



**Università degli Studi di Verona**

**Dipartimento di Informatica**

**Corso di laurea in Informatica**

**Anno accademico 2019/2020**

# **DOCUMENTAZIONE PROGETTO INGEGNERIA DEL SOFTWARE**

---

Sistema informatico per gestire il servizio di spesa on-line di un supermercato

**“eBuy”**

**Feola Emanuele – VR437205**

**Carra Mattia - VR429609**

**Timofte Robert Octavian – VR429581**

# Indice

//TODO (alla fine)

# Introduzione

L'obiettivo dell'elaborato è lo sviluppo di un prototipo software per la gestione di un sistema per la spesa online.

In questa relazione ci proponiamo di raccogliere la documentazione sviluppata e di fornire spiegazioni dettagliate sulle scelte progettuali ed implementative.

## Requisiti

Si vuole progettare un sistema informatico per gestire il servizio di spesa on-line di un supermercato.

Per accedere al sistema, gli utenti devono essere registrati. Per gli utenti si memorizzano nome, cognome, indirizzo, CAP, città, numero di telefono ed email. Ogni utente registrato accede con email e password ed ha eventualmente associata una tessera fedeltà per la raccolta punti. Ogni tessera fedeltà ha un numero identificativo, una data di emissione e il totale dei punti raccolti. Gli utenti possono specificare un metodo di pagamento preferito (con carta di credito, con paypal o alla consegna).

Quando l'utente vuole effettuare una spesa, accede al sistema con le sue credenziali ed inizia a comporre il carrello della spesa. I prodotti disponibili sono visualizzati per reparto (Carne, Pesce, Frutta e Verdura, Alimentari, ...). Per ogni prodotto viene visualizzato il nome, la marca, la quantità contenuta nella confezione, il prezzo e una immagine che lo rappresenta. Per ogni prodotto si visualizza anche se è disponibile o meno. Potranno essere inseriti nel carrello solo prodotti disponibili al momento della spesa. Se un utente inserisce un prodotto che al momento della conferma della spesa non risulta più disponibile, il sistema segnala la cosa al cliente ed elimina il prodotto dal carrello. Dopo aver confermato la spesa, l'utente sceglie data e orario della consegna visualizzando le opzioni possibili.

L'utente può visualizzare il carrello per modificare la quantità dei prodotti inseriti o rimuovere qualche prodotto. L'utente può ricercare i prodotti per tipo (uova, biscotti, pasta), per marca o per eventuali caratteristiche (senza glutine, bio, senza latte, ...). I prodotti possono essere visualizzati in ordine di prezzo crescente o decrescente o in ordine alfabetico rispetto alla marca.

Per ogni spesa si memorizzano il codice (univoco), la data prevista per la consegna, insieme all'intervallo di tempo in cui la spesa potrà essere consegnata, i prodotti che la compongono e in quale quantità con prezzo unitario e totale di ogni prodotto, l'utente che l'ha effettuata, il costo totale, il tipo di pagamento (carta di credito, paypal o alla consegna) e il saldo punti. Ad ogni spesa vengono accreditati sulla tessera fedeltà un numero di punti pari agli euro spesi nella spesa considerata.

Il sistema deve permettere agli utenti di accedere al loro profilo, modificare i dati anagrafici, verificare il saldo punti e lo stato delle loro spese. La spesa può essere confermata, in preparazione o consegnata. Ogni utente può vedere tutte le spese che ha effettuato nel tempo con il dettaglio dei prodotti acquistati.

I responsabili del reparto spesa on-line devono autenticarsi per poter accedere al sistema e devono poter verificare lo stato delle spese e provvedere all'inserimento delle informazioni relative ai prodotti. Il sistema memorizza la quantità di prodotti disponibili in magazzino. Per i responsabili di reparto si memorizzano la matricola, i dati anagrafici, il ruolo e le credenziali di accesso (login e password).

# Requisiti funzionali

## ➤ Requisito funzionale n° 1: Utenti

Tutti gli utenti sono opportunatamente autenticati al sistema, prima che possano accedere a tutte le funzionalità specifiche.

## ➤ Requisito funzionale n° 2: Responsabili del reparto

Devono autenticarsi per poter accedere al sistema e devono poter verificare lo stato delle spese e provvedere all'inserimento delle informazioni relative ai prodotti.

