

UNINADELIVERY

Documentazione di progetto di

Object Orientation

Università degli Studi di Napoli Federico II



Corso di studi in Informatica A.A. 2023/2024

AUTORI:

Fabrizio Apuzzo, Maria Della Valle

DOCENTE:

Professore Sergio Di Martino

Data emissione: 4 febbraio 2024

SINTESI DEI CONTENUTI

Progetto relativo alla creazione dell'applicativo Java UninaDelivery per la gestione della spedizione di merci, la sua progettazione tramite l'uso del diagramma delle classi di design con annessa documentazione comprensiva di esempi di Sequence Diagram.

SOMMARIO

<u>Capitolo 1 – Analisi Del Problema</u>	<u>1</u>
1.1 Obiettivo	1
1.2 Definizione	1
1.3 Funzionalità	2
<u>Capitolo 2 – Progettazione</u>	<u>3</u>
2.1 Considerazioni sul dominio	3
2.2 Design dell'applicazione	4
<u>Capitolo 3 – Implementazione</u>	<u>7</u>
3.1 Tecnologie utilizzate	7
3.2 Programmazione ad Oggetti.....	7
3.3 Diagramma delle classi di design	8
3.4 Esempi di Sequence Diagram	10

Capitolo 1 | ANALISI DEL PROBLEMA

1.1 Obiettivo

Si sviluppa un sistema informativo, composto da un applicativo Java dotato di GUI (Swing), per la gestione di UninaDelivery.

1.2 Definizione

UninaDelivery è un sistema di spedizione merci su scala regionale. Opera su più sedi, ognuna con i propri mezzi di trasporto, corrieri e operatori. Questi ultimi si occupano di gestire le spedizioni, in base agli ordini da consegnare, alla data e all'orario in cui è preferibile farlo, tenendo conto della disponibilità dei mezzi di trasporto in base al carico e alla presenza di corrieri adatti. Tali ordini sono effettuati da acquirenti iscritti ad UninaDelivery, i quali possono inserire più unità di più prodotti, sempre in base alla disponibilità nelle sedi, e impostare una fascia oraria in cui vogliono sia consegnato l'ordine.

1.3 Funzionalità

L'applicativo di UninaDelivery permette ai propri operatori, previa autorizzazione, di:

- Visualizzare un report statistico mensile degli ordini giunti alla propria sede, selezionando mese ed anno, comprensivo di:
 - Numero medio di ordini al giorno effettuati nel mese;
 - Ordini consegnati o da consegnare nel mese con il maggior numero di prodotti;
 - Ordini consegnati o da consegnare nel mese con il minor numero di prodotti.
- Generare una nuova spedizione per la propria sede, con i seguenti step:
 - Visualizzando gli ordini pronti ad essere spediti;
 - Filtrandoli come meglio crede secondo un determinato acquirente e/o data e orario di consegna;
 - Selezionando gli ordini che vuole inserire nella spedizione;
 - Scegliendo la data, l'orario di partenza e l'orario di arrivo della spedizione;
 - Selezionando il mezzo di trasporto che consegnerà gli ordini e il corriere che guiderà tale mezzo, in base alla loro disponibilità temporale, la capienza del mezzo e la patente del corriere.
 - Confermando, infine, la creazione della spedizione.

Capitolo 2 | PROGETTAZIONE

2.1 Considerazioni sul dominio

È importante, prima di dare una spiegazione alle scelte di design effettuate nella progettazione, fare riferimento ai dati su cui opera l'applicativo e alla loro organizzazione interna. Operando UninaDelivery su una base di dati esterna a Java, sono riportate qui le informazioni più importanti sulla struttura del database, utili a comprendere scelte di progettazione effettuate successivamente.

Il database dà ampio spazio agli aspetti che interessano la gestione delle spedizioni, è mappato invece solo lo stretto necessario della parte relativa all'effettuazione degli ordini.

Ogni sede dispone di una gamma di mezzi di trasporto. Si differenziano secondo:

- la tipologia, strettamente connessa al massimo carico (peso) che possono trasportare;
- la patente richiesta per poterli condurre.

