



Laurea in Informatica - Università degli Studi di Salerno Corso di  
Ingegneria del Software - Prof. A. De Lucia



Data: 08/10/2024

### TEAM

Gianluca Fusco 0512116485  
Antonio Giorgio 0512106036  
Cristian Di Popolo 0512105370

# Revision History

Data	Versione	Descrizione	Autore
13/12/2024	1.0	Introduzione e packages	G. Fusco A. Giorgio C. Di Popolo
16/12/2024	1.1	Interfaccia delle classi	G. Fusco A. Giorgio C. Di Popolo

# **Indice**

## **1. Introduzione**

1.1 Object Design Trade-Offs

1.2 Linee guida per l'interfaccia

1.3 Definizioni, acronimi e abbreviazioni

## **2. Packages**

2.1 Adapter

2.2 Model

2.3 View

2.4 Api

## **3. Interfaccia delle classi**

# 1 INTRODUZIONE

## 1.1 Object Design Trade-Offs

### **Comprensibilità vs Tempo:**

Data la scarsa disponibilità di tempo, il codice non verrà scritto dando priorità alla comprensibilità ma verrà scritto in funzione del tempo del relativo task.

### **Riusabilità vs Costi:**

Dato il budget designato per il progetto, pari a zero. Non saranno integrati sistemi che prevedano un costo di utilizzo.

### **Sicurezza vs Efficienza:**

Dal punto di vista della sicurezza, abbiamo implementato un sistema di autenticazione basato su username e password crittografata con algoritmi sicuri come BCrypt, evitando l'archiviazione in chiaro. Inoltre, il controllo degli accessi è regolato da ruoli, garantendo che ogni utente acceda solo alle risorse autorizzate. Dal punto di vista dell'efficienza, queste misure essenziali bilanciano protezione dei dati e rispetto delle scadenze, offrendo una base sicura e migliorabile per il progetto.

### **Trasparenza vs Efficienza:**

Per la gestione dei dati, abbiamo scelto MySQL come sistema di database, privilegiando la trasparenza e l'affidabilità nelle operazioni. MySQL offre un DBMS che garantisce consistenza, semplificando l'interazione con i dati e riducendo il rischio di errori nella persistenza. Questo approccio, sebbene comporti un lieve sacrificio in termini di efficienza rispetto a soluzioni meno strutturate, ci consente di mantenere un sistema comprensibile, affidabile e facile da mantenere, in linea con le risorse disponibili e gli obiettivi del progetto.

## 1.2 Linee guida per l'interfaccia

### - Classi e Interfacce

#### **Nomi delle Classi e Interfacce:**

Devono seguire la convenzione UpperCamelCase.

I nomi devono descrivere chiaramente il ruolo della classe o interfaccia (ad esempio OrderManager, ProductService).

### - Metodi

#### **Nomi dei Metodi:**

Devono seguire la convenzione lowerCamelCase.

I nomi devono descrivere la funzionalità del metodo e includere un verbo (es.: addProduct, updateOrderStatus).

Metodi getter e setter devono seguire la regola di nominazione get[nomeAttributo] e set[nomeAttributo].

### - Variabili

#### **Nomi delle Variabili:**

Devono seguire la convenzione lowerCamelCase.

Le variabili di istanza devono essere private.

Le variabili locali possono essere dichiarate insieme, ma ogni variabile d'istanza deve essere dichiarata singolarmente.

### - Convenzioni di Gestione degli Errori

Le condizioni di errore devono lanciare eccezioni invece di restituire valori di errore.

Utilizzare eccezioni specifiche e personalizzate per l'applicazione.

### - Altre Linee Guida

#### **Indentazione del codice:**

Utilizzare 4 spazi o una tabulazione per l'indentazione.

### **1.3 Definizioni, acronimi e abbreviazioni**

**APP:** Applicazione di acquisto online per prodotti alimentari.

#### **ADM**

Amministratore: utente responsabile della gestione della piattaforma e del catalogo prodotti.

#### **USR**

Utente generico: qualsiasi utilizzatore della piattaforma.

#### **CUS**

Cliente: utente loggato che ha la possibilità di acquistare prodotti.

#### **PROD**

Prodotto: dati e informazioni relative a un prodotto alimentare disponibile per l'acquisto.

#### **CAT**

Catalogo: l'elenco aggiornato di prodotti disponibili sulla piattaforma.

#### **CAR**

Carrello: spazio dove l'utente aggiunge i prodotti prima dell'acquisto.

#### **API**

Application Programming Interface: interfaccia per l'interazione tra sistemi diversi.

#### **UI**

User Interface: interfaccia utente che permette l'interazione con l'applicazione.

#### **JSON**

JavaScript Object Notation: formato di dati per lo scambio di informazioni tra client e server.

