

Progetto di Ingegneria del Software

Sistema informatico per la gestione del magazzino di una catena di negozi di articoli sportivi

17 luglio 2018



UNIVERSITÀ di VERONA

A.A. 2017/2018

CdL in Bioinformatica

Docenti:

Combi Carlo

Oliboni Barbara

Posenato Roberto

Studenti:

Douglas Enrico Elio

Gozzi Federico

Treccani Mirko

INDICE

| | |
|--|-----------|
| Introduzione | 2 |
| <i>Testo elaborato</i> | 2 |
| <i>Obiettivi e idee progettuali</i> | 2 |
| <i>Requisiti funzionali</i> | 3 |
| <i>Database</i> | 4 |
| Use Cases | 5 |
| <i>Use case 1 – Responsabile</i> | 5 |
| <i>Use case 2 – Segreteria</i> | 6 |
| <i>Use case 3 – Magazziniere</i> | 7 |
| Activity diagrams | 8 |
| <i>Activity diagram 1 – Responsabile</i> | 8 |
| <i>Activity diagram 2 – Segreteria</i> | 9 |
| <i>Activity diagram 3 – Magazziniere</i> | 10 |
| Sequence diagrams | 11 |
| <i>Sequence diagram 1 – Responsabile</i> | 11 |
| <i>Sequence diagram 2 – Segreteria</i> | 13 |
| <i>Sequence diagram 3 – Magazziniere</i> | 15 |
| Class diagrams | 17 |
| <i>Class diagram generale</i> | 17 |
| Testing e validazione | 18 |

INTRODUZIONE

Testo dell'elaborato

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.

Obiettivi e idee progettuali

L'obiettivo è la progettazione di un software per la gestione di un magazzino e dei relativi ordini per il rifornimento di una catena di negozi di articoli sportivi.

Il sistema è ideato per l'accesso di utenti aventi uno dei tre ruoli indicati di seguito: il responsabile negozio (indicato dal codice 1), il segretario (indicato con codice 2) e il magazziniere (indicato con codice 3). Ogni ruolo permette l'esecuzione di determinate mansioni e quindi l'accesso a diverse aree del software.

Il sistema fa riferimento a un database inizializzato in PostgreSQL composto da sette tabelle. L'interfaccia del software è stata implementata in Java, utilizzando le relative librerie Swing.

In seguito all'estrazione delle specifiche indicate dal testo, abbiamo scelto uno sviluppo ibrido del software, ovvero unendo la modalità plan-driven alla modalità agile. Abbiamo infatti

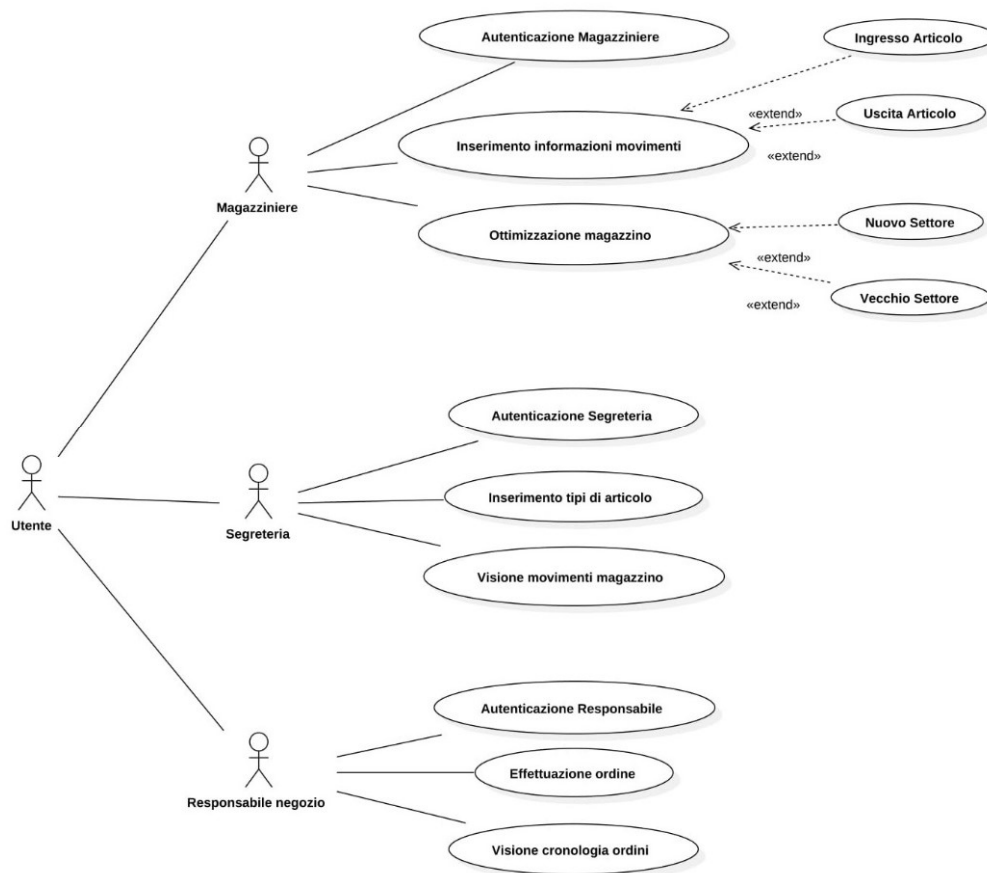
pianificato le attività all’inizio del lavoro, ma abbiamo modificato il processo durante la progettazione stessa del software. Il processo plan-driven è stato più utilizzato per la compilazione dei casi d’uso e dei diagrammi mentre l’approccio agile è stato più utilizzato per quanto riguarda la parte di implementazione del codice.

Requisiti funzionali

Il software consente di gestire il magazzino e i relativi ordini di entrata e uscita verso negozi di una catena di articoli sportivi.

Le funzioni del software sono distinte in base al ruolo di cui gli utenti del negozio sono in possesso. Ogni utente può utilizzare il software solamente in seguito a login.