Il corriere e l'operatore sono entrambi caratterizzati da un codice fiscale e un salario adeguato e lavorano per una sola sede. Il corriere ha anche una patente, adatta a guidare mezzi specifici. L'operatore invece è abilitato a gestire le spedizioni sugli ordini relativi solo alla propria sede, con i suoi mezzi e i suoi corrieri. Per accedere e gestire tali dati utilizza una mail aziendale con password associata.

UninaDelivery dispone di una vastità di articoli. Non tutti i prodotti sono presenti nel catalogo di tutte le sedi. Se un prodotto appartiene al catalogo di una sede, allora esiste una relazione tra i due.

I prodotti possono essere inclusi negli ordini, in quantità variabile. Non per forza tutte le unità di un prodotto devono essere presenti nella stessa sede. Se, ad esempio, un cliente ordina 10 divani, ma nella sede più vicina a lui ce ne sono solo 7, automaticamente gli altri 3 saranno reperiti dalla successiva sede che ha disponibilità, dividendo l'ordine in due ordini differenti, o se, ad esempio, per la programmazione decisa dall'acquirente non è possibile spedire gli articoli per mancanza di corrieri, si sceglierà la sede successiva che può recapitare l'ordine in tempo. Non è mappata né la vicinanza delle sedi all'acquirente né la gestione di tali situazioni poiché, in entrambi i casi presentati e simili, sarà un applicativo diverso (quello che si occupa dell'interazione con il cliente, ad esempio un sito web) a gestire tali situazioni. L'operatore, dunque, vedrà sempre e solo gli ordini relativi alla propria sede, senza porsi problemi riguardanti le altre sedi e la disponibilità degli articoli.

Come già visto, quando genera l'ordine, l'acquirente decide una programmazione, inserisce i prodotti da spedire e specifica l'indirizzo al quale vanno recapitati. I clienti sono registrati tramite la propria mail a UninaDelivery, possono inserire più metodi di pagamento (o nessuno, assumendo che siano registrati senza aver effettuato ancora alcun ordine) e scegliere di aggiungere un recapito telefonico.

Tutti gli ordini sono salvati nel database e possono essere catalogati in base al momento della loro effettuazione, allo stato in cui sono e alla loro programmazione. L'operatore può a questo punto generare più spedizioni, con un orario di partenza e uno di arrivo, inserendovi un numero variabile di ordini in base alla disponibilità di mezzi di trasporto e corrieri adatti a recapitarli.

Sono lasciate all'operatore la scelta e la possibilità di far partire spedizioni di lunghezza variabile, in base alla distanza sia fisica sia temporale fra gli ordini selezionati e alla loro quantità. Sarà lui, con buonsenso, a rendere massima l'efficienza delle spedizioni scegliendo mezzi e orari adatti, aiutato dall'interazione con il database che svolge invece compiti più semplici che non necessitano discrezione dell'operatore.

2.2 Design dell'applicazione

Avendo ora le informazioni necessarie, possono essere evidenziare le scelte di progettazione proprie dell'applicativo che sono state prese in funzione dell'interpretazione del dominio descritta al [Capitolo 1](#) e della conseguente progettazione del database. È possibile fare riferimento all'UML Class Diagram del seguente paragrafo per comprendere al meglio le scelte effettuate in fase di progettazione, e viceversa.

2.2.1 | Fase di login

L'operatore, caratterizzato da una e-mail aziendale e una password, prima di poter effettuare qualsiasi operazione, deve essere autorizzato da UninaDelivery. Egli potrà, in dei campi appositi, inserire la propria e-mail completa (oppure solo il proprio nome utente) e la propria password. Eseguito un controllo sulla coerenza dei dati inseriti, e tenuto conto della sede per cui lavora l'operatore, così da mostrargli solo informazioni relative ad essa, l'applicativo gli permetterà di entrare nella sua area di lavoro.

2.2.2| Visualizzazione del report

Per visualizzare il report statistico mensile, l'operatore dovrà scegliere a quale mese di quale anno farà riferimento il report. Allora chiederà ad UninaDelivery di calcolare le statistiche e gli saranno mostrati i risultati.