## **CSS**

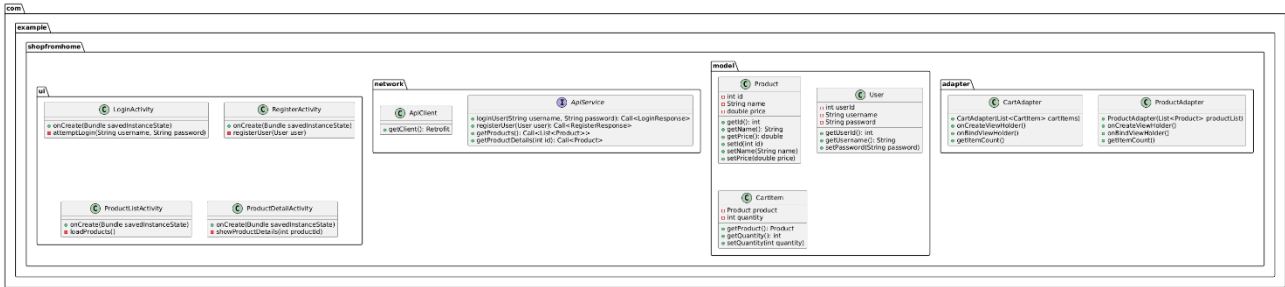
Cascading Style Sheets: fogli di stile per la formattazione dell'interfaccia utente.

## **JS**

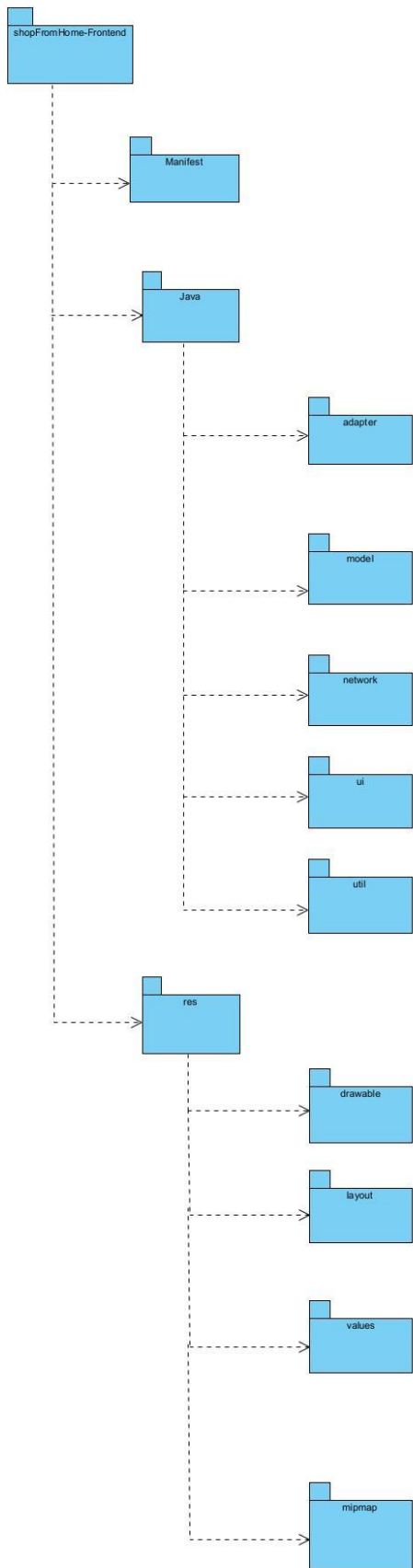
JavaScript: linguaggio di programmazione per funzioni interattive nel frontend.

