministratore
- 2.4 Schema concettuale finale

Capitolo 3 - Progettazione logica

- 3.1 Stima del volume dei dati
- 3.2 Descrizione delle operazioni principali e stima della loro frequenza
- 3.3 Schemi di navigazione e tabelle degli accessi
- 3.4 Raffinamento dello schema
- 3.5 Analisi delle ridondanze
- 3.6 Traduzione delle entità e associazioni in relazioni
- 3.7 Schema relazionale finale
- 3.8 Traduzione delle operazioni in query SQL

Capitolo 4 – Progettazione dell'applicazione

- 4.1 Descrizione dell'architettura dell'applicazione realizzata

Capitolo 1- Analisi

1.1 intervista

Proposta di Commissione per il Sistema di Gestione Ordini e Consegne

Obiettivo della Commissione:

Progettare un sistema integrato per la gestione degli ordini, delle consegne e delle recensioni, che garantisca un'efficiente interazione tra utenti, fattorini e ristoranti, e permetta una gestione chiara e operativa delle transazioni e delle consegne.

Componenti del Sistema:

1. Gestione Utenti:

- **Registrazione e Autenticazione:** Gli utenti possono registrarsi compilando un modulo con Nome, Cognome, Nome utente (univoco), Password e Indirizzo per la consegna. Gli utenti già registrati possono accedere al sistema tramite login con il proprio username e password.
- **Portafoglio:** Ogni utente dispone di un portafoglio con un credito iniziale di 100 euro al momento della registrazione. Gli utenti possono visualizzare e gestire il loro saldo.
- **Recensioni:** Gli utenti possono fornire recensioni sui ristoranti, includendo un voto da 1 a 5 e un commento facoltativo.
- **Ordini:** Gli utenti possono effettuare ordini presso ristoranti disponibili. Ogni ordine deve includere il ristorante scelto, la quantità di ogni elemento ordinato, e la data e l'ora dell'ordine. Il prezzo dell'ordine è la somma dei prezzi degli elementi ordinati e della tariffa di consegna, che è proporzionale al prezzo dell'ordine.

2. Gestione Fattorini:

- **Visualizzazione e Accettazione Ordini:** I fattorini possono accettare ordini in stato "sospeso" che sono stati effettuati negli ultimi 60 minuti, oltre a visualizzarne dettagli quali l'elenco delle pietanze ordinate e le relative quantità e il compenso potenziale.
- **Consegna:** Ogni fattorino può consegnare solo gli ordini che ha accettato e ha un'ora di tempo per completare la consegna.

3. Gestione Ristoranti:

- **Orari di Apertura e Chiusura:** I ristoranti devono specificare i propri orari di apertura e chiusura.

- **Menù:** Ogni ristorante deve elencare le pietanze disponibili, comprese le tipologie di cucina.
- **Indirizzo:** Deve essere memorizzato l'indirizzo completo del ristorante (via, numero civico e città).

4. Autenticazione e Ruoli:

- Gli utenti possono essere autenticati in tre modalità:
 - **Amministratore:** Esiste un solo amministratore con accesso privilegiato.
 - **Utente:** Colui che effettua gli ordini.
 - **Fattorino:** Colui che si occupa della consegna del cibo.

5. Privilegi amministratore

- L'amministratore deve poter monitorare il corretto comportamento della user base. Deve poter eliminare recensioni se ritiene siano volgari o non pertinenti, e può visualizzare varie informazioni aggregate sui dati presenti, come ad esempio:
 1. Il piatto preferito dai consumatori
 2. Il ristorante con più ordini
 3. Il fattorino con più consegne
 4. Il ristorante con più recensioni negative
 5. La tipologia di cucina più gettonata

Dettagli della Commissione:

La commissione ha lo scopo di delineare e definire le specifiche tecniche e operative per il sistema sopra descritto. Dovrà considerare la creazione di un'interfaccia user-friendly, garantire la sicurezza dei dati sensibili e facilitare un'efficiente gestione delle transazioni. La commissione dovrà inoltre stabilire i criteri per la gestione dei portafogli utente, il monitoraggio e la gestione degli ordini, e la supervisione delle attività dei fattorini e dei ristoranti.

1.2 Estrazione dei concetti principali

Termine	Breve descrizione	Eventuali sinonimi
Utente	Colui che invia ordini, scrive recensioni e in generale il target user dell'applicazione.	Consumatore
Fattorino	Colui che utilizza l'applicazione a scopo di lucro, quindi consegnando ordini a domicilio.	Deliverer
Ordine	L'insieme di pietanze e bevande richieste a un determinato ristorante	Comanda
Amministratore	Colui che ha il livello di accesso più alto, ha permessi che gli altri utenti non hanno.	Moderatore

A seguito della lettura e comprensione dei requisiti, si procede redigendo un testo che ne riassume tutti i concetti e in particolare ne estragga quelli principali eliminando le ambiguità sopra rilevate:

Gli **utenti** devono potersi registrare compilando un modulo con le seguenti informazioni: Nome, Cognome, Nome utente (univoco), Password e Indirizzo per la consegna.

Se già registrati, gli utenti devono potersi autenticare attraverso **login** utilizzando il proprio username e password.

Ogni **utente** dispone di un portafoglio, nel quale è memorizzato il credito disponibile.

Ogni **utente** ha inizialmente 100 euro gratuiti all'iscrizione da spendere.

Gli **utenti** possono fornire recensioni per ciascun ristorante.

Una **recensione** include un voto (da 1 a 5) e un commento facoltativo.

Gli **utenti** possono effettuare ordini presso i ristoranti disponibili.

Un **ordine** può contenere solo elementi provenienti da un singolo ristorante.

Ogni **ordine** deve specificare il ristorante scelto, la quantità di ogni elemento ordinato, la data e l'ora dell'ordine.

Il **prezzo** di ogni ordine è costituito dalla somma dei prezzi degli elementi scelti a cui si somma la tariffa per la consegna, che è proporzionale al prezzo dell'ordine.

