

Progetto di Ingegneria del Software

Prototipo di un'applicazione per la gestione di un magazzino.

A cura di
Enrico Pasquali, Alessio Zampieri

Indice

1	Specifiche	2
2	Introduzione	2
3	Requisiti utente	3
3.1	Usecase Diagram	3
3.1.1	Personale	4
3.1.2	Segreteria amministrativa	4
3.1.3	Responsabile negozio	5
3.1.4	Magazzinieri	6
3.2	Sequence diagram	7
4	Requisiti di sistema	8
4.1	Class Diagram	8
4.1.1	Controller	8
4.1.2	Model	9
4.1.3	View	10
4.2	Sequence diagram delle classi	11
4.2.1	Segreteria amministrativa	11
4.2.2	Responsabile negozio	11
4.2.3	Magazziniere	11
5	Modelli di sistema	12
5.1	Activity diagram	12
5.1.1	Segreteria amministrativa	12
5.1.2	Responsabile negozio	12
5.1.3	Magazziniere	13
6	Progettazione e sviluppo	13
6.1	I Pattern usati	13
7	Struttura del database	13
8	Testing	13

1 Specifiche

Si vuole progettare un sistema informatico per gestire il magazzino di una catena di negozi di articoli sportivi.

Il negozio vende articoli di diversa tipologia, raggruppati per sport. Per ogni tipo articolo si registra: un nome univoco, una descrizione, lo sport, e i materiali utilizzati per produrlo.

Il sistema registra tutti gli articoli in magazzino memorizzando per ogni articolo: il tipo di articolo, un codice univoco, il prezzo e la data di produzione.

Gli articoli in magazzino vengono gestiti dal sistema che registra per ogni ingresso in magazzino: un codice interno univoco, la data e tutti articoli entrati e le loro posizioni in magazzino.

Per ogni uscita il sistema registra: la data e il numero di bolla (univoco), tutti gli articoli usciti, il negozio che li ha ordinati e lo spedizioniere che li ritira.

Per ogni negozio della catena il sistema registra: il codice fiscale, il nome, l'indirizzo e la città.

Il sistema memorizza inoltre gli ordini dei negozi registrando: il negozio che ha effettuato l'ordine, un codice ordine univoco, la data dell'ordine, i tipi di articolo ordinati e per ogni tipo di articolo la quantità ordinata e il prezzo totale.

Quando un ordine viene evaso si registra un'uscita dal magazzino che viene collegata all'ordine al quale si riferisce. Si suppone che per ogni ordine evaso si abbia una sola uscita dal magazzino.

Per ogni tipo di articolo il sistema memorizza esplicitamente alla fine di ogni mese dell'anno la quantità di articoli ricevuti in magazzino e la quantità di articoli usciti.

Il sistema deve permettere ai magazzinieri di inserire le informazioni relative ai movimenti di ingresso e uscita dal magazzino. I magazzinieri, inoltre, possono spostare un articolo da una posizione ad un'altra del magazzino, al fine di ottimizzare l'occupazione del magazzino.

La segreteria amministrativa della catena di negozi è responsabile dell'inserimento dei tipi di articolo. Essa può accedere al sistema e visualizzare i movimenti di magazzino rispetto agli ordini dei vari negozi. Tutti gli utenti sono opportunamente autenticati dal sistema, prima che possano accedere alle funzionalità specifiche.

I responsabili dei negozi possono accedere al sistema per effettuare gli ordini e per avere un riassunto degli ordini passati.

2 Introduzione

Si vuole progettare una base di dati per gestire le informazioni relative alla gestione di un magazzino di una catena di negozi di articoli sportivi. Gli attori principali del sistema sono:

- La segreteria amministrativa
- I responsabili di ciascun negozio della catena
- I magazzinieri

Il seguente documento parte quindi da una descrizione ad un alto livello d'astrazione dei concetti fondamentali del sistema, attraverso la presentazione di use case e sequence diagram, per poi entrare nel dettaglio con lo schema delle classi e la descrizione dei principali pattern usati. Verrà anche poi presentato lo schema ER (entità-relazione) e lo schema fisico della base di dati creata per la gestione del sistema. L'applicazione prevede l'autenticazione in modo elementare di tutti gli utenti che interagiscono con il sistema. Seguiranno infine alcuni esempi di utilizzo dell'applicazione.