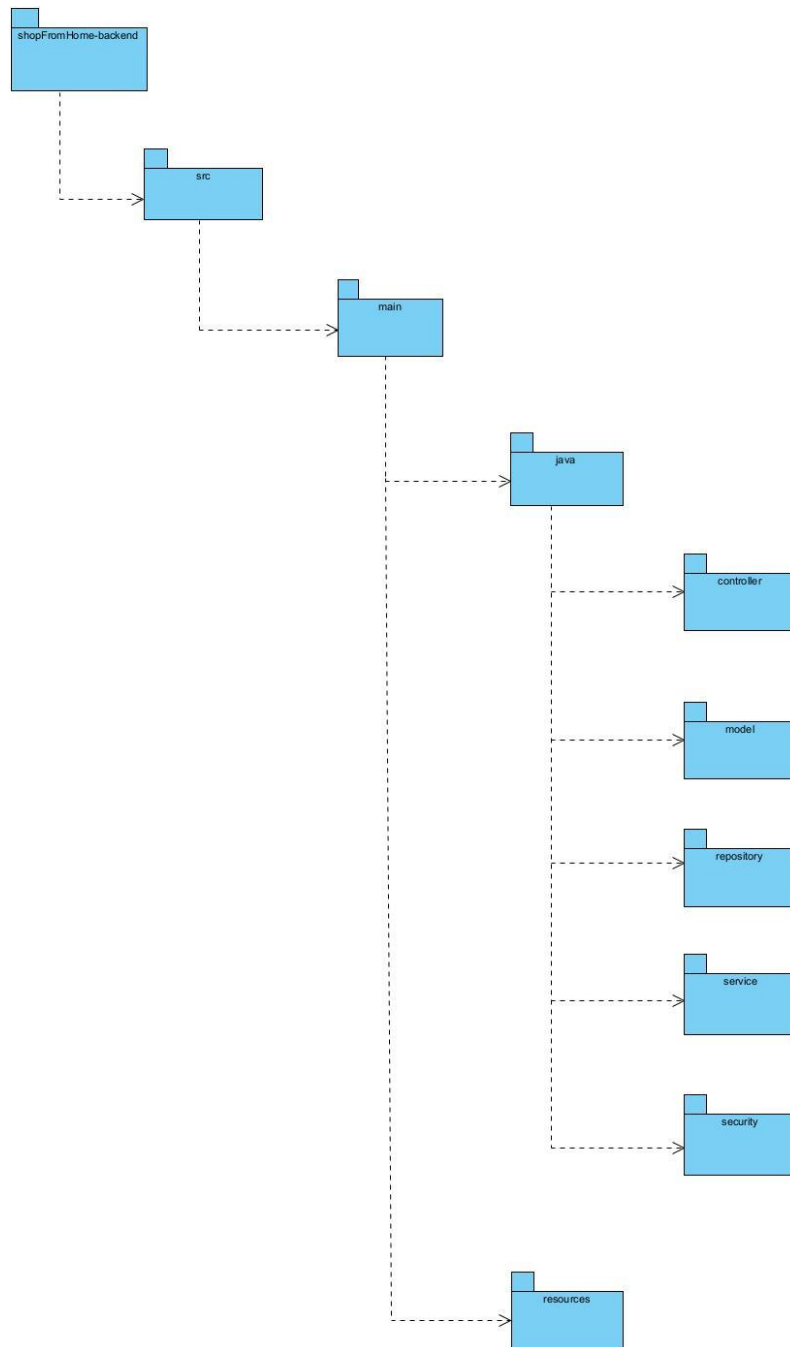
# 2 PACKAGES

Il sistema ShopFromHome è diviso in packages nel modo seguente:  
Frontend:







[illegible]

### 3 CLASS INTERFACE

Adapter:

Nome Classe	ProductAdapter	
Descrizione	Adapter per la visualizzazione della lista dei prodotti nel frontend.	
Metodi	onCreateViewHolder(viewType: int) onBindViewHolder(holder: ViewHolder, position: int) getItemCount()	
Invariante di classe	/	
Nome Metodo		Descrizione
onCreateViewHolder(viewType: int)		Crea una nuova vista per un prodotto.
onBindViewHolder(holder: ViewHolder, position: int)		Associa i dati del prodotto alla view.
getItemCount()		Restituisce il numero di prodotti.

Nome Classe	CartAdapter	
Descrizione	Adapter per la visualizzazione delle categorie nel frontend.	
Metodi	onCreateViewHolder(viewType:int) onBindViewHolder(holder: ViewHolder, position: int) getItemCount()	
Invariante di classe	/	

Nome Metodo	Descrizione
<b>onCreateViewHolder(viewType: int)</b>	Crea una vista per un articolo del carrello.
<b>onBindViewHolder(holder: ViewHolder, position: int)</b>	Associa i dati del carrello alla vista.
<b>getItemCount()</b>	Restituisce il numero di articoli nel carrello.

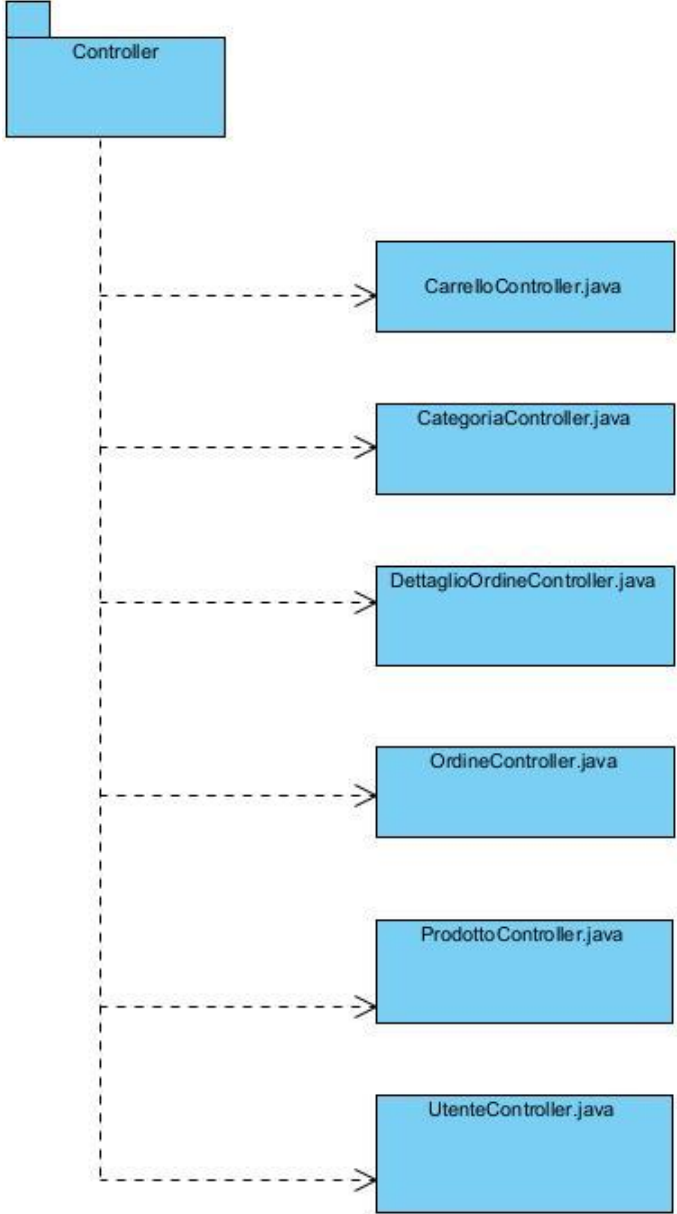
Nome Classe	CategoryAdapter
Descrizione	Adapter per la visualizzazione delle categorie nel frontend.
Metodi	onCreateViewHolder(viewType:int) onBindViewHolder(holder: ViewHolder, position: int) getItemCount()
Invariante di classe	/