Vengono mostrati:

- Numero medio di ordini effettuati ogni giorno nel mese selezionato
- Ordini con il numero massimo e minimo di prodotti consegnati o da consegnare nel mese selezionato

Degli ordini con numero massimo e minimo di prodotti saranno visualizzati soltanto e-mail dell'acquirente, data di effettuazione e indirizzo di consegna. Da qui potrà decidere se modificare anno e/o mese per mostrare altri report o se tornare alla pagina iniziale per svolgere altre operazioni.

2.2.3| Creazione di una nuova spedizione

L'operatore avrà la possibilità di avviare il processo di creazione di una spedizione solo se sono presenti nuovi ordini. Tale processo sarà diviso in momenti differenti:

- un primo momento in cui l'operatore può filtrare e selezionare gli ordini da spedire;
- un secondo momento in cui, se non si sono riscontrati errori, può selezionare data, orario di partenza e orario di arrivo, mezzo di trasporto e corriere per la spedizione.

Il passaggio da un momento all'altro non è irreversibile. La conferma di creazione di spedizione, invece, lo è. All'avvio del processo, dunque, all'operatore saranno mostrati tutti gli ordini confermati che non sono stati ancora spediti, e di essi le seguenti informazioni: e-mail dell'acquirente, data e fascia oraria in cui è preferibile consegnarli, e peso. Egli potrà, se vuole e quante volte vuole, inserire negli appositi campi i filtri (e-mail, data, orario di inizio e/o orario di fine consegna) e applicarli, così da poter visualizzare solo gli ordini che li rispettano. UninaDelivery terrà conto degli ordini selezionati, così da poter:

- aggiornare costantemente il peso totale degli ordini e mostrarlo all'operatore. Questo è particolarmente utile se, ad esempio, il peso totale supera il massimo possibile, cioè quando non c'è alcun mezzo di trasporto nella sede che abbia una capienza superiore a tale peso. In questo caso, si impedisce all'operatore di passare al momento successivo del processo.
- mostrare all'operatore il numero dei mezzi disponibili a trasportare quel peso, che abbiano ognuno almeno un corriere disponibile abilitato a guidarli. Tale numero è mostrato solo se gli ordini selezionati

hanno delle fasce orarie e delle date compatibili fra loro, cioè solo se è possibile creare una sola spedizione che consegna tutti gli ordini selezionati senza rallentamenti. Altrimenti, tale incoerenza verrà notata (Attenzione! Ciò non impedisce la continuazione del processo) e sarà l'operatore a decidere come gestirla.

- banalmente, conservare le informazioni sugli ordini, così che non solo le merci giuste vengano caricate con successo (si suppone che un altro applicativo sia a supporto di tali operazioni) ma così che anche quando l'operatore torni a generare un'ulteriore spedizione, non gli siano mostrati ad esempio ordini già spediti o corrieri occupati.

Selezionati gli ordini, all'operatore verranno mostrati tutti i mezzi di trasporto della sua sede abilitati a trasportare il peso totale degli ordini e non appena avrà selezionato data, orario di partenza e orario di arrivo della spedizione, potrà scegliere il mezzo, scelto fra i disponibili in quell'orario secondo i criteri già indicati. Di tale mezzo verranno visualizzati i corrieri disponibili a guidare il veicolo e l'operatore potrà scegliere di selezionarne uno.

È importante notare che una spedizione non potrà essere creata se sarà verificata almeno una delle seguenti condizioni:

- non è selezionato alcun ordine;
- l'orario di partenza della spedizione è successivo a quello di arrivo;
- la data selezionata per la partenza è precedente al giorno attuale;
- l'arrivo previsto è a più di nove ore dalla partenza selezionata.

Selezionato tutto il necessario e rispettati i vincoli, l'operatore potrà creare la spedizione, a seguito di una conferma.

Si noti che il software è deliberatamente progettato per non vincolare l'operatore a rispettare gli orari della consegna, a patto che le decisioni da lui prese non violino vincoli dettati dal dominio, sarà l'operatore a gestire le consegne al massimo delle sue capacità sfruttando al meglio le risorse a disposizione della sua sede.

