

PROGETTO 5: CARRELLO CON PIÙ FORNITORI E ORDINE

Chiara Auriemma 10722613 956170

Giacomo Ballabio 10769576 959913



HTML

SPECIFICATION

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.

Entità Relazioni Attributi

SPECIFICATION

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.

SPECIFICATION

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.

DATABASE SCHEMA

```
CREATE TABLE `user` (  
  `Mail` varchar(30) NOT NULL,  
  `Password` char(8) NOT NULL,  
  `Name` varchar(15) NOT NULL,  
  `Surname` varchar(15) NOT NULL,  
  `Address` varchar(50) NOT NULL,  
  PRIMARY KEY (`Mail`), KEY `Address` (`Address`))
```

DATABASE SCHEMA

```
CREATE TABLE `product` (  
  `Code` int NOT NULL AUTO_INCREMENT,  
  `Name` varchar(30) NOT NULL,  
  `Description` varchar(100) DEFAULT NULL,  
  `Category` varchar(30) NOT NULL,  
  `Photo` varchar(30) DEFAULT NULL,  
  PRIMARY KEY (`Code`))
```

DATABASE SCHEMA

```
CREATE TABLE `visualize` (  
  `MailUser` varchar(30) NOT NULL,  
  `ProdCode` int NOT NULL,  
  `Date` date NOT NULL,  
  `Time` time NOT NULL,  
  PRIMARY KEY (`MailUser`,`ProdCode`,`Date`,`Time`),  
  KEY `MailUser` (`MailUser`),  
  KEY `ProdCode` (`ProdCode`),  
  CONSTRAINT `visualize_ibfk_1` FOREIGN KEY (`MailUser`) REFERENCES `user` (`Mail`),  
  CONSTRAINT `visualize_ibfk_2` FOREIGN KEY (`ProdCode`) REFERENCES `product` (`Code`))
```


DATABASE SCHEMA

```
CREATE TABLE `supplier` (  
  `Code` int NOT NULL,  
  `Name` varchar(30) NOT NULL,  
  `Score` int NOT NULL DEFAULT '0',  
  `FreeShipping` float DEFAULT NULL,  
  PRIMARY KEY (`Code`), KEY `Supplier` (`Name`))
```

DATABASE SCHEMA

```
CREATE TABLE `sold_by` (  
  `ProdCode` int NOT NULL,  
  `Price` float NOT NULL DEFAULT '0',  
  `Supcode` int NOT NULL,  
  PRIMARY KEY (`ProdCode`,`Supcode`),  
  KEY `SupCode_idx` (`Supcode`),  
  CONSTRAINT `ProdCode` FOREIGN KEY (`ProdCode`) REFERENCES `product` (`Code`),  
  CONSTRAINT `SupCode` FOREIGN KEY (`Supcode`) REFERENCES `supplier` (`Code`))
```

DATABASE SCHEMA

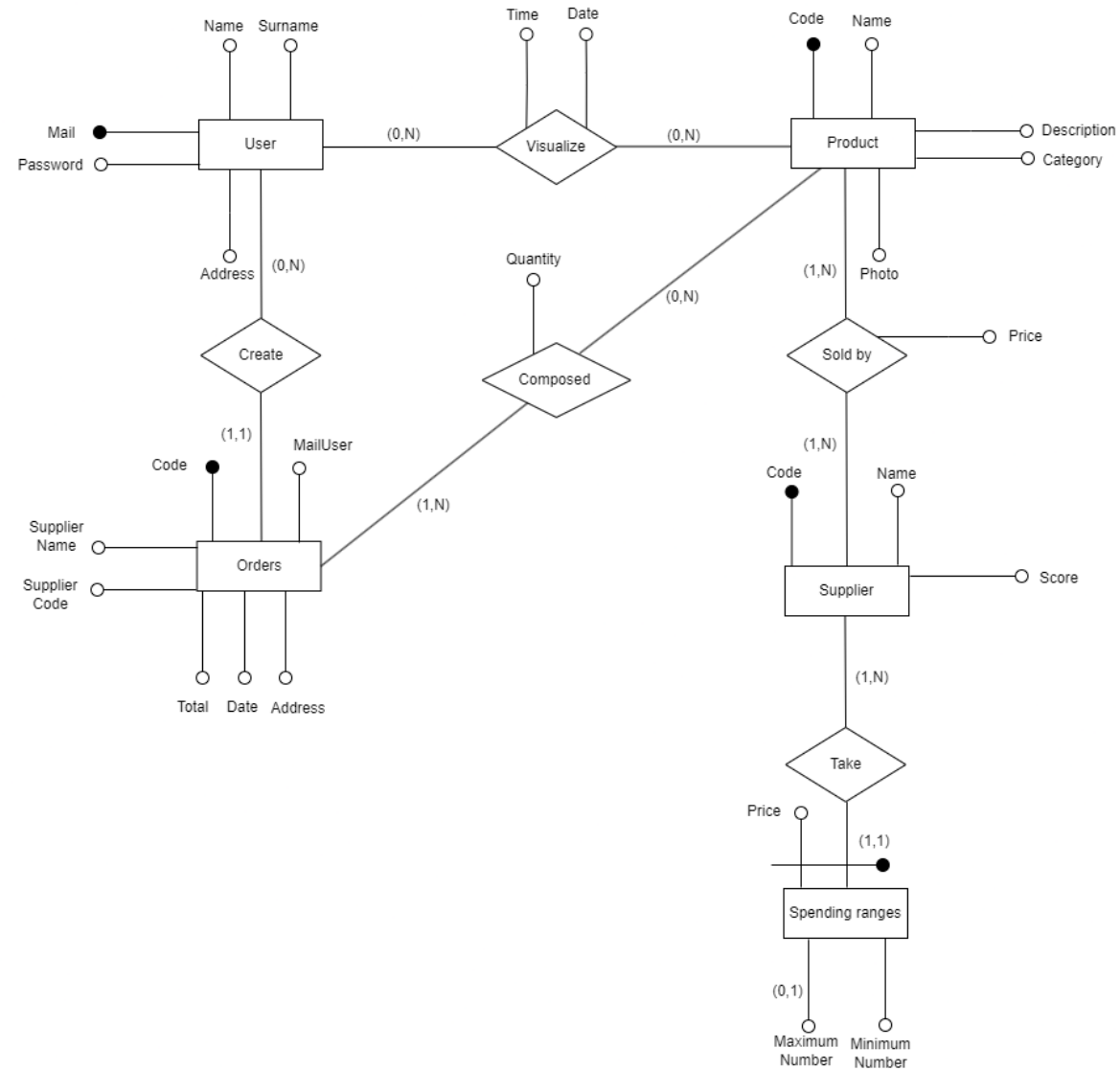
```
CREATE TABLE `spending_ranges` (  
  `SupCode` int NOT NULL,  
  `Price` float NOT NULL DEFAULT '0',  
  `MaximumN` int DEFAULT NULL,  
  `MinimumN` int NOT NULL,  
  PRIMARY KEY (`SupCode`,`Price`),  
  CONSTRAINT `Code` FOREIGN KEY (`SupCode`) REFERENCES `supplier` (`Code`))
```

DATABASE SCHEMA

```
CREATE TABLE `orders` (  
  `Code` int NOT NULL AUTO_INCREMENT,  
  `MailUser` varchar(30) NOT NULL,  
  `Supplier` varchar(30) NOT NULL,  
  `Total` float NOT NULL,  
  `Date` date NOT NULL,  
  `Address` varchar(50) NOT NULL,  
  `SupCode` int NOT NULL,  
  PRIMARY KEY (`Code`),  
  KEY `MailUser_idx` (`MailUser`),  
  KEY `Supplier_idx` (`Supplier`),  
  KEY `Address_idx` (`Address`),  
  KEY `SupCode_idx` (`SupCode`),  
  CONSTRAINT `Address` FOREIGN KEY (`Address`) REFERENCES `user` (`Address`),  
  CONSTRAINT `MailUser` FOREIGN KEY (`MailUser`) REFERENCES `user` (`Mail`),  
  CONSTRAINT `SCode` FOREIGN KEY (`SupCode`) REFERENCES `supplier` (`Code`),  
  CONSTRAINT `Supplier` FOREIGN KEY (`Supplier`) REFERENCES `supplier` (`Name`))
```

DATABASE SCHEMA

```
CREATE TABLE `composed` (  
  `OrderCode` int NOT NULL,  
  `ProductCode` int NOT NULL,  
  `Quantity` int NOT NULL,  
  PRIMARY KEY (`OrderCode`,`ProductCode`),  
  KEY `ProductCode_idx` (`ProductCode`),  
  KEY `OrderCode_idx` (`OrderCode`),  
  CONSTRAINT `composed_ibfk_1` FOREIGN KEY (`OrderCode`) REFERENCES `orders` (`Code`),  
  CONSTRAINT `ProductCode` FOREIGN KEY (`ProductCode`) REFERENCES `product` (`Code`))
```



DATABASE

APPLICATION REQUIREMENTS ANALYSIS

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**.