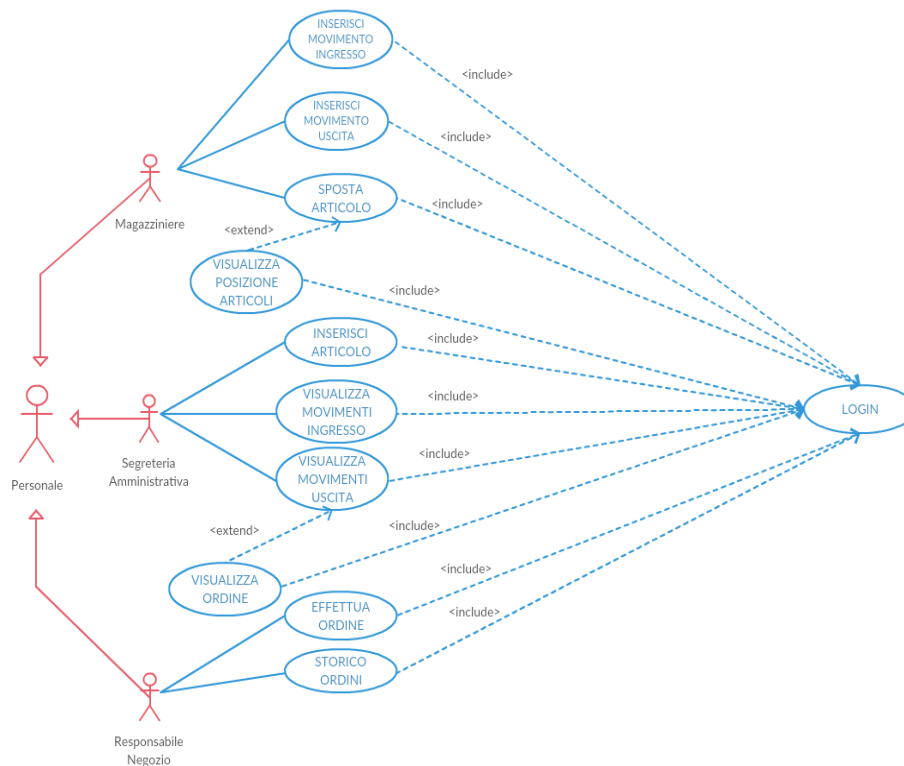
3 Requisiti utente

Vengono evidenziati coloro che sono coinvolti nel progetto e nel successivo utilizzo del prototipo.

STAKEHOLDERS: Segreteria amministrativa, magazzinieri, responsabili dei negozi, progettisti del sistema.

3.1 Usecase Diagram

Sono presentati di seguito il diagramma dei casi d'uso e le relative schede.



3.1.1 Personale

CASO D'USO: LOGIN
ID: LOG
ATTORI: PERSONALE
PRECONDIZIONI: 1. Il personale vuole accedere al sistema.
SEQUENZA: 1. Il caso d'uso inizia quando il personale inserisce username e password a. Il sistema controlla se i dati inseriti sono corretti i. Se non sono corretti, il sistema avvisa tramite un messaggio d'errore ii. Se sono corretti, il sistema mostra la schermata dedicata al personale autenticato
POSTCONDIZIONI: 1. Viene mostrata la schermata dedicata al personale autenticato