Nome Metodo	Descrizione
<b>onCreateViewHolder(viewType: int)</b>	Crea una vista per una categoria.
<b>onBindViewHolder(holder: ViewHolder, position: int)</b>	Associa i dati della categoria alla vista.
<b>getItemCount()</b>	Restituisce il numero di categorie.

View:

Nome Classe	CategoryActivity	
Descrizione	Gestisce l'interfaccia utente per la gestione delle categorie.	
Metodi	onCreate()  onAddCategoryClick()  onUpdateCategoryClick()  onDeleteCategoryClick()  displayCategories()	
Invariante di classe	/	
Nome Metodo		Descrizione
onCreate()		Inizializza l'interfaccia utente per la gestione delle categorie.
onAddCategoryClick()		Metodo per aggiungere una nuova categoria.
onUpdateCategoryClick()		Metodo per modificare una categoria esistente.
onDeleteCategoryClick()		Metodo per eliminare una categoria.
displayCategories()		Visualizza la lista delle categorie.

Controller:



Nome Classe	CarrelloController	
Descrizione	Gestisce le operazioni sul carrello degli utenti.	
Metodi	aggiungiProdotto(Long idCarrello, DettagliCarrello dettagliCarrello)  rimuoviProdotto(Long idCarrello, Long idProdotto)  aggiornaQuantita(Long idCarrello, Long idProdotto, BigDecimal nuovaQuantita)  getCarrelliByUtente(Long idUtente)  getDettagliCarrello(Long idCarrello)	
Invariante di classe	/	
Nome Metodo		Descrizione
aggiungiProdotto(Long idCarrello, DettagliCarrello dettagliCarrello)		Aggiunge un prodotto al carrello specificato.
rimuoviProdotto(Long idCarrello, Long idProdotto)		Rimuove un prodotto dal carrello.
aggiornaQuantita(Long idCarrello, Long idProdotto, BigDecimal nuovaQuantita)		Aggiorna la quantità di un prodotto nel carrello.
getCarrelliByUtente(Long idUtente)		Recupera i carrelli associati a un utente.
getDettagliCarrello(Long idCarrello)		Recupera i dettagli del carrello specificato.

<b>Nome Classe</b>	<b>CategoriaController</b>	
<b>Descrizione</b>	Gestisce le categorie.	
<b>Metodi</b>	getAllCategories() createCategory(Categoria categoria) updateCategory(Long id, Categoria categoria) deleteCategory(Long id)	
<b>Invariante di classe</b>	/	
<b>Nome Metodo</b>		<b>Descrizione</b>
getAllCategories()		Restituisce tutte le categorie.
createCategory(Categoria categoria)		Crea una nuova categoria.
updateCategory(Long id, Categoria categoria)		Aggiorna una categoria esistente.
deleteCategory(Long id)		Elimina una categoria.

<b>Nome Classe</b>	<b>DettaglioOrdineController</b>	
<b>Descrizione</b>	Gestisce i dettagli dell'ordine.	
<b>Metodi</b>	getAllDettagliOrdine() getDettaglioOrdineById(Long id) createDettaglioOrdine(DettaglioOrdine dettaglioOrdine) deleteDettaglioOrdine(Long id)	
<b>Invariante di classe</b>	/	



Nome Metodo	Descrizione
<b>getAllDettagliOrdine()</b>	Restituisce tutti i dettagli degli ordini.
<b>getDettaglioOrdineById(Long id)</b>	Recupera un dettaglio ordine per ID.
<b>createDettaglioOrdine(DettaglioOrdine dettaglioOrdine)</b>	Crea un nuovo dettaglio ordine.
<b>deleteDettaglioOrdine(Long id)</b>	Elimina un dettaglio ordine.

Nome Classe	OrdineController	
Descrizione	Gestisce gli ordini.	
Metodi	<b>getAllOrdini()</b> <b>creaOrdine(Ordine ordine)</b> <b>updateOrderStatus(Long ordineId, String stato)</b> <b>getOrdiniByStato(StatoOrdine stato)</b> <b>getOrdiniByUtente(Long utenteId)</b> <b>getOrdineById(Long ordineId)</b> <b>annullaOrdine(Long ordineId)</b>	
Invariante di classe	/	
Nome Metodo	Descrizione	
<b>getAllOrdini()</b>	Restituisce tutti gli ordini.	
<b>creaOrdine(Ordine ordine)</b>	Crea un nuovo ordine.	