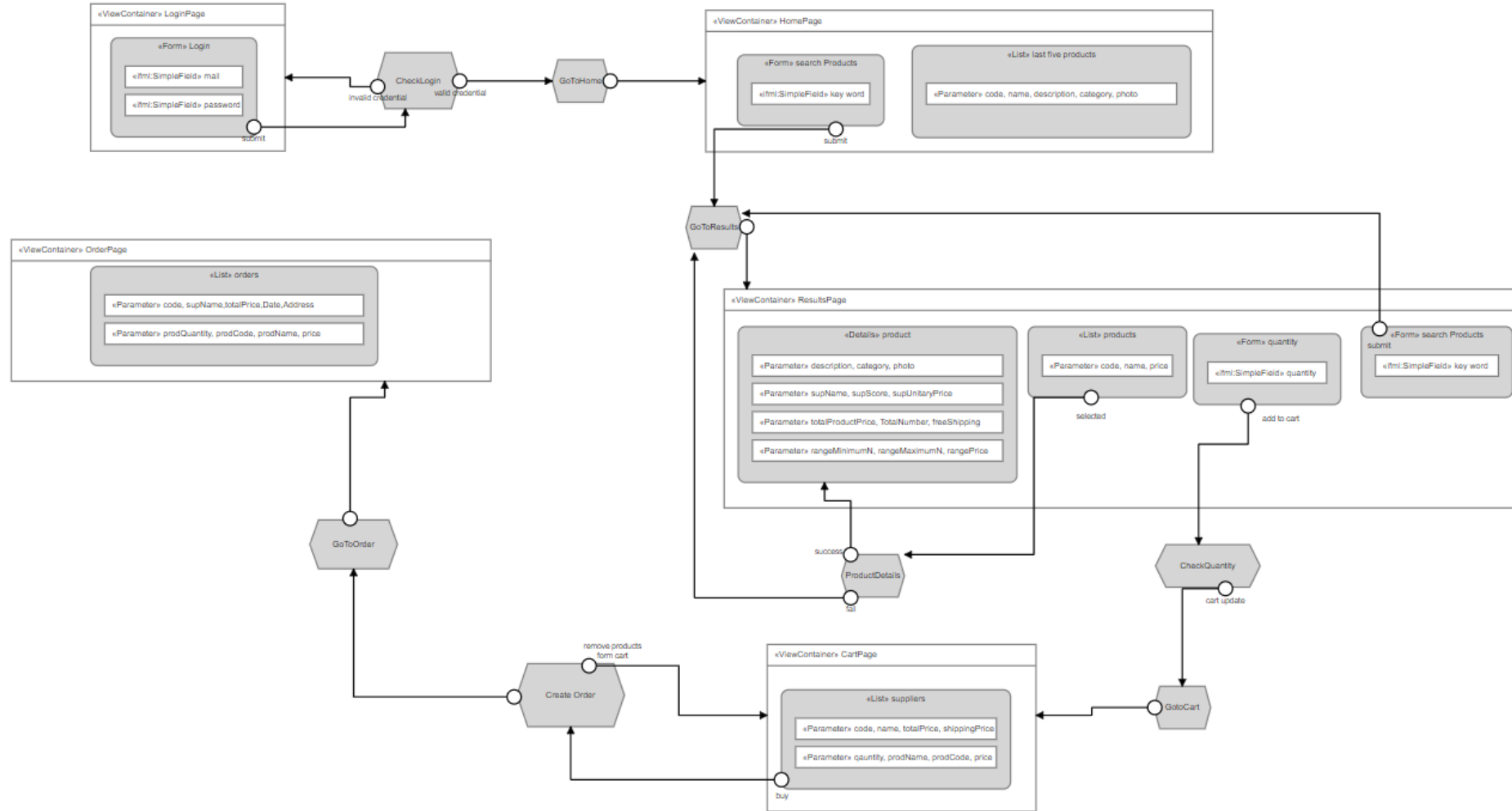
APPLICATION REQUIREMENTS ANALYSIS

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.

APPLICATION REQUIREMENTS ANALYSIS

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.

IFML



COMPONENTS

Beans:

- CartSupplier
- User
- Order
- Product
- SpendingRanges
- Supplier

Filter:

- LoginChecker

DAO:

- OrderDao
- ProductDao
- SpendingRangesDao
- SupplierDao
- UserDao

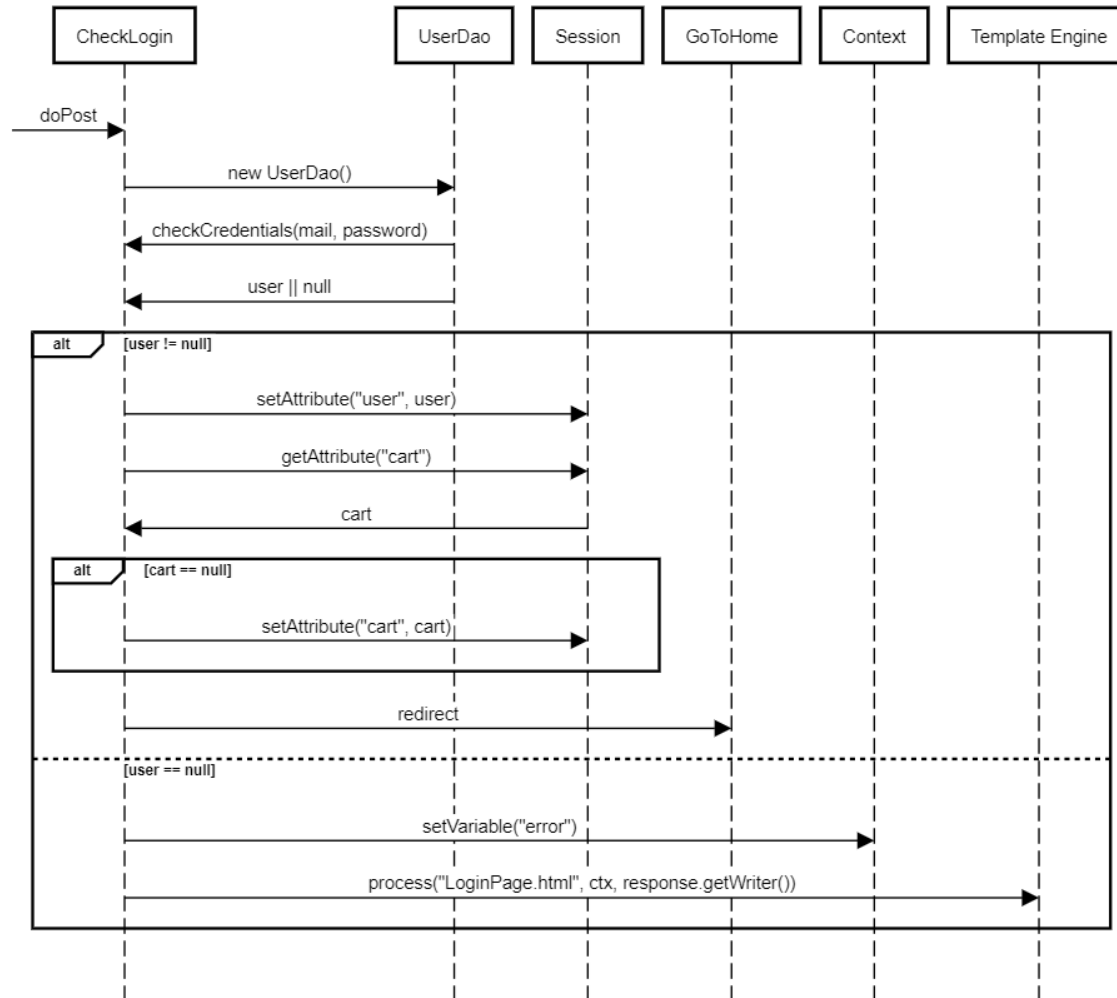
Controllers:

- CheckLogin
- CheckQuantity
- CreateOrder
- GetImage
- GoToCart
- GoToHome
- GoToOrder
- GoToResults
- Logout
- ProductDetails

View:

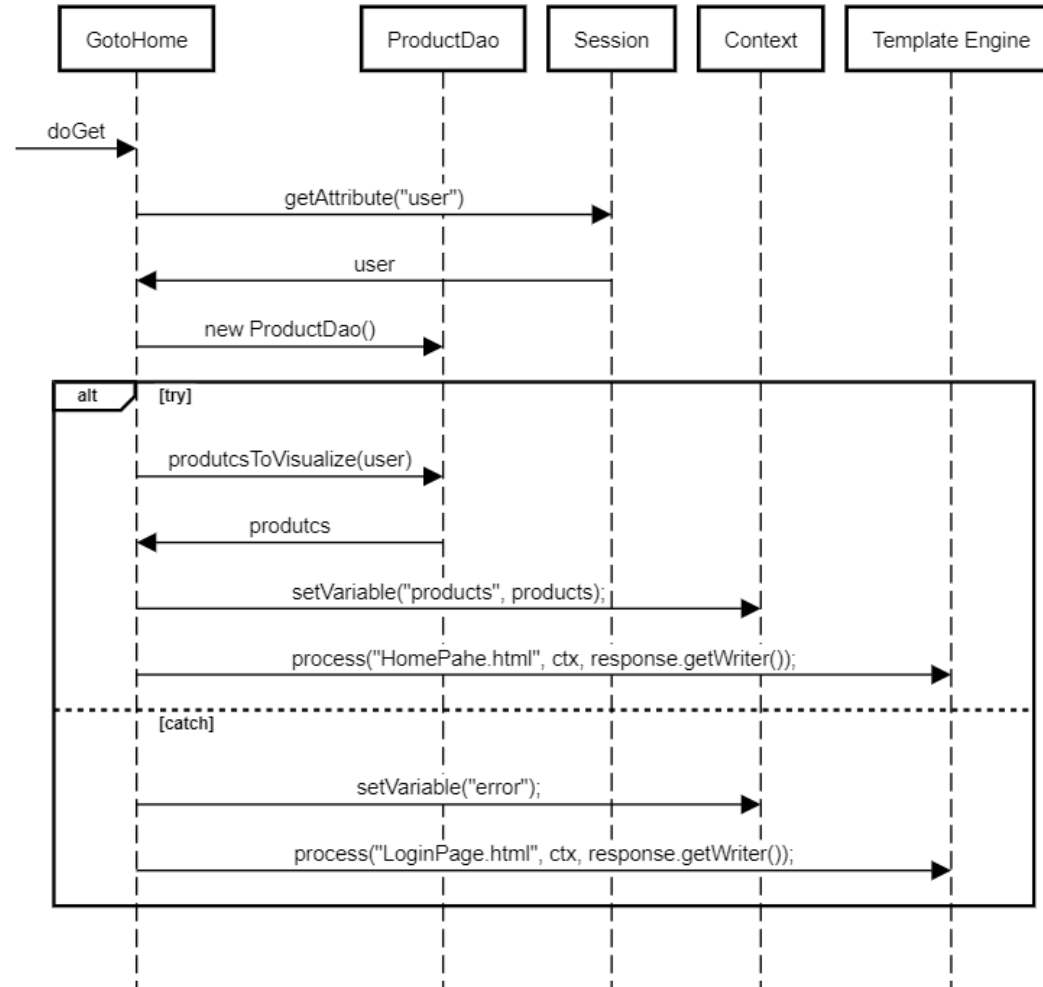
- CartPage.html
- HomePage.html
- OrderPage.html
- ResultsPage.html
- LoginPage.html