# UML

In questa sezione sono presentati i diagrammi richiesti per la documentazione del prototipo.

## Use Case Diagram

Gli use case sono stati suddivisi in due macro aree definite dalla tipologia di persona che effettua il login.



Di seguito le schede descrittive degli use case sopra citati.

➤ Caso d'uso n° 1: Persona

***Scheda di specifica***

<u>Attori</u>	Persone
<u>Pre-condizioni</u>	Nessuna
<u>Operazioni permesse</u>	Inserimento dati personali per registrarsi al sistema
<u>Sequenza</u>	<ol style="list-style-type: none"><li>1) La persona preme il pulsante "Register" per iniziare la procedura di registrazione al sistema</li><li>2) Si apre una schermata "Edit User", dove la persona può inserire i propri dati personali</li><li>3) Premere il tasto "Ok" per confermare l'operazione oppure "Annulla" per non proseguire</li></ol>
<u>Post-condizioni</u>	Diventare un nuovo utente

**Scheda di specifica**

<u>Attori</u>	Utenti
<u>Pre-condizioni</u>	Aver effettuato l'accesso al sistema
<u>Operazioni permesse</u>	<ul style="list-style-type: none"> <li>➤ Gestione profilo</li> <li>➤ Gestione carta fedeltà</li> <li>➤ Gestione metodo di pagamento</li> <li>➤ Gestione spesa (in particolare gestione carrello, gestione consegna e ricerca prodotti)</li> </ul>
<u>Sequenza</u>	<ol style="list-style-type: none"> <li>1) L'utente entra nel menu principale</li> <li>2) Inizia a fare la spesa scegliendo i prodotti tra quelli disponibili e li aggiunge al carrello premendo il pulsante "Add to Cart"               <ol style="list-style-type: none"> <li>a. Può visualizzare il carrello della spesa in "Shopping" → "Shopping Cart"</li> <li>b. Può visualizzare l'elenco degli ordini effettuati in "Shopping" → "All expenses"</li> <li>c. Può visualizzare i punti ottenuti sulla carta fedeltà in "Fidelity Card" → "View"                   <ol style="list-style-type: none"> <li>i. Se non ha la carta fedeltà, il sistema ne crea automaticamente una</li> <li>ii. Può decidere di non volerla premendo su "Remove"</li> </ol> </li> <li>d. Può visualizzare i prodotti per caratteristica (inizialmente non settata poiché opzionale)</li> </ol> </li> <li>3) Può gestire il profilo su "Profile"</li> <li>4) Al termine effettua il logout</li> </ol>
<u>Post-condizioni</u>	Nuova successione di operazioni effettuate