Una volta inviato al ristorante, l'ordine sarà in stato "sospeso" fino a quando un fattorino non lo accetta per la consegna.

I **fattorini** possono visualizzare e accettare ordini in stato "sospeso", e che siano stati fatti da meno di un'ora.

I **fattorini** possono visualizzare i dettagli dell'ordine.

Ogni **fattorino** può consegnare solo gli ordini che ha precedentemente accettato, e ha un ora di tempo per consegnarli una volta accettati.

Ogni **ristorante** deve poter specificare i propri orari di apertura e chiusura.

Devono essere elencate tutte le pietanze disponibili presso ciascun ristorante, inclusa la tipologia di cucina.

È necessario memorizzare l'**indirizzo** completo del ristorante (via, numero civico e città).

Gli **utenti** si possono autenticare in tre modi:

- **Amministratore:** Esiste un solo amministratore.
- **Utente:** Colui che ordina il cibo.
- **Fattorino:** Colui che consegna il cibo.

Segue un elenco delle principali azioni richieste:

1. Registrazione degli utenti
2. Login degli utenti
3. Visualizzazione dell'elenco dei ristoranti
4. Visualizzazione del menù di un ristorante
5. Visualizzazione delle recensioni di un ristorante

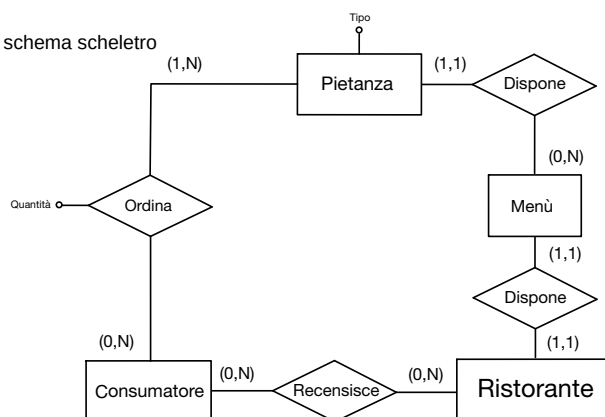
6. Inserimento di un ordine
7. Inserimento di una recensione
8. Visualizzazione degli ordini accettabili
9. Visualizzazione degli ordini accettati
10. Accettazione degli ordini da parte dei fattorini
11. Consegna degli ordini da parte dei fattorini
12. Eliminazione di una recensione
13. Visualizzazione del ristorante più scelto
14. Visualizzazione del piatto più scelto
15. Visualizzazione della tipologia di cucina più scelta
16. Visualizzazione del ristorante con le recensioni peggiori
17. Visualizzazione del fattorino con più consegne
18. Visualizzazione del compenso per un certo ordine

Capitolo 2 - Progettazione concettuale

2.1 Consumatori

2.1.1 Progettazione dello schema E/R

Viene inizialmente proposto il seguente schema scheletro



2.1.2 Raffinamenti proposti

L'entità Consumatore rappresenta una estensione di una più generica entità Utente, si decide quindi di aggiungere tale entità.

L'entità menu rappresenta in realtà l'insieme delle pietanze (cibo o bevande) che il ristorante offre ai consumatori. Viene perciò eliminata e sostituita con una nuova entità Elemento_Menù, che a sua volta si specializza in Cibo, oppure in Bevanda.

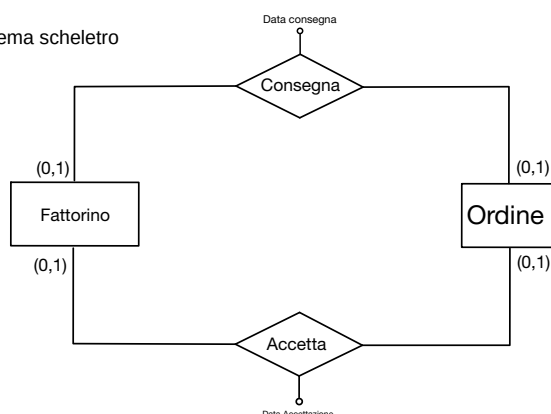
Si decide di reificare l'associazione "Ordina" in una nuova entità Ordine che si relaziona con Consumatore, elemento menu e ristorante, al fine di storicizzarne i dati. L'entità ordine è identificata attraverso la combinazione fra il timestamp dell'ordine, e le foreign key del consumatore ordinante e la foreign key del ristorante di pertinenza. Essa si relaziona a Elemento_Menù attraverso l'associazione Riferimento, a cui aggiungo l'attributo "quantità", poichè in un ordine ogni scelta è accompagnata da una rispettiva quantità.

Si decide similmente di reificare l'associazione "Recensisce" nell'entità Recensione, con un fine analogo al precedente, ovvero rendere visibili le recensioni passate ai determinati ristoranti. Questa nuova entità Recensione è in relazione con il Consumatore che la scrive e con il ristorante di riferimento. L'entità è quindi identificata dalla combinazione della data che corrisponde all'inserimento, dalla foreign key di Consumatore e dalla foreign key di Ristorante.

2.2 Fattorini

2.2.1 Progettazione dello schema E/R

Viene inizialmente proposto il seguente schema scheletro



L'entità Fattorino rappresenta una estensione di una più generica entità Utente, si decide quindi di aggiungere tale entità.

Non si modella il fatto un fattorino possa consegnare soltanto ordini che abbia precedentemente accettato. E nemmeno il fatto che un fattorino non possa consegnare un ordine accettato da un altro fattorino.