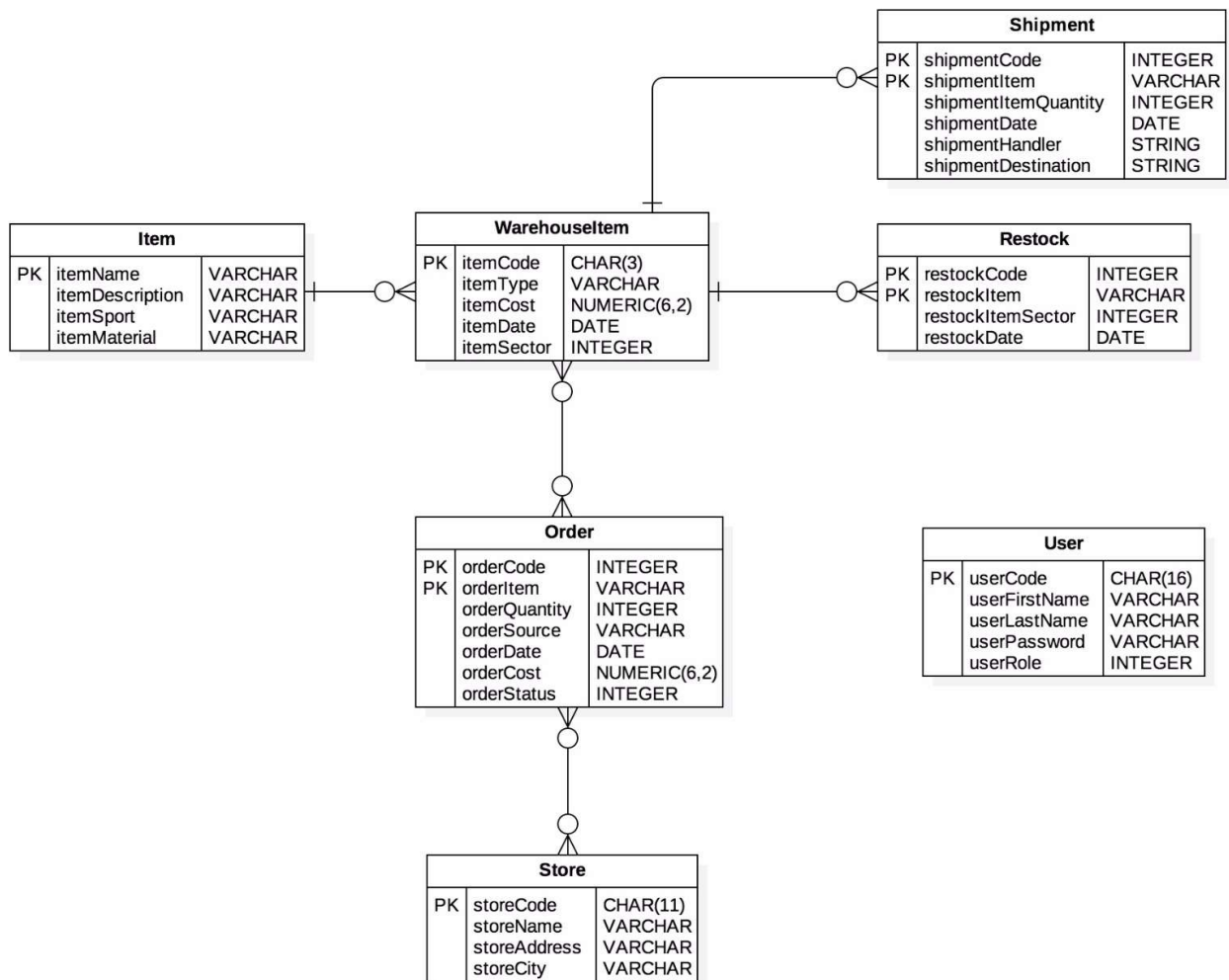
- per il responsabile del negozio (codice: 1), solamente in seguito ad autenticazione, è possibile effettuare gli ordini e visionare la cronologia degli ordini precedentemente effettuati.
- per gli utenti della segreteria (codice: 2), solamente in seguito ad autenticazione, è possibile inserire i possibili tipi di articolo, con le relative informazioni, che si possono trovare nel magazzino e visionare i movimenti sia in entrata che in uscita del magazzino.
- per i magazzinieri (codice: 3), solamente in seguito ad autenticazione, è possibile inserire informazioni riguardanti i movimenti sia in ingresso che in uscita dal magazzino per le varie tipologie di articolo e ottimizzare il magazzino in termini di spazio, aggiornando la posizione di un articolo in un dato settore.



Database

Il software si appoggia su un database implementato mediante lo script *schematabelle.sql*, scritto il linguaggio SQL. Tale script definisce il codice per la costruzione delle tabelle nel database.

In seguito alla costruzione delle tabelle nel database, le abbiamo popolate separatamente, mediante il software PostgreSQL. In questo modo, ci assicuriamo di avere un database di partenza corretto e completo, evitando eventuali errori nell'inserimento dei dati che potrebbero produrre errori durante l'esecuzione di parti di codice.



USE CASES

Use case 1 – Responsabile

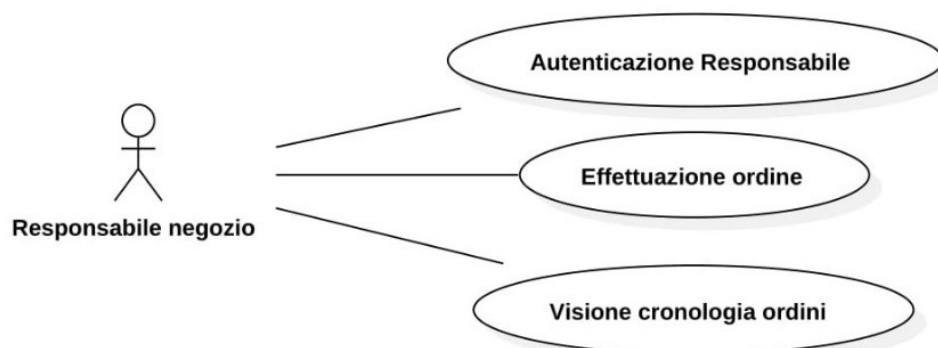
Il primo caso d'uso analizzato è quello relativo agli utenti aventi ruolo di responsabile di negozio.

Per poter utilizzare il software, l'utente deve effettuare l'autenticazione. In base alle sue credenziali, l'utente è riconosciuto come responsabile del negozio e identificato con il codice 1.