Capitolo 3 | IMPLEMENTAZIONE

3.1 Tecnologie utilizzate

L'applicativo UninaDelivery è sviluppato nel linguaggio Java, con il supporto dei seguenti strumenti:

- Eclipse, per la stesura del codice;
- Apache Maven, per la gestione delle librerie;
- Github, per collaborare attraverso l'utilizzo del repository;
- Discord, per comunicare.

Sono utilizzate le seguenti librerie esterne:

- com.postgresql, per permettere la comunicazione e il trasferimento dei dati dall'applicativo alla base di dati;
- com.github.lgooddatepicker, per permettere all'operatore di selezionare intuitivamente date e orari;
- com.miglayout, Layout Manager versatile per Java Swing.

3.2 Programmazione ad Oggetti

L'impiego del paradigma di programmazione Object-Oriented, in combinazione con l'euristica Entity-Boundary-Control e l'utilizzo dei design pattern DAO (Data Access Object) e DTO (Data Transfer Object) garantiscono un rapido ed elegante sviluppo del codice per la gestione dell'applicativo, fornendo una maggiore robustezza, flessibilità e una più sostenibile manutenibilità del codice.

Le classi di UninaDelivery seguono l'euristica EBC secondo le seguenti chiavi:

- Le Entities fanno riferimento alle classi del dominio necessarie al funzionamento dell'applicativo;

- Le Boundaries si interfacciano con l'utente, mostrandogli informazioni e accettando suoi input;
- Esiste un unico Controller, che si occupa di gestire tutte le classi costituenti dell'applicativo, sia Entities, sia Boundaries, e tutte le interazioni possibili fra di loro.

I design pattern DAO e DTO permettono invece al software di essere in costante comunicazione con la base di dati sottostante e di garantire la persistenza dei dati. Per ogni classe DTO facente parte del dominio esiste una classe DAO equivalente che permette di applicare modifiche e/o recuperare informazioni dal Database.

3.3 Diagramma delle classi di design

3.3.1 | Alcune precisazioni

È qui di seguito presentato il diagramma delle classi UML di UninaDelivery. Esso segue l'euristica EBC e i design pattern DTO e DAO nei modi sopraindicati.