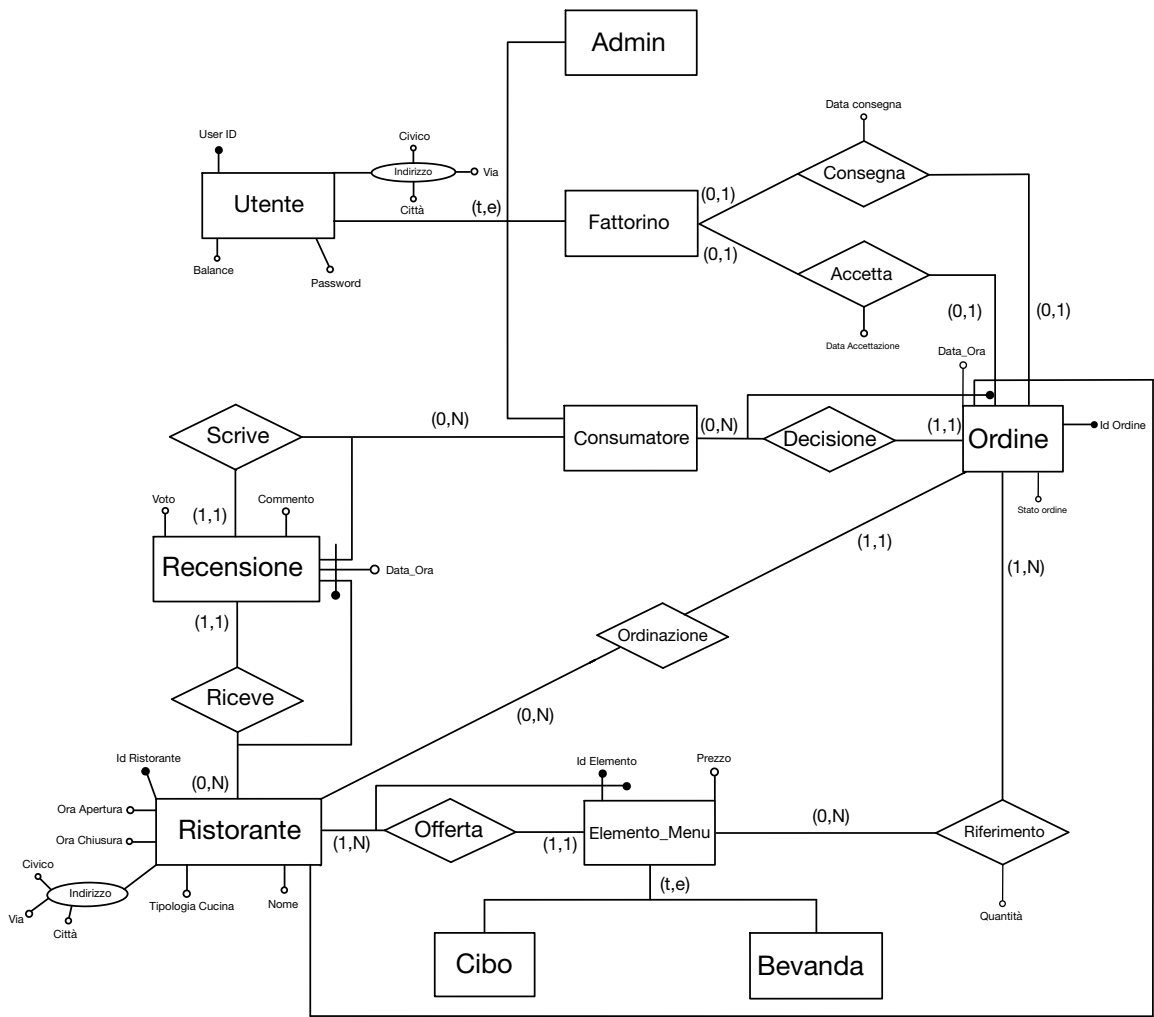
Lo schema finale risulta invariato.

2.3 Amministratore

2.3.1 Note sull'amministratore

Nello scenario preso in esame si suppone che l'amministratore sia uno solo, esso è un utente con username e password prestabiliti, poichè è l'amministratore a popolare il database con i ristoranti ad esempio.

2.4 Schema concettuale finale



Capitolo 3 - Progettazione logica

3.1 Stima del volume dei dati

Concetto	Costrutto	Volume
Consumatore	E	400
Recensione	E	200
Ordine	E	1,000
Ordinazione	R	1,000
Ristorante	E	20
Elemento_Menu	E	400

Concetto	Costrutto	Volume
Fattorino	E	50
Admin	E	1
Accetta	R	800
Consegna	R	700
Decisione	R	1,000
Offerta	R	400
Riferimento	R	5,000
Riceve	R	200
Scrive	R	200

3.2 Descrizione delle operazioni principali e stima della loro frequenza

Identifichiamo le principali operazioni e stimiamo la frequenza giornaliera di ciascuna:

#	Operazione	Frequenza giornaliera
1	Registrazione degli utenti	5
2	Login degli utenti	60
3	Visualizzazione dell'elenco dei ristoranti	60
4	Visualizzazione del menù di un ristorante	15
5	Visualizzazione delle recensioni di un ristorante	20
6	Inserimento di un ordine	10
7	Inserimento di una recensione	3
8	Visualizzazione degli ordini accettabili	40
9	Visualizzazione degli ordini accettati	12
10	Accettazione degli ordini da parte dei fattorini	8
11	Consegna degli ordini da parte dei fattorini	7
12	Eliminazione di una recensione	1
13	Visualizzazione del ristorante più scelto	1
14	Visualizzazione del piatto più scelto	1
15	Visualizzazione della tipologia di cucina più scelta	1
16	Visualizzazione del ristorante con le recensioni peggiori	1
17	Visualizzazione del fattorino con più consegne	1
18	Visualizzazione del compenso per un certo ordine	30

3.3 Schemi di navigazione e tabelle degli accessi

OP-1 Registrazione degli utenti

Concetto	Costrutto	Accessi	Frequenza = 5
Utente	E	1L	
Utente	E	1S	
$1L + 1S = 3$			
$5 \times 3 = 15$			

OP-2 Login degli utenti

Concetto	Costrutto	Accessi	Frequenza = 60
Utente	E	1L	
$1L = 1$			
60			

OP-3 Visualizzazione dell'elenco dei ristoranti

Concetto	Costrutto	Accessi	Frequenza = 60
Ristorante	E	1L	
20L			
$20 \times 60 = 1.200$			

OP-4 Visualizzazione del menù di un ristorante

Ci sono 20 ristoranti e 400 elementi, quindi ogni ristorante ha in media 20 elementi nel menu