➤ Caso d'uso n° 3: Responsabile del reparto

***Scheda di specifica***

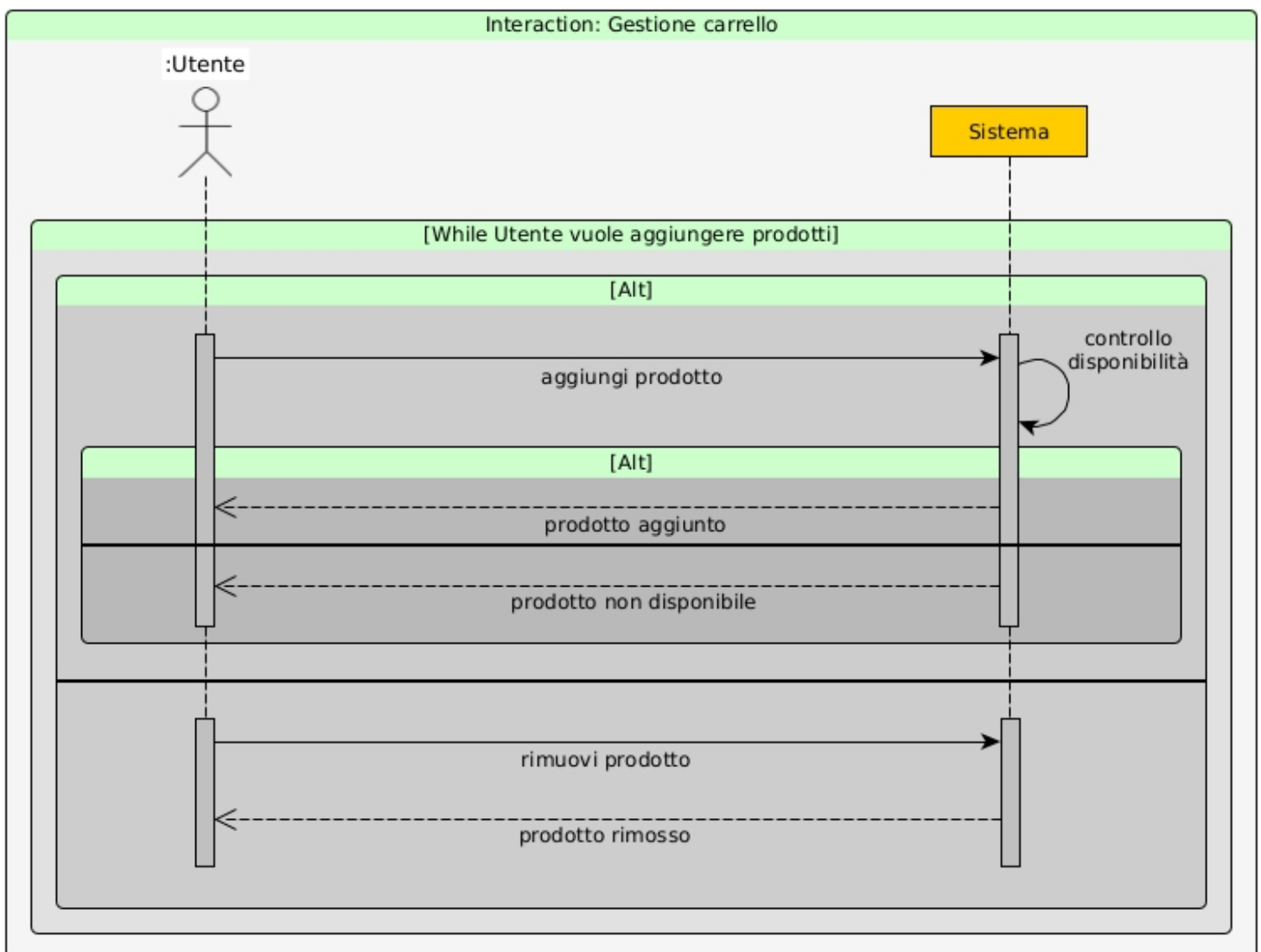
<u>Attori</u>	Responsabili del reparto
<u>Pre-condizioni</u>	Aver effettuato l'accesso al sistema
<u>Operazioni permesse</u>	<ul style="list-style-type: none"> <li>➤ Gestione spese</li> <li>➤ Gestione prodotti</li> </ul>
<u>Sequenza</u>	<ol style="list-style-type: none"> <li>1) Il manager entra nel menu principale</li> <li>2) Su "Product" visualizza i prodotti del supermercato               <ol style="list-style-type: none"> <li>a. Può aggiungere un prodotto con "New"                   <ol style="list-style-type: none"> <li>i. Compila tutte le informazioni relative, scegliendo anche un'immagine e specificando la disponibilità</li> </ol> </li> <li>b. Può modificare un prodotto specifico con "Edit"</li> <li>c. Può cancellare un prodotto selezionato con "Delete"</li> </ol> </li> <li>3) Può visualizzare tutti gli ordini evasi su "Expenses"</li> <li>4) Può aggiungere un altro Manager con "New", specificandone il ruolo</li> <li>5) Con "Edit" modifica il suo account</li> <li>6) Effettua il "Logout"</li> </ol>
<u>Post-condizioni</u>	Nuova successione di operazioni effettuate