Una volta autenticato, il responsabile di negozio può eseguire due azioni: effettuare un ordine oppure prendere visione della cronologia degli ordini passati.

| | |
|-----------------|--|
| CASO D'USO | Effettuazione dell'ordine |
| NUMERO | 1A |
| ATTORE | Responsabile del negozio |
| PRE-CONDIZIONI | L'utente fa login come responsabile |
| AZIONI | 1) Autenticazione 2) Effettuazione di un ordine 3) Visione della cronologia degli ordini |
| POST-CONDIZIONI | Viene inoltrato l'ordine al magazzino, tramite la memorizzazione nel database |

| | |
|-----------------|---|
| CASO D'USO | Visualizzazione della cronologia |
| NUMERO | 1B |
| ATTORE | Responsabile del negozio |
| PRE-CONDIZIONI | L'utente fa login come responsabile |
| AZIONI | 1) Autenticazione 2) Effettuazione di un ordine 3) Visione della cronologia degli ordini |
| POST-CONDIZIONI | Viene mostrata la cronologia degli ordini precedenti, mediante le informazioni contenute nel database |



Use case 2 – Segreteria

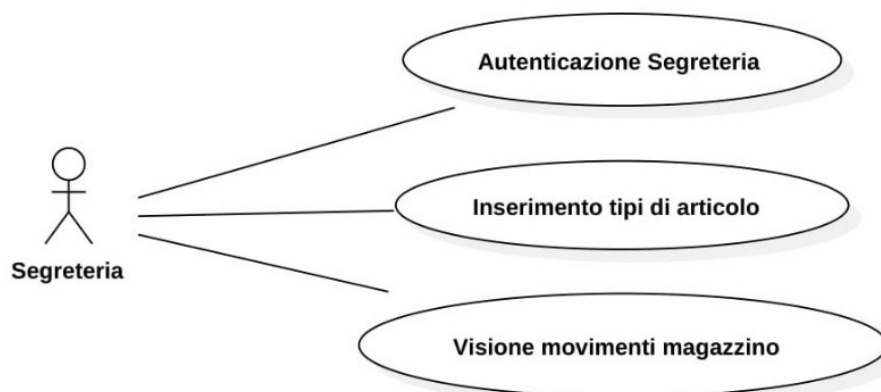
Il secondo caso d'uso analizzato è quello relativo agli utenti aventi ruolo all'interno della segreteria, e quindi presenti in qualità di segretari.

In seguito ad autenticazione possono utilizzare il software per le funzioni a loro dedicate. Vengono infatti riconosciuti come segretari e identificati dal codice 2.

Effettuato l'accesso, la segreteria può effettuare due azioni: inserire i diversi tipi di articolo che si possono trovare in magazzino oppure visionare tutti i movimenti effettuati dal magazzino, sia in entrata che in uscita.

| | |
|-----------------|--|
| CASO D'USO | Inserimento tipo di articolo |
| NUMERO | 2 |
| ATTORE | Segreteria |
| PRE-CONDIZIONI | L'utente fa login come segreteria |
| AZIONI | 1) Autenticazione 2) Inserimento tipo di articolo 3) Visione dei movimenti del magazzino |
| POST-CONDIZIONI | Il tipo di articolo viene memorizzato nel database |

| | |
|-----------------|--|
| CASO D'USO | Visualizzazione dei movimenti del magazzino |
| NUMERO | 2 |
| ATTORE | Segreteria |
| PRE-CONDIZIONI | L'utente fa login come segreteria |
| AZIONI | 1) Autenticazione 2) Inserimento tipo di articolo 3) Visione dei movimenti del magazzino |
| POST-CONDIZIONI | Viene mostrato all'utente l'insieme dei movimenti del magazzino, memorizzati nel database |



Use case 3 – Magazziniere

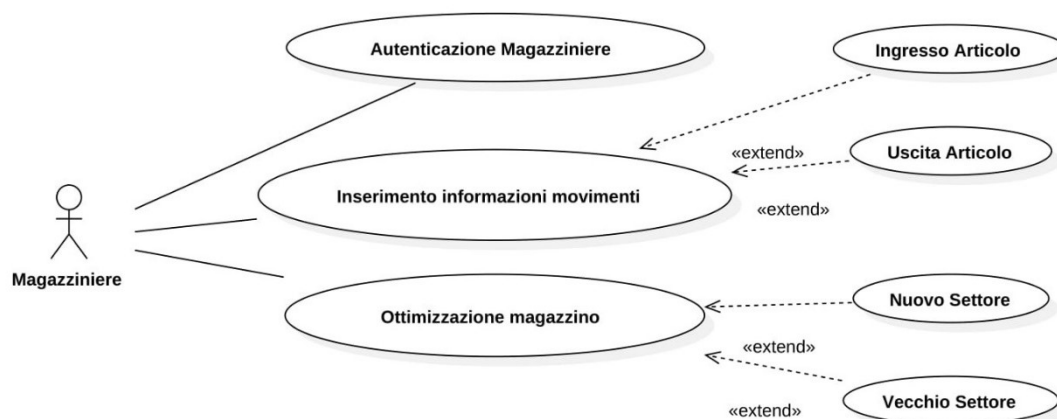
Infine, il terzo caso d'uso preso in considerazione è quello relativo agli utenti aventi ruolo di magazziniere.

Solamente in seguito ad autenticazione, il software riconosce tali utenti come addetti al magazzino e li identifica con il codice 3.

Quindi, i magazzinieri possono effettuare le seguenti azioni: inserire informazioni circa i movimenti in ingresso e in uscita di determinati articoli, oppure ottimizzare gli spazi all'interno del magazzino, spostando gli articoli in un nuovo settore e quindi modificare i vecchi settori di appartenenza.