Concetto	Costrutto	Accessi	Frequenza = 15
Ristorante	E	1L	
Offerta	R	20L	
Elemento_Menu	E	20L	
$41 \times 15 = 615$			

OP-5 Visualizzazione delle recensioni di un ristorante

Ogni ristorante ha in media 10 recensioni (200/20)

Concetto	Costrutto	Accessi	Frequenza = 20
Ristorante	E	1L	
Riceve	R	10L	
Recensione	E	10L	
$21 \times 20 = 420$			

OP-6 Inserimento di un ordine

Ogni ordine fa riferimento in media a 5 elementi con le rispettive quantità

Concetto	Costrutto	Accessi	Frequenza = 10
Ordine	E	1S	
Riferimento	R	5S	
6S			
$12 \times 10 = 120$			

OP-7 Inserimento di una recensione

Concetto	Costrutto	Accessi	Frequenza = 3
Scrivo	R	1S	
Recensione	E	1S	
Riceve	R	1S	
3S			
$6 \times 3 = 18$			

OP-8 Visualizzazione degli ordini accettabili

Concetto	Costrutto	Accessi	Frequenza = 40
Ordine	E	1000L	
Accetta	R	1000L	
2000L			
$2000 \times 40 = 80.000$			

OP-9 Visualizzazione degli ordini accettati

Concetto	Costrutto	Accessi	Frequenza = 12
Accetta	R	800	

Concetto	Costrutto	Accessi	Frequenza = 12
Consegna	R	700	
Ordine	E	1.000	
$12 \times 2.500 = 30.000$			

OP-10 Accettazione degli ordini da parte dei fattorini

Concetto	Costrutto	Accessi	Frequenza = 8
Accetta	R	1S	
$8 \times 2 = 16$			

OP-11 Consegna degli ordini da parte dei fattorini

Concetto	Costrutto	Accessi	Frequenza = 7
Consegna	R	1S	
$7 \times 2 = 14$			

OP-12 Eliminazione di una recensione

Concetto	Costrutto	Accessi	Frequenza = 1
Scrivo	R	1S	
Recensione	E	1S	
Ricevo	R	1S	
6			

OP-13 Visualizzazione del ristorante più scelto

Concetto	Costrutto	Accessi	Frequenza = 1
Ristorante	E	20L	
Ordinazione	R	1000L	
1.020			

OP-14 Visualizzazione del piatto più scelto

Concetto	Costrutto	Accessi	Frequenza = 1
Elemento_Menu	E	400L	
Riferimento	R	1000L	
1400			

OP-15 Visualizzazione della tipologia di cucina più scelta

Concetto	Costrutto	Accessi	Frequenza = 1
Ristorante	E	20L	
Ordinazione	R	1000L	
1020			

OP-16 Visualizzazione del ristorante con le recensioni peggiori

Concetto	Costrutto	Accessi	Frequenza = 1
Ristorante	E	20L	
Ricevo	R	200L	
Recensione	E	200L	
420			

OP-17 Visualizzazione del fattorino con più consegne

Concetto	Costrutto	Accessi	Frequenza = 1
Fattorino	E	50L	
Consegna	R	700L	
750			

OP-18 Visualizzazione del compenso di un ordine

Ogni ordine è composto in media da 5 elementi

Concetto	Costrutto	Accessi	Frequenza = 1
Ordine	E	1L	
Riferimento	R	5L	
6			

3.4 Raffinamento dello schema

Eliminazione delle gerarchie

Per l'eliminazione della gerarchia Utente si è scelto di adottare l'approccio del collasso verso l'alto, creando quindi l'attributo Ruolo nell'entità Utente. Il motivo della scelta è stato il fatto che gli identificatori di Consumatore e Fattorino devono essere univoci nella loro totalità, i.e. non ci può essere un Consumatore con lo stesso identificatore di un Fattorino, l'attributo Ruolo assume quindi la funzione di un livello di autorizzazione.

E' stato adottato lo stesso approccio anche per quanto riguarda l'entità Elemento, che presenta quindi un attributo "Tipo" che può assumere uno solo tra i due valori "Cibo" e "Bevanda", la scelta viene dal fatto che in questo contesto la visualizzazione del tipo di un Elemento sia solo a fine informativo, ovvero fornire più informazioni al consumatore.

Eliminazione degli attributi composti

Nello schema è presente un attributo composto nell'entità Utente e nell'Entità ristorante, in entrambi i casi l'attributo viene scomposto nei suoi sotto attributi.

Scelta delle chiavi primarie

Nello schema sono già evidenziate senza ambiguità la maggior parte delle chiavi primarie per le entità. Nel caso di ordine viene scelto di introdurre la chiave primaria ID ordine, al fine di facilitare le operazioni di join.

Eliminazione degli identificatori esterni

Nello schema E/R sono eliminate le seguenti relazioni

- Scrive, importando Username in Recensione
- Riceve, importando RistoranteID in Recensione
- Offerta, importando RistoranteID in Elemento_Menu
- Decisione, importando Username in Ordine
- Ordinazione, importando Id Ristorante in Ordine
- Riferimento, reificata importando Id Elemento da Elemento_Menu e l'identificatore di Ordine
- Accetta, reificata importando Id Ordine da Ordine e Username da Fattorino
- Consegna, trasportata sulla reificazione della associazione precedente, e poi eliminata importando l'attributo "dataConsegna" nella reificazione come attributo opzionale.

3.5 Analisi delle ridondanze

Si è scelto di non introdurre ridondanze al fine di facilitare le operazioni che fanno uso di dati derivati, come ad esempio la raccolta dei vari dati statistici fatta dall'amministratore, oppure da operazioni quali il calcolo del costo totale di ordine. La scelta è principalmente motivata dalla volontà di mantenere un'organizzazione dei dati chiara e coerente, a causa della natura dinamica dell'applicazione nell'utilizzo reale, oltre al fatto che si suppone che le operazioni di aggregazione dei dati eseguite dall'amministratore siano molto meno frequenti di quelle eseguite da fattorini e consumatori.