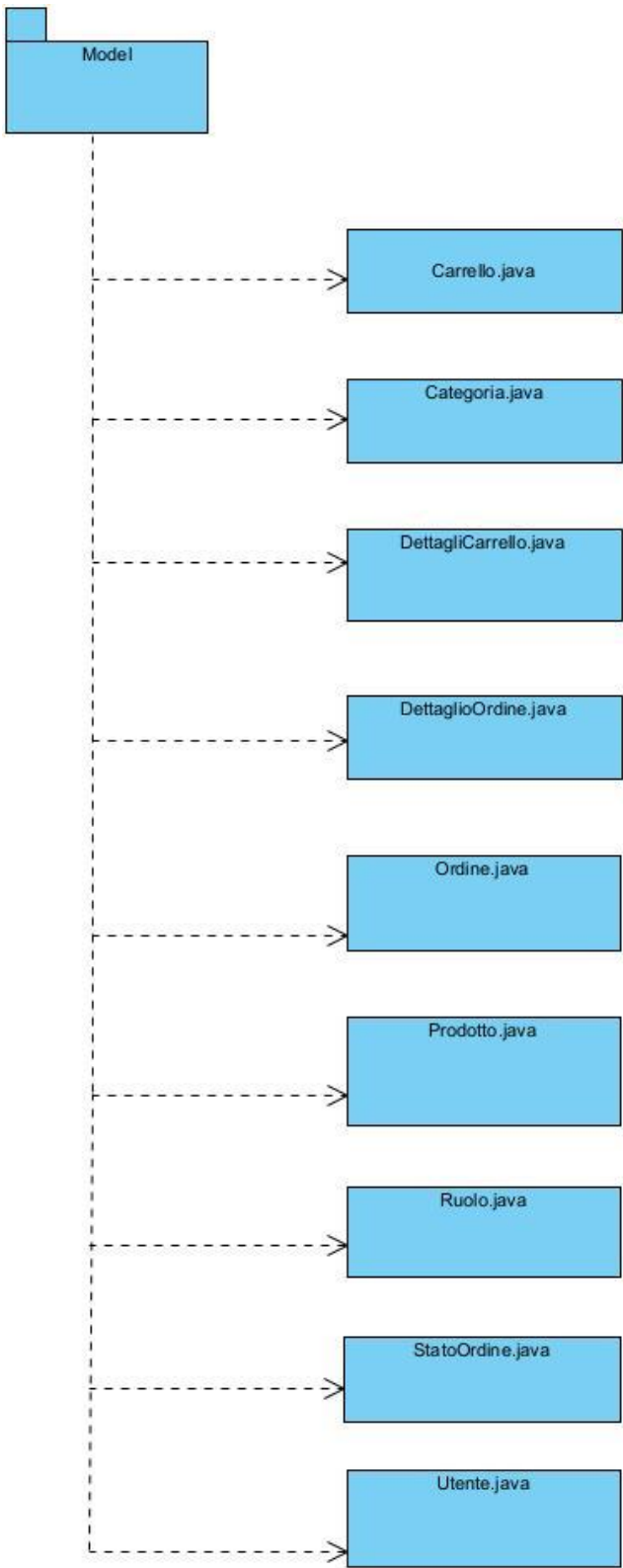
<b>updateOrderStatus(Long ordineId, String stato)</b>	Aggiorna lo stato di un ordine.
<b>getOrdiniByStato(StatoOrdine stato)</b>	Recupera ordini per stato.
<b>getOrdiniByUtente(Long utenteId)</b>	Recupera ordini per utente.
<b>getOrdineById(Long ordineId)</b>	Recupera un ordine per ID.
<b>annullaOrdine(Long ordineId)</b>	Annulla un ordine.

Nome Classe	ProdottoController	
Descrizione	Gestisce le operazioni sui prodotti.	
Metodi	getAllProducts() getProductsByCategory(Long categoriaId) searchProducts(String query) createProduct(Prodotto prodotto) updateProduct(Long id, Prodotto prodotto) deleteProduct(Long id) getProductById(Long id)	
Invariante di classe	/	
Nome Metodo		Descrizione
getAllProducts()		Restituisce tutti i prodotti.
getProductsByCategory(Long categoriaId)		Recupera prodotti per categoria.
searchProducts(String query)		Cerca prodotti in base a una query.
createProduct(Prodotto		Crea un nuovo prodotto.

<b>prodotto)</b>	
<b>updateProduct(Long id, Prodotto prodotto)</b>	Aggiorna un prodotto esistente.
<b>deleteProduct(Long id)</b>	Elimina un prodotto.
<b>getProductById(Long id)</b>	Recupera un prodotto per ID.

<b>Nome Classe</b>	<b>UtenteController</b>	
<b>Descrizione</b>	Gestisce l'utente.	
<b>Metodi</b>	registerUser (Utente utente) getUserByEmail(String email) getUtentiByRuolo(Ruolo ruolo) loginUser (Utente userInput)	
<b>Invariante di classe</b>	/	
<b>Nome Metodo</b>	<b>Descrizione</b>	
<b>registerUser (Utente utente)</b>	Registra un nuovo utente.	
<b>getUserByEmail(String email)</b>	Recupera un utente per email.	
<b>getUtentiByRuolo(Ruolo ruolo)</b>	Recupera utenti per ruolo.	
<b>loginUser (Utente userInput)</b>	Effettua il login di un utente.	

