

Travaini Alessandro



# PROGETTO DI TECNOLOGIE INFORMATICHE PER IL WEB

POLITECNICO DI MILANO  
INGEGNERIA INFORMATICA

---

Codice persona: 10742196

Matricola: 955634

Docente: Fraternali Piero

Anno accademico: 2022-2023

## Sommario

1 Specifica .....	3
1.1 Esercizio 5: carrello con più fornitori e ordine .....	3
1.1.1 Versione HTML pura .....	3
1.1.2 Versione con JavaScript .....	4
2 Analisi della specifica .....	5
2.1 Analisi dei dati .....	5
2.1.1 Codice SQL .....	6
2.1.2 Diagramma E-R .....	8
2.2 Analisi funzionale (HTML) .....	9
2.2.1 Riepilogo dell'analisi funzionale.....	10
2.2.2 Design dell'applicazione web .....	12
2.2.3 Componenti .....	13
2.2.4 Sequence diagrams.....	15
2.3 Analisi funzionale (JavaScript) .....	25
2.3.1 Riepilogo dell'analisi funzionale.....	26
2.3.2 Design dell'applicazione web .....	28
2.3.3 Componenti .....	29
2.3.4 Eventi e azioni.....	30
2.3.5 Controllori ed event handlers.....	31
2.3.6 View e view components.....	32
2.3.7 Sequence diagrams.....	33
3 Strumenti utilizzati.....	40

# 1 Specifica

## 1.1 Esercizio 5: carrello con più fornitori e ordine

### 1.1.1 Versione HTML pura

Un'applicazione di commercio elettronico consente all'utente (acquirente) di visualizzare un catalogo di prodotti venduti da diversi fornitori, inserire prodotti in un carrello della spesa e creare un ordine di acquisto a partire dal contenuto del carrello. Un prodotto ha un codice (campo chiave), un nome, una descrizione, una categoria merceologica e una foto. Lo stesso prodotto (cioè codice prodotto) può essere venduto da più fornitori a prezzi differenti. Un fornitore ha un codice, un nome, una valutazione da 1 a 5 stelle e una politica di spedizione. Un utente ha un nome, un cognome, un'e-mail, una password e un indirizzo di spedizione. La politica di spedizione precisa il prezzo della spedizione in base al numero di articoli ordinati. Ogni fornitore è libero di definire fasce di spesa. Una fascia di spesa ha un numero minimo, un numero massimo e un prezzo. Ad esempio: da 1 a 3 articoli 15€, da 4 a 10 articoli 20€, oltre a 10 articoli, ecc. Oltre alla fascia di spesa, il fornitore può anche indicare un importo in euro oltre al quale la spedizione è gratuita. Se il totale supera la soglia per la gratuità della spedizione, la spedizione è gratuita indipendentemente dal numero di articoli. Dopo il login, l'utente accede a una pagina HOME che mostra (come tutte le altre pagine) un menù con i link HOME, CARRELLO, ORDINI, un campo di ricerca e una lista degli ultimi cinque prodotti visualizzati dall'utente. Se l'utente non ha visualizzato almeno cinque prodotti, la lista è completata con prodotti in offerta scelti a caso in una categoria di default. L'utente può inserire una parola chiave di ricerca nel campo di input e premere INVIO. A seguito dell'invio compare una pagina RISULTATI con prodotti che contengono la chiave di ricerca nel nome o nella descrizione. L'elenco mostra solo il codice, il nome del prodotto e il prezzo minimo di vendita del prodotto da parte dei fornitori che lo vendono (lo stesso prodotto può essere venduto da diversi fornitori a prezzi diversi e l'elenco mostra il minimo valore di tali prezzi). L'elenco è ordinato in modo crescente in base al prezzo minimo di vendita del prodotto da parte dei fornitori che lo offrono. L'utente può selezionare mediante un click un elemento dell'elenco e visualizzare nella stessa pagina i dati completi e l'elenco dei fornitori che lo vendono a vari prezzi (questa azione rende il prodotto "visualizzato"). Per ogni fornitore in tale elenco compaiono: nome, valutazione, prezzo unitario, fasce di spesa di spedizione, importo minimo della spedizione gratuita e il numero dei prodotti e valore totale dei prodotti di quel fornitore che l'utente ha già messo nel carrello. Accanto all'offerta di ciascun fornitore compare un campo di input intero (quantità) e un bottone METTI NEL CARRELLO. L'inserimento nel carrello di una quantità maggiore di zero di prodotti comporta l'aggiornamento del contenuto del carrello e la visualizzazione della pagina CARRELLO. Questa mostra i prodotti inseriti, raggruppati per fornitore. Per ogni fornitore nel carrello si vedono la lista dei prodotti, il prezzo totale dei prodotti e il prezzo della spedizione calcolato in base alla politica del fornitore. Per ogni fornitore compare un bottone ORDINA. Premere il bottone comporta l'eliminazione dei prodotti del fornitore dal carrello e la creazione di un ordine corrispondente. Un ordine ha un codice, il nome del fornitore, l'elenco dei prodotti, un valore totale composto dalla somma del valore dei prodotti e delle spese di spedizione, una data di spedizione e l'indirizzo di spedizione dell'utente. I valori degli attributi di un ordine sono memorizzati esplicitamente nella base di dati indipendentemente dai dati del carrello. In ogni momento l'utente può accedere tramite il menu alle pagine HOME, ORDINI e CARRELLO. La pagina ORDINI mostra l'elenco ordinato per data decrescente degli ordini con tutti i dati associati. L'applicazione NON salva il carrello nella base di dati ma solo gli ordini.

### 1.1.2 Versione con JavaScript

Si realizzi un'applicazione client server web che estende e/o modifica le specifiche precedenti come segue:

- Dopo il login dell'utente, l'intera applicazione è realizzata con un'unica pagina.
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica del contenuto da aggiornare a seguito dell'evento.
- L'applicazione memorizza il contenuto del carrello a lato client.
- Nella pagina RISULTATI l'elenco dettagliato dei prodotti già nel carrello da parte di un fornitore compare mediante una finestra sovrapposta quando si passa con il mouse sopra il numero che indica quanti prodotti del medesimo fornitore sono già nel carrello.

## 2 Analisi della specifica

### 2.1 Analisi dei dati

Legenda:

- **Entità;**
- **Relazioni;**
- **Attributi;**

Un'applicazione di commercio elettronico consente all'**utente** (acquirente) di visualizzare un catalogo di prodotti venduti da diversi fornitori, inserire prodotti in un carrello della spesa e creare un ordine di acquisto a partire dal contenuto del carrello. Un **prodotto** ha un **codice** (campo chiave), un **nome**, una **descrizione**, una **categoria merceologica** e una **foto**. Lo stesso prodotto (cioè codice prodotto) può essere **venduto** da più fornitori a **prezzi** differenti. Un **fornitore** ha un **codice**, un **nome**, una **valutazione** da 1 a 5 stelle e una politica di spedizione. Un utente ha un **nome**, un **cognome**, un'**e-mail**, una **password** e un **indirizzo di spedizione**. La **politica di spedizione** precisa il prezzo della spedizione in base al numero di articoli ordinati. Ogni fornitore è libero di definire fasce di spesa. Una **fascia di spesa** ha un numero minimo, un numero massimo e un prezzo. Ad esempio: da 1 a 3 articoli 15€, da 4 a 10 articoli 20€, oltre a 10 articoli, ecc. Oltre alla fascia di spesa, il fornitore può anche indicare un importo in euro oltre al quale la spedizione è gratuita. Se il totale supera la soglia per la gratuità della spedizione, la spedizione è gratuita indipendentemente dal numero di articoli. Un **ordine** ha un **codice**, il **nome del fornitore**, l'**elenco dei prodotti**, un **valore totale** composto dalla somma del valore dei prodotti e delle spese di spedizione, una **data di spedizione** e l'**indirizzo di spedizione** dell'utente. I valori degli attributi di un ordine sono memorizzati esplicitamente nella base di dati indipendentemente dai dati del carrello.

### 2.1.1 Codice SQL

```
CREATE TABLE `indirizzo` (  
  `Id` int NOT NULL AUTO_INCREMENT,  
  `Citta` varchar(32) NOT NULL,  
  `Via` varchar(32) NOT NULL,  
  `Cap` char(5) NOT NULL,  
  `Numero` int NOT NULL,  
  PRIMARY KEY (`Id`)  
);
```

```
CREATE TABLE `utente` (  
  `IdUtente` int NOT NULL AUTO_INCREMENT,  
  `Nome` varchar(16) NOT NULL,  
  `Cognome` varchar(16) NOT NULL,  
  `Email` varchar(64) NOT NULL,  
  `Password` varchar(16) NOT NULL,  
  `IdIndirizzo` int NOT NULL,  
  PRIMARY KEY (`IdUtente`),  
  KEY `fk_indirizzo_idx` (`IdIndirizzo`),  
  CONSTRAINT `fk_indirizzo` FOREIGN KEY (`IdIndirizzo`) REFERENCES `indirizzo`  
  (`Id`)  
);
```

```
CREATE TABLE `politicaSpedizione` (  
  `Id` int NOT NULL AUTO_INCREMENT,  
  `Soglia` int DEFAULT NULL,  
  PRIMARY KEY (`Id`)  
);
```

```
CREATE TABLE `fascia` (  
  `IdFascia` int NOT NULL AUTO_INCREMENT,  
  `Prezzo` float NOT NULL,  
  `Min` int NOT NULL,  
  `Max` int NOT NULL,  
  PRIMARY KEY (`IdFascia`)  
);
```

```
CREATE TABLE `fornitore` (  
  `Id` int NOT NULL AUTO_INCREMENT,  
  `Nome` varchar(16) NOT NULL,  
  `Valutazione` varchar(8) NOT NULL,  
  `IdPoliticaSpedizione` int NOT NULL,  
  PRIMARY KEY (`Id`),  
  KEY `fk_politica_fornitore_idx` (`IdPoliticaSpedizione`),  
  CONSTRAINT `fk_politica_fornitore` FOREIGN KEY (`IdPoliticaSpedizione`)  
  REFERENCES `politicaSpedizione` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE `prodotto` (  
  `Id` int NOT NULL AUTO_INCREMENT,  
  `Nome` varchar(32) NOT NULL,  
  `Descrizione` varchar(1024) NOT NULL,  
  `Categoria` varchar(16) NOT NULL,  
  `Immagine` longblob,  
  PRIMARY KEY (`Id`)  
);
```

```
CREATE TABLE `ordine` (  
  `Id` int NOT NULL AUTO_INCREMENT,
```

```

`Totale` float NOT NULL,
`Data` timestamp DEFAULT NULL,
`IdIndirizzo` int NOT NULL,
`IdUtente` int NOT NULL,
`IdFornitore` int NOT NULL,
PRIMARY KEY (`Id`),
KEY `fk_indirizzo_ordine_idx` (`IdIndirizzo`),
KEY `fk_utente_ordine_idx` (`IdUtente`),
KEY `fk_fornitore_ordine_idx` (`IdFornitore`),
CONSTRAINT `fk_fornitore_ordine` FOREIGN KEY (`IdFornitore`) REFERENCES
`fornitore` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `fk_indirizzo_ordine` FOREIGN KEY (`IdIndirizzo`) REFERENCES
`indirizzo` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `fk_utente_ordine` FOREIGN KEY (`IdUtente`) REFERENCES `utente`
(`IdUtente`) ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE `contenuto` (
`IdOrdine` int NOT NULL,
`IdProdotto` int NOT NULL,
`Quantita` int NOT NULL,
PRIMARY KEY (`IdOrdine`,`IdProdotto`),
KEY `fk_ordine_contenuto_idx` (`IdOrdine`),
KEY `fk_prodotto_contenuto_idx` (`IdProdotto`),
CONSTRAINT `fk_ordine_contenuto` FOREIGN KEY (`IdOrdine`) REFERENCES
`ordine` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `fk_prodotto_contenuto` FOREIGN KEY (`IdProdotto`) REFERENCES
`prodotto` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE `composizioneFasce` (
`IdPoliticaComp` int NOT NULL,
`IdFasceComp` int NOT NULL,
PRIMARY KEY (`IdPoliticaComp`,`IdFasceComp`),
KEY `fk_politica_composizione_idx` (`IdPoliticaComp`),
KEY `fk_range_composizione_idx` (`IdFasceComp`),
CONSTRAINT `fk_politica_composizione` FOREIGN KEY (`IdPoliticaComp`)
REFERENCES `politicaSpedizione` (`Id`) ON DELETE CASCADE ON UPDATE
CASCADE,
CONSTRAINT `fk_range_composizione` FOREIGN KEY (`IdFasceComp`) REFERENCES
`fascia` (`IdFascia`) ON DELETE CASCADE ON UPDATE
CASCADE
);

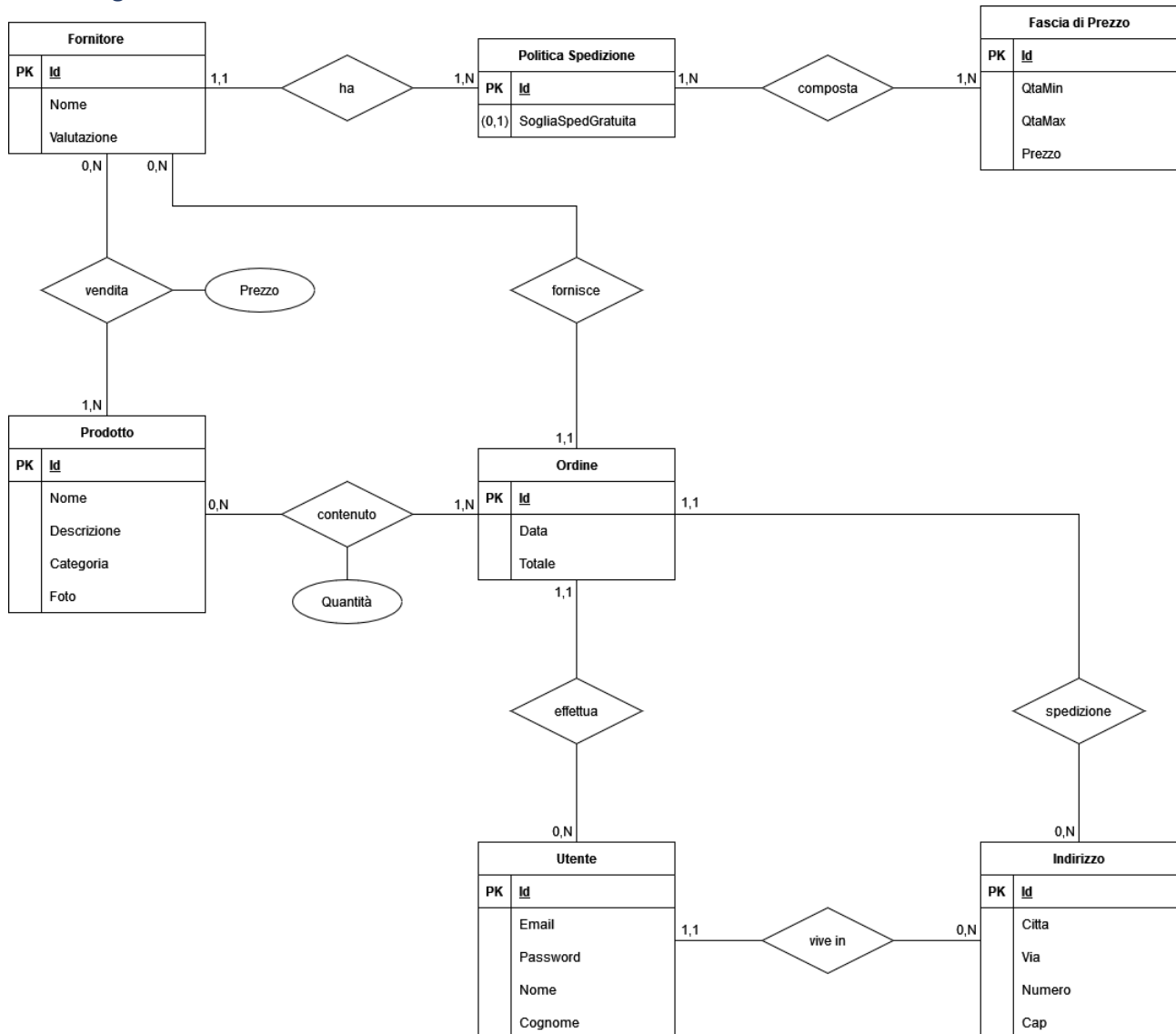
```

```

CREATE TABLE `vendita` (
`IdFornitore` int NOT NULL,
`IdProdotto` int NOT NULL,
`Prezzo` float NOT NULL,
PRIMARY KEY (`IdFornitore`,`IdProdotto`),
KEY `fk_fornitore_vendita_idx` (`IdFornitore`),
KEY `fk_prodotto_vendita_idx` (`IdProdotto`),
CONSTRAINT `fk_fornitore_vendita` FOREIGN KEY (`IdFornitore`) REFERENCES
`fornitore` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `fk_prodotto_vendita` FOREIGN KEY (`IdProdotto`) REFERENCES
`prodotto` (`Id`) ON DELETE CASCADE ON UPDATE CASCADE
);