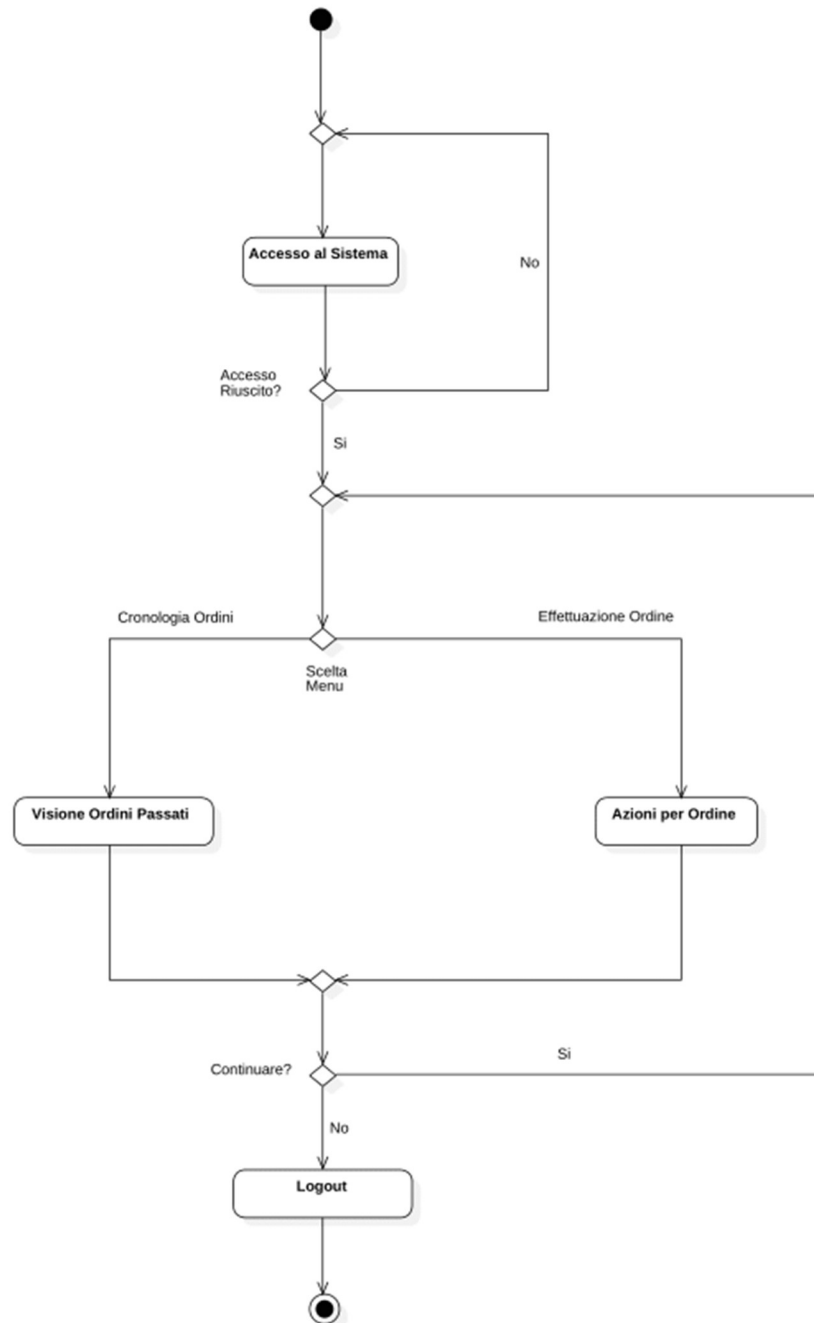
| | |
|-----------------|---|
| CASO D'USO | Inserimento dei movimenti |
| NUMERO | 3 |
| ATTORE | Magazziniere |
| PRE-CONDIZIONI | L'utente fa login come magazziniere |
| AZIONI | 1) Autenticazione 2) Inserimento movimenti 3) Ottimizzazione del magazzino |
| POST-CONDIZIONI | Se il movimento è di ingresso, sono memorizzati gli articoli inseriti. Se il movimento è di uscita, viene evaso l'ordine memorizzato. |

| | |
|-----------------|--|
| CASO D'USO | Ottimizzazione del magazzino |
| NUMERO | 3 |
| ATTORE | Magazziniere |
| PRE-CONDIZIONI | L'utente fa login come segreteria |
| AZIONI | 1) Autenticazione 2) Inserimento movimenti 3) Ottimizzazione del magazzino |
| POST-CONDIZIONI | L'articolo scelto viene memorizzato nella nuova posizione inserita |