Model:



Nome Classe	Carrello
Descrizione	Rappresenta entità carrello.
Metodi	Getter Setter
Invariante di classe	/

Nome Classe	Categoria
Descrizione	Rappresenta entità categoria.
Metodi	Getter Setter
Invariante di classe	/

Nome Classe	DettagliCarrello
Descrizione	Rappresenta entità dettagli carrello.
Metodi	Getter Setter
Invariante di classe	/

Nome Classe	DettaglioOrdine
Descrizione	Rappresenta entità dettaglio ordine.
	Getter

<b>Metodi</b>	Setter
<b>Invariante di classe</b>	/

<b>Nome Classe</b>	<b>Ordine</b>
<b>Descrizione</b>	Rappresenta entità ordine.
<b>Metodi</b>	Getter Setter
<b>Invariante di classe</b>	/

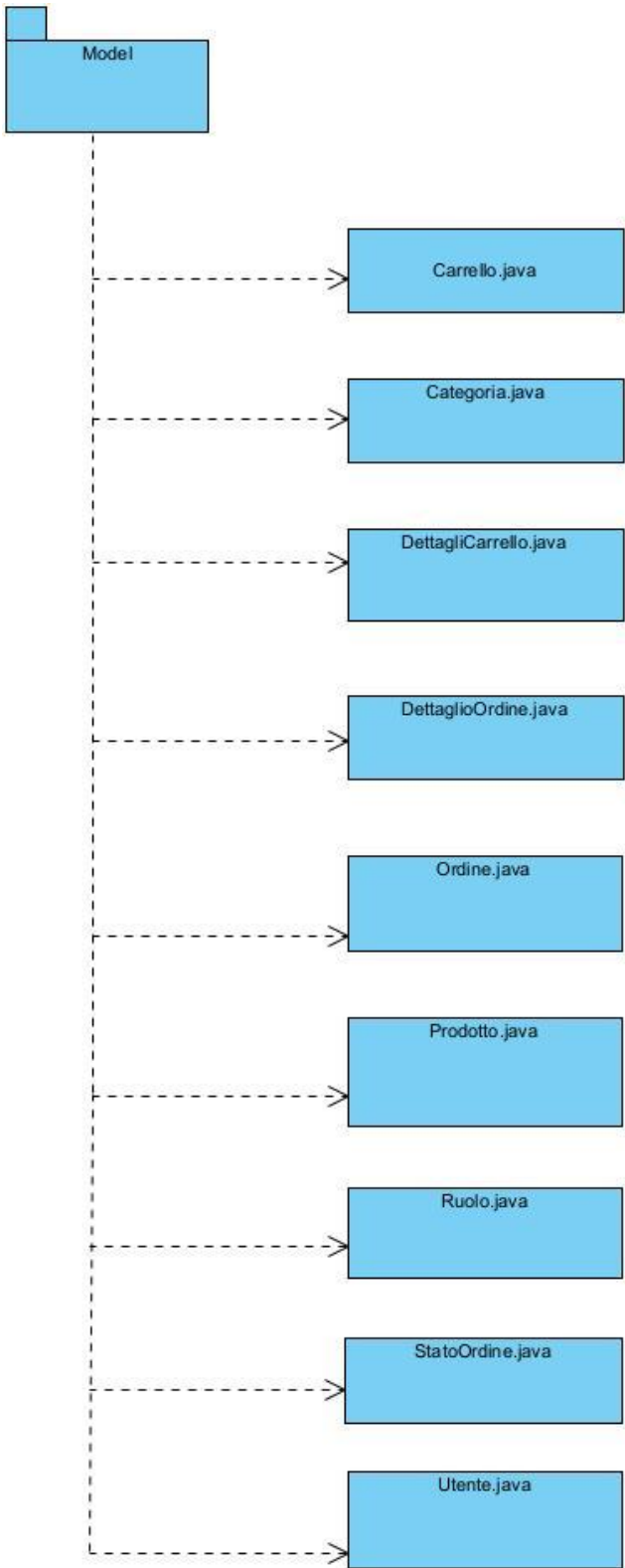
<b>Nome Classe</b>	<b>Prodotto</b>
<b>Descrizione</b>	Rappresenta entità prodotto.
<b>Metodi</b>	Getter Setter
<b>Invariante di classe</b>	/

<b>Nome Classe</b>	<b>Ruolo</b>
<b>Descrizione</b>	Rappresenta entità ruolo.
<b>Metodi</b>	Getter Setter
<b>Invariante di classe</b>	/

Nome Classe	StatoOrdine
Descrizione	Rappresenta entità stato ordine.
Metodi	Getter Setter
Invariante di classe	/

Nome Classe	Utente
Descrizione	Rappresenta entità utente.
Metodi	Getter Setter
Invariante di classe	/

Service:





Nome Classe	UtenteService	
Descrizione	Rappresenta un utente nel sistema.	
Metodi	createUser (Utente utente) findByEmail(String email) deleteUser (Long id) findByRuolo(Ruolo ruolo)	
Invariante di classe	/	
Nome Metodo		Descrizione
createUser (Utente utente)		Crea un nuovo utente.
findByEmail(String email)		Trova un utente per email.
deleteUser (Long id)		Elimina un utente.
findByRuolo(Ruolo ruolo)		Trova utenti per ruolo.