```

## 2.1.2 Diagramma E-R





## 2.2 Analisi funzionale (HTML)

Legenda:

- **Pagine;**
- View components;
- **Azioni** (eseguite dalle servlet);
- **Eventi** (provenienti dall'utente);

Dopo il **login**<sup>1</sup>, l'utente accede a una pagina **HOME** che mostra (come tutte le altre pagine) un menù con i link HOME, CARRELLO, ORDINI, un campo di ricerca e una lista degli ultimi cinque prodotti visualizzati dall'utente. Se l'utente non ha visualizzato almeno cinque prodotti, la lista è completata con prodotti in offerta scelti a caso in una categoria di default. L'utente può **inserire una parola chiave di ricerca** nel campo di input e **premere INVIO**. A seguito dell'invio **compare una pagina RISULTATI** con prodotti che contengono la chiave di ricerca nel nome o nella descrizione. L'elenco mostra solo il codice, il nome del prodotto e il prezzo minimo di vendita del prodotto da parte dei fornitori che lo vendono (lo stesso prodotto può essere venduto da diversi fornitori a prezzi diversi e l'elenco mostra il minimo valore di tali prezzi). L'elenco è ordinato in modo crescente in base al prezzo minimo di vendita del prodotto da parte dei fornitori che lo offrono. L'utente può **selezionare mediante un click** un elemento dell'elenco e **visualizzare nella stessa pagina** i dati completi e l'elenco dei fornitori che lo vendono a vari prezzi (questa azione **rende il prodotto "visualizzato"**). Per ogni fornitore in tale elenco compaiono: nome, valutazione, prezzo unitario, fasce di spesa di spedizione, importo minimo della spedizione gratuita e il numero dei prodotti e valore totale dei prodotti di quel fornitore che l'utente ha già messo nel carrello. Accanto all'offerta di ciascun fornitore compare un campo di input intero (quantità) e un bottone METTI NEL CARRELLO. L'**inserimento nel carrello** di una quantità maggiore di zero di prodotti comporta l'**aggiornamento del contenuto del carrello** e la **visualizzazione della pagina CARRELLO**. Questa mostra i prodotti inseriti, raggruppati per fornitore. Per ogni fornitore nel carrello si vedono la lista dei prodotti, il prezzo totale dei prodotti e il prezzo della spedizione calcolato in base alla politica del fornitore. Per ogni fornitore compare un bottone ORDINA. **Premere il bottone** comporta l'**eliminazione dei prodotti del fornitore dal carrello** e la **creazione di un ordine corrispondente**. In ogni momento l'utente può **accedere tramite il menu** alle pagine HOME, ORDINI e CARRELLO. La pagina **ORDINI** mostra l'elenco ordinato per data decrescente degli ordini con tutti i dati associati.

Completamento specifiche:

- Il form di login è la pagina iniziale;
- La quantità di prodotti nel carrello e negli ordini non può essere negativa;
- La quantità di prodotti nel carrello e negli ordini è minore alle 999 unità;
- I dati di spedizione per un ordine devono essere inseriti;
- Il **click sul pulsante ORDINA** **visualizza la sezione ORDINI**;

Aggiunta di specifiche:

- Messaggio di benvenuto personalizzato;
- Il **click su un prodotto nella home** ne **effettua la ricerca e mostra la pagina RISULTATI** di tale prodotto;
- Pulsante di logout;
- Pagina di registrazione alla piattaforma;
- Pulsante "Indietro" nella pagina di registrazione;

---

<sup>1</sup> Il form di login è contenuto nella pagina di login.

### 2.2.1 Ricapitolazione dell'analisi funzionale

#### Pagine e view components:

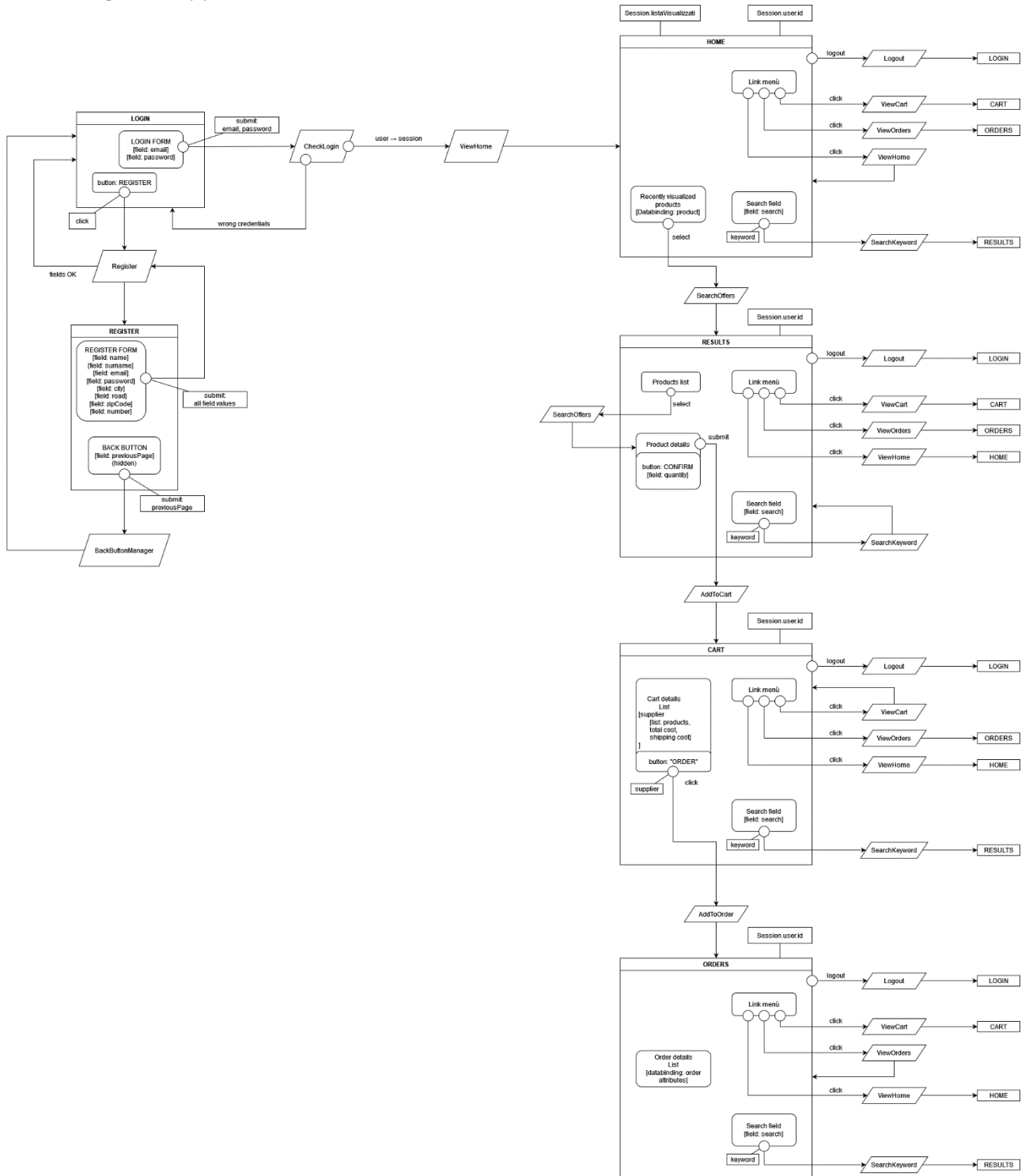
- **Login:**
  - Form di login;
  - Pulsante LOGIN;
  - Link per registrazione;
- **Register:**
  - Form di registrazione;
  - Pulsante REGISTER;
  - Pulsante "Indietro";
- **Home:**
  - Menù:
    - Pulsante HOME;
    - Pulsante CART;
    - Pulsante ORDERS;
    - Campo di ricerca;
    - Pulsante SEARCH;
    - Messaggio di benvenuto personalizzato;
    - Pulsante LOGOUT;
  - Lista degli ultimi cinque prodotti visualizzati;
- **Results:**
  - Menù;
  - Lista dei prodotti corrispondenti alla ricerca;
- **Cart:**
  - Menù;
  - Lista dei carrelli:
    - Lista dei prodotti per ogni carrello;
    - Pulsante "ORDER";
- **Orders:**
  - Menù;
  - Lista degli ordini;

#### Eventi e azioni:

- **Pressione pulsante di LOGIN:**
  - Verifica credenziali;
- **Pressione link di registrazione:**
  - Caricamento pagina di registrazione;
- **Pressione pulsante "Indietro" nella pagina di registrazione:**
  - Caricamento pagina di login;
- **Pressione pulsante REGISTER:**
  - Creazione utente;
  - Caricamento pagina di login;
- **Pressione tasto SEARCH:**
  - Ricerca con parola chiave;
  - Caricamento pagina RESULTS;

- Click su un prodotto:
  - Visualizzazione nella stessa pagina dei dati completi del prodotto e l'elenco dei fornitori con tale oggetto;
  - Aggiunta prodotto alla lista di prodotti visualizzati;
- Click su pulsante ADD TO CART:
  - Aggiornamento carrello;
  - Visualizzazione carrello;
- Click su pulsante ORDER:
  - Creazione dell'ordine;
  - Rimozione dei prodotti dal carrello;
  - Visualizzazione pagina ORDERS;
- Click HOME:
  - Caricamento pagina HOME;
- Click CART:
  - Caricamento pagina CART;
- Click ORDERS:
  - Caricamento pagina ORDERS;
- Click LOGOUT:
  - Logout;

## 2.2.2 Design dell'applicazione web



### 2.2.3 Componenti

Model objects (beans):

- Address;
- Cart;
- Order;
- Product;
- ShippingRange;
- Supplier;
- User;

Data Access Objects (DAO):