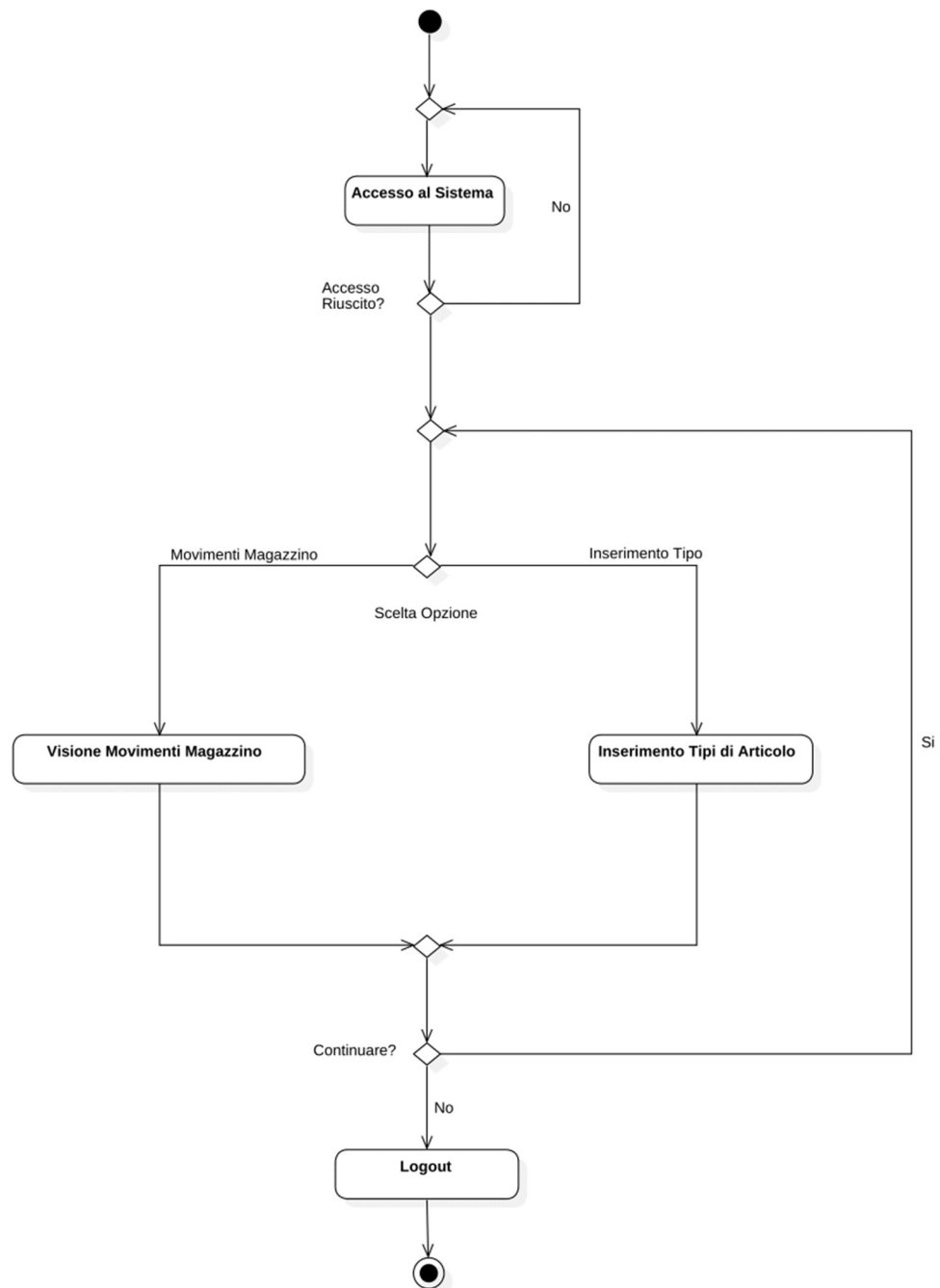


ACTIVITY DIAGRAMS

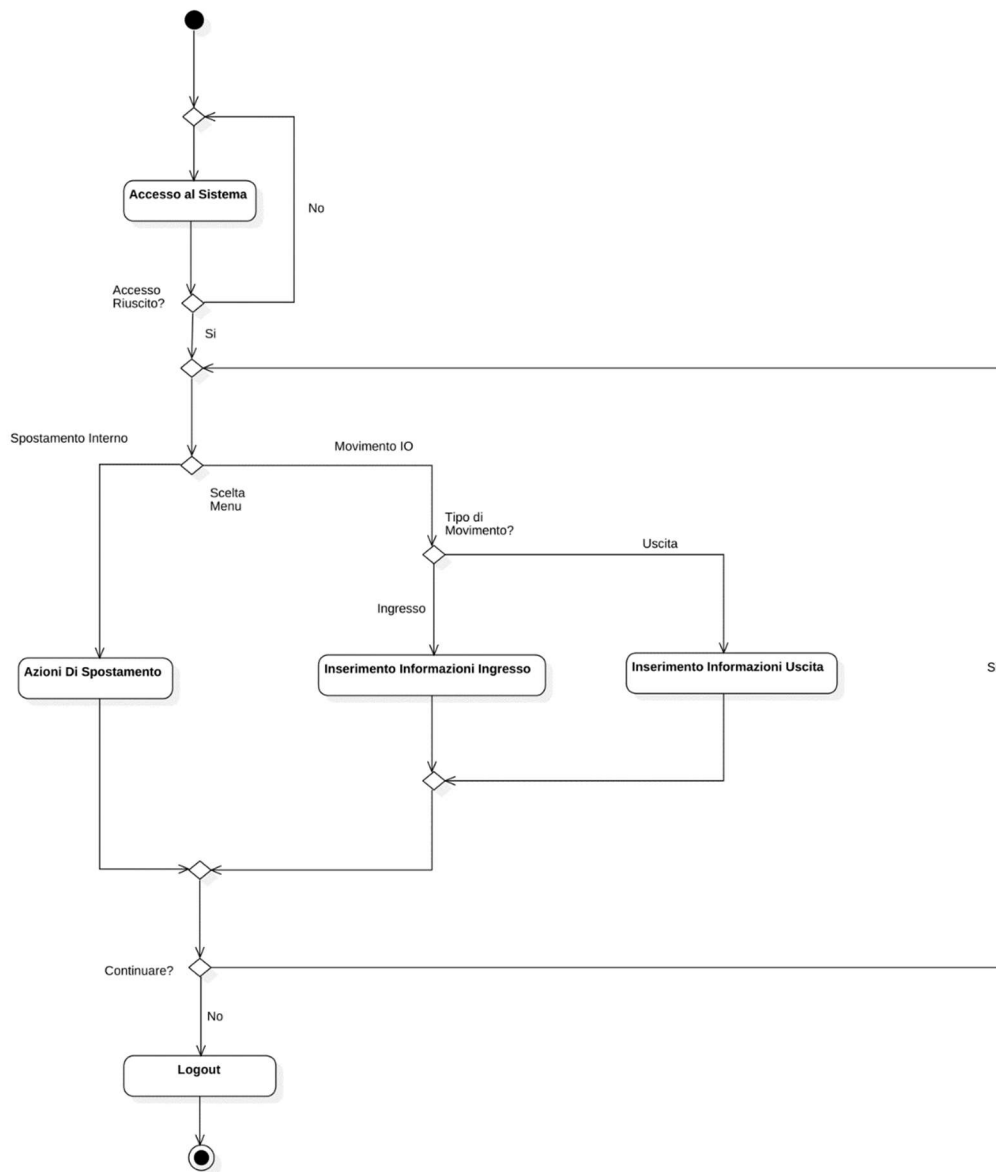
Activity diagram 1 – Responsabile



Activity diagram 2 – Segreteria

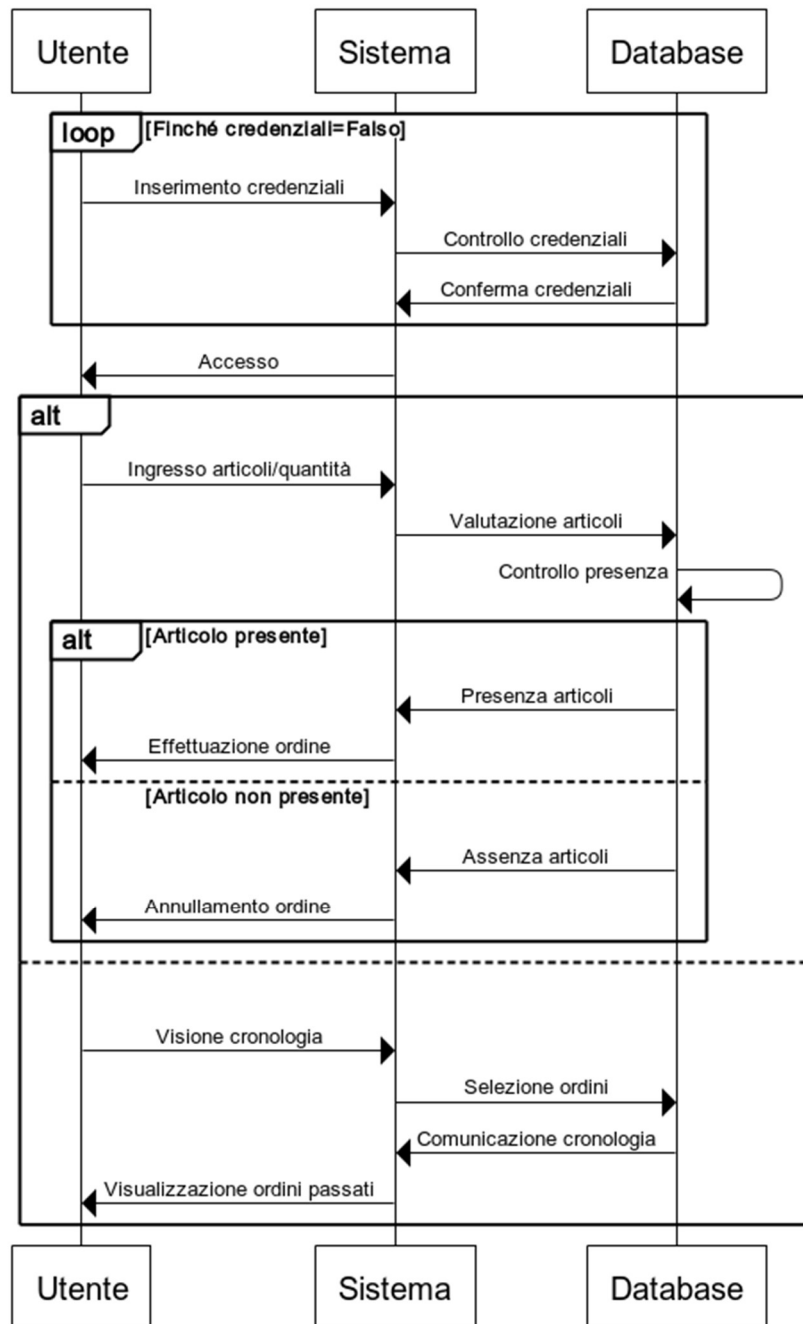


Activity diagram 3 – Magazziniere



SEQUENCE DIAGRAMS

Sequence diagram 1 – Responsabile



Il diagramma di sequenza del responsabile mostra come questo utente si interfacci al sistema, e quindi al database che ne è la fonte di informazione, per poter svolgere le mansioni a lui assegnate.

Prima di qualsiasi attività, il responsabile deve effettuare la procedura di log-in, inserendo le proprie credenziali. La verifica delle credenziali passa quindi dal sistema al database, il quale paragona le credenziali in memoria con quelle inserite. Se le credenziali sono errate, il sistema

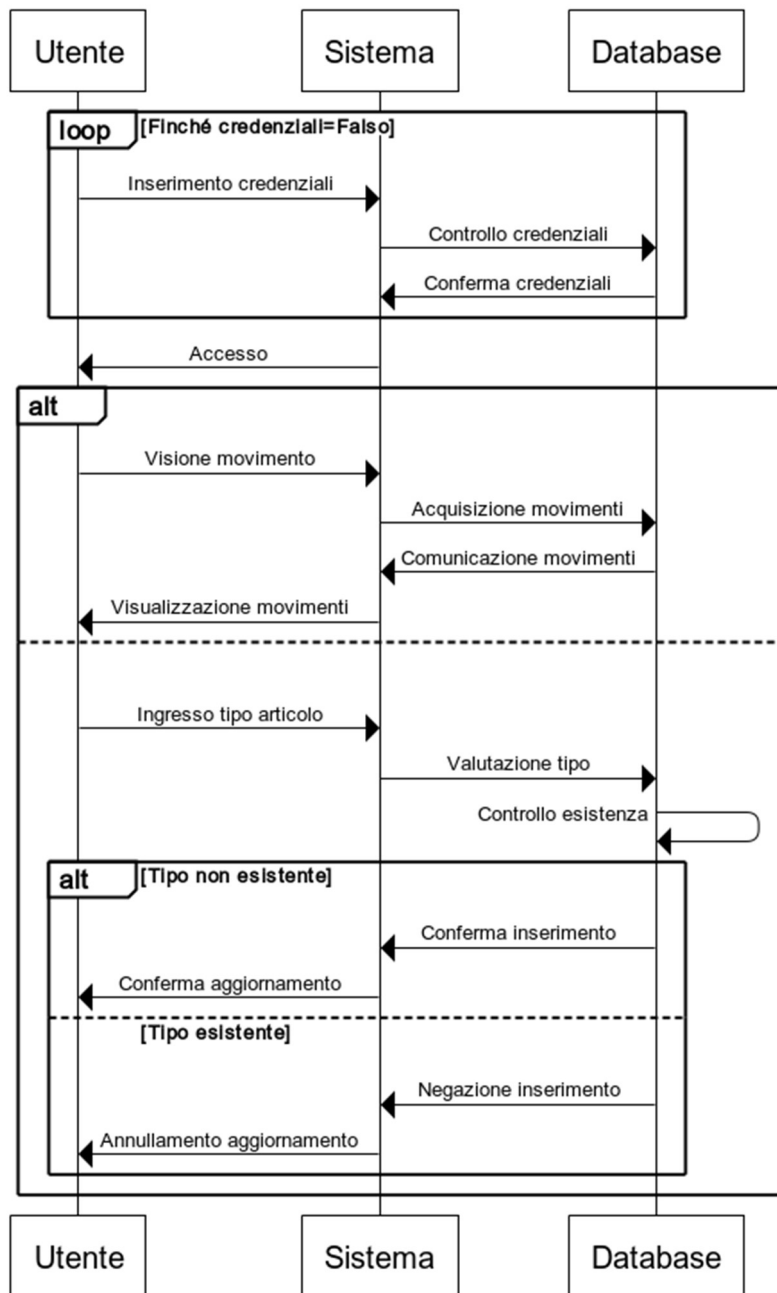
richiede all'utente di inserire nuove credenziali, come mostrato dal frammento contenete la condizione di loop; altrimenti l'utente effettua l'accesso in qualità di responsabile.

Successivamente, il responsabile può eseguire due azioni differenti, inserite all'interno del frammento contenente la condizione alt.

Se decide di effettuare un ordine, il responsabile inserisce nel sistema gli articoli con le relative quantità. Il sistema valuta quindi gli articoli nel database, il quale ne controlla la presenza, in base al quantitativo richiesto. Se questi articoli sono presenti, il database ne comunica la presenza al sistema, che quindi permette l'effettuazione dell'ordine. Altrimenti, se gli articoli richiesti non sono presenti, il database comunica l'assenza al server, il quale annulla l'ordine.