Nome Classe	CategoriaService	
Descrizione	Rappresenta una categoria nel sistema.	
Metodi	getAllCategories() createCategory(Categoria categoria) updateCategory(Long id, Categoria categoria) deleteCategory(Long id)	
Invariante di classe	/	
Nome Metodo		Descrizione
getAllCategories()		Restituisce tutte le categorie.

<b>createCategory(Categoria categoria)</b>	Crea una nuova categoria.
<b>updateCategory(Long id, Categoria categoria)</b>	Aggiorna una categoria esistente.
<b>deleteCategory(Long id)</b>	Elimina una categoria.

Nome Classe	ProdottoService	
Descrizione	Rappresenta un prodotto nel sistema.	
Metodi	getAllProducts() getProductsByCategory(Long categoriaId) createProduct(Prodotto prodotto) updateProduct(Long id, Prodotto prodotto) deleteProduct(Long id) searchProducts(String query) getProductById(Long id)	
Invariante di classe	/	
Nome Metodo		Descrizione
getAllProducts()		Restituisce tutti i prodotti.
getProductsByCategory(Long categoriaId)		Recupera prodotti per categoria.
createProduct(Prodotto prodotto)		Crea un nuovo prodotto.
updateProduct(Long id,		Aggiorna un prodotto esistente.

<b>Prodotto prodotto)</b>	
<b>deleteProduct(Long id)</b>	Elimina un prodotto.
<b>searchProducts(String query)</b>	Cerca prodotti in base a una query.
<b>getProductById(Long id)</b>	Recupera un prodotto per ID.

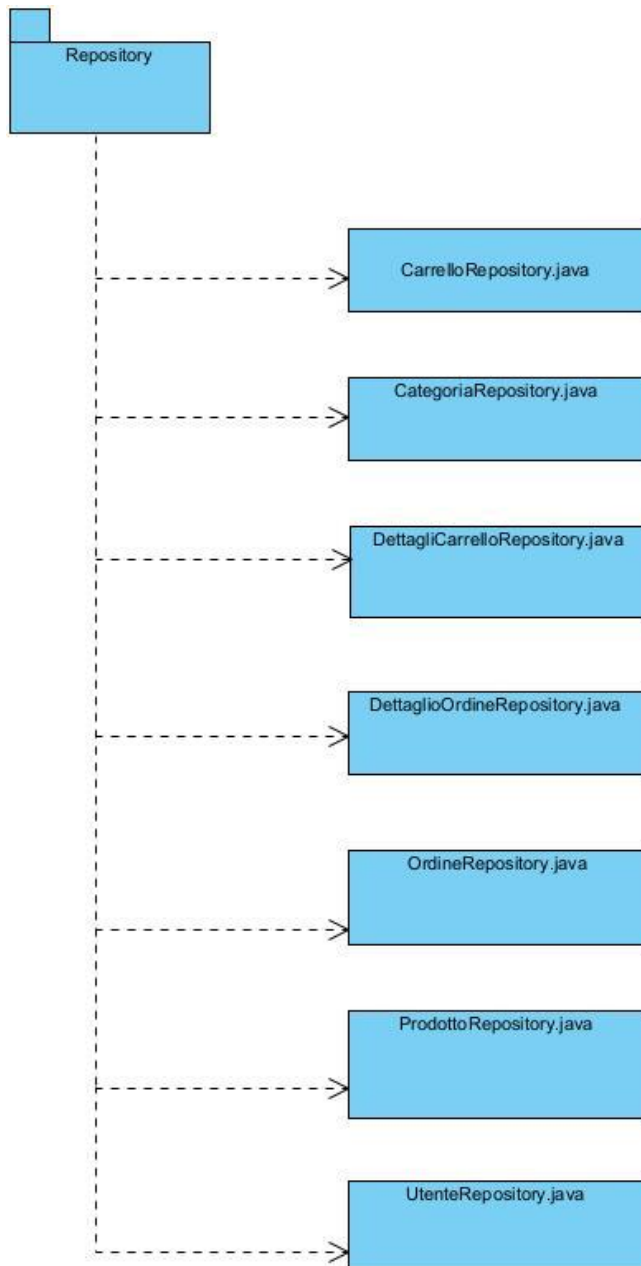
Nome Classe	CarrelloService	
Descrizione	Rappresenta il carrello.	
Metodi	aggiungiProdottoAlCarrello(Long idCarrello, DettagliCarrello dettagliCarrello)  rimuoviProdottoDalCarrello(Long idCarrello, Long idProdotto)  modificaQuantitaProdotto(Long idCarrello, Long idProdotto, BigDecimal nuovaQuantita)  getCarrelliByUtente(Long idUtente)  getDettagliCarrelloById(Long idCarrello)	
Invariante di classe	/	
Nome Metodo		Descrizione
aggiungiProdottoAlCarrello(Long idCarrello, DettagliCarrello dettagliCarrello)		Aggiunge un prodotto al carrello.
rimuoviProdottoDalCarrello(Long idCarrello, Long idProdotto)		Rimuove un prodotto dal carrello.
modificaQuantitaProdotto(Long idCarrello, Long idProdotto, BigDecimal nuovaQuantita)		Modifica la quantità di un prodotto nel carrello.