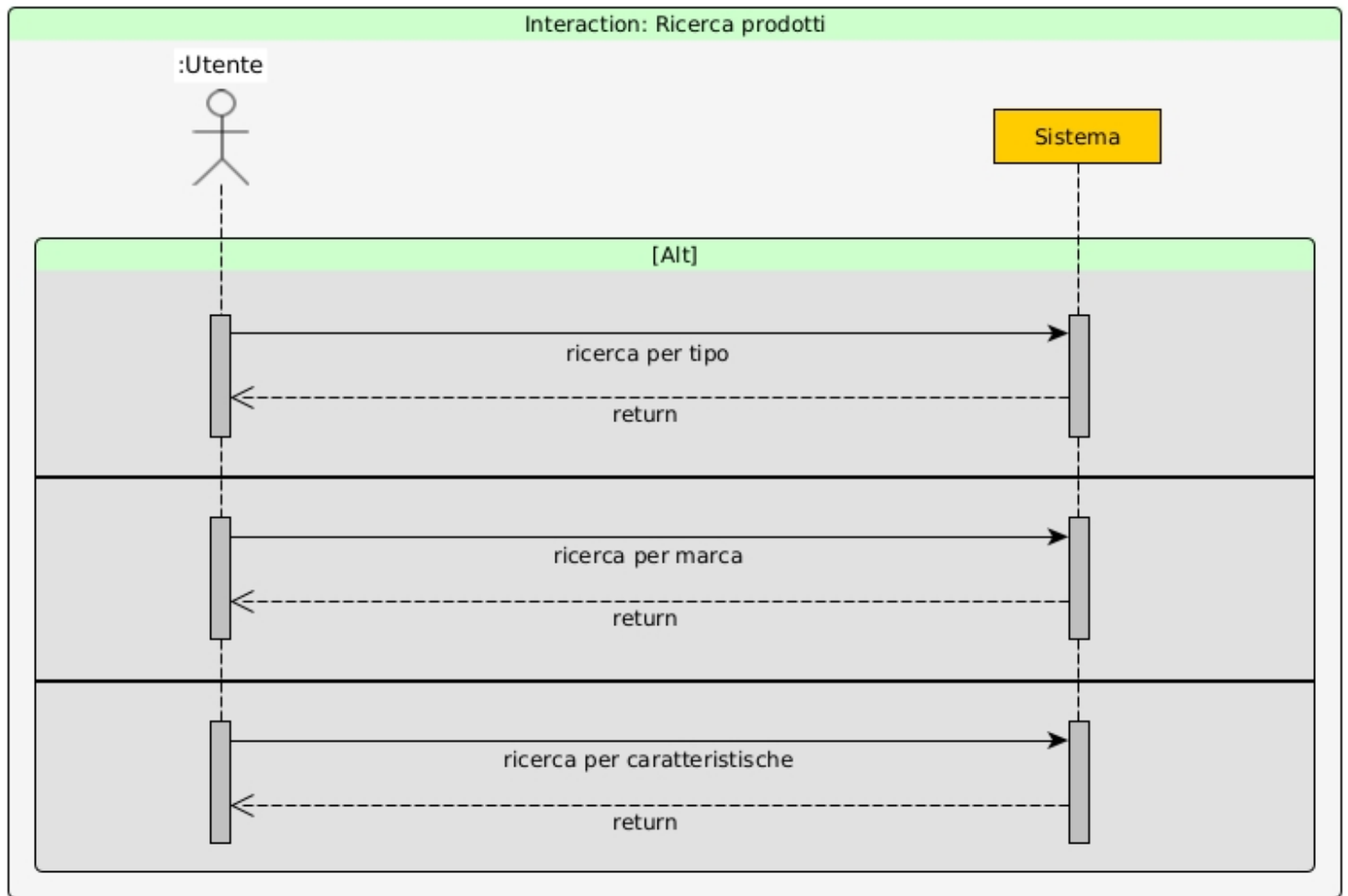
## Sequence Diagram (relativo a Use Case)

Un Sequence Diagram è un diagramma previsto dall'UML utilizzato per descrivere uno scenario. Uno scenario è una determinata sequenza di azioni in cui tutte le scelte sono state già effettuate.