- AddressDAO:
  - getAddressById(int id);
  - getAddressByParameters(String citta, String via, String cap, int numero);
  - addAddress(String citta, String via, String cap, int numero)
- OrderDAO:
  - getOrdersOfUserById(int id);
  - addOrder(float total, int addressId, int userId, int supplierId, List<Product> products);
- ProductDAO:
  - getProductById(int id);
  - getProductOffersById(int id);
  - getProductByIdAndSupplier(int productId, int supplierId);
  - getProductsByKeyword(String keyword);
  - getProducts(Queue<Integer> alreadyDisplayed, int quantity);
  - existsProduct(int id);
- SupplierDAO:
  - getSupplierById(int id);
  - existsSupplier(int id);
- UserDAO:
  - checkCredentials(String email, String password);
  - addUser(String name, String surname, String email, String password, int addressId);

Controllori (servlets):

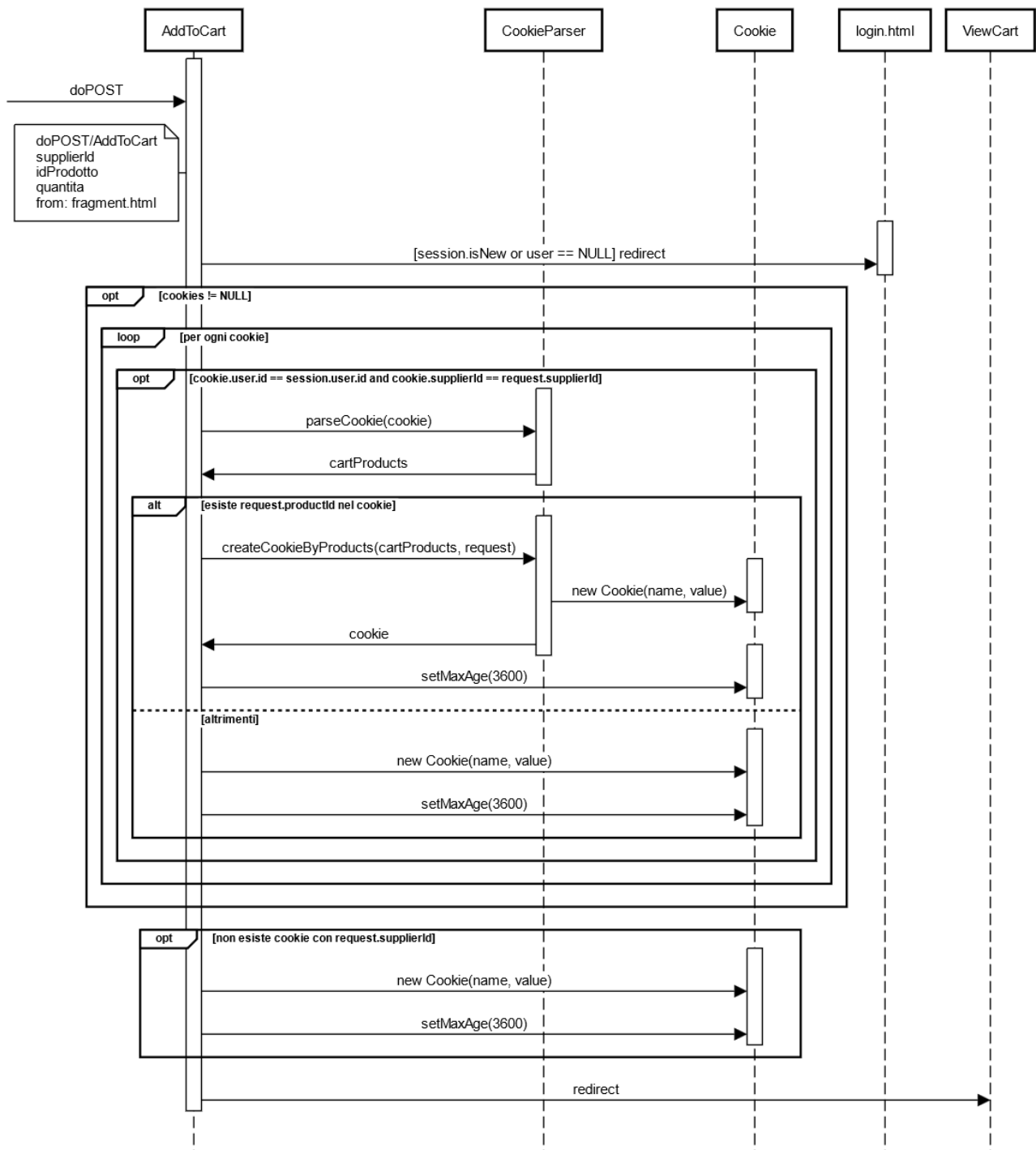
- AddToCart;
- AddToOrder;
- BackButtonManager;
- CheckLogin;
- Logout;
- Register;
- SearchKeyword;
- SearchOffers;
- SearchProduct;
- ViewCart;
- ViewHome;
- ViewOrders;

Pagine (templates):

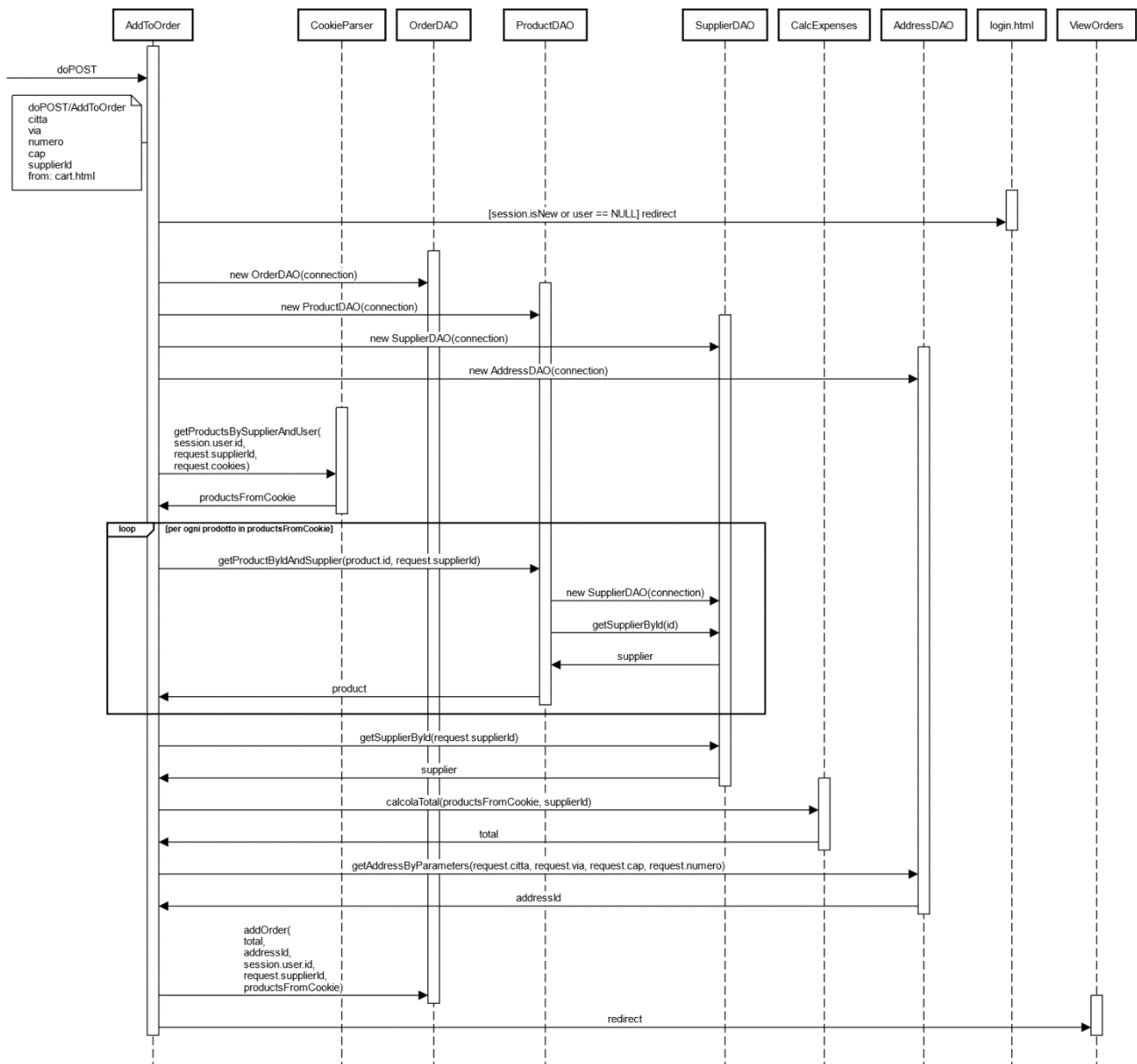
- Login;
- Cart;
- Fragment;
- Home;
- Orders;
- Register;
- Results;