Invece, se il responsabile decide di prendere visione della cronologia degli ordini, lo richiede al sistema. Il sistema seleziona quindi gli ordini nel database, il quale restituisce gli ordini relativi al responsabile che ne ha fatto richiesta. Infine, il sistema mostra al responsabile gli ordini che egli ha effettuato in passato.

Sequence diagram 2 – Segreteria



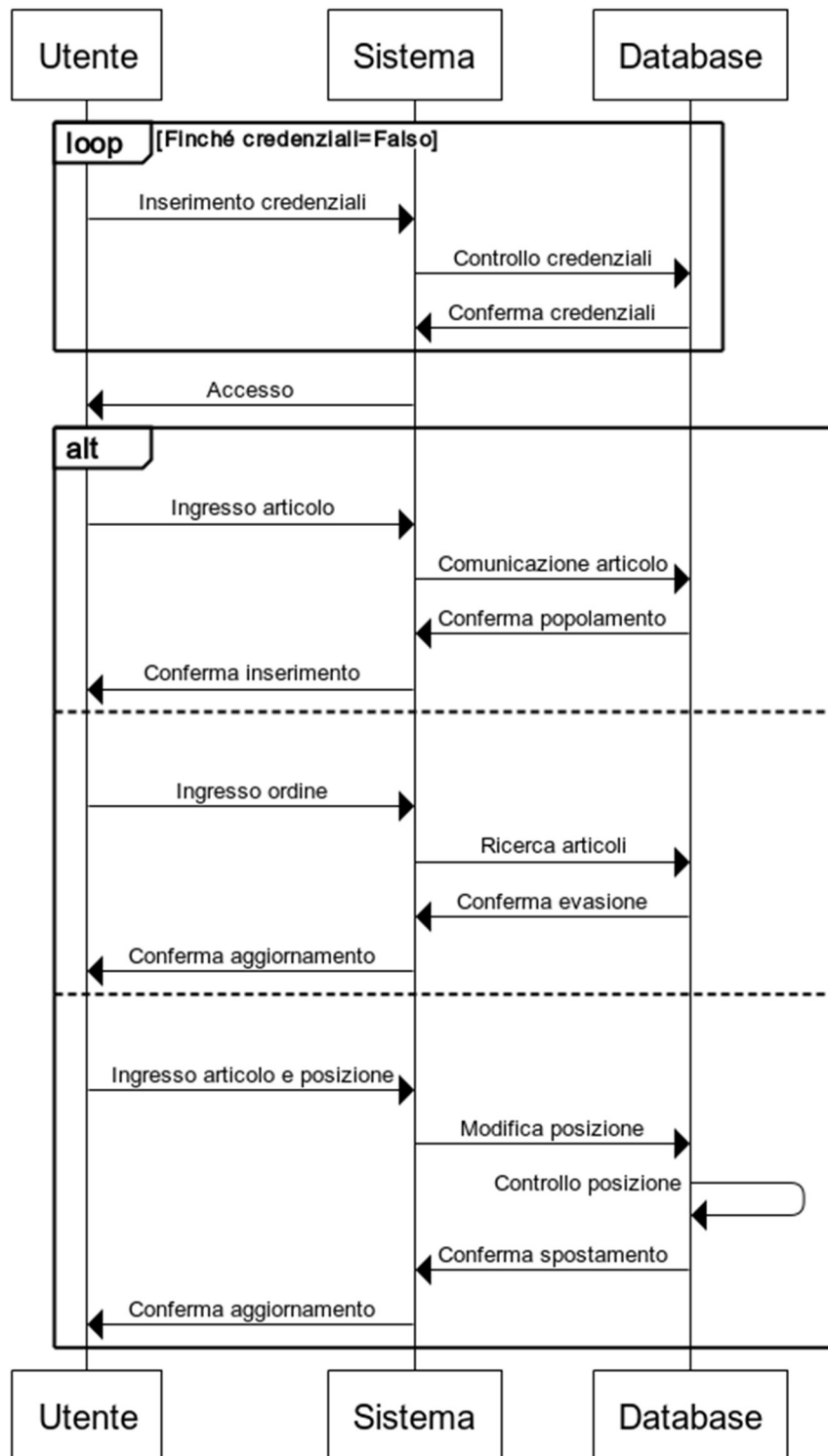
Il diagramma di sequenza della segreteria mostra la comunicazione tra utenti, sistema e database per gli utenti registrati come segretari.

In modo analogo a quanto accade per il responsabile, gli utenti aventi mansioni di segreteria devono prima effettuare la procedura di log-in. Se questa va a buon fine, possono procedere a effettuare le azioni a loro consentite, nell'apposito frammento con condizione alt.

Se la segreteria necessita di visualizzare i movimenti effettuati, ne fa richiesta al sistema. Il sistema quindi richiede i dati al database, il quale comunica i movimenti al sistema; quest'ultimo, infine, mostra i movimenti effettuati alla segreteria.

Invece, se deve inserire un nuovo tipo di articolo, la segreteria comunica al sistema il tipo di articolo. Il sistema rivolge quindi la richiesta al database, il quale controlla l'esistenza del tipo di articolo, tra quelli salvati. Se l'articolo non esiste, il database inserisce il nuovo tipo di articolo e ne viene data comunicazione al sistema, il quale comunica all'utente l'aggiornamento dei tipi. Altrimenti, il database non permette l'inserimento del nuovo tipo, dando risposta negativa al sistema e, quindi, all'utente.

Sequence diagram 3 – Magazziniere



Il diagramma di sequenza dei magazzinieri mostra la comunicazione tra gli utenti aventi tale ruolo, il sistema e il database contenente i dati.

Come per i precedenti due casi, anche gli utenti con ruolo di magazziniere devono eseguire la procedura di log-in; una volta inserite le credenziali corrette, i magazzinieri possono quindi effettuare le azioni di loro competenza, indicate all'interno del frammento con condizione alt.