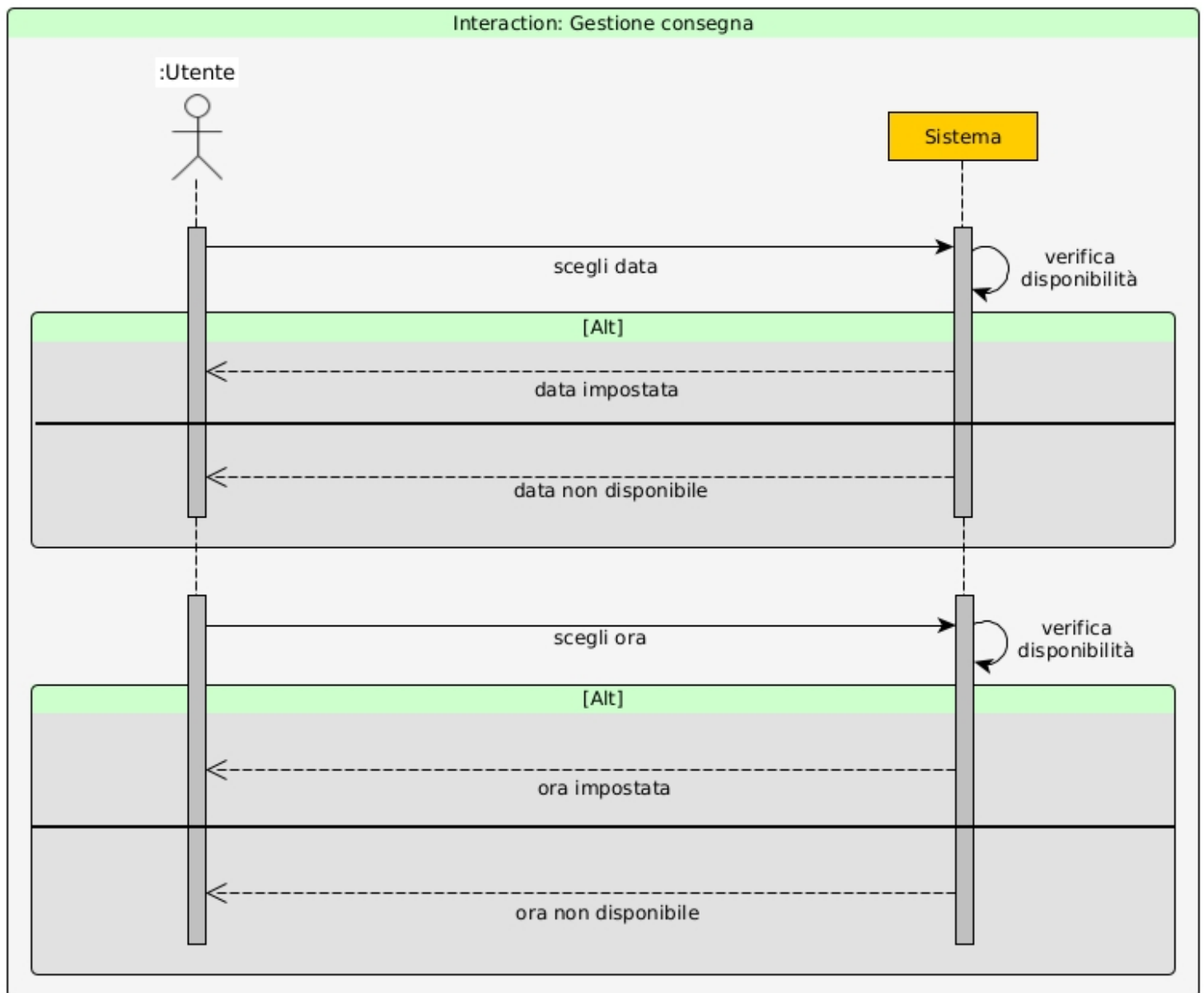
### Gestione carrello



## Ricerca prodotto



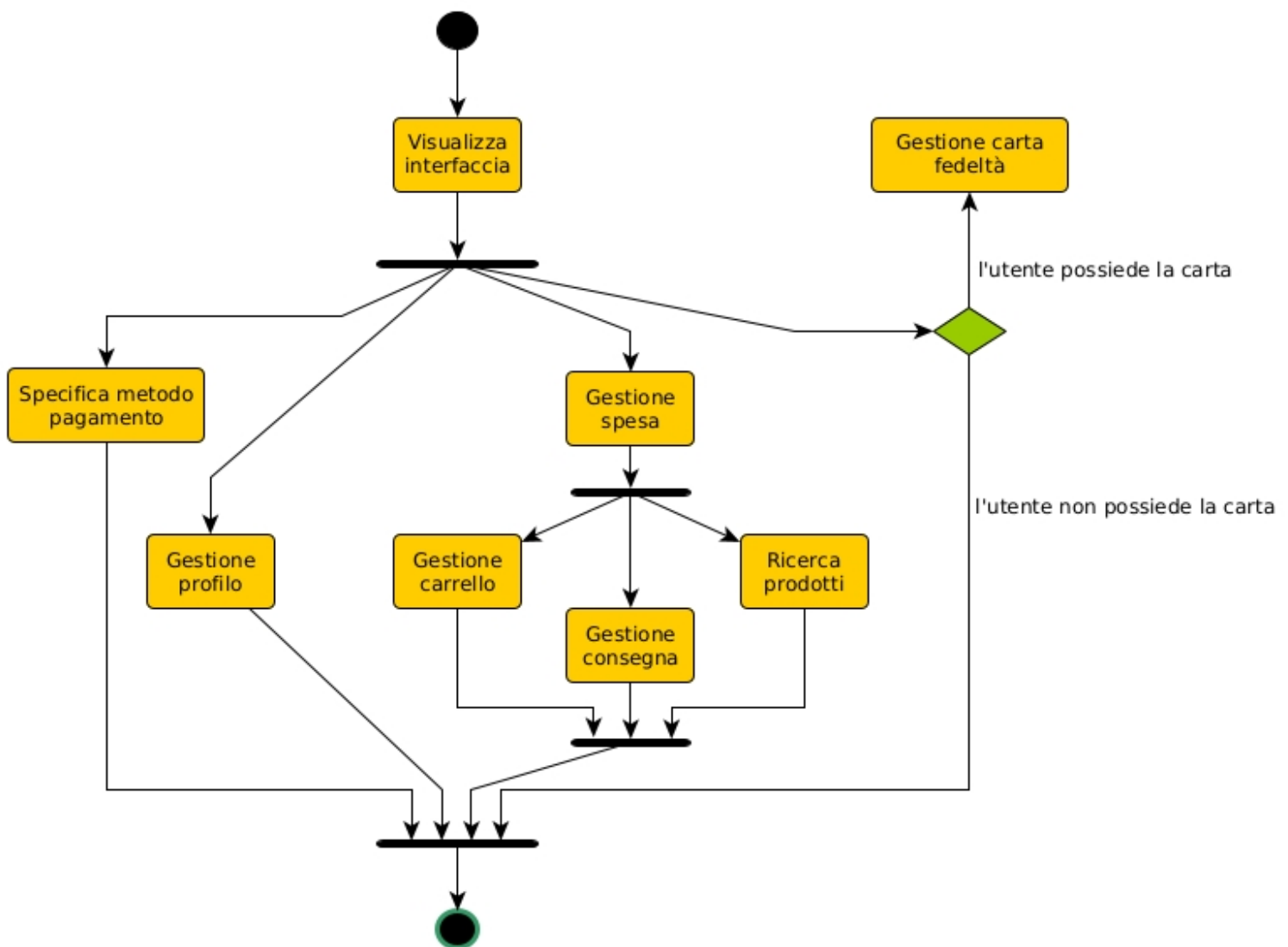
## Gestione consegna

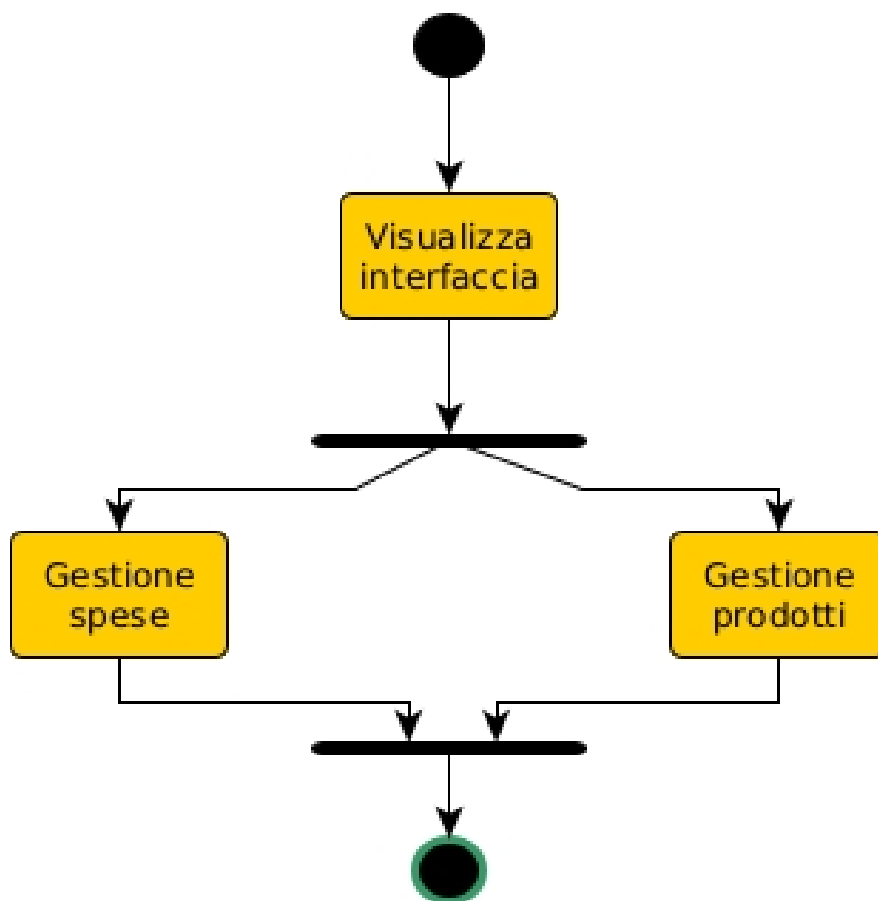


# Activity Diagram

Gli activity diagram di seguito servono a esplicitare meglio il flusso d'esecuzione degli use case descritti in precedenza.

Utente





## Class Diagram

I diagrammi delle classi sono uno dei tipi di diagrammi che possono comparire in un modello UML. In termini generali, consentono di descrivere tipi di entità, con le loro caratteristiche e le eventuali relazioni fra questi tipi.

//todo io domani quando tutto è finito

## Sequence Diagram (relativo al sw progettato)

//todo io domani

## Scelte progettuali

### Sviluppo

Abbiamo scelto di sviluppare il prototipo utilizzando il linguaggio Java vista la sua grande flessibilità e la sua estesa