## 2.2.4 Sequence diagrams

### AddToCart:

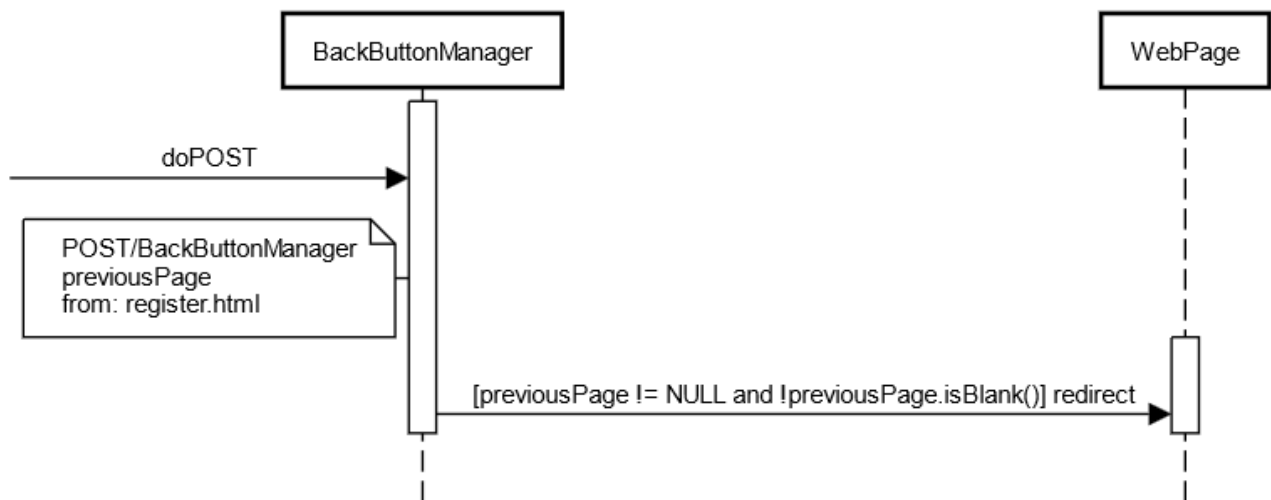


## AddToOrder:

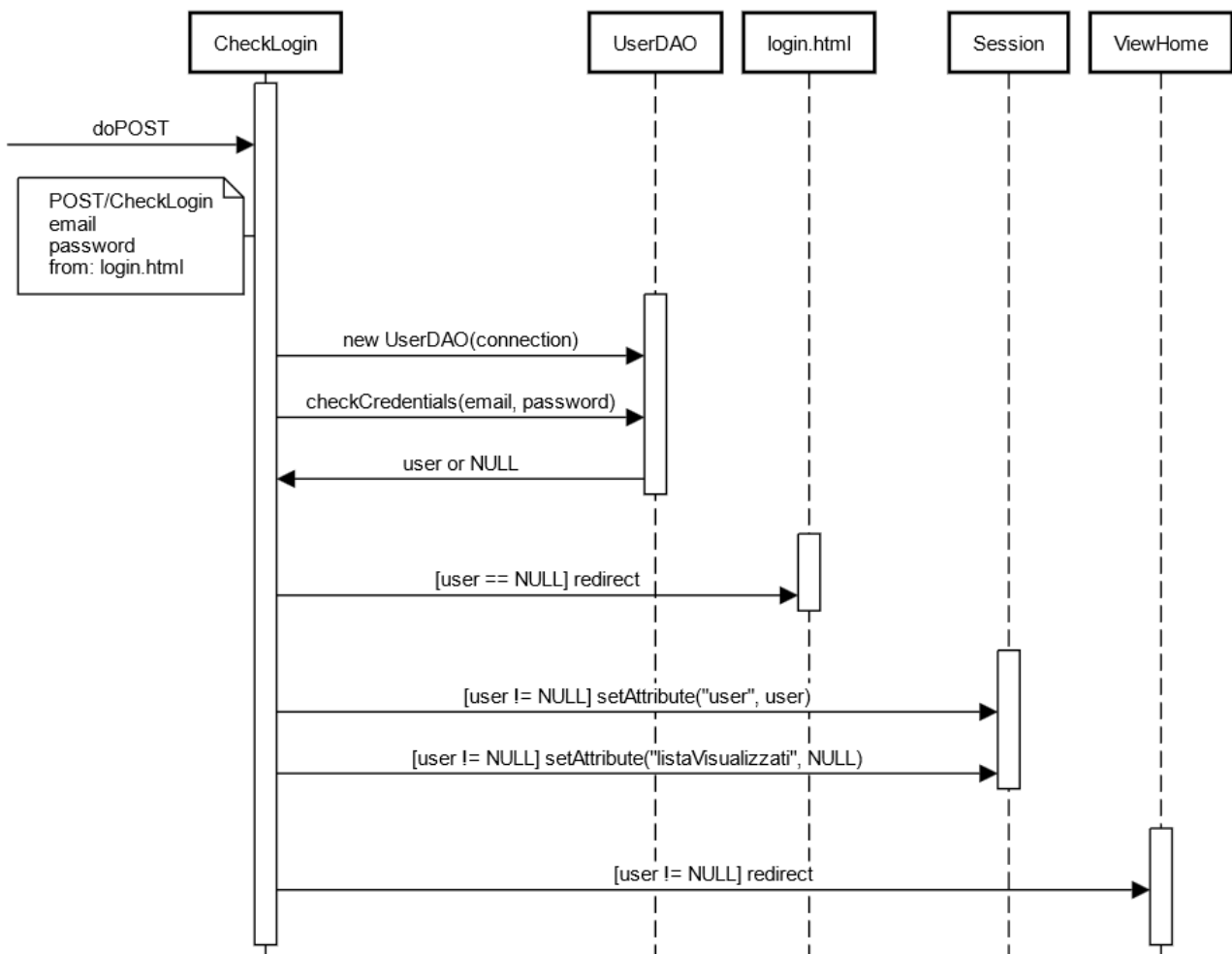




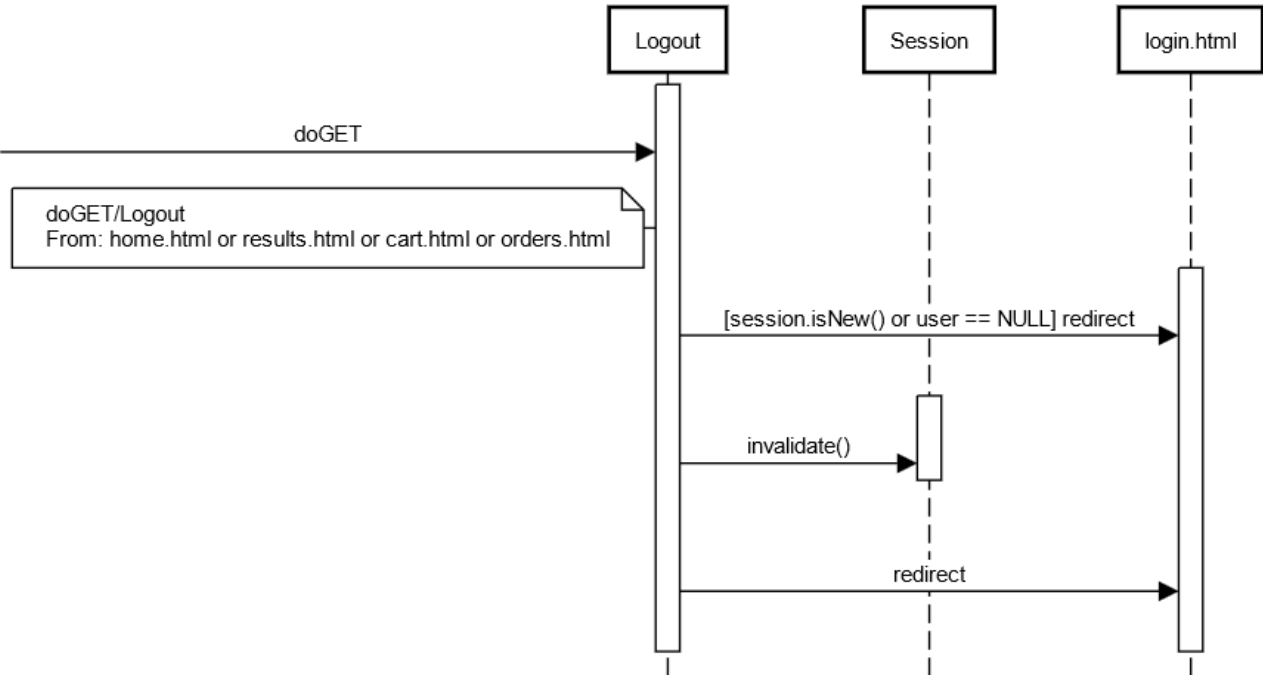
### BackButtonManager:



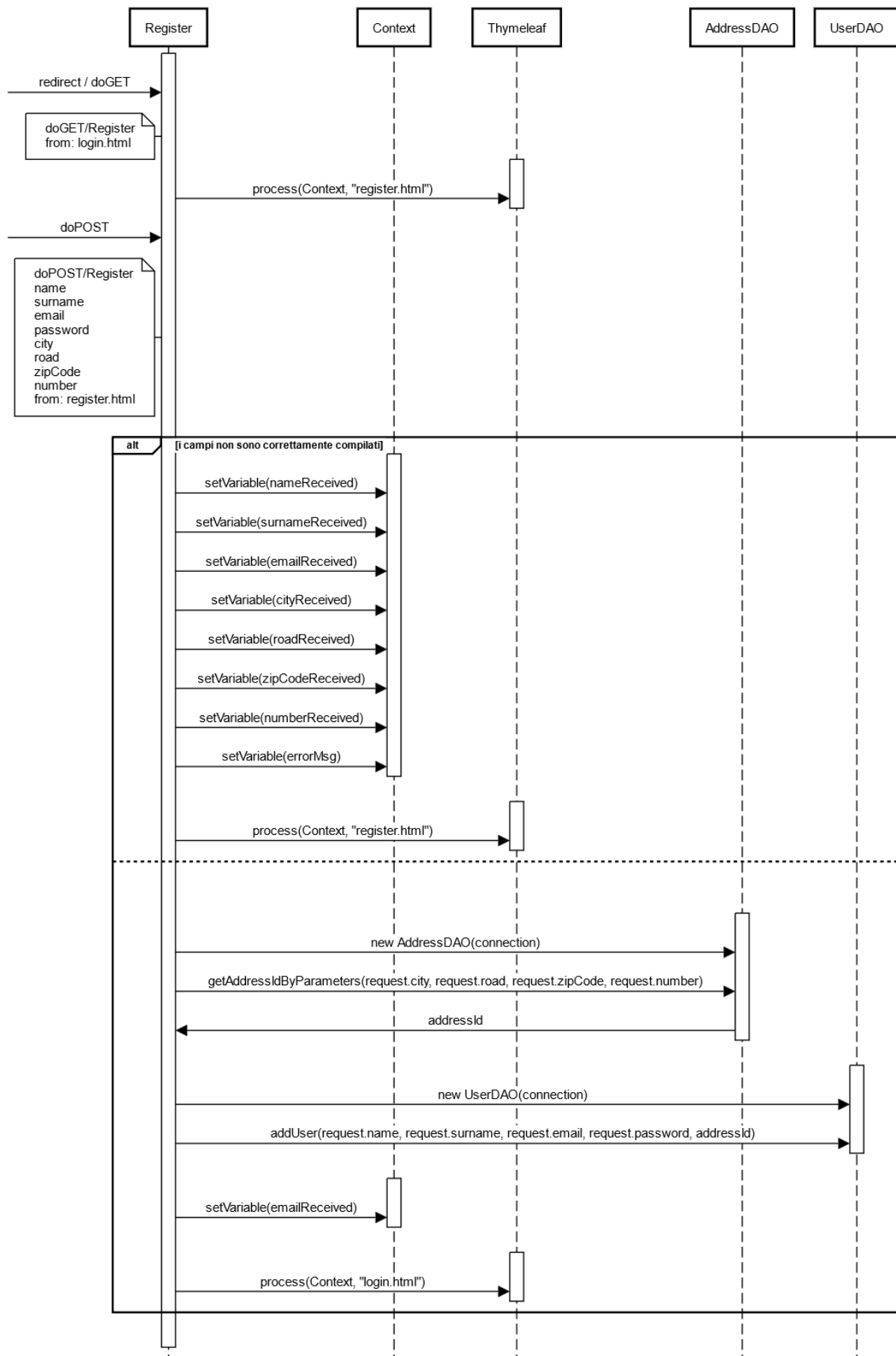
### CheckLogin:



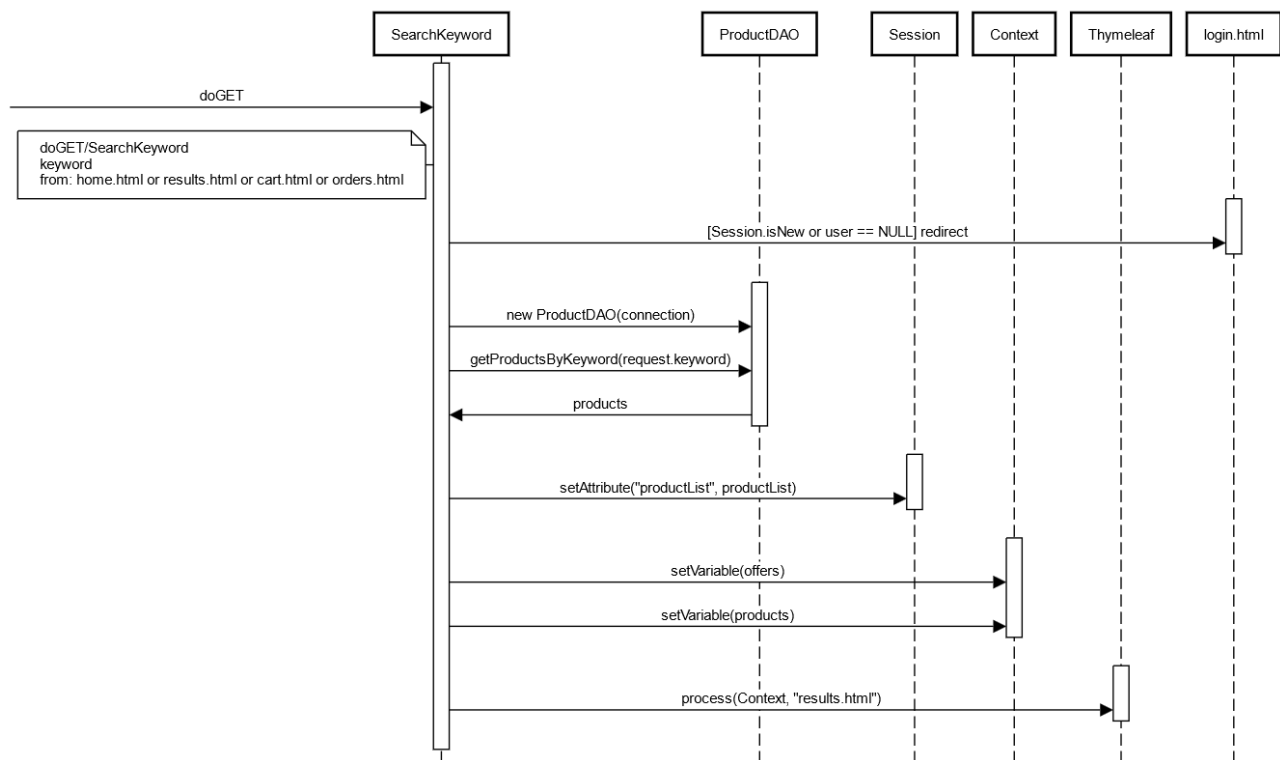
**Logout:**



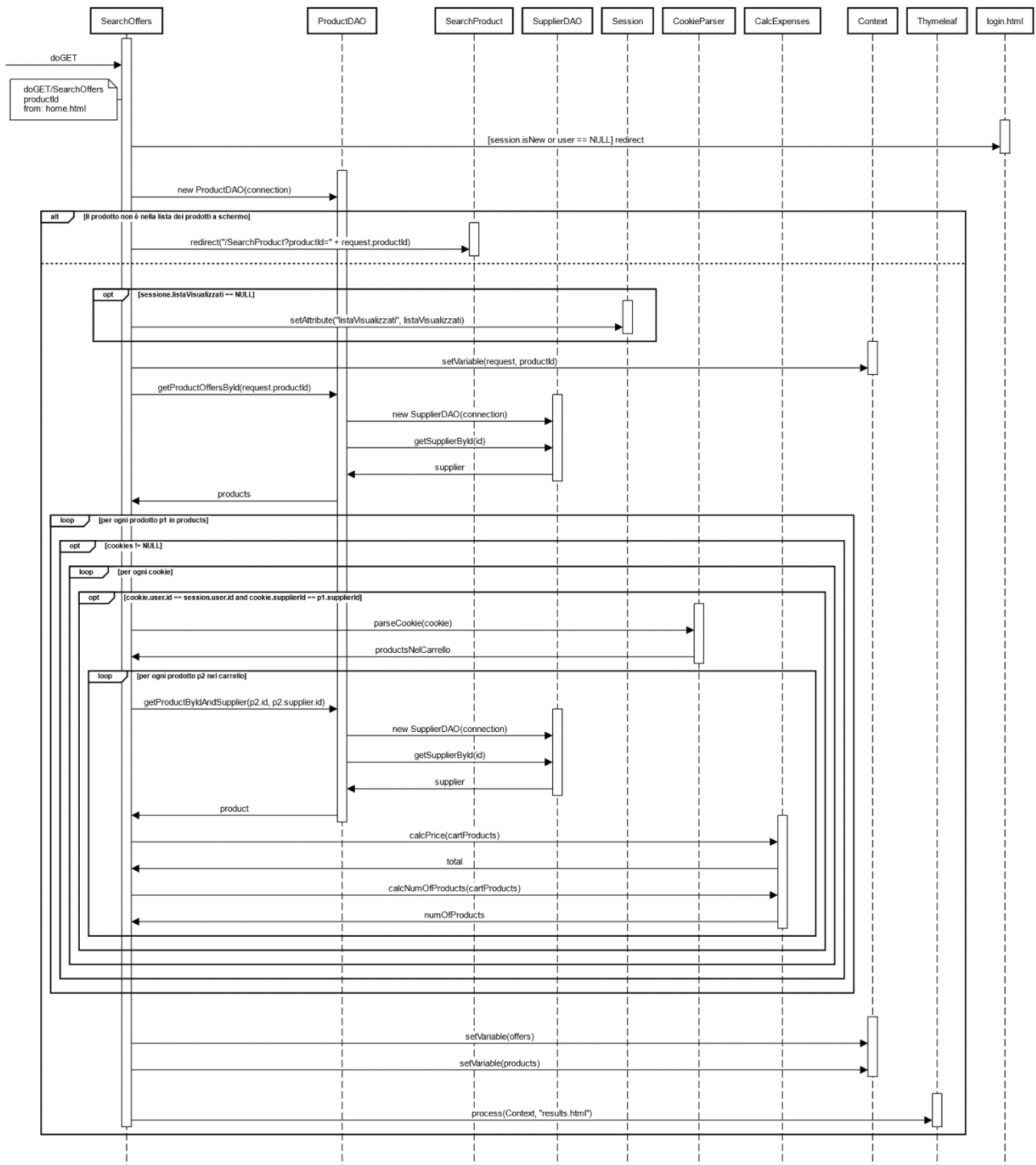
## Register:



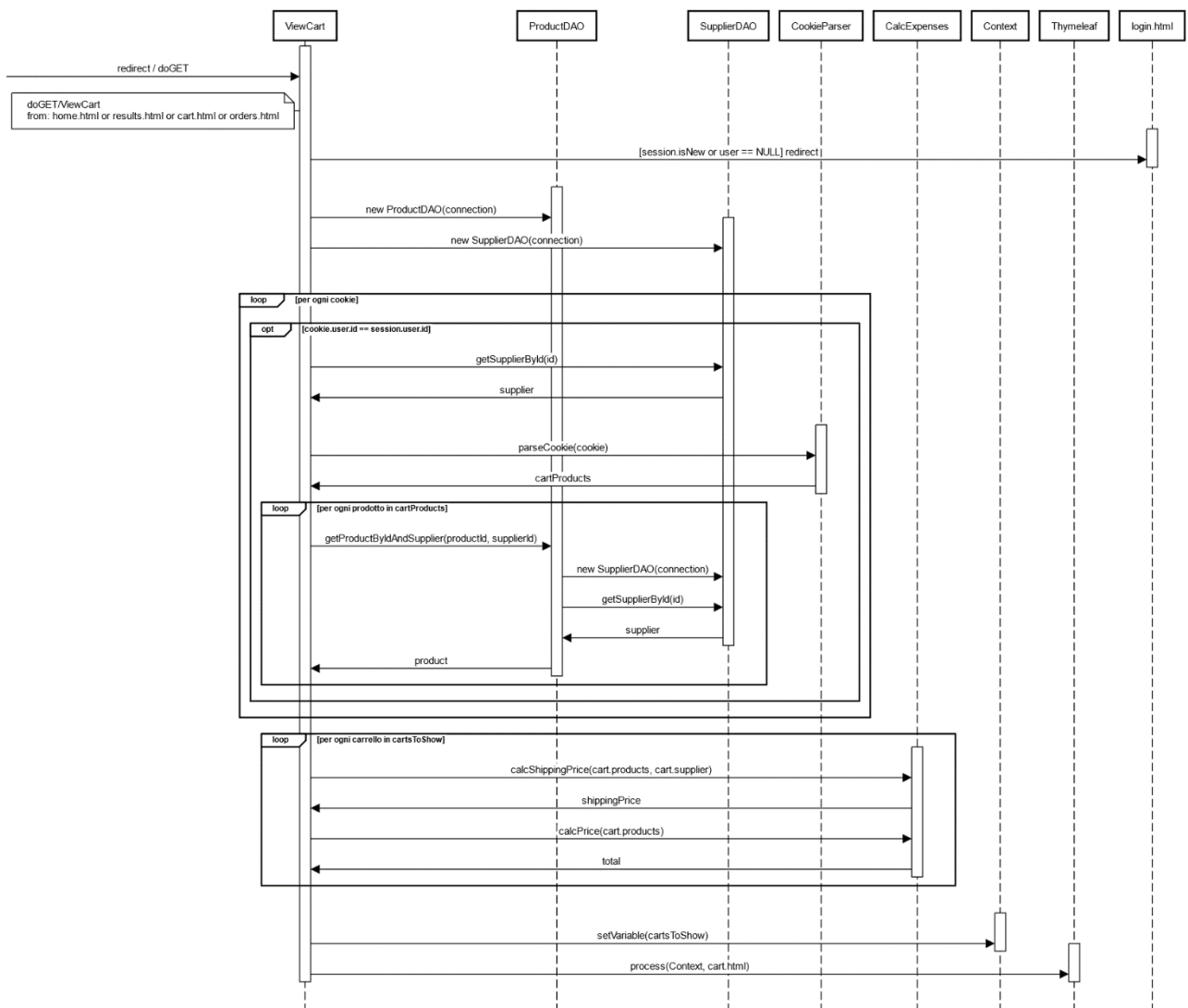
**SearchKeyword:**



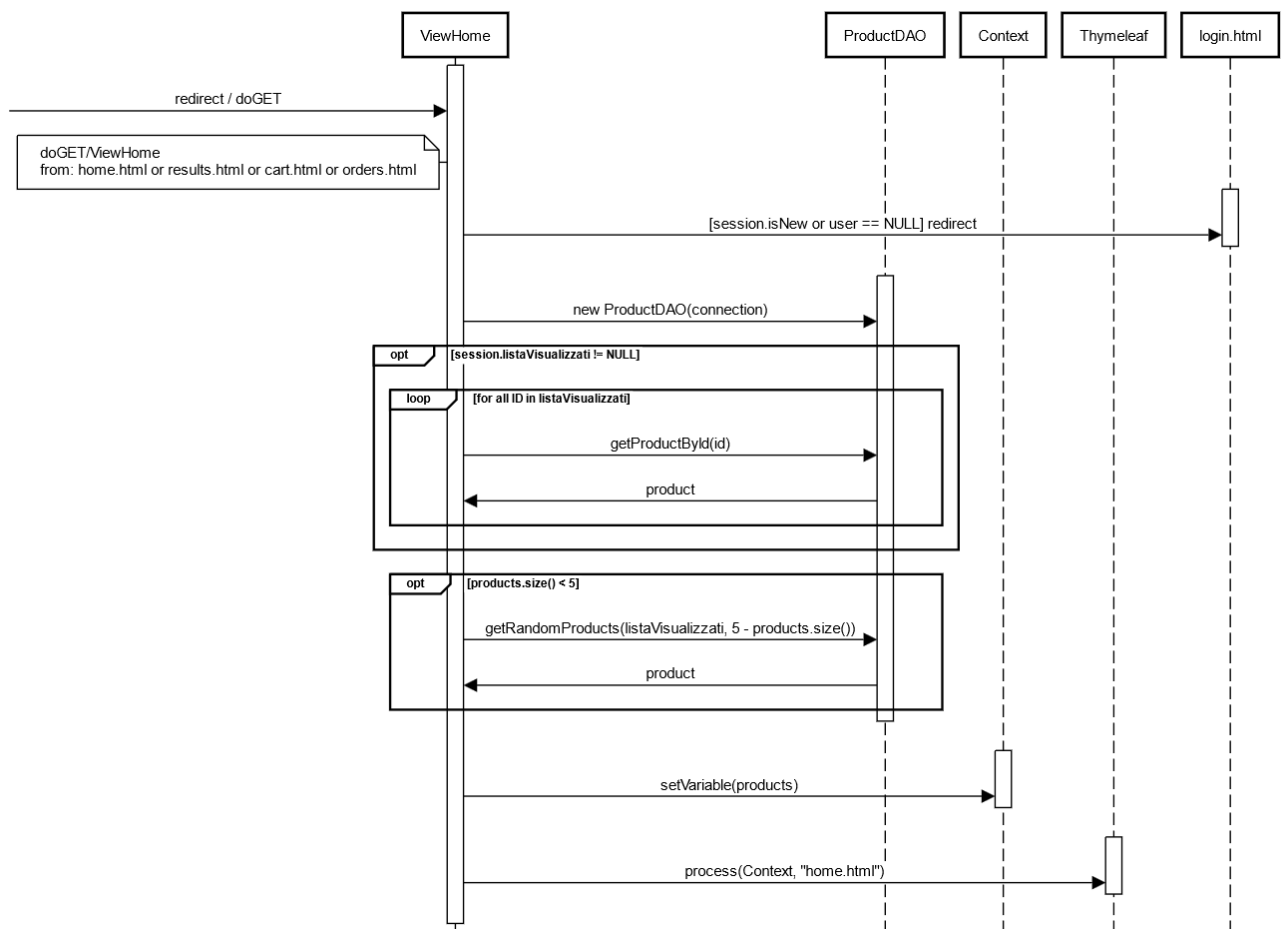
## SearchOffers:



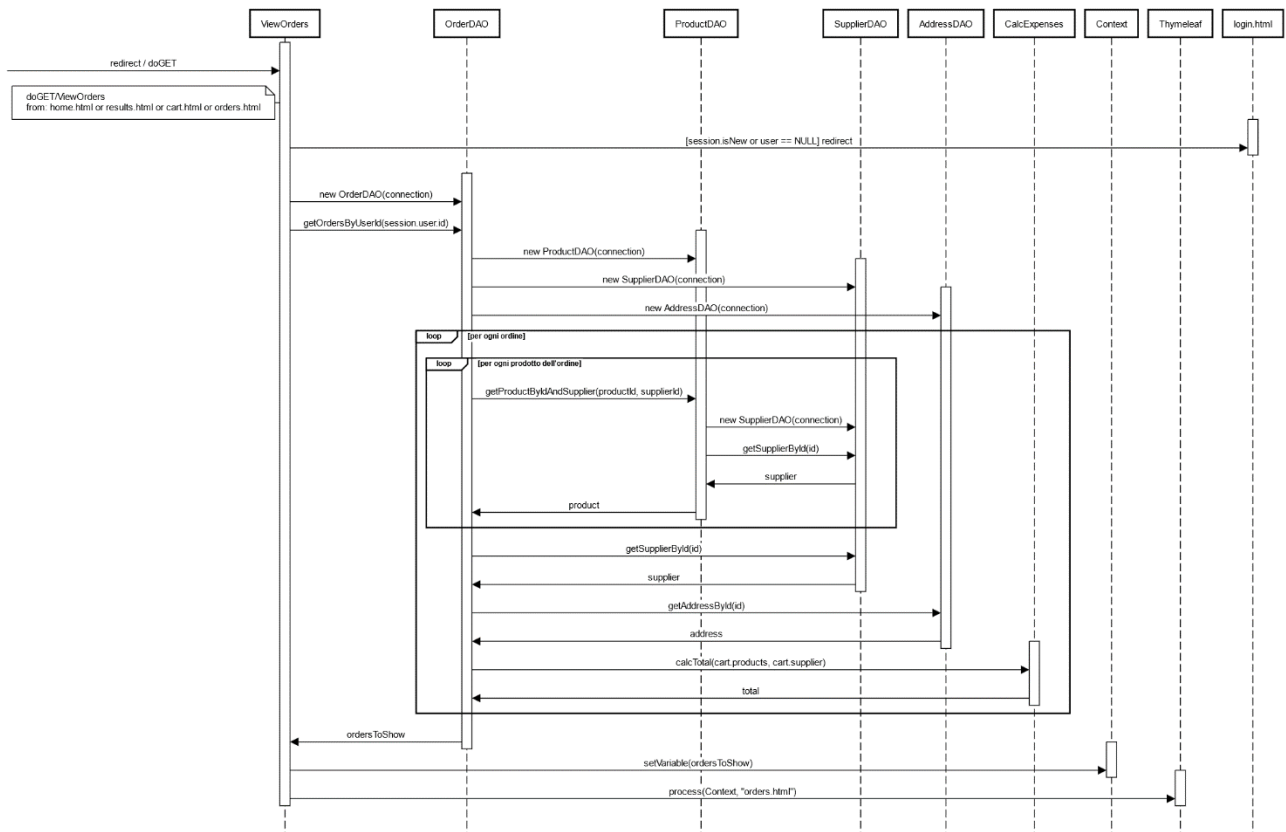
## ViewCart:



## ViewHome:



## ViewOrders:





## 2.3 Analisi funzionale (JavaScript)

Legenda:

- **Pagine;**
- View components;
- **Azioni** (eseguite dalle servlet);
- **Eventi** (provenienti dall'utente);

Dopo il **login**<sup>2</sup>, l'utente accede a una pagina **HOME** che mostra (come tutte le altre pagine) un menù con i link HOME, CARRELLO, ORDINI, un campo di ricerca e una lista degli ultimi cinque prodotti visualizzati dall'utente. Se l'utente non ha visualizzato almeno cinque prodotti, la lista è completata con prodotti in offerta scelti a caso in una categoria di default. L'utente può **inserire una parola chiave di ricerca** nel campo di input e **premere INVIO**. A seguito dell'invio **compare una pagina RISULTATI** con prodotti che contengono la chiave di ricerca nel nome o nella descrizione. L'elenco mostra solo il codice, il nome del prodotto e il prezzo minimo di vendita del prodotto da parte dei fornitori che lo vendono (lo stesso prodotto può essere venduto da diversi fornitori a prezzi diversi e l'elenco mostra il minimo valore di tali prezzi). L'elenco è ordinato in modo crescente in base al prezzo minimo di vendita del prodotto da parte dei fornitori che lo offrono. L'utente può **selezionare mediante un click** un elemento dell'elenco e **visualizzare nella stessa pagina** i dati completi e l'elenco dei fornitori che lo vendono a vari prezzi (questa azione **rende il prodotto "visualizzato"**). Per ogni fornitore in tale elenco compaiono: nome, valutazione, prezzo unitario, fasce di spesa di spedizione, importo minimo della spedizione gratuita e il numero dei prodotti e valore totale dei prodotti di quel fornitore che l'utente ha già messo nel carrello. Accanto all'offerta di ciascun fornitore compare un campo di input intero (quantità) e un bottone METTI NEL CARRELLO. L'**inserimento nel carrello** di una quantità maggiore di zero di prodotti comporta l'**aggiornamento del contenuto del carrello** e la **visualizzazione della pagina CARRELLO**. Questa mostra i prodotti inseriti, raggruppati per fornitore. Per ogni fornitore nel carrello si vedono la lista dei prodotti, il prezzo totale dei prodotti e il prezzo della spedizione calcolato in base alla politica del fornitore. Per ogni fornitore compare un bottone ORDINA. **Premere il bottone** comporta l'**eliminazione dei prodotti del fornitore dal carrello** e la **creazione di un ordine corrispondente**. In ogni momento l'utente può **accedere tramite il menu** alle pagine HOME, ORDINI e CARRELLO. La pagina ORDINI **mostra** l'elenco ordinato per data decrescente degli ordini con tutti i dati associati.