Login



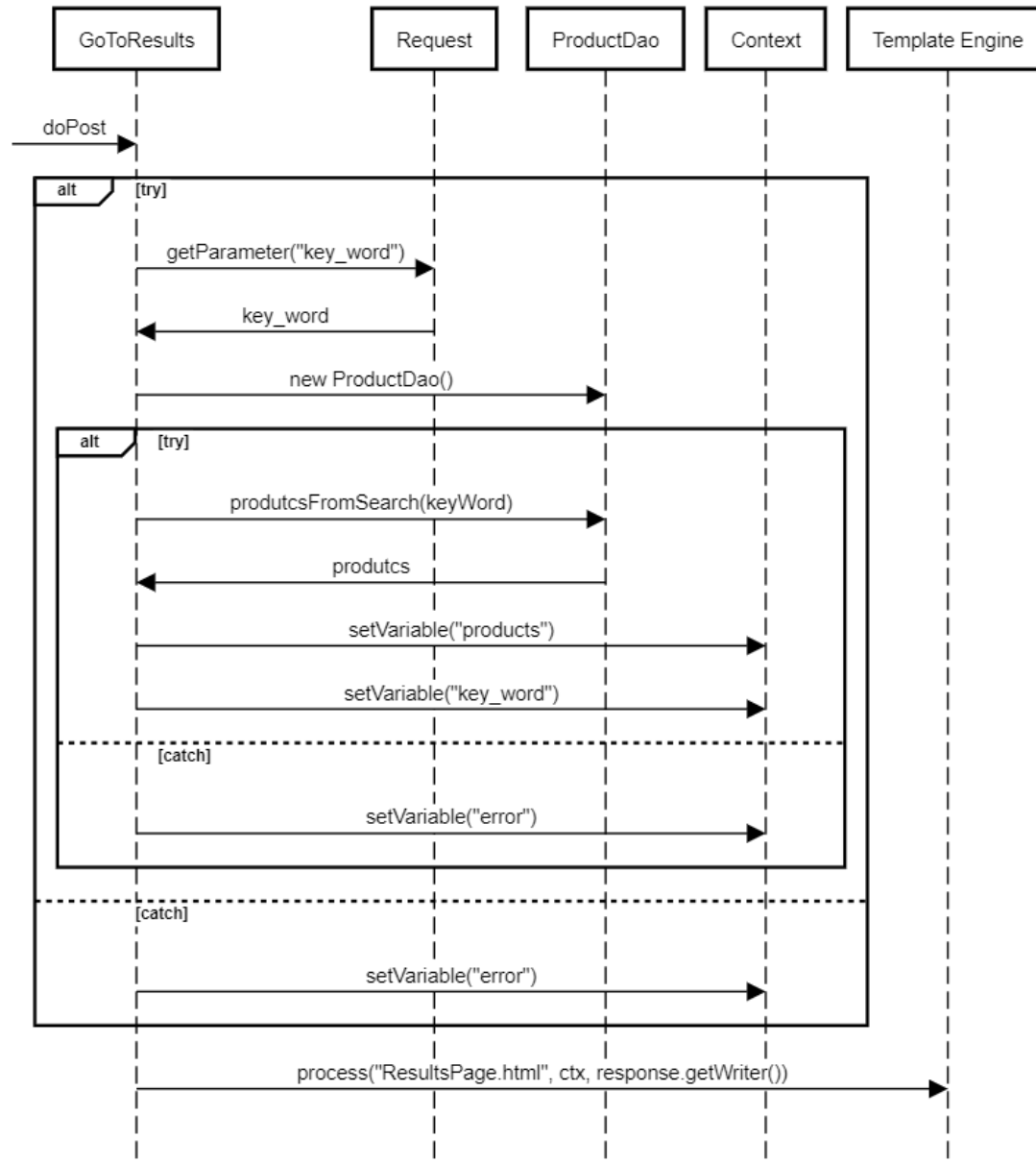
LOGIN

GoToHome

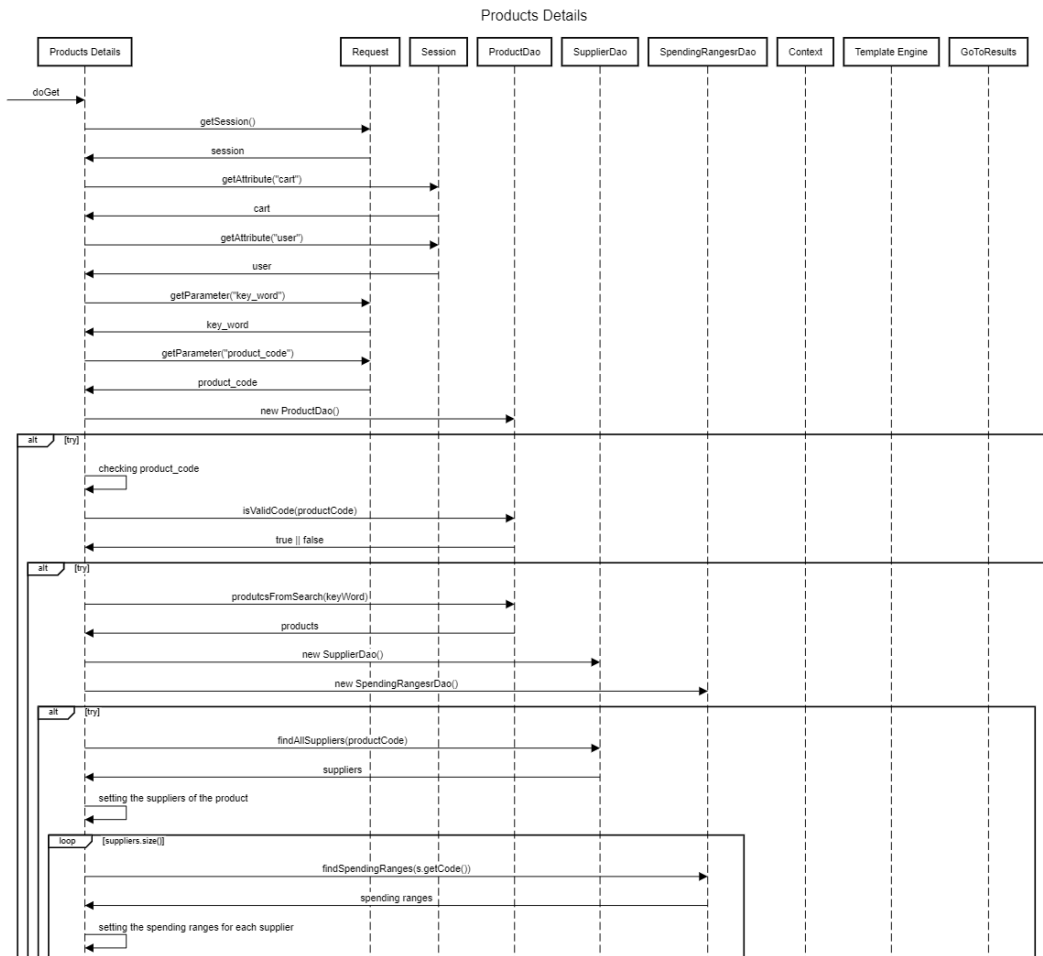


GO TO HOME

GoToResults

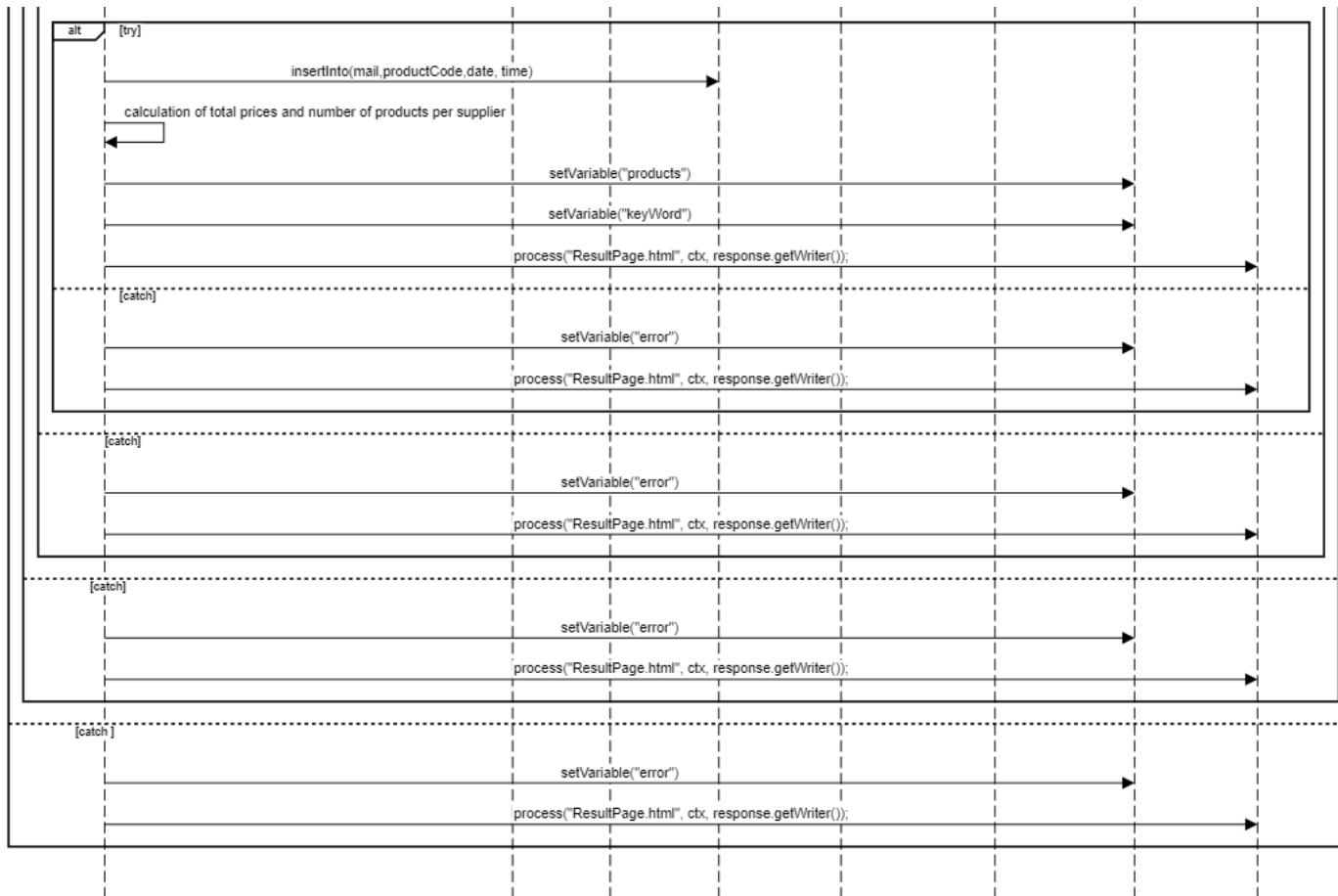


GO TO RESULTS



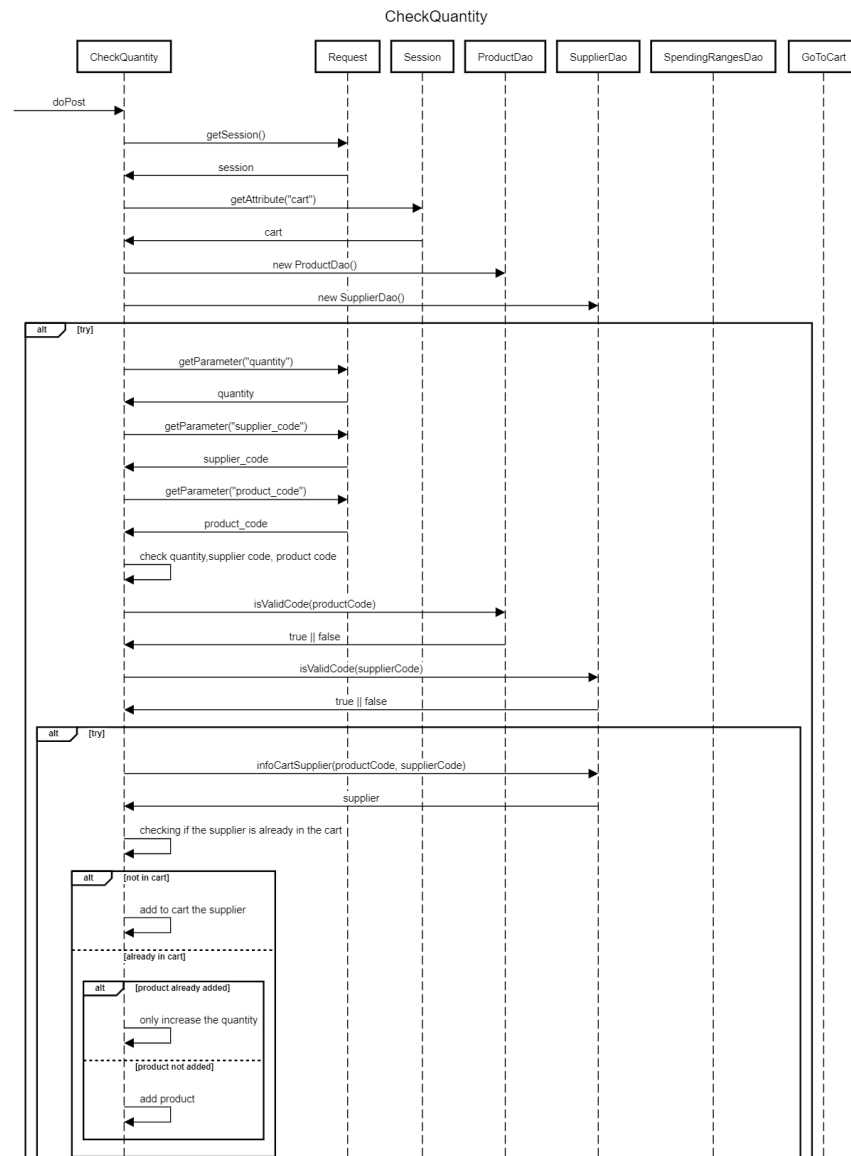
PRODUCT DETAILS

Parte 1



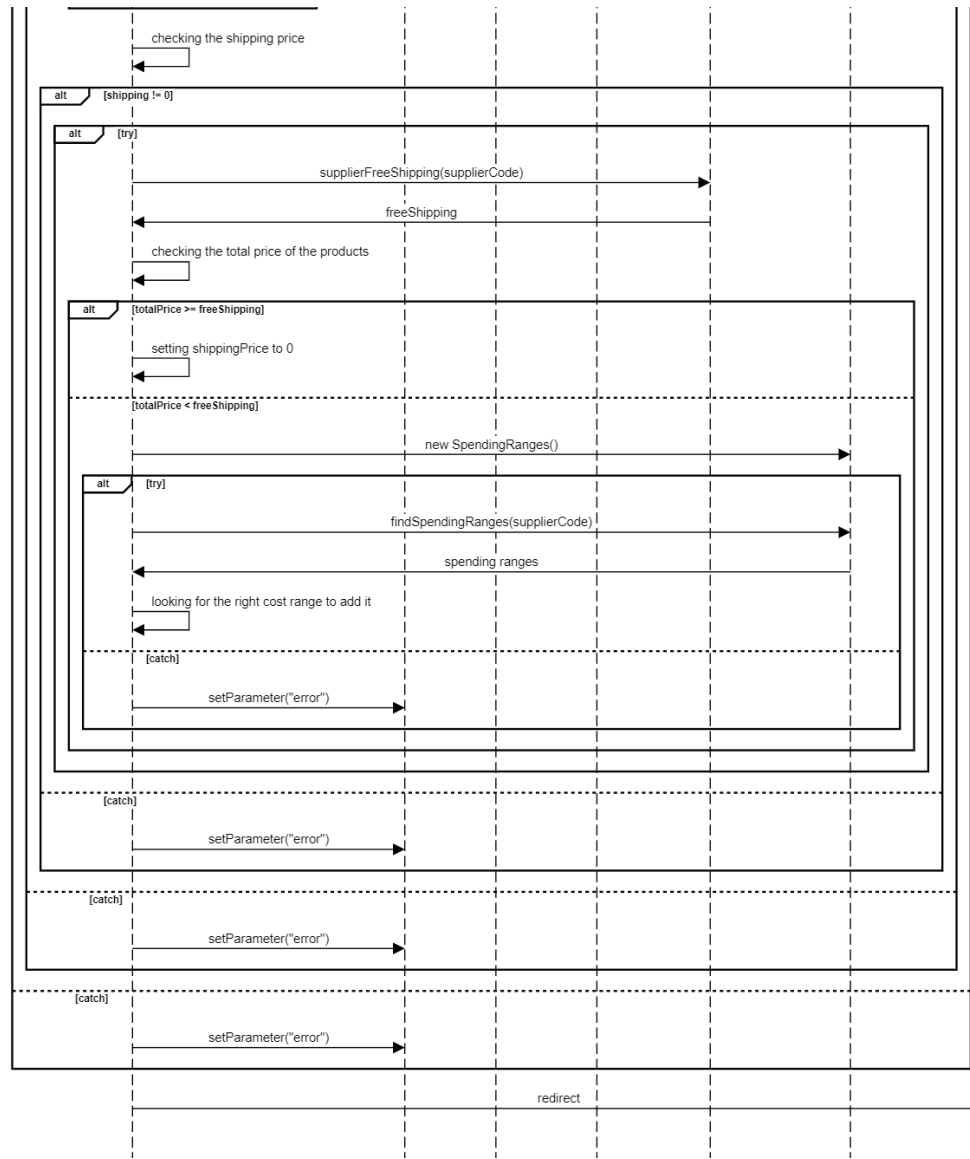
PRODUCT DETAILS

Parte 2



CHECK QUANTITY

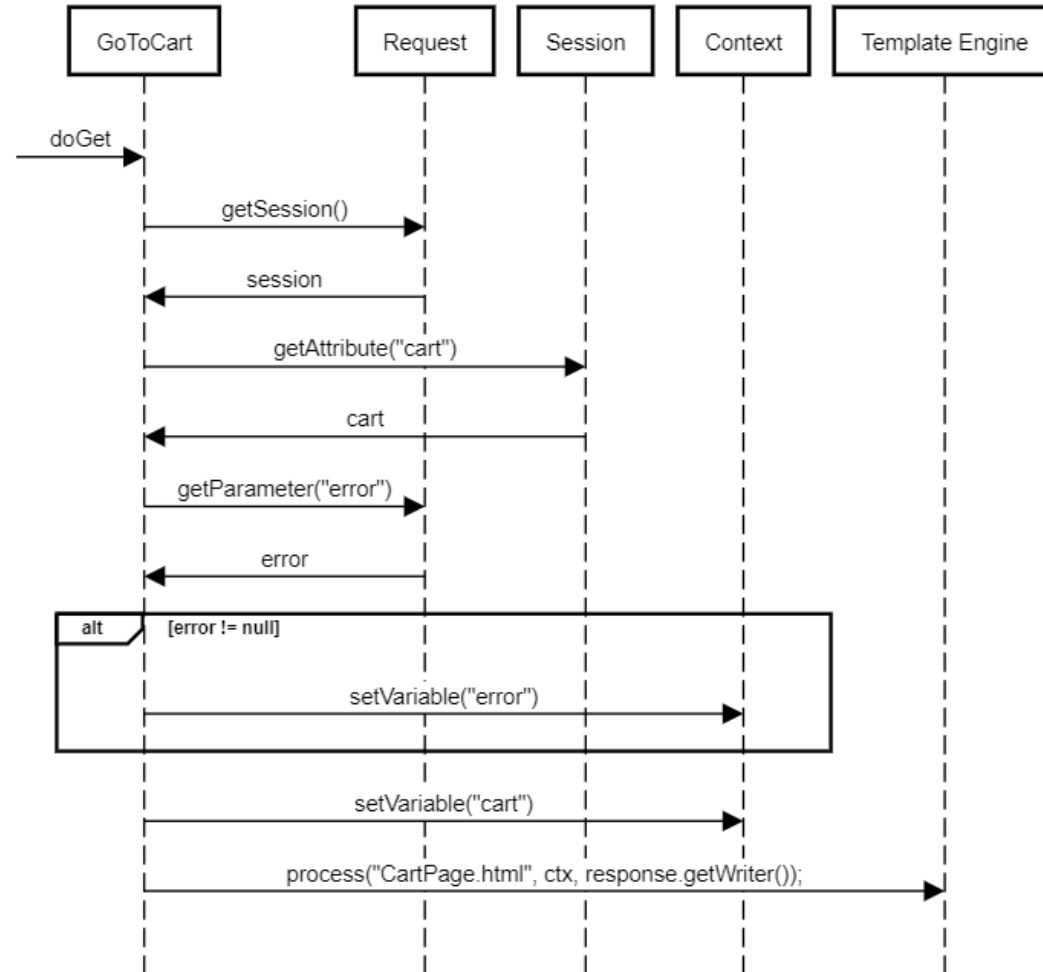
Parte 1



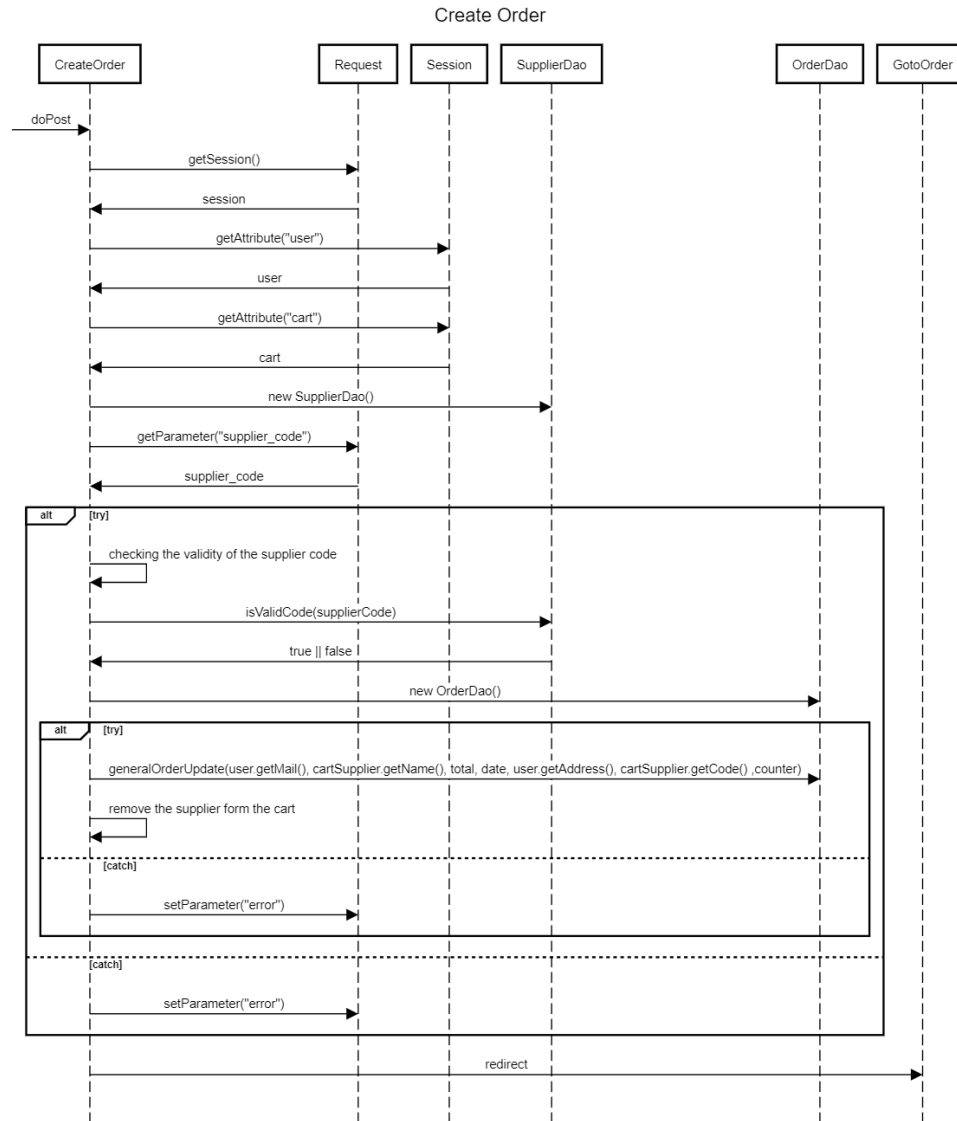
CHECK QUANTITY

Parte 2

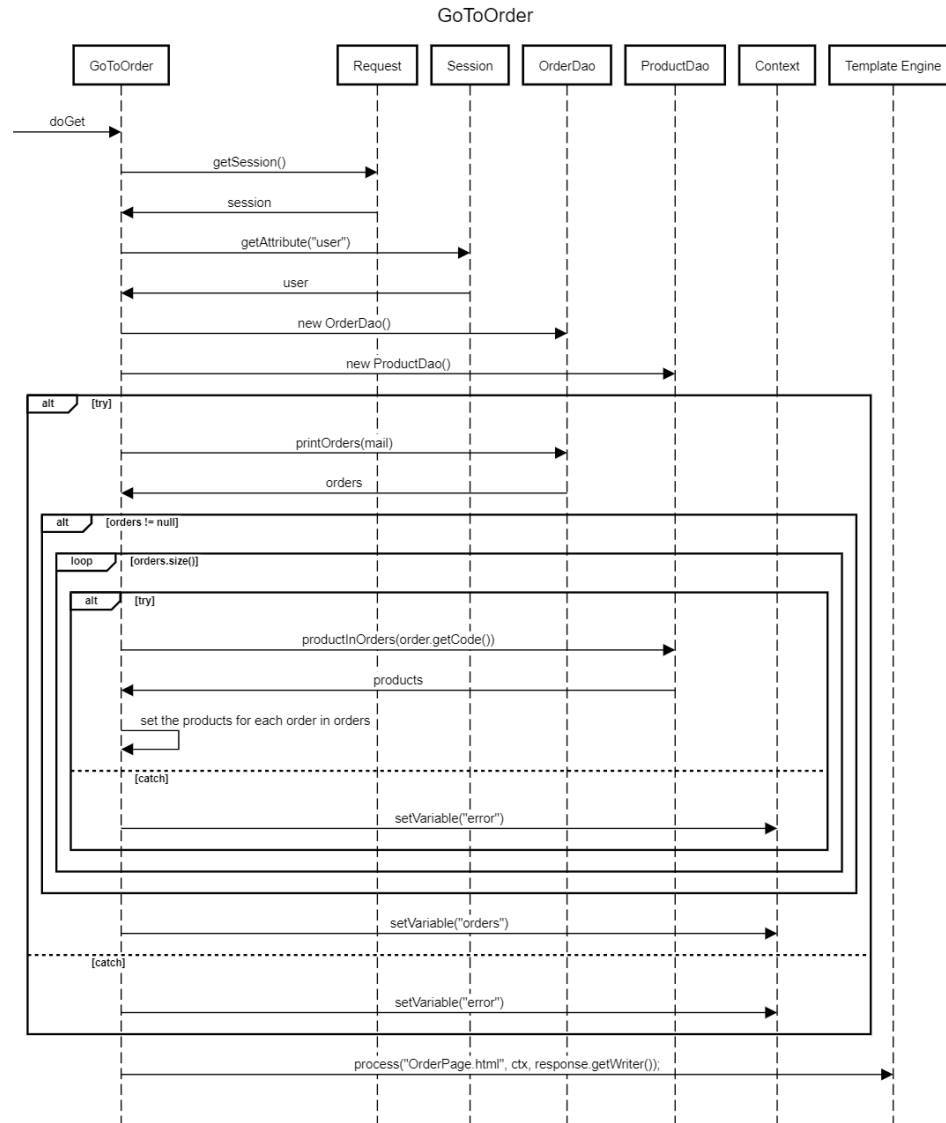
GoToCart



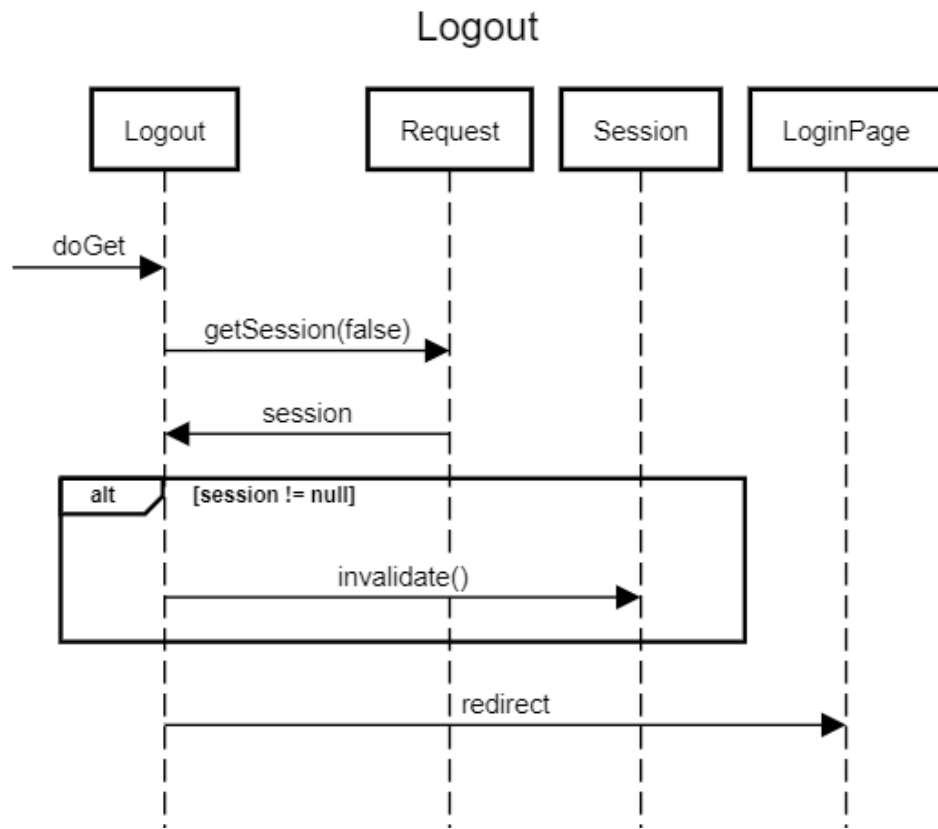
GO TO CART



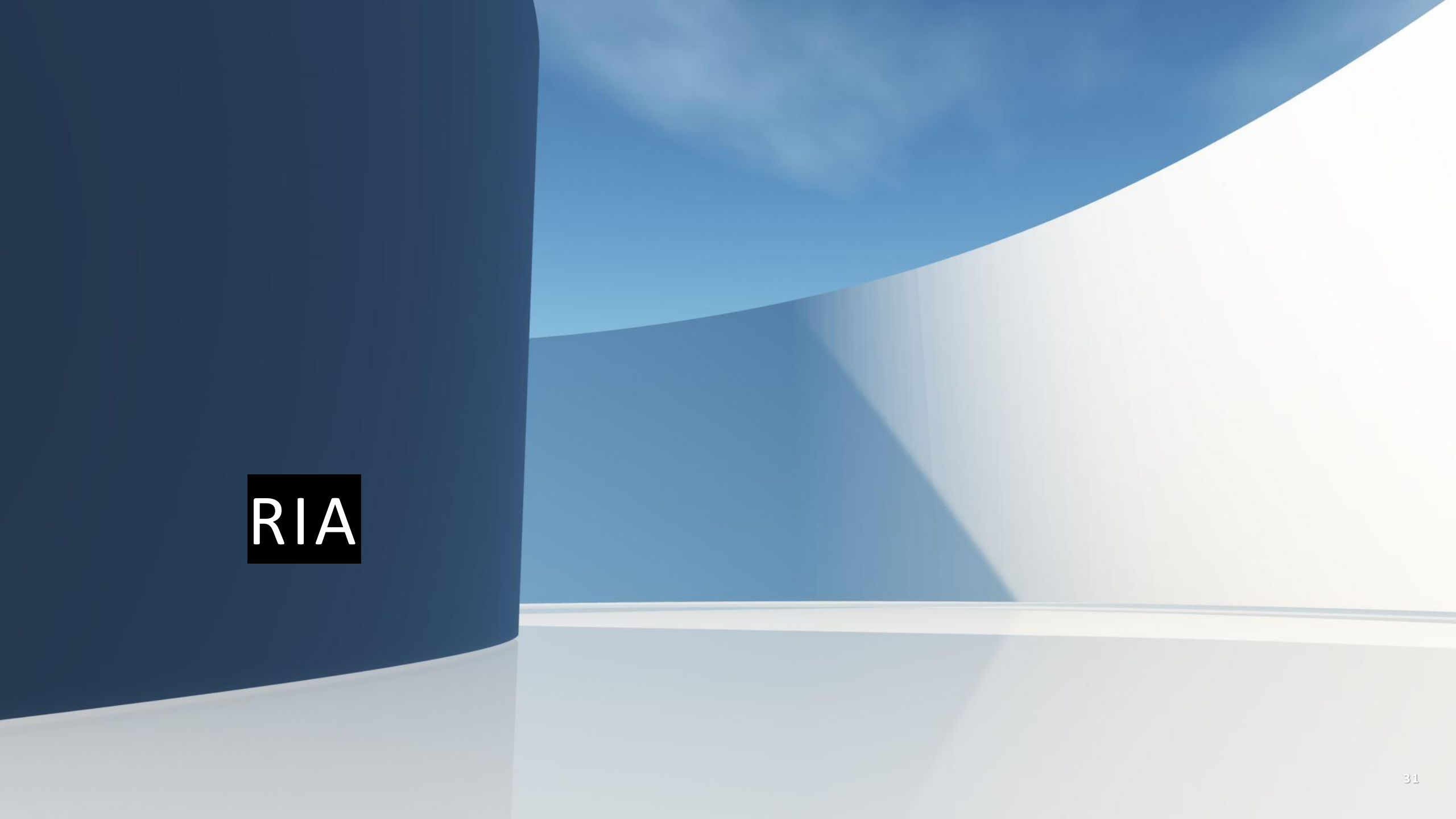
CREATE ORDER



GO TO ORDER



LOGOUT