3.6 Traduzione di entità e associazioni in relazioni

ASSEGNAZIONI_CONSEGNE(OrdineID: ORDINI, FattorinoID: UTENTI, DataOraAssegnazione, DataOraConsegna*)

DETTAGLI_ORDINI(OrdineID: ORDINI, ElementoMenuID: ELEMENTI, Quantità)

ELEMENTI(ElementoMenuID, Nome, Prezzo, Tipo, RistoranteID: RISTORANTI)

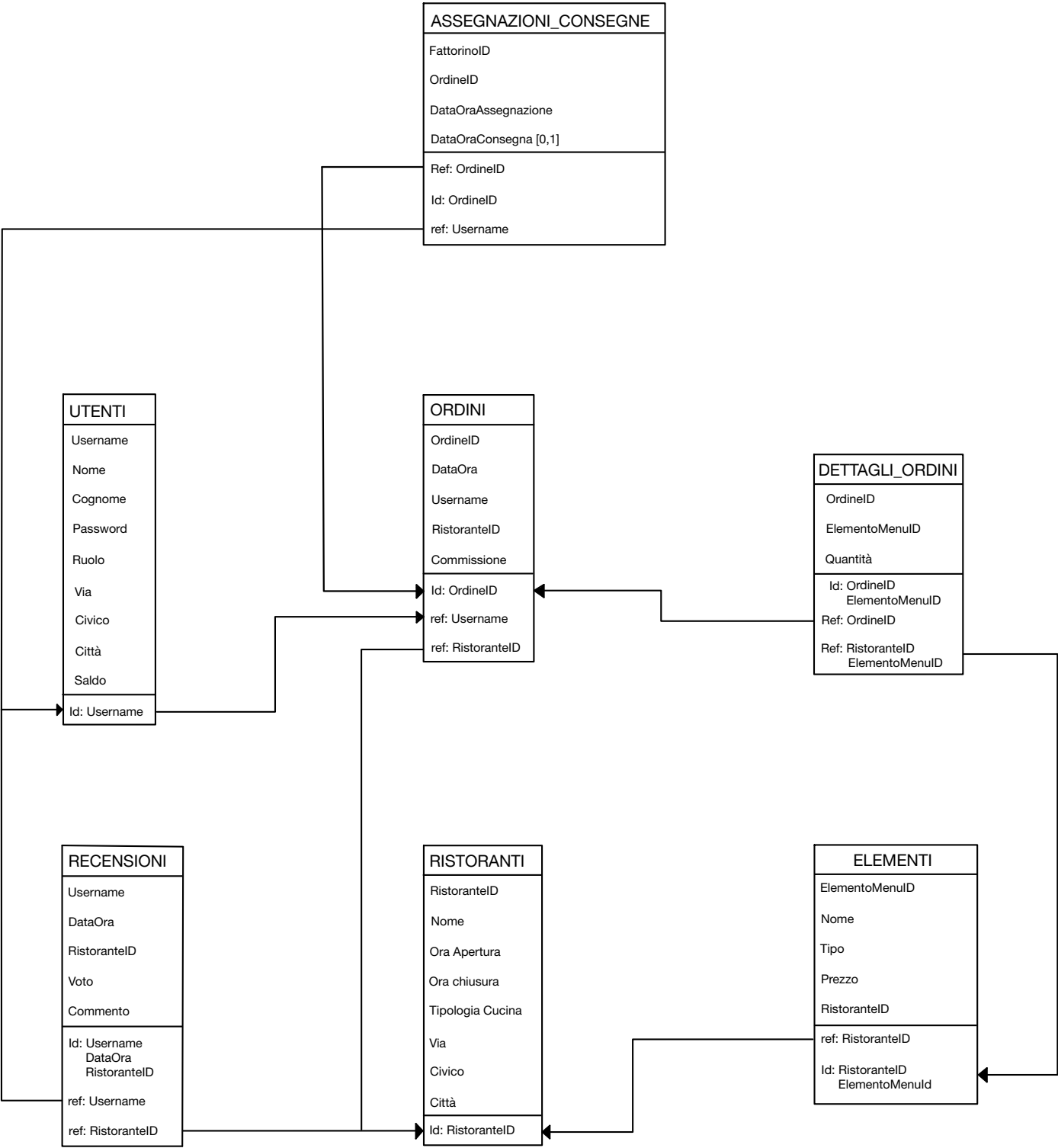
RECENSIONI(Username: UTENTI, DataOra, RistoranteID: RISTORANTI, Voto, Commento)

ORDINI(OrdineID, DataOra, Username: UTENTI, RistoranteID: RISTORANTI, Commissione)
UNIQUE(DataOra, Username, RistoranteID)

RISTORANTI(RistoranteID, Nome, OraApertura, OraChiusura, TipologiaCucina, IndirizzoVia, IndirizzoCivico, IndirizzoCittà)

UTENTI(Username, Nome, Cognome, Password, IndirizzoVia, IndirizzoCivico, IndirizzoCittà, Ruolo, Balance)

3.7 Schema relazionale finale



Traduzione delle operazioni in query SQL

OP-1 Registrazione degli utenti

Quando si iscrive un nuovo utente, è prima necessario controllare che lo username scelto sia disponibile

```
SELECT u.username
FROM UTENTI u
WHERE u.username = ?
```

Se la query non ritorna alcun valore, si può procedere con l'inserimento del nuovo utente

```
INSERT INTO UTENTI (Username, Password, Nome, Cognome, IndirizzoVia, IndirizzoCivico, IndirizzoCittà, Ruolo)
VALUES ( ?, ?, ?, ?, ?, ?, ?, ? )
```

OP-2 Login degli utenti

Per eseguire il login con username e password è sufficiente controllare che ci sia una corrispondenza in UTENTI, se la query ritorna un valore il login è andato a buon fine.

```
SELECT u.username
FROM UTENTI u
WHERE u.username = ? AND u.password = ?
```