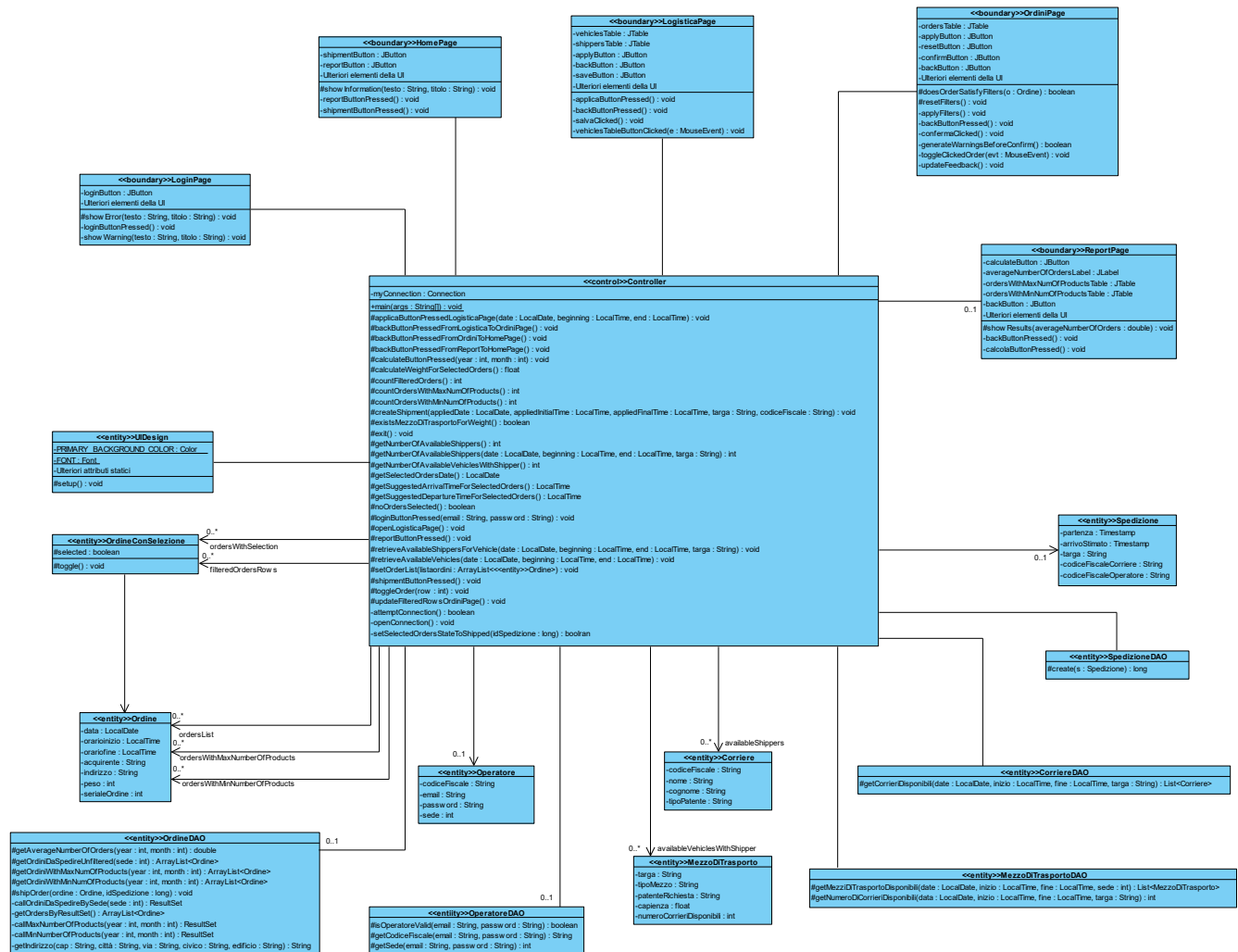
Ogni classe presente nel diagramma, con conseguenti attributi, metodi e associazioni, è implementata nel codice Java, fatta eccezione per la classe di supporto UIDesign, non presente nel diagramma ma solo nel codice, utilizzata per migliorare la piacevolezza dell'interazione del software con l'utente.

Viceversa, sono presenti solo elementi che hanno un riscontro fisico nel diagramma: classi e associazioni che hanno senso di esistere nel dominio UninaDelivery, ma non sono necessarie al funzionamento dell'applicativo, sono omesse.

Per semplificare la leggibilità del diagramma, sono state prese alcune scelte da tenere in considerazione durante la sua consultazione:

- Talora non sia specificata la molteplicità di un'associazione, si assuma essa equivalga ad 1;
- Non sono presenti costruttori, metodi getter e metodi setter semplici, pur essendo presenti nel codice;
- Sono evidenziati solo alcuni dei più importanti attributi delle Boundaries, i restanti sono omessi;
- Sono omesse anche delle classi innestate delle Boundaries, utilizzate per l'aggiornamento in tempo reale delle tabelle di informazioni mostrate all'utente.