3.1.2 Segreteria amministrativa

CASO D'USO: INSERIMENTO TIPO ARTICOLO
ID: NEW_ART
ATTORI: SEGRETERIA AMMINISTRATIVA
PRECONDIZIONI: 1. La segreteria amministrativa deve inserire un tipo di articolo nel database
SEQUENZA: 1. Il caso d'uso inizia quando la segreteria seleziona "Inserisci nuovo articolo". 2. Il sistema chiede il nome univoco da utilizzare per il nuovo tipo articolo a. Se il nome esiste già, il sistema mostra un messaggio di errore e la segreteria deve inserirne un altro 3. Il sistema chiede di inserire una descrizione per il tipo articolo 4. Il sistema chiede di inserire il tipo di sport per l'articolo 5. Il sistema chiede di inserire i materiali utilizzati per produrre l'articolo 6. Il sistema conferma l'avvenuta aggiunta del nuovo tipo articolo
POSTCONDIZIONI 1. Il database è aggiornato con il nuovo tipo articolo al suo interno

CASO D'USO: VISUALIZZA MOVIMENTI INGRESSO
ID: MOVE_IN_LIST
ATTORI: SEGRETERIA AMMINISTRATIVA
PRECONDIZIONI: 1. La segreteria amministrativa vuole visualizzare i movimenti in ingresso del magazzino
SEQUENZA: 1. Il caso d'uso inizia quando la segreteria amministrativa seleziona "Visualizza movimenti in ingresso" 2. Il sistema mostra l'elenco degli ingressi del magazzino
POSTCONDIZIONI: 1. La segreteria è a conoscenza dei movimenti in ingresso del magazzino

CASO D'USO: VISUALIZZA MOVIMENTI USCITA
ID: MOVE_OUT_LIST
ATTORI: SEGRETERIA AMMINISTRATIVA
PRECONDIZIONI: <ol style="list-style-type: none"> 1. La segreteria amministrativa vuole visualizzare i movimenti in uscita del magazzino
SEQUENZA: <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando la segreteria amministrativa seleziona "Visualizza movimenti in uscita" 2. Il sistema mostra l'elenco delle uscite del magazzino
POSTCONDIZIONI: <ol style="list-style-type: none"> 1. La segreteria è a conoscenza dei movimenti in uscita del magazzino

3.1.3 Responsabile negozio

CASO D'USO: EFFETTUA ORDINE
ID: ORDINA
ATTORI: RESPONSABILE NEGOZIO
PRECONDIZIONI: <ol style="list-style-type: none"> 1. Il responsabile di un negozio della catena necessita di effettuare un ordine
SEQUENZA: <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il responsabile del negozio seleziona "Nuovo ordine" 2. Il sistema chiede all'utente di inserire il nome dell'articolo da inserire nel carrello 3. Il sistema chiede all'utente di inserire la quantità desiderata per ogni tipologia di articolo nel carrello 4. Il sistema mostra il riepilogo dell'ordine assieme al prezzo totale 5. Il sistema chiede conferma all'utente <ol style="list-style-type: none"> a. Se l'utente seleziona "Conferma", il sistema mostrerà l'avviso "Ordine confermato" b. Se l'utente seleziona "Modifica ordine", il sistema mostrerà la schermata di riepilogo
POSTCONDIZIONI: <ol style="list-style-type: none"> 1. L'ordine viene inviato al magazzino

CASO D'USO: STORICO ORDINI
ID: STORICO
ATTORI: RESPONSABILE NEGOZIO
PRECONDIZIONI: <ol style="list-style-type: none"> 1. Il responsabile del negozio vuole visualizzare l'elenco degli ordini passati
SEQUENZA: <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il responsabile del negozio seleziona "Storico ordini" 2. Il sistema mostra l'elenco degli ordini passati
POSTCONDIZIONI: <ol style="list-style-type: none"> 1. Il responsabile negozio è a conoscenza degli ordini passati

3.1.4 Magazzinieri

CASO D'USO: SPOSTA ARTICOLO
ID: MOVE
ATTORI: MAGAZZINIERE
PRECONDIZIONI: <ol style="list-style-type: none"> 1. Il magazziniere vuole modificare la posizione di un articolo nel magazzino
SEQUENZA: <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il magazziniere seleziona "Sposta articolo" 2. Il sistema chiede il codice dell'articolo da spostare <ol style="list-style-type: none"> a. Se il codice non è nel database, il sistema mostra un messaggio di errore 3. Il sistema chiede di scegliere tra le posizioni libere in magazzino 4. Il magazziniere seleziona una posizione libera 5. Include "VISUALIZZA POSIZIONE ARTICOLI"
POSTCONDIZIONI: <ol style="list-style-type: none"> 1. Il database è aggiornato con la nuova posizione dell'articolo