OP-3 Visualizzazione dell'elenco dei ristoranti

```
SELECT r.RistoranteID, r.Nome, r.TipologiaCucina
FROM RISTORANTI r
ORDER BY r.TipologiaCucina
```

OP-4 Visualizzazione del menù di un ristorante

```
SELECT m.ElementoMenuID, m.Prezzo, m.Nome, m.Tipo
FROM ELEMENTI m
WHERE m.RistoranteID = ?
ORDER BY m.Tipo
```

OP-5 Visualizzazione delle recensioni di un ristorante

```
SELECT *
FROM RECENSIONI r
WHERE r.RistoranteID = ?
```

OP-6 Inserimento di ordini

Per inserire un ordine va prima inserito il record nella tabella ORDINI

```
INSERT INTO ORDINI (Username, RistoranteID, Commissione)
VALUES ( ?, ?, ? )
```

e per ogni elemento ordinato si inseriscono i vari record in DETTAGLI_ORDINI per specificare la quantità di ciascuno.

```
INSERT INTO DETTAGLI_ORDINI (OrdineID, ElementoMenuID, Quantità)
VALUES ( ?, ?, ? )
```

OP-7 Inserimento di recensioni

```
INSERT INTO RECENSIONI (RistoranteID, Voto, Commento, Username)
VALUES ( ?, ?, ?, ? )
```

OP-8 Visualizzazione degli ordini accettabili

La query identifica gli ordini che possono essere accettati dai fattorini. Un ordine è considerato accettabile se non è già stato assegnato a un fattorino e se è stato effettuato negli ultimi 60 minuti. La query esclude gli ordini già assegnati, selezionando solo quelli che rientrano nel range di tempo specificato e ordina i risultati in ordine decrescente in base alla data e ora dell'ordine.

```
SELECT o.OrdineID, o.Username, o.RistoranteID
FROM ORDINI o
WHERE o.OrdineID NOT IN (SELECT ao.OrdineID FROM ASSEGNAZIONI_CONSEGNE ao)
AND o.DataOra BETWEEN DATE_SUB(NOW(), INTERVAL 1 HOUR) AND NOW()
ORDER BY o.DataOra DESC
```

OP-9 Visualizzazione degli ordini accettati

La query elenca gli ordini che sono stati accettati da un fattorino specifico ma non ancora consegnati. La query filtra gli ordini basandosi sul fattorino identificato (indicato con ?) e considera solo quelli che sono stati assegnati a questo fattorino negli ultimi 60 minuti, ma che non sono ancora stati consegnati. I risultati sono ordinati in ordine decrescente in base alla data e ora dell'ordine.

```
SELECT o.OrdineID, o.Username, o.RistoranteID
FROM ORDINI o
WHERE o.OrdineID IN (
  SELECT ao.OrdineID
  FROM ASSEGNAZIONI_CONSEGNE ao
  WHERE ao.FattorinoID = ?
  AND ao.DataOraConsegna IS NULL
  AND ao.DataOraAssegnazione
  BETWEEN DATE_SUB(NOW(), INTERVAL 1 HOUR) AND NOW()
)
ORDER BY o.DataOra DESC
```

OP-10 e OP-11 Accettazione e consegna degli ordini da parte dei fattorini

Per accettare un ordine si inserisce l'OrdineID in ASSEGNAZIONI_CONSEGNE, senza specificare una data di consegna che sarà setta a NULL, mentre per consegnarlo si setterà la data di consegna dell'orde specificato al valore corrente.

```
INSERT INTO ASSEGNAZIONI_CONSEGNE (OrdineID, FattorinoID)
VALUES ( ?, ? )
```

```
UPDATE ASSEGNAZIONI_CONSEGNE
SET DataOraConsegna = CURRENT_TIMESTAMP
WHERE OrdineID = ?
```

Sussiste però il seguente problema: dato che la visualizzazione da parte di un fattorino degli ordini disponibili e di quelli accetati è asincrona rispetto all'accettazione e la consegna di questi ultimi, è necessario aggiungere un trigger che controlli che i vincoli impostati in lettura siano rispettati anche in scrittura, altrimenti sarebbe possibile accettare e consegnare ordini ben'oltre la soglia di un'ora desiderata, a patto che rimangano visualizzabili nella interfaccia grafica.

I check implementati nel file DDL sono i seguenti

```
DELIMITER //

CREATE TRIGGER check_accept_order
BEFORE INSERT ON ASSEGNAZIONI_CONSEGNE
FOR EACH ROW
BEGIN
  DECLARE valid_assignment BOOLEAN;

  SET valid_assignment = (
    SELECT COUNT(*) = 0
    FROM ASSEGNAZIONI_CONSEGNE ao
    WHERE ao.OrdineID = NEW.OrdineID
    AND ao.FattorinoID = NEW.FattorinoID
    AND ao.DataOraAssegnazione BETWEEN DATE_SUB(NOW(), INTERVAL 1 HOUR) AND NOW()
  );

  IF NOT valid_assignment THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Order assignment time exceeds 1 hour';
  END IF;
END;
//
```

```

CREATE TRIGGER check_deliver_order
BEFORE UPDATE ON ASSEGNAZIONI_CONSEGNE
FOR EACH ROW
BEGIN
    DECLARE valid_delivery BOOLEAN;

    SET valid_delivery = (
        SELECT COUNT(*)
        FROM ASSEGNAZIONI_CONSEGNE ao
        WHERE ao.OrdineID = NEW.OrdineID
        AND ao.DataOraAssegnazione BETWEEN DATE_SUB(NOW(), INTERVAL 1 HOUR) AND NOW()
    ) > 0;

    IF NOT valid_delivery THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Order delivery time exceeds 1 hour from assignment';
    END IF;
END;
//

DELIMITER ;

```

OP-12 Eliminazione di una recensione

```

DELETE FROM RECENSIONI
WHERE RistoranteID = ? AND Username = ? AND DataOra = ?