Si realizzi un'applicazione client server web che estende e/o modifica le specifiche precedenti come segue:

- Dopo il login dell'utente, l'intera applicazione è realizzata con un'unica pagina.
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica del contenuto da aggiornare a seguito dell'evento.
- L'applicazione memorizza il contenuto del carrello a lato client.
- Nella pagina RISULTATI l'elenco dettagliato dei prodotti già nel carrello da parte di un fornitore **compare** mediante una finestra sovrapposta quando si **passa con il mouse sopra il numero che indica quanti prodotti del medesimo fornitore sono già nel carrello**.

Completamento specifiche:

- Il form di login è la pagina iniziale;
- La quantità di prodotti nel carrello e negli ordini non può essere negativa;
- La quantità di prodotti nel carrello e negli ordini è minore alle 999 unità;
- I dati di spedizione per un ordine devono essere inseriti;
- Il **click sul pulsante ORDINA visualizza la sezione ORDINI**;

---

<sup>2</sup> Il form di login è contenuto nella pagina di login.

Aggiunta di specifiche:

- Messaggio di benvenuto personalizzato;
- Il **click su un prodotto nella home** ne **effettua la ricerca e mostra la pagina RISULTATI** di tale prodotto;
- Pulsante di logout;

### 2.3.1 Ricapitolazione dell'analisi funzionale

#### Pagine e view components:

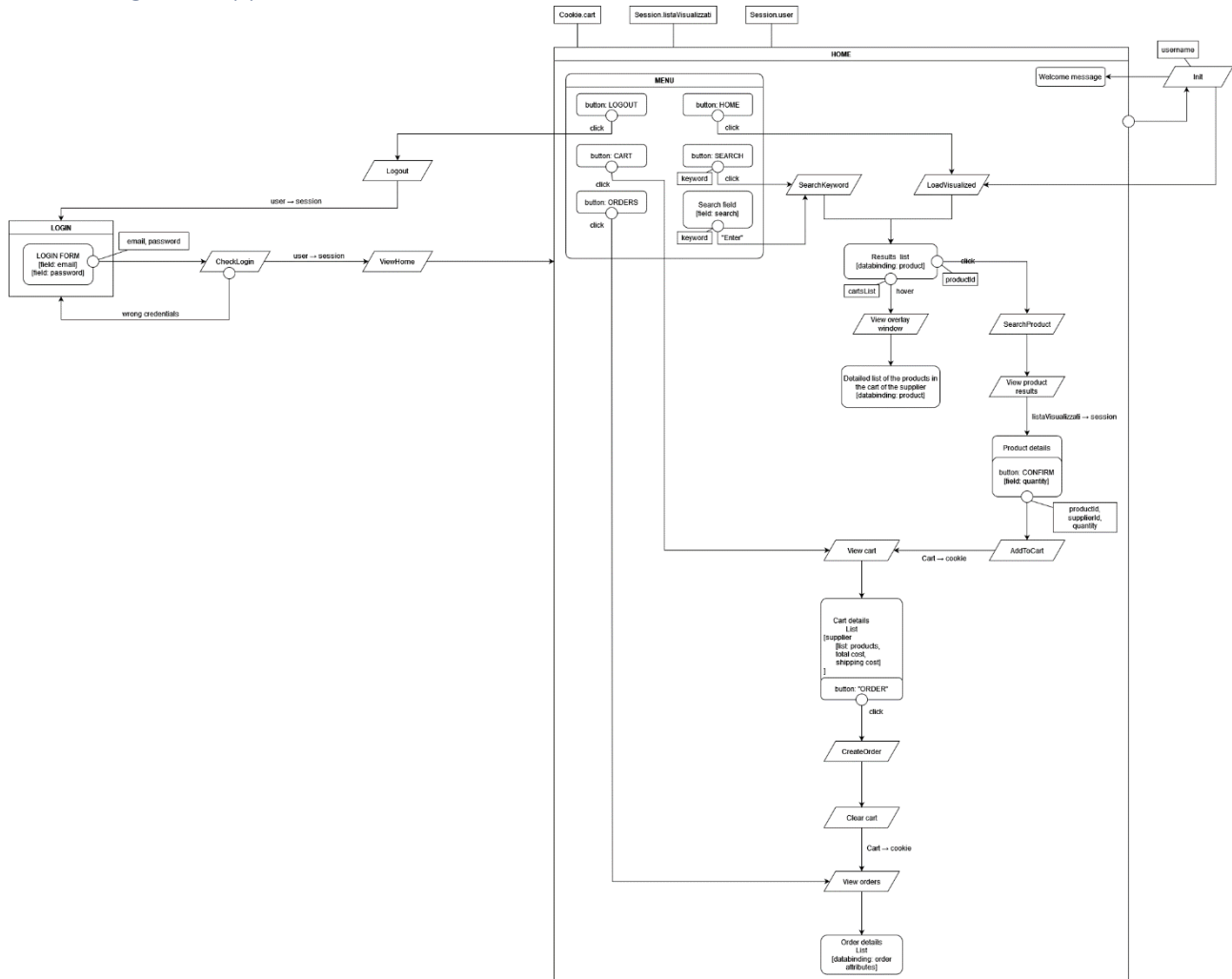
- **Login:**
  - Form di login;
  - Pulsante LOGIN;
- **Home:**
  - Menù:
    - Pulsante HOME;
    - Pulsante CART;
    - Pulsante ORDERS;
    - Campo di ricerca;
    - Pulsante SEARCH;
    - Pulsante LOGOUT;
  - Messaggio di benvenuto personalizzato;
  - Lista degli ultimi cinque prodotti visualizzati;
  - Lista risultati;
  - Elenco dettagliato prodotti già nel carrello per tale fornitore;
  - Campo di input "QUANTITY";
  - Pulsante "ADD TO CART";
  - Lista prodotti nel carrello;
  - Pulsante "ORDER";
  - Lista ordini;

#### Eventi e azioni:

- **Pressione pulsante di LOGIN:**
  - **Verifica credenziali;**
- **Pressione tasto SEARCH:**
  - **Ricerca con parola chiave;**
- **Click su un prodotto:**
  - **Visualizzazione nella stessa pagina dei dati completi del prodotto e l'elenco dei fornitori con tale oggetto;**
  - **Aggiunta prodotto alla lista di prodotti visualizzati;**
- **Click su pulsante ADD TO CART:**
  - **Aggiornamento carrello;**
  - **Caricamento pagina CART;**
- **Click su pulsante ORDER:**
  - **Creazione dell'ordine;**
  - **Rimozione dei prodotti dal carrello;**
  - **Caricamento pagina ORDERS;**
- **Click HOME:**
  - **Caricamento pagina HOME;**

- Click CART:
  - Caricamento pagina CART;
- Click ORDERS:
  - Caricamento pagina ORDERS;
- Click LOGOUT:
  - Logout;

## 2.3.2 Design dell'applicazione web



### 2.3.3 Componenti

Model objects (beans):

- Address;
- Cart;
- Order;
- Product;
- ShippingRange;
- Supplier;
- User;

Data Access Objects (DAO):

- AddressDAO:
  - getAddressById(int id);
  - getAddressByParameters(String citta, String via, String cap, int numero);
  - addAddress(String citta, String via, String cap, int numero)
- OrderDAO:
  - getOrdersOfUserById(int id);
  - addOrder(float total, int addressId, int userId, int supplierId, List<Product> products);
- ProductDAO:
  - getProductById(int id);
  - getProductOffersById(int id);
  - getProductByIdAndSupplier(int productId, int supplierId);
  - getProductsByKeyword(String keyword);
  - getProducts(Queue<Integer> alreadyDisplayed, int quantity);
  - existsProduct(int id);
- SupplierDAO:
  - getSupplierById(int id);
  - existsSupplier(int id);
- UserDAO:
  - checkCredentials(String email, String password);
  - addUser(String name, String surname, String email, String password, int addressId);

Controllori (servlets):

- AddToCart;
- CheckLogin;
- CreateOrder;
- LoadCart;
- LoadVisualized;
- Logout;
- SearchKeyword;
- SearchProduct;
- ViewOrders;

Pagine (templates):

- Login;
- Home;

### 2.3.4 Eventi e azioni

Gli elementi in **grassetto** vengono inviati sotto forma di testo in formato JSON.

N.	Client side		Server side	
	Evento	Azione	Evento	Azione
1	Login → login form → submit	Login	POST (username, password)	Controllo credenziali
2	Home → load	Equivalente a EVENTO 3	-	-
3	Home → menù → pulsante HOME	Carica prodotti recenti	POST ( <b>listaVisualizzati</b> )	Estrazione dei 5 prodotti precedentemente visualizzati (ove necessario, utilizzare prodotti scelti casualmente)
4	Home → menù → campo ricerca → INVIO	Ricerca keyword	GET (keyword)	Estrazione lista di prodotti contenenti la keyword
5	Home → menù → pulsante ORDINI	Visualizzazione ordini	GET ()	Estrazione degli ordini effettuati dall'utente
6	Home → menù → pulsante CARRELLO	Visualizzazione carrello	POST ( <b>cartList</b> )	Estrazione dei prodotti all'interno di ogni carrello dei fornitori
7	Home → lista risultati → click elemento	<ul style="list-style-type: none"> <li>Visualizzazione dati elemento</li> <li>Aggiunta elemento a listaVisualizzati</li> </ul>	GET (productId)	Estrazione informazioni relative al prodotto
8	Home → lista risultati → hover elemento	Visualizzazione finestra sovrapposta con dettaglio carrello	POST ( <b>cartList</b> )	Estrazione dei prodotti all'interno di un singolo carrello
9	Home → lista risultati → elemento → pulsante ADD TO CART	<ul style="list-style-type: none"> <li>Aggiornamento carrello</li> <li>EVENTO 6</li> </ul>	-	-
10	Home → lista prodotti nel carrello → pulsante ORDINA	<ul style="list-style-type: none"> <li>Creazione dell'ordine</li> <li>Svuotamento del carrello</li> <li>EVENTO 5</li> </ul>	POST ( <b>Cart</b> , shippingAddress)	Creazione dell'ordine all'interno del database
11	Home → menù → pulsante LOGOUT	Logout	GET ()	Logout

### 2.3.5 Controllori ed event handlers

N.	Client side		Server side	
	Evento	Controllore (funzione)	Evento	Controllore (servlet)
1	Login → login form → submit	makeCall()	POST (username, password)	Login
2	Home → load	Equivalente a EVENTO 3	-	-
3	Home → menù → pulsante HOME	makeCall()	POST ( <b>listaVisualizzati</b> )	LoadVisualized
4	Home → menù → campo ricerca → INVIO	makeCall()	GET (keyword)	SearchKeyword
5	Home → menù → pulsante ORDINI	makeCall()	GET ()	ViewOrders
6	Home → menù → pulsante CARRELLO	makeCall()	POST ( <b>cartList</b> )	LoadCart
7	Home → lista risultati → click elemento	<ul style="list-style-type: none"> <li>addToVisualized(productId)</li> <li>makeCall()</li> </ul>	GET (productId)	SearchProduct
8	Home → lista risultati → hover elementi carrello	makeCall()	POST ( <b>cartList</b> )	LoadCart
9	Home → lista risultati → elemento → pulsante ADD TO CART	<ul style="list-style-type: none"> <li>addCookieProduct()</li> <li>EVENTO 6</li> </ul>	-	-
10	Home → lista prodotti nel carrello → pulsante ORDINA	<ul style="list-style-type: none"> <li>makeCall()</li> <li>clearCartCookies()</li> <li>EVENTO 5</li> </ul>	POST ( <b>Cart</b> , shippingAddress)	CreateOrder
11	Home → menù → pulsante LOGOUT	makeCall()	GET ()	Logout

### 2.3.6 View e view components

#### Login

- Form di login
  - Submit del form, controllo dei campi e gestione della risposta;