CASO D'USO: INSERISCI MOVIMENTO INGRESSO
ID: NEW_MOVE_IN
ATTORI: MAGAZZINIERE
PRECONDIZIONI: <ol style="list-style-type: none"> 1. Il magazziniere deve registrare un nuovo ingresso in magazzino
SEQUENZA <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il magazziniere seleziona "Nuovo ingresso" 2. Il sistema chiede di identificare il nuovo ingresso tramite un codice univoco <ol style="list-style-type: none"> a. Il magazziniere inserisce un codice identificativo univoco <ol style="list-style-type: none"> i. Se esiste già, il sistema mostra un messaggio di errore 3. Il sistema chiede il codice di ogni articolo presente nell'ordine <ol style="list-style-type: none"> a. Il magazziniere inserisce il codice di ogni articolo dell'ordine <ol style="list-style-type: none"> i. Se il codice non è corretto, il sistema mostra un messaggio di errore b. Per ogni articolo, il sistema chiede di inserirne la posizione in magazzino 4. Il sistema chiede al magazziniere di inserire la data <ol style="list-style-type: none"> a. Il magazziniere inserisce una data tramite menu a tendina
POSTCONDIZIONI: <ol style="list-style-type: none"> 1. Il database viene aggiornato con un nuovo ingresso in magazzino

CASO D'USO: INSERISCI MOVIMENTO USCITA
ID: NEW_MOVE_OUT
ATTORI: MAGAZZINIERE
PRECONDIZIONI: <ol style="list-style-type: none"> 1. Il magazziniere deve registrare una nuova uscita in magazzino
SEQUENZA <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando il magazziniere seleziona "Nuova uscita" 2. Il sistema chiede di identificare la nuova uscita tramite il numero di bolla 3. Il sistema chiede il codice di ogni articolo presente nell'ordine <ol style="list-style-type: none"> a. Il magazziniere inserisce il codice di ogni articolo dell'ordine <ol style="list-style-type: none"> i. Se il codice non è corretto, il sistema mostra un messaggio di errore 4. Il sistema chiede di inserire il nome del negozio che ha effettuato l'ordine <ol style="list-style-type: none"> a. Il magazziniere sceglie un negozio dal menu a tendina 5. Il sistema chiede di inserire la data <ol style="list-style-type: none"> a. Il magazziniere inserisce la data tramite menu a tendina 6. Il sistema chiede di inserire lo spedizioniere <ol style="list-style-type: none"> a. Il magazziniere sceglie lo spedizioniere tramite menu a tendina
POSTCONDIZIONI: <ol style="list-style-type: none"> 1. Il database viene aggiornato con una nuova uscita dal magazzino

3.2 Sequence diagram

I *sequence diagram* dettagliano la sequenza temporale degli eventi che non è possibile rappresentare con gli *use case*.

4 Requisiti di sistema

4.1 Class Diagram

Presentati di seguito i *class diagram* sviluppati per il software.

In un primo momento, viene visualizzata la struttura generale dove sono evidenziate le relazioni presenti tra le classi. Successivamente si entrerà nel dettaglio di ciascuna.

Il progetto è suddiviso in tre package:

- **controller**, il quale contiene le classi relative al controllo dei dati.
- **model**, il quale contiene le classi per la gestione e recupero dati interfacciandosi con la base di dati.
- **view**, il quale implementa la parte di visualizzazione grafica.

I tre package comunicano quindi seguendo il pattern del *Model-View-Controller* (**MVC**).

In figura è mostrata la struttura generale dove: il controller risulta essere in relazione con il model, ricevendo gli eventi dalla view e coordinando le operazioni necessarie.

4.1.1 Controller

Ogni attore è quindi gestito nelle sue funzionalità da un Controller distinto.