3.3.2 | Class Diagram UML

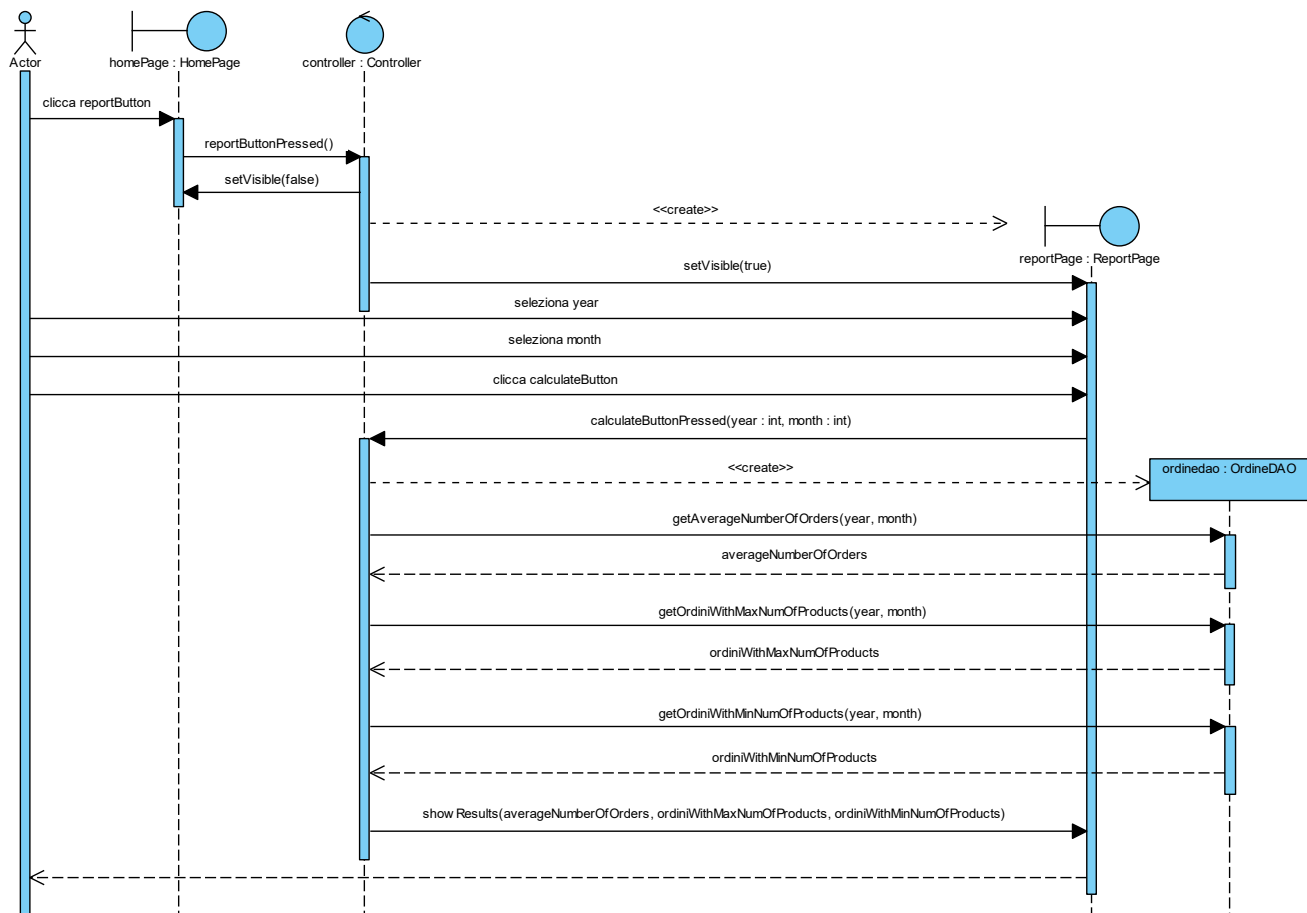


3.4 Esempi di Sequence Diagram

Sono mostrati di seguito due Sequence Diagram UML, atti a mostrare un possibile percorso effettuato dall'utente nell'utilizzo delle due funzionalità principali dell'applicativo. Essendo i Sequence Diagram uno strumento per mostrare lo scheletro del funzionamento di un'operazione senza perdersi fra le righe di codice, tecnicismi sono omessi in favore di un rendimento più astratto del procedimento, ad esempio l'utilizzo di "Show Errors" in sostituzione dei metodi specifici di Swing per mostrare all'utente un errore che impedisce il continuo di un'azione.

Si omettono inoltre, per leggibilità, i controlli più banali di correttezza degli input, ad esempio i controlli che si accertano che l'orario di arrivo della spedizione appena creata sia maggiore dell'orario di partenza, presenti invece nel codice.

3.4.1 | Visualizzazione del report statistico



Creazione di una nuova spedizione