diffusione in quello che è il mercato odierno. Nonostante questo, vista la nostra voglia di ampliare le nostre conoscenze del linguaggio al di fuori degli argomenti trattati a lezione, abbiamo deciso di utilizzare la libreria grafica JavaFX invece di Swing consapevoli del fatto che non avremmo potuto far uso di un designer di interfacce grafiche. La libreria supporta completamente gli stili CSS, da noi poco utilizzati in quanto soddisfatti del risultato ottenuto con lo stile standard dei componenti.

### Metodologia di sviluppo

Il progetto è stato interamente sviluppato utilizzando GitHub per l'hosting del repository. Abbiamo seguito la metodologia di sviluppo Agile: il lavoro è stato diviso in due macroaree (area logica e GUI).

A sua volta ogni area è stata divisa in task più piccoli in modo da alleggerire il carico di lavoro e testati man mano che venivano inseriti nel codice. I design pattern sono stati scelti nella fase di sviluppo in base a quelle che erano le nostre necessità al fine di ottenere un codice performante ma allo stesso tempo leggibile.

Ogni settimana organizzavamo dei meeting zoom per confrontarci sul lavoro fatto e per discutere le varie idee sul progetto, in modo tale da migliorare il pair programming.

Inoltre abbiamo utilizzato Trello, un tool online che permette di schedare ogni compito da eseguire e di gestire il lavoro nel modo migliore possibile.



## Gestione dei dati (Serializzazione)

//TODO (appena avete fatto integro la docu)

## Organizzazione della GUI

//TODO magari aggiungere le schermate principali con strumento di cattura (me ne occupo io)

## Pattern utilizzati

//TODO (appena avete fatto integro la docu)

## Testing e validazione

//TODO JUnit ? Javadoc?