```

OP-13 Visualizzazione dei 5 ristoranti più scelti

La query conta il numero di ordini effettuati presso ciascun ristorante, ordina i risultati in ordine decrescente in base al numero di ordini, e restituisce solo i primi cinque ristoranti. La tabella `ORDINI` viene utilizzata per aggregare gli ordini, e la tabella `RISTORANTI` per ottenere i dettagli dei ristoranti.

```

SELECT r.RistoranteID, r.Nome, COUNT(r.RistoranteID) AS NumeroOrdini
FROM ORDINI o
JOIN RISTORANTI r ON o.RistoranteID = r.RistoranteID
GROUP BY r.RistoranteID
ORDER BY NumeroOrdini DESC
LIMIT 5

```

OP-14 Visualizzazione del piatto più scelto

La query calcola la quantità totale ordinata di ciascun piatto e lo ordina in ordine decrescente per quantità. Viene restituito solo il piatto con la quantità totale più alta. La query utilizza le tabelle `DETTAGLI_ORDINI`, `ELEMENTI` e `RISTORANTI` per raccogliere e aggregare i dati relativi ai piatti e ai ristoranti.

```

SELECT e.Nome AS Nome, r.Nome AS Ristorante, SUM(d.Quantità) AS QuantitàTotale
FROM DETTAGLI_ORDINI d
JOIN ORDINI o ON d.OrdineID = o.OrdineID
JOIN ELEMENTI e ON d.ElementoMenuID = e.ElementoMenuID AND o.RistoranteID = e.RistoranteID
JOIN RISTORANTI r ON o.RistoranteID = r.RistoranteID
WHERE e.Tipo = 'Cibo'
GROUP BY e.Nome, r.Nome
ORDER BY QuantitàTotale DESC
LIMIT 1

```

OP-15 Visualizzazione della tipologia di cucina più scelta

La query conta il numero di ordini per ciascuna tipologia di cucina, ordina i risultati in ordine decrescente in base al numero di ordini e restituisce solo la tipologia di cucina con il numero più alto. Le tabelle `ORDINI` e `RISTORANTI` sono utilizzate per raccogliere e aggregare i dati relativi alle tipologie di cucina.

```

SELECT r.TipologiaCucina, COUNT(r.TipologiaCucina) AS NumeroOrdini
FROM ORDINI o
JOIN RISTORANTI r ON o.RistoranteID = r.RistoranteID
GROUP BY r.TipologiaCucina
ORDER BY NumeroOrdini DESC
LIMIT 1

```

OP-16 Visualizzazione del ristorante con le recensioni peggiori

Al fine di calcolare il ristorante con le recensioni peggiori, si utilizza un algoritmo che consiste nel calcolare la media (modificata) nel seguente modo: Sia $X \in [1, 2, 3, 4, 5]$ il voto di una recensione, allora

$$\overline{X}_{Adj} = \frac{1 + \sum_{i=1}^n x_i}{n + 1}$$

Viene quindi tenuto conto anche del numero delle recensioni, oltre che della media campionaria.

```
SELECT r.Nome, subquery.adjusted_average
FROM RISTORANTI r
JOIN(
    SELECT r.RistoranteID, (SUM(r.Voto) + 1) / (COUNT(*) + 1) AS adjusted_average
    FROM RECENSIONI r
    GROUP BY r.RistoranteID) subquery ON r.RistoranteID = subquery.RistoranteID
ORDER BY subquery.adjusted_average ASC;
```

OP-17 Visualizzazione dei 5 fattorini con più consegne

La query conta il numero di consegne per ciascun fattorino, ordina i risultati in ordine decrescente in base al numero di consegne, e restituisce solo i primi cinque fattorini. La tabella ASSEGNAZIONI_CONSEGNE è utilizzata per aggregare le consegne per ciascun fattorino.

```
SELECT ac.FattorinoID, COUNT(ac.FattorinoID) AS NumeroConsegne
FROM ASSEGNAZIONI_CONSEGNE ac
GROUP BY ac.FattorinoID
ORDER BY NumeroConsegne DESC
LIMIT 5
```

OP-18 Visualizzazione del compenso di un ordine

Il compenso viene determinato dalla somma dei prezzi degli elementi ordinati moltiplicati per le rispettive quantità, e successivamente viene applicata la commissione associata all'ordine. La query unisce le tabelle DETTAGLI_ORDINI, ELEMENTI e ORDINI per ottenere i dettagli necessari e calcolare il compenso finale. La clausola WHERE filtra i dati per un identificativo di ordine specifico.

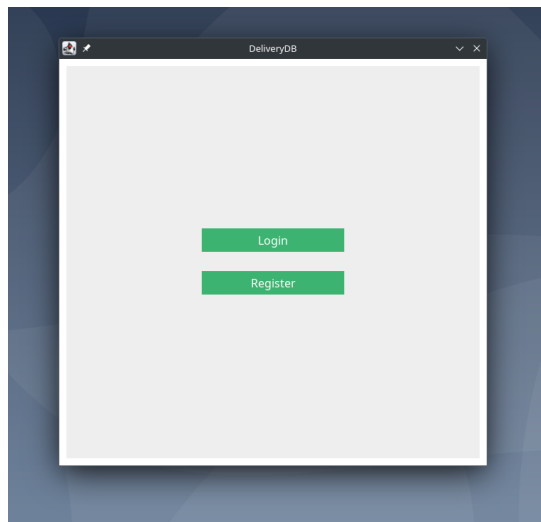
```
SELECT SUM(m.Prezzo * d.Quantità) * o.Commissione AS CompensoOrdine
FROM DETTAGLI_ORDINI d
JOIN ORDINI o ON d.OrdineID = o.OrdineID
JOIN ELEMENTI m ON d.ElementoMenuID = m.ElementoMenuID AND o.RistoranteID = m.RistoranteID
WHERE d.OrdineID = ?
```