RIA

JAVASCRIPT-ONLY SPECIFICATIONS

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.

COMPONENTS

Beans:

- CartSupplier
- Order
- Product
- SpendingRanges
- Supplier
- User

Dao:

- OrderDao
- ProductDao
- SpendingRangesDao
- SupplierDao
- UserDao

Controller:

- CheckLogin
- CreateOrder
- GoToHome
- GoToOrder
- GoToResults
- Logout
- ProductsDetails

View:

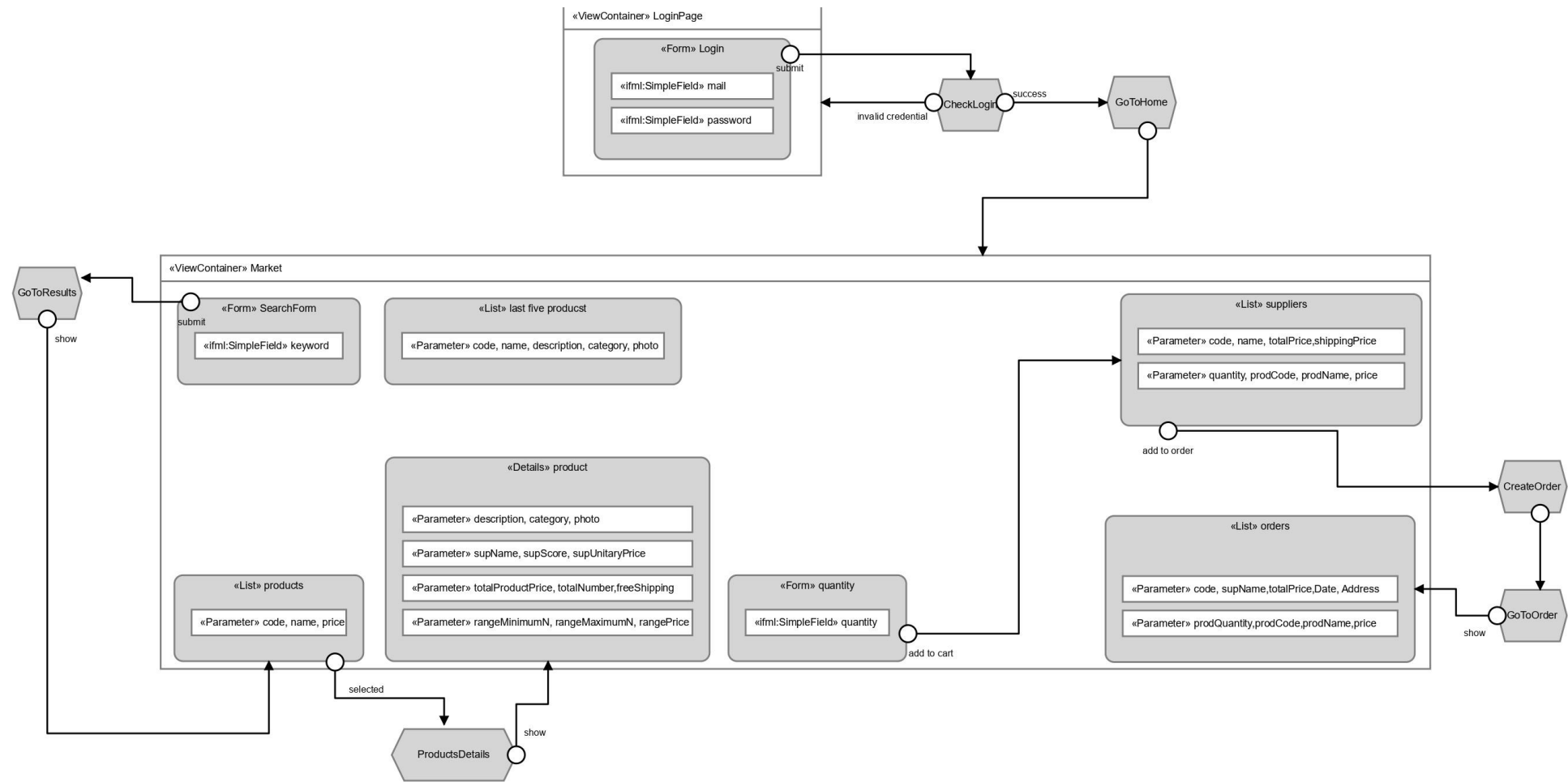
- LoginPage.html
- Market.html

Filter:

- LoginChecker

JAVASCRIPT COMPONENTS

- **PageOrchestrator:** initializes and displays the page components
- **VisualizeProduct:** takes care of displaying the table with the last 5 products visualized
- **SearchForm:** shows the product search form
- **VisualizeSearchProduct:** shows a list of products that contain the search keyword in the name or description
- **VisualizeCartProduct:** shows the cart
- **VisualizeOrderProduct:** shows the order history



EVENTS & ACTIONS

CLIENT SIDE		SERVER SIDE	
EVENT	ACTION	EVENT	ACTION
LoginPage.html → login form → Submit	Check submitted datas	POST (mail, password)	mail & password check
Market → load	Show the Home Page	GET	Retrieves the user's last five visualized products
Market → Search form → Submit	Asking for products that match with the keyword	POST(keyword)	Retrieves the products that contains the keyword in the name or description
Market → link details	Asking for details about a product	POST(productCode)	Retrieves the details about the product clicked
Market → mouseover details	Show a new window with the supplier's products already in cart		
Market → add to cart button → Submit	Add the product to cart and show the cart		

EVENTS & ACTIONS

CLIENT SIDE		SERVER SIDE	
EVENT	ACTION	EVENT	ACTION
Market→ add to order button→Submit	Asking for insert an order and remove form the cart the cartSupplier with his products	POST(cartSupplier)	Create a new order and retrieves a list with the orders of the user
Market→link Logout	Delete the cart	GET	Invalidate session