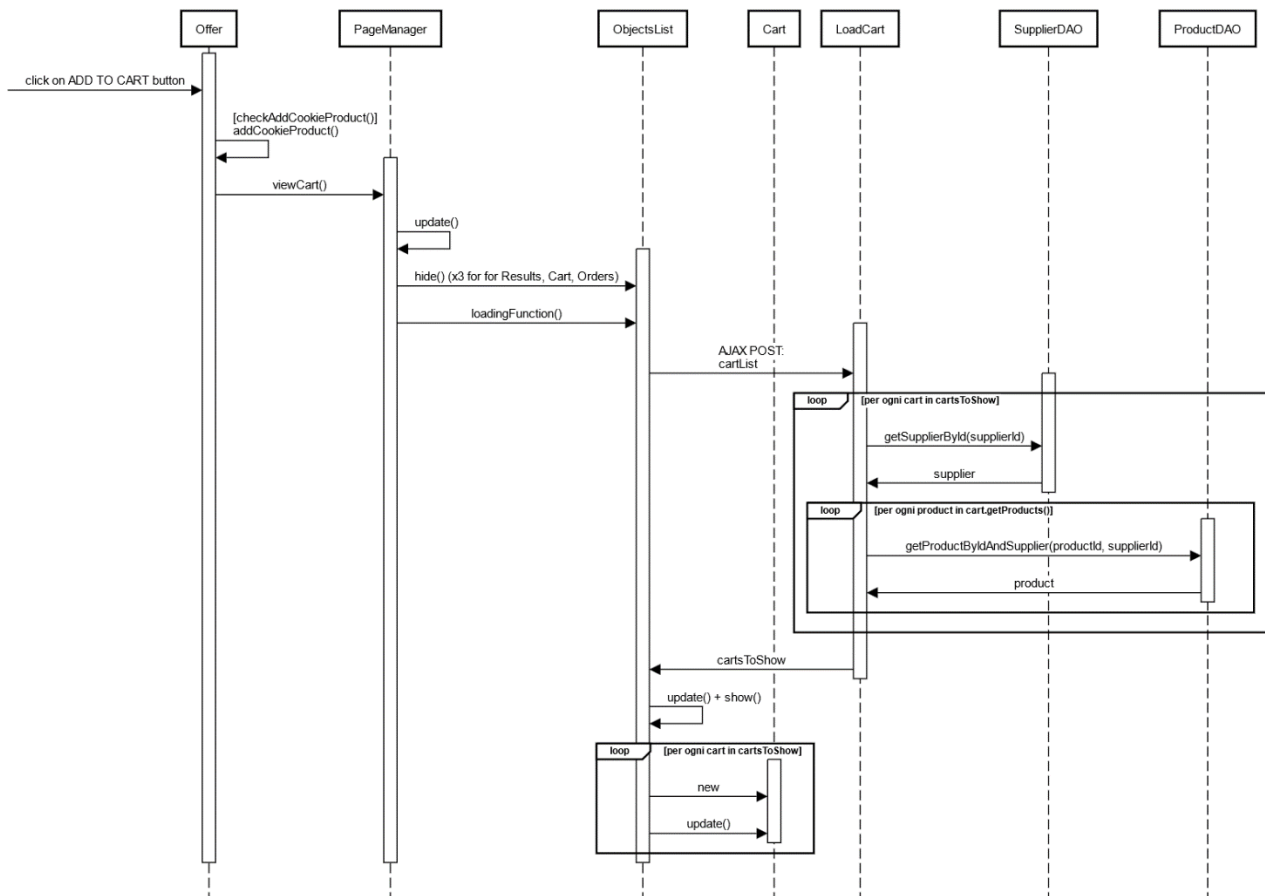
#### Home

- ObjectsList: generico costruttore delle liste di oggetti.
  - Update(): richiama a cascata la funzione Update() degli oggetti contenuti nella lista;
  - Show(): visualizza la lista;
  - Hide(): nasconde la lista;
  - isHidden(): ritorna `vero` se la lista è nascosta, `falso` altrimenti;
- Product, Order, Cart, Offer: i vari oggetti contenuti nelle liste.
  - Update(): carica le informazioni relative all'elemento e crea il codice HTML per la sua visualizzazione.
- Menu
  - addEvents(): aggiunge i listener ai pulsanti della barra di ricerca e li associa alle funzioni dedicate per una corretta navigazione;
- WelcomeMessage
  - Show(): visualizza il messaggio di benvenuto personalizzato;
- PageManager
  - Init(): inizializza i componenti della pagina, quali il messaggio di benvenuto, il menù, la lista dei risultati, dei carrelli e degli ordini;
  - Update(): aggiorna il contenuto della pagina, nascondendo tutte le liste;
  - viewHome(), viewCart(), viewOrders(), viewResults(): visualizza le liste di prodotti, carrelli, ordini o risultati;

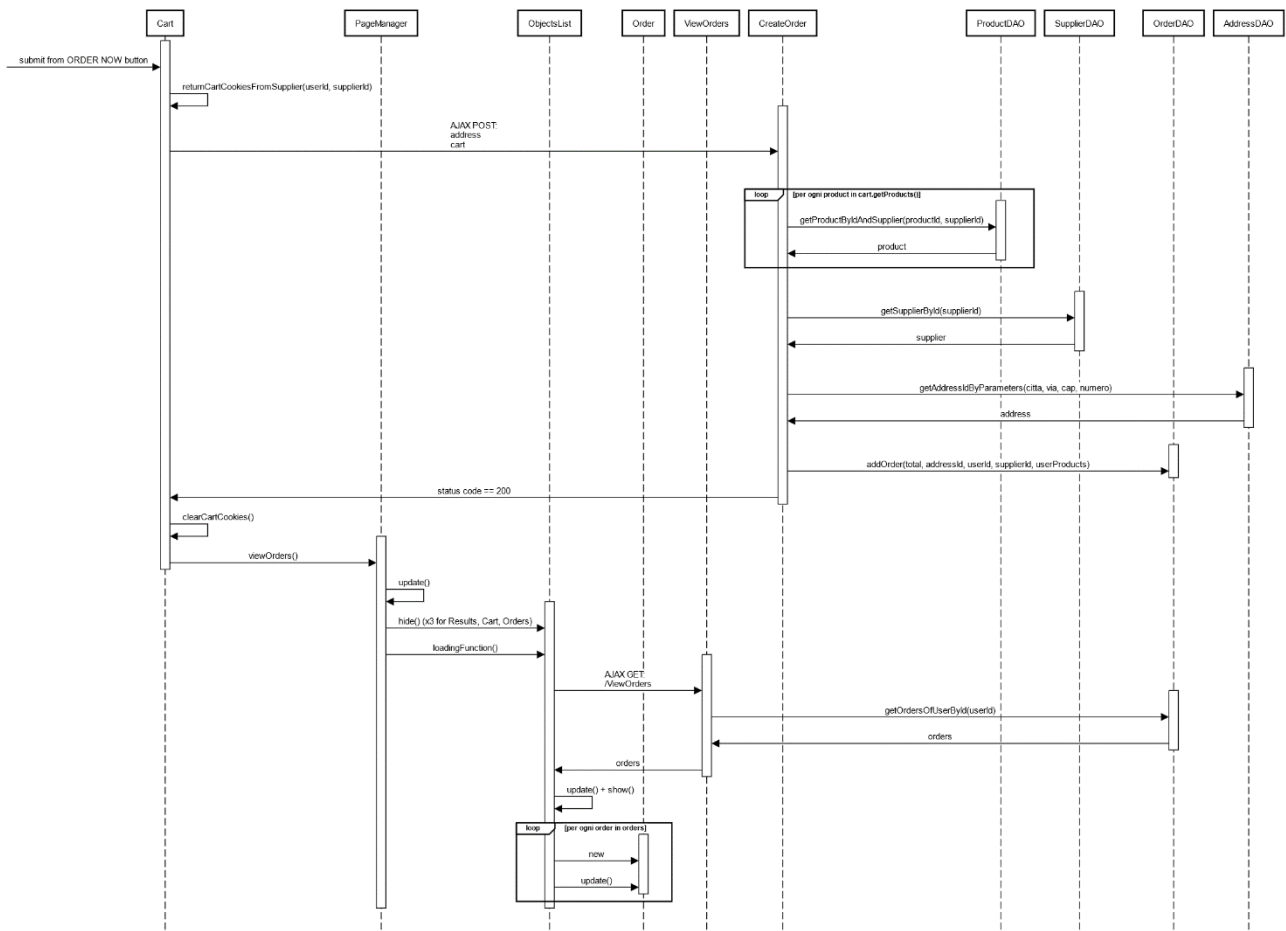


## 2.3.7 Sequence diagrams

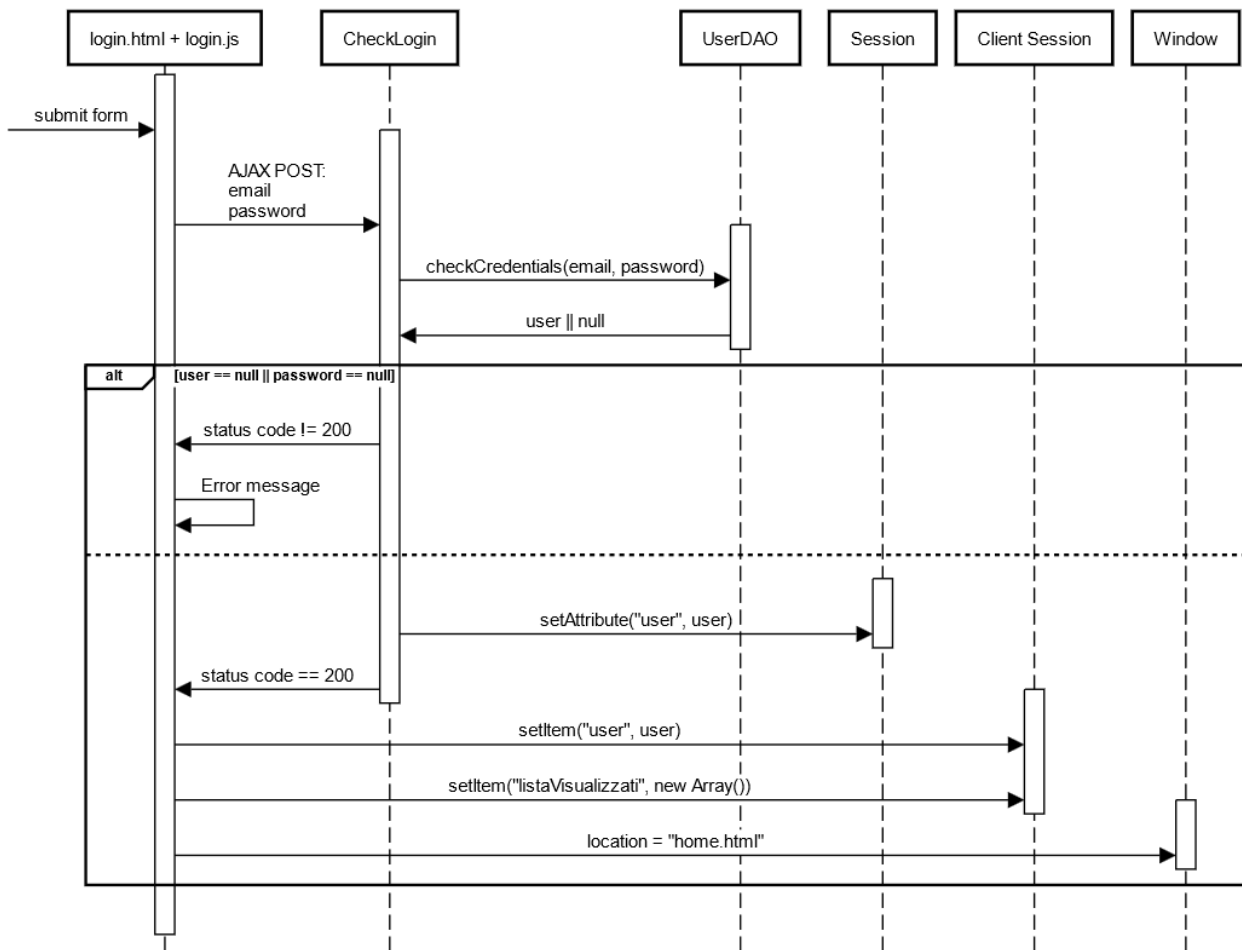
### AddToCart:



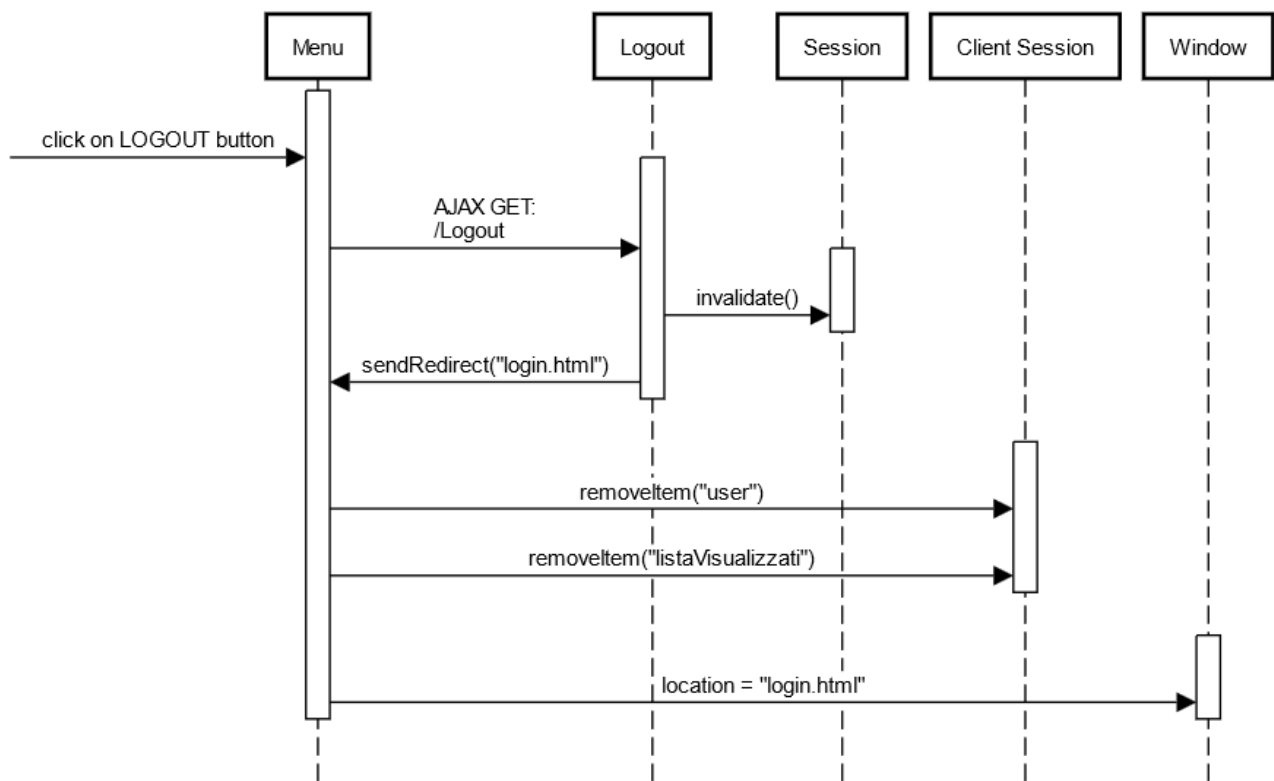
## CreateOrder:



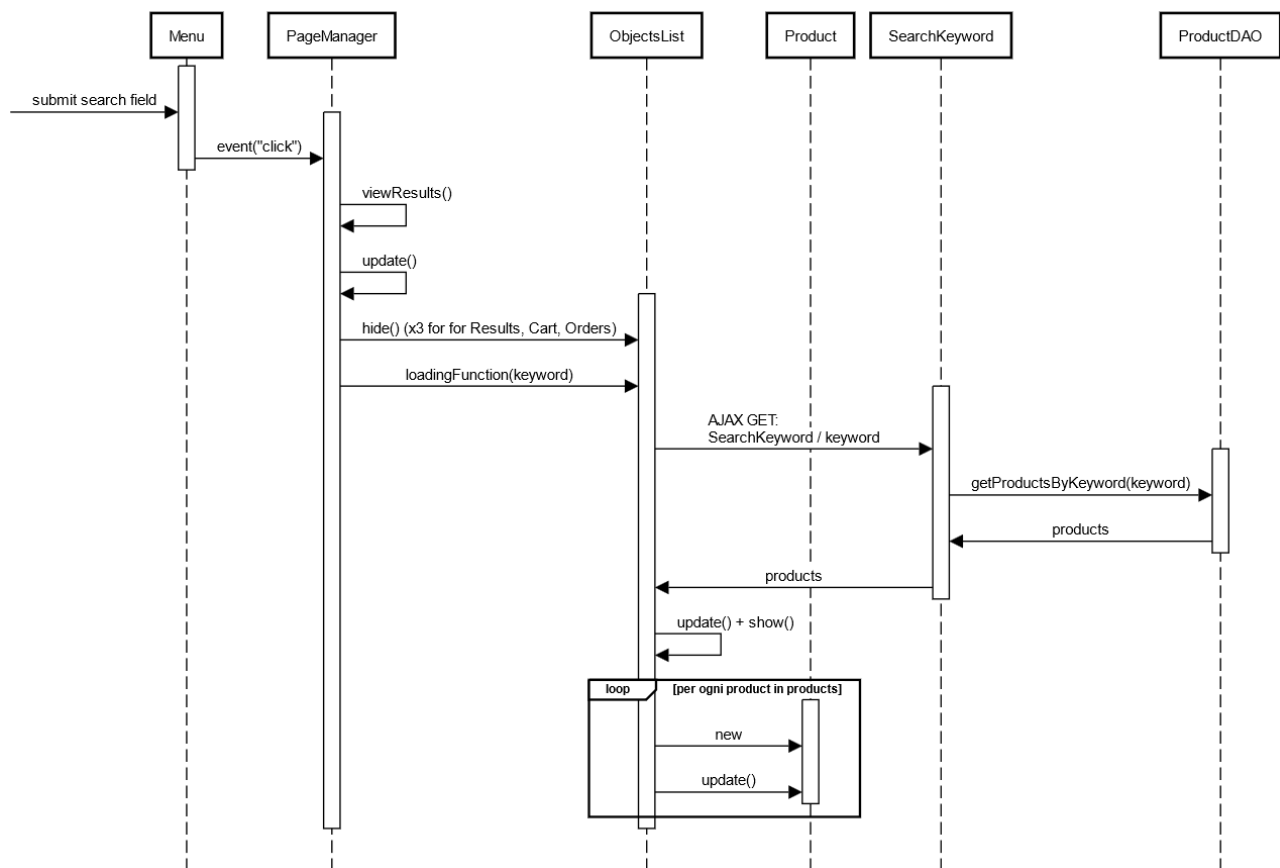
## Login:



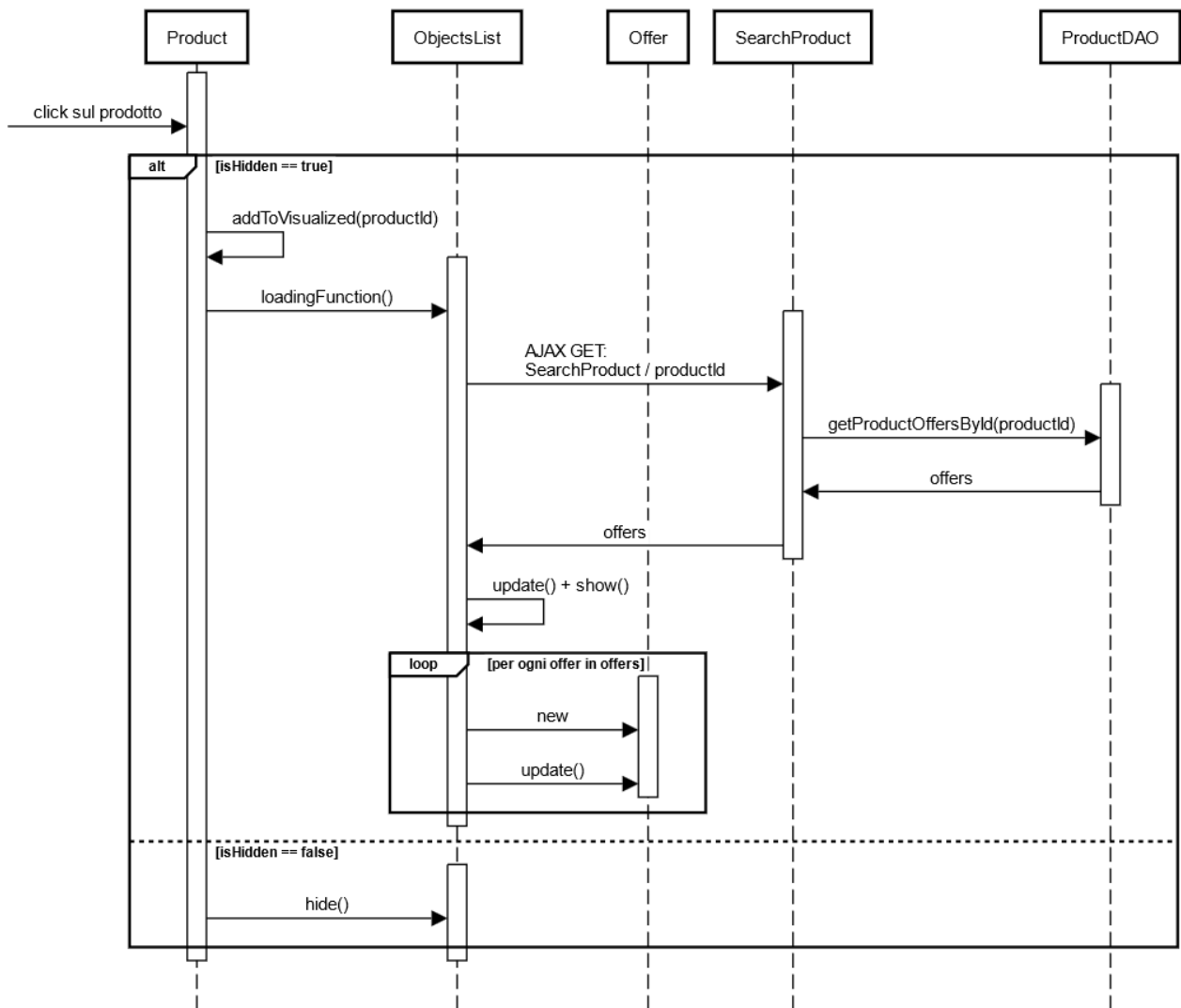
## Logout:



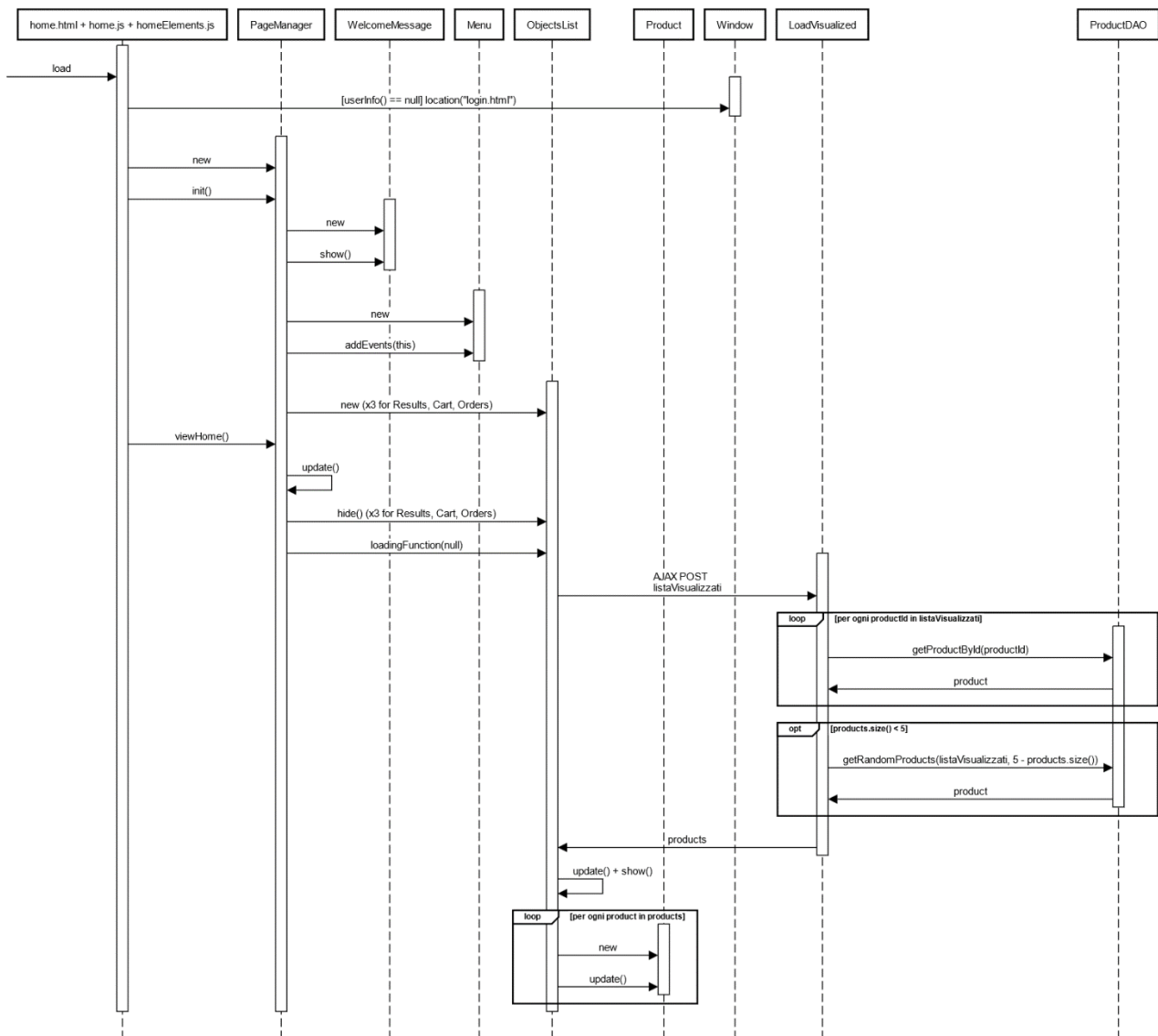
**SearchKeyword:**



## SearchProduct:



## ViewHome:



### 3 Strumenti utilizzati

IDE	Eclipse IDE 2022-12
Server	Apache Tomcat v9.0
Gestione database	MySQL Workbench
Browser	Firefox 119
Creazione sequence diagrams	<a href="https://sequencediagram.org/">https://sequencediagram.org/</a>
Creazione altri diagrammi	<a href="https://draw.io">draw.io</a>