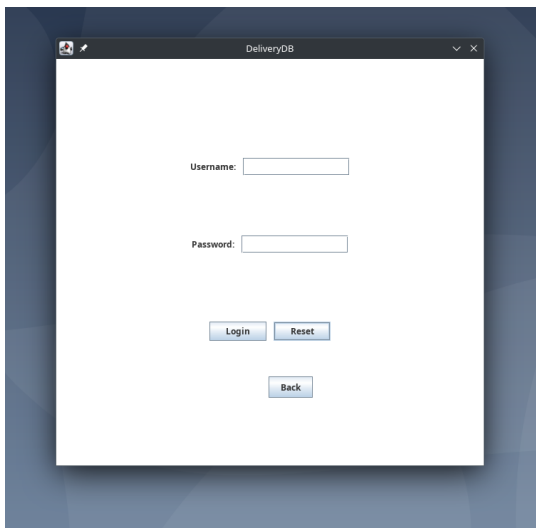
Capitolo 4 - Progettazione dell'applicazione

4.1 Descrizione dell'architettura dell'applicazione realizzata

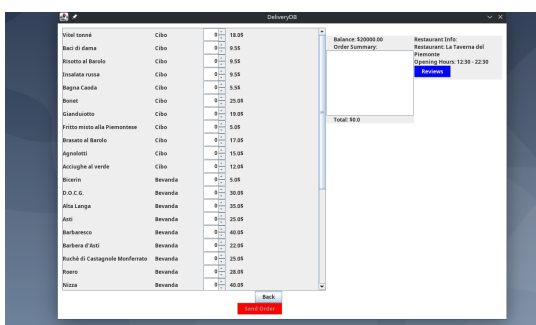
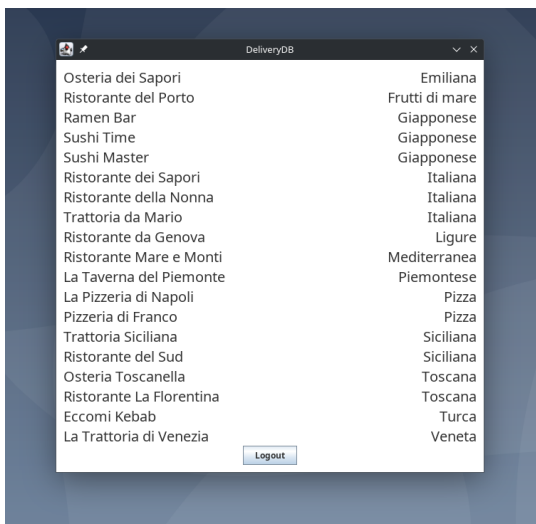
L'applicazione per interfacciarsi al database è stata realizzata in Java, sfruttando lo strumento di ORM MySQL Connector; il database risiede in locale e il DBMS usato è MySQL. L'applicazione utilizza Swing per le interfacce grafiche, le quali sono organizzate in schermate, le quali possiedono un controller dedicato ciascuna, al fine di rendere chiara la responsabilità e funzione di ciascun interfaccia nell'applicazione. Vengono utilizzati dei DAO (Database Access Objects) al fine di semplificare la traduzione fra modello relazionale e modello a oggetti.

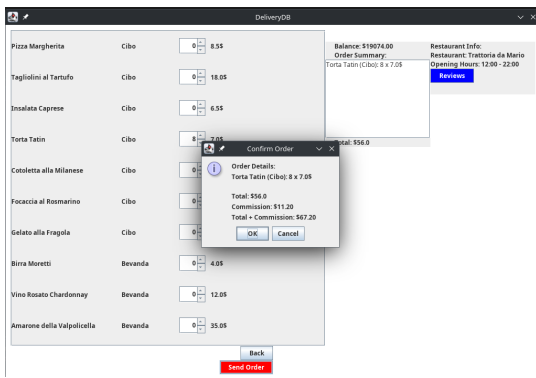
A seguito alcuni screenshot delle GUI realizzate per l'autenticazione.



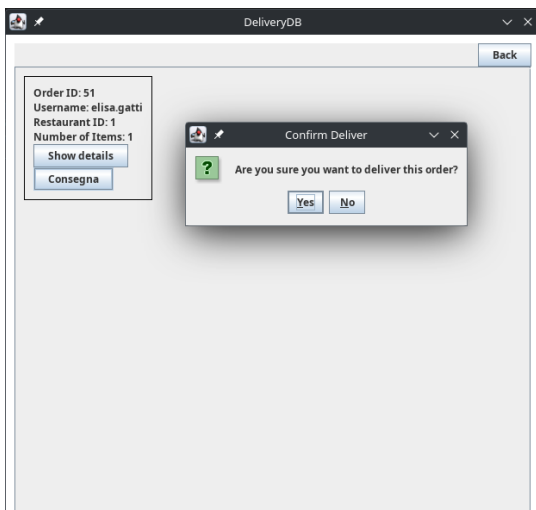
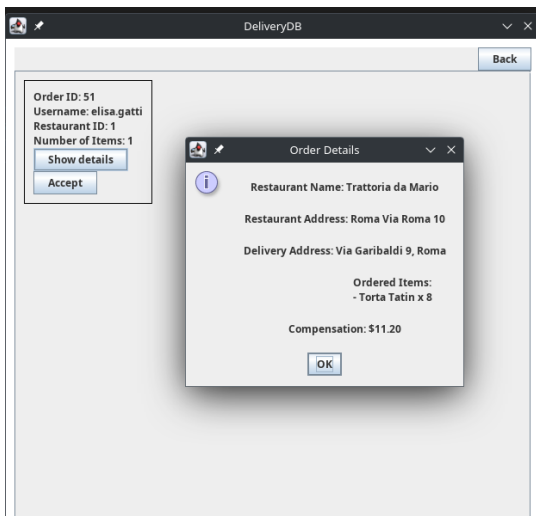


Per i consumatori sono state realizzate le seguenti GUI, che forniscono tutti i dettagli necessari al fine di comporre un ordine e leggere/scrivere recensioni





Per i fattorini, sono predisposti invece gli strumenti per accettare e/o consegnare degli ordini



All'amministratore è fornito un pannello di controllo generico al fine di visualizzare le varie informazioni derivate dai vari dati, e di moderare il contenuto delle recensioni di ciascun ristorante.