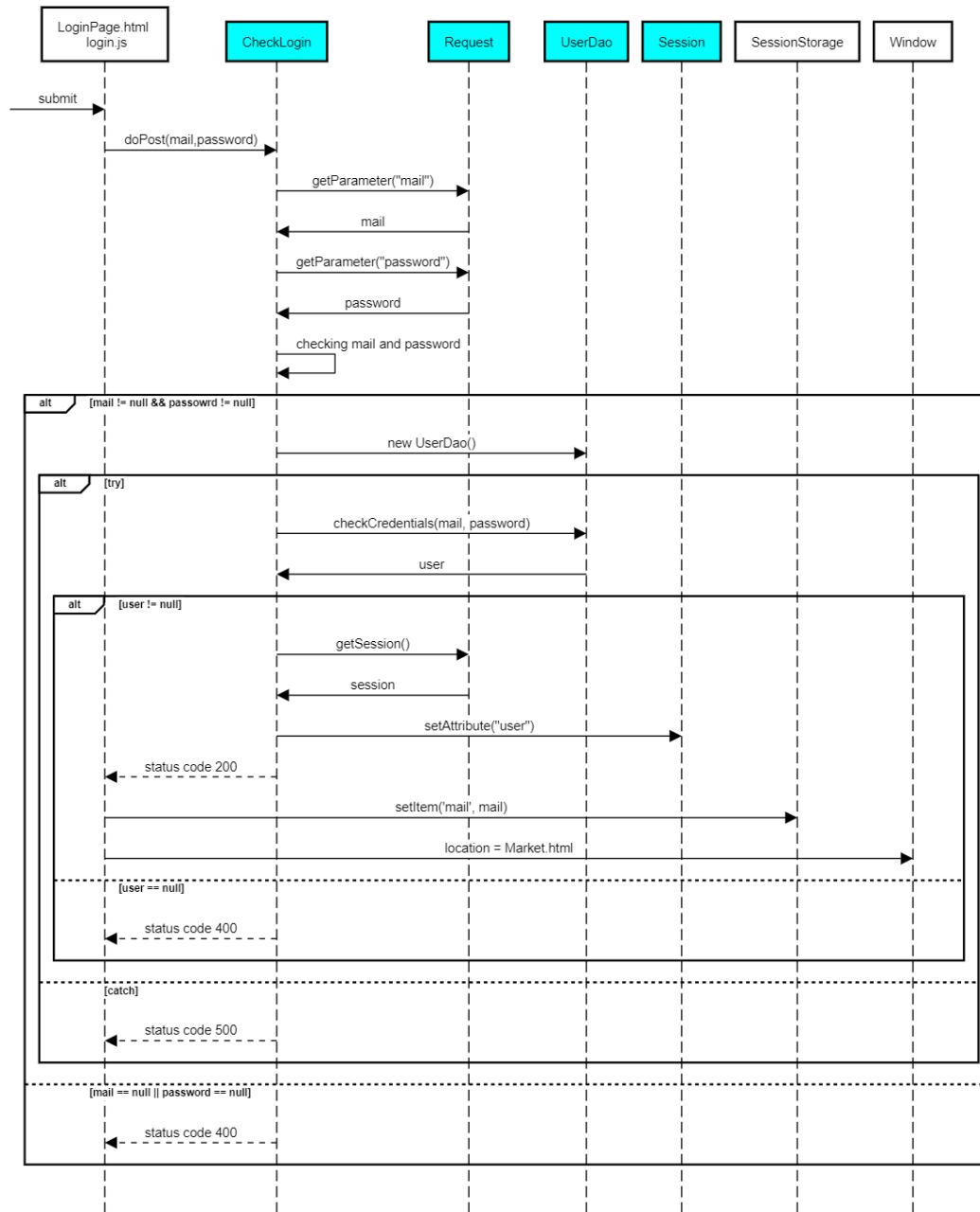
EVENTS & CONTROLLERS

CLIENT SIDE		SERVER SIDE	
EVENT	CONTROLLER	EVENT	CONTROLLER
LoginPage.html → login form → Submit	makeCall	POST (mail, password)	CheckLogin
Market → load	PageOrchestrator, VisualizeProduct	GET	GoToHome
Market → Search form → Submit	SearchForm, VisualizeSearchProduct	POST(keyword)	GoToResults
Market → link details	VisualizeSearchProduct	POST(productCode)	ProductsDetails
Market → mouseover details	VisualizeSearchProduct		
Market → add to cart button → Submit	VisualizeSearchProduct, VisualizeCartProduct		

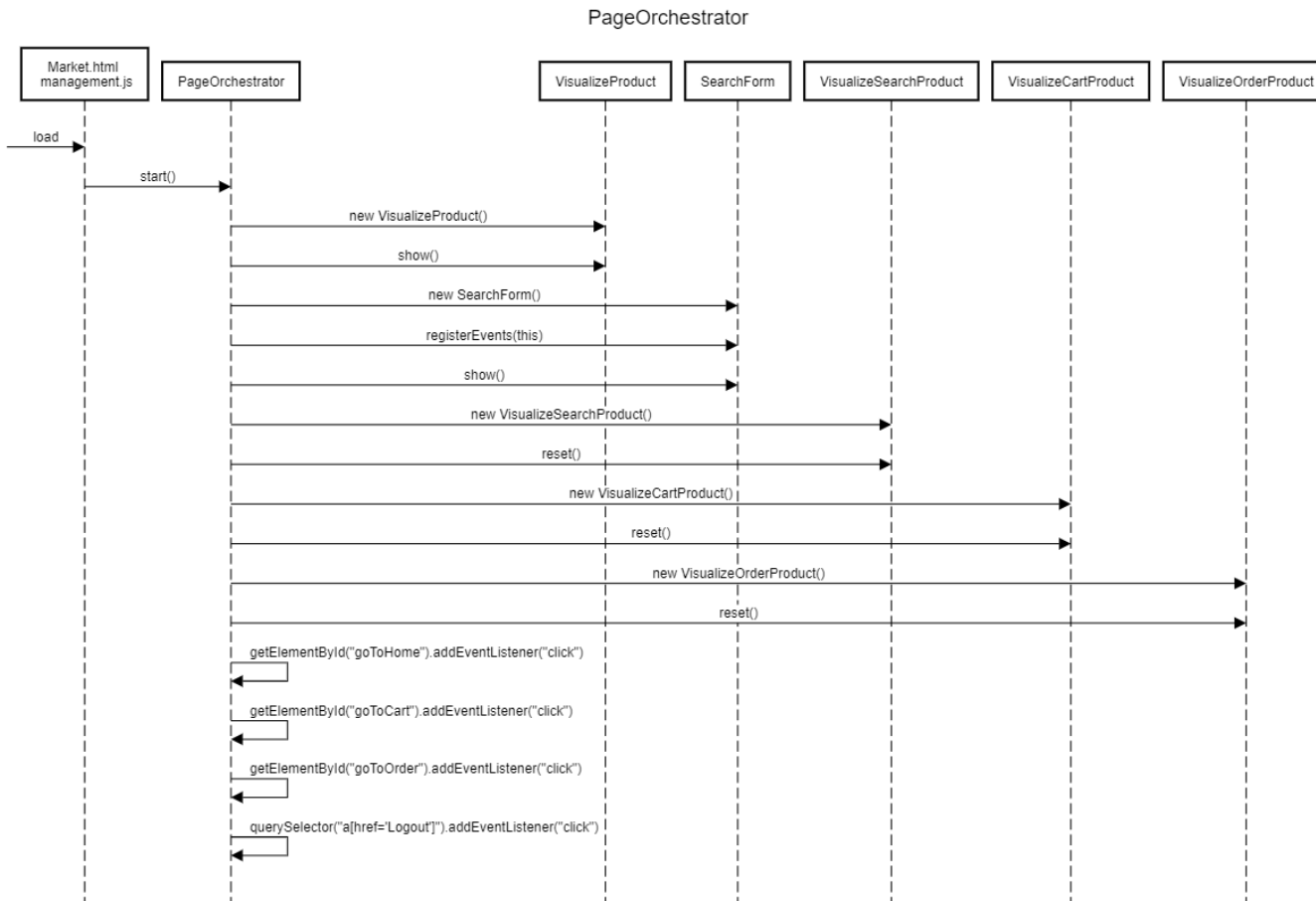
EVENTS & CONTROLLERS

CLIENT SIDE		SERVER SIDE	
EVENT	CONTROLLER	EVENT	CONTROLLER
Market→ add to order button→Submit	VisualizeCartProduct, VisualizeOrderProduct	POST(cartSupplier)	Create Order GoToOrder
Market→link Logout	PageOrchestrator	GET	Logout

Login