4.1.2 Model

L'implementazione del *Model* invece risulta abbastanza semplice, in quanto la parte importante di gestione e memorizzazione dei dati è concentrata nella *Base di Dati*.

Ogni classe esegue le query a cui fa riferimento per poi incapsulare i dati letti e passarli al controller, view e model corrispondente in modo più agevole.

4.1.3 View

Il package *View* contiene le classi che si occupano della realizzazione dell'interfaccia grafica.

4.2 Sequence diagram delle classi

Qui i *Sequence Diagram* che descrivono a più basso livello il comportamento delle classi.

4.2.1 Segreteria amministrativa

4.2.2 Responsabile negozio

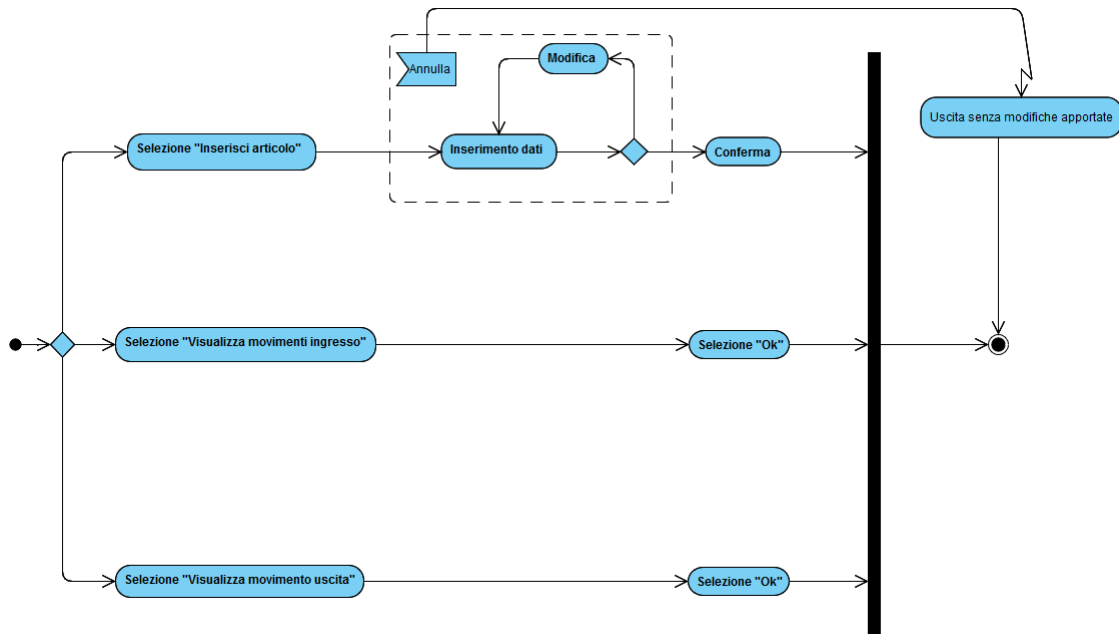