<b>getCarrelliByUtente(Long idUtente)</b>	Recupera i carrelli di un utente.
<b>getDettagliCarrelloById(Long idCarrello)</b>	Recupera i dettagli di un carrello.

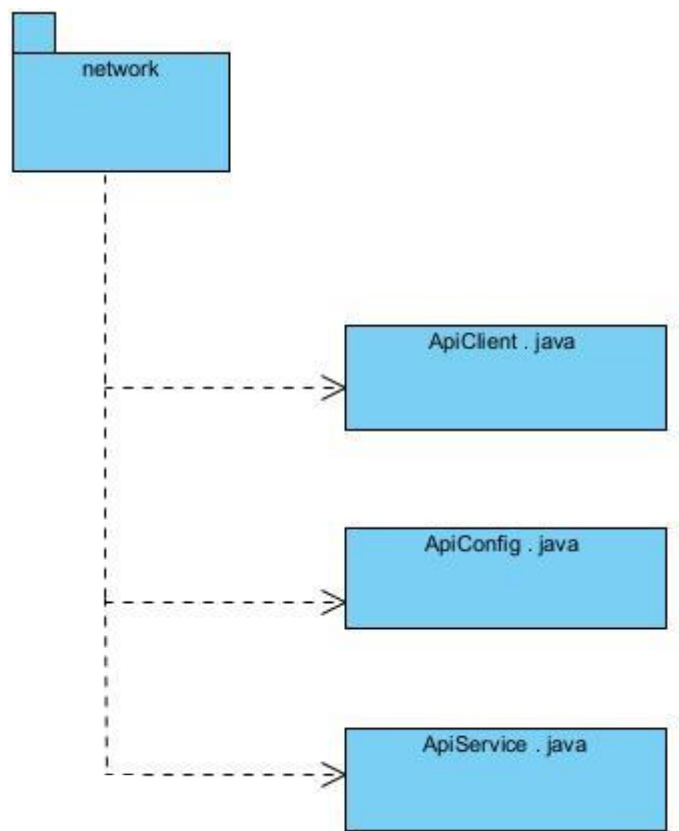
Nome Classe	DettaglioOrdineService	
Descrizione	Rappresenta i dettagli dell'ordine.	
Metodi	getAlldettagliOrdine() getDettaglioOrdineById(Long id) saveDettaglioOrdine(DettaglioOrdine dettaglioOrdine) deleteDettaglioOrdine(Long id)	
Invariante di classe	/	
Nome Metodo		Descrizione
getAlldettagliOrdine()		Restituisce tutti i dettagli ordine.
getDettaglioOrdineById(Long id)		Recupera un dettaglio ordine per ID.
saveDettaglioOrdine(DettaglioOrdine dettaglioOrdine)		Salva un nuovo dettaglio ordine.
deleteDettaglioOrdine(Long id)		Elimina un dettaglio ordine.

<b>Nome Classe</b>	<b>OrdineService</b>	
<b>Descrizione</b>	Rappresenta gli ordini.	
<b>Metodi</b>	salvaOrdine(Ordine ordine) updateOrderStatus(Long ordineId, StatoOrdine stato) findByStato(StatoOrdine stato) findByUtenteId(Long utenteId) getOrdineById(Long ordineId) getAllOrdini() annullaOrdine(Long ordineId)	
<b>Invariante di classe</b>	/	
<b>Nome Metodo</b>		<b>Descrizione</b>
salvaOrdine(Ordine ordine)		Salva un nuovo ordine.
updateOrderStatus(Long ordineId, StatoOrdine stato)		Aggiorna lo stato di un ordine.
findByStato(StatoOrdine stato)		Trova ordini per stato.
findByUtenteId(Long utenteId)		Trova ordini per ID utente.
getOrdineById(Long ordineId)		Recupera un ordine per ID.
getAllOrdini()		Recupera tutti gli ordini.
annullaOrdine(Long ordineId)		Annulla un ordine.

## Repository:



API:



Nome Classe	ApiService
Descrizione	Gestisce le chiamate API per interagire con il backend.
Metodi	getProducts() getProductById(id: int) createProduct(product: Product) updateProduct(id: int, product: Product) deleteProduct(id: int)
Invariante di classe	/

Nome Metodo	Descrizione
getProducts()	Recupera la lista di tutti i prodotti.

<b>getProductById(id: int)</b>	Recupera un prodotto per ID.
<b>createProduct(product: Product)</b>	Crea un nuovo prodotto.
<b>updateProduct(id: int, product: Product)</b>	Modifica un prodotto esistente.
<b>deleteProduct(id: int)</b>	Elimina un prodotto.

<b>Nome Classe</b>	<b>ApiConfig</b>	
<b>Descrizione</b>	Configura le impostazioni per le chiamate API.	
<b>Metodi</b>	initialize()	
<b>Invariante di classe</b>	/	
<b>Nome Metodo</b>		<b>Descrizione</b>
<b>initialize()</b>		Inizializza le impostazioni per le chiamate API.