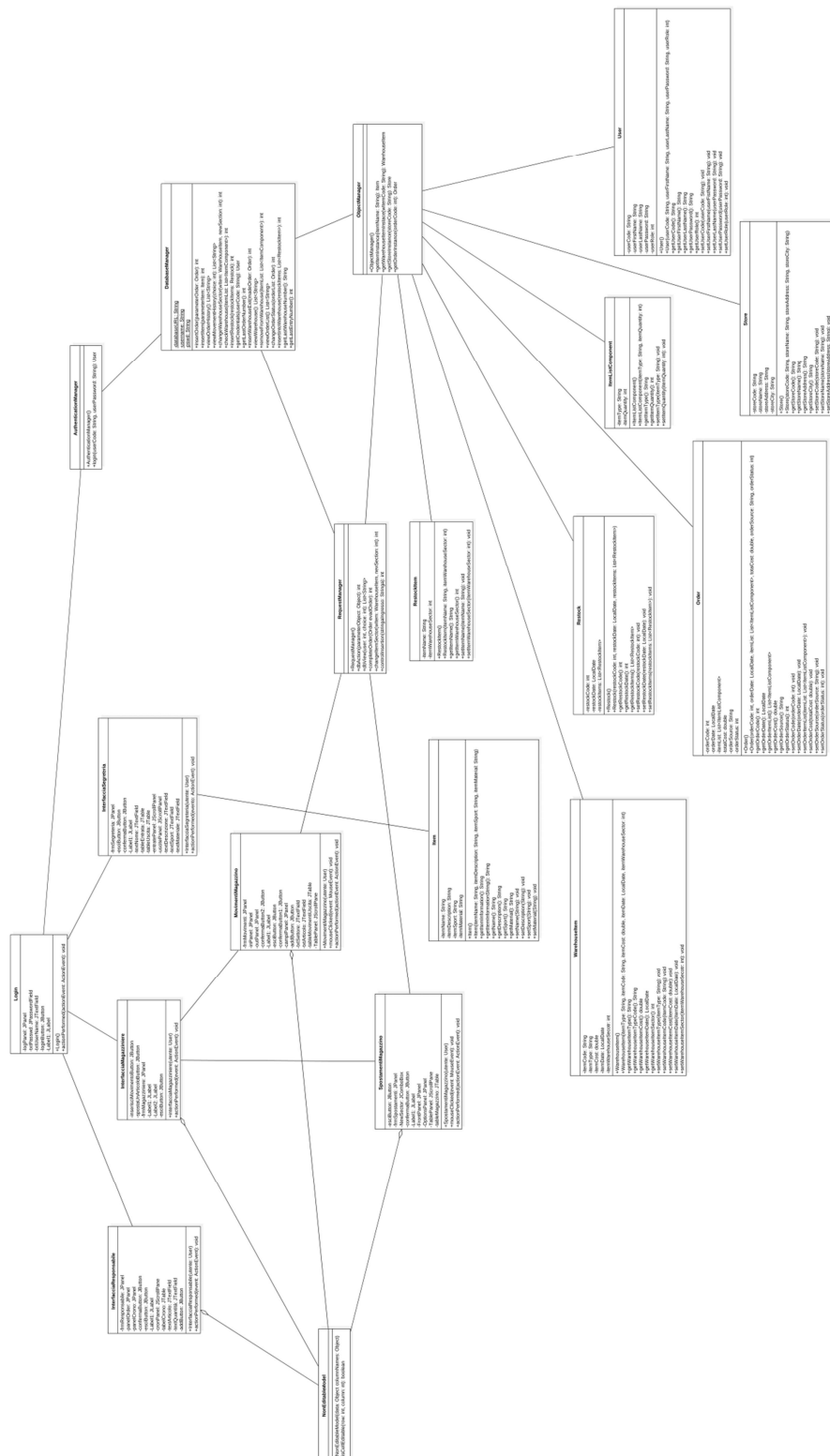
In primo luogo, i magazzinieri possono gestire i movimenti in ingresso. Per fare questo, inseriscono nel sistema l'articolo inserito nel magazzino, il quale lo comunica al database; questo memorizza l'articolo e dà conferma al sistema, il quale assicura l'utente sull'effettiva memorizzazione dell'articolo.

In secondo luogo, i magazzinieri possono evadere gli ordini in uscita. Per compiere questa seconda operazione, essi comunicano al sistema l'ordine da evadere; il sistema ricerca quindi nel database gli articoli relativi all'ordine e il database conferma l'evasione al sistema che, a sua volta, conferma l'effettuazione dell'ordine all'utente.

Infine, i magazzinieri possono modificare la posizione di un articolo. Per fare questo, comunicano al sistema l'articolo da spostare, contenente già l'attuale posizione e la nuova posizione. Il sistema comunica quindi al database la posizione da modificare e la posizione sostitutiva. Il database controlla l'effettiva correttezza dei dati e quindi conferma al sistema l'effettivo spostamento dell'articolo. Infine, il sistema comunica all'utente l'aggiornamento della posizione dell'articolo inserito.

Class diagram

Class diagram generale



Testing e validazione

Il test del software è stato eseguito in modo contemporaneo allo sviluppo del codice.

È stato inizialmente configurato il database relativo al software da implementare mediante il programma PostgreSQL e successivamente è stato popolato. Quindi sono stati effettuati dei test sull'effettivo funzionamento del database: una volta popolato con oggetti noti, è stata controllata la reale presenza di questi all'interno delle tabelle del database.

Terminata la configurazione del database, il processo di sviluppo del software è seguito con la stesura del codice. Gradualmente, sono stati corretti gli errori riportati a terminale dalle diverse parti implementate, come, ad esempio, le diverse finestre implementate mediante le librerie Swing di Java. Inoltre, al termine di ogni script in linguaggio Java compilato, è stato effettuato un test, utilizzando la comunicazione con il database, in modo da verificare l'effettivo funzionamento della singola componente del software.

Dopo aver sviluppato le singole componenti, il database è stato totalmente spopolato, in modo da evitare che il codice fosse stato costruito ad hoc sugli oggetti inseriti e, quindi, è stato ripopolato in modo differente rispetto al precedente popolamento. Quindi è stata verificata la funzionalità del sistema implementato nella sua totalità, cioè valutando l'insieme delle sue parti, correggendo gli errori che si sono presentati.

Infine, è stato avviato un leggero processo di raffinamento del codice, apportando alcune migliorie grafiche, rispetto alla versione originale e ad ulteriori controlli mediante costrutti condizionali nel codice.

Infine, è stato svuotato nuovamente il database, per eliminare le credenziali di prova utilizzate, ed è stato popolato assegnando ad ogni membro del gruppo una funzione e quindi accedendo come segue:

username: Codice Fiscale

password: Nome