4.2.3 Magazziniere

5 Modelli di sistema

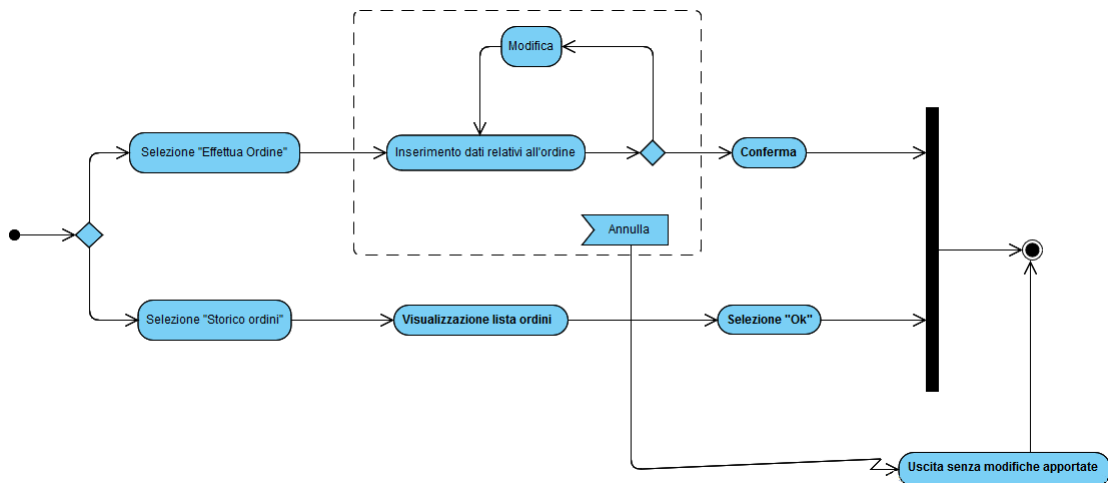
5.1 Activity diagram

Qui i modelli di sistema, ovvero, gli *activity diagram*.

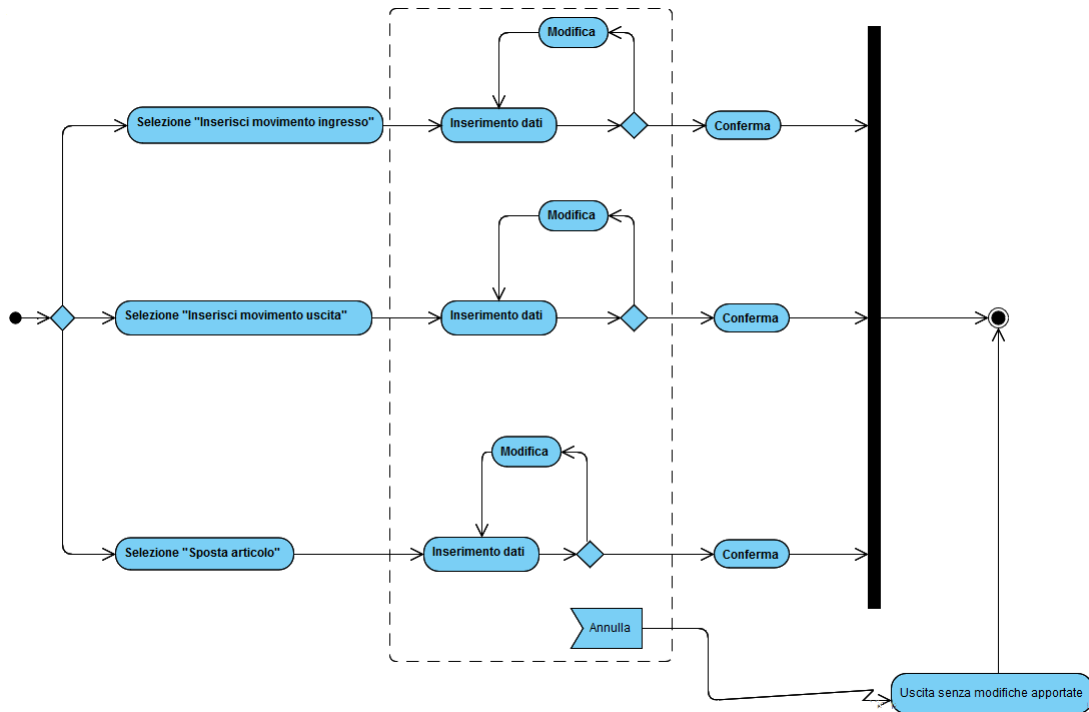
5.1.1 Segreteria amministrativa



5.1.2 Responsabile negozio



5.1.3 Magazziniere



6 Progettazione e sviluppo

6.1 I Pattern usati

7 Struttura del database

SCHEMA CONCETTUALE

SCHEMA RELAZIONALE

I vincoli d'integrità sono: (*DAMODIFICARE*)

Locale.proprietario → Personale.username

Comande.cameriera → Personale.username

Comande.tavolo → Tavoli.numTavolo

Comande.piatto → Listino.piatto

Ricette.piatto → Listino.piatto

Ricette.ingrediente → magazzino.nomeIngrediente

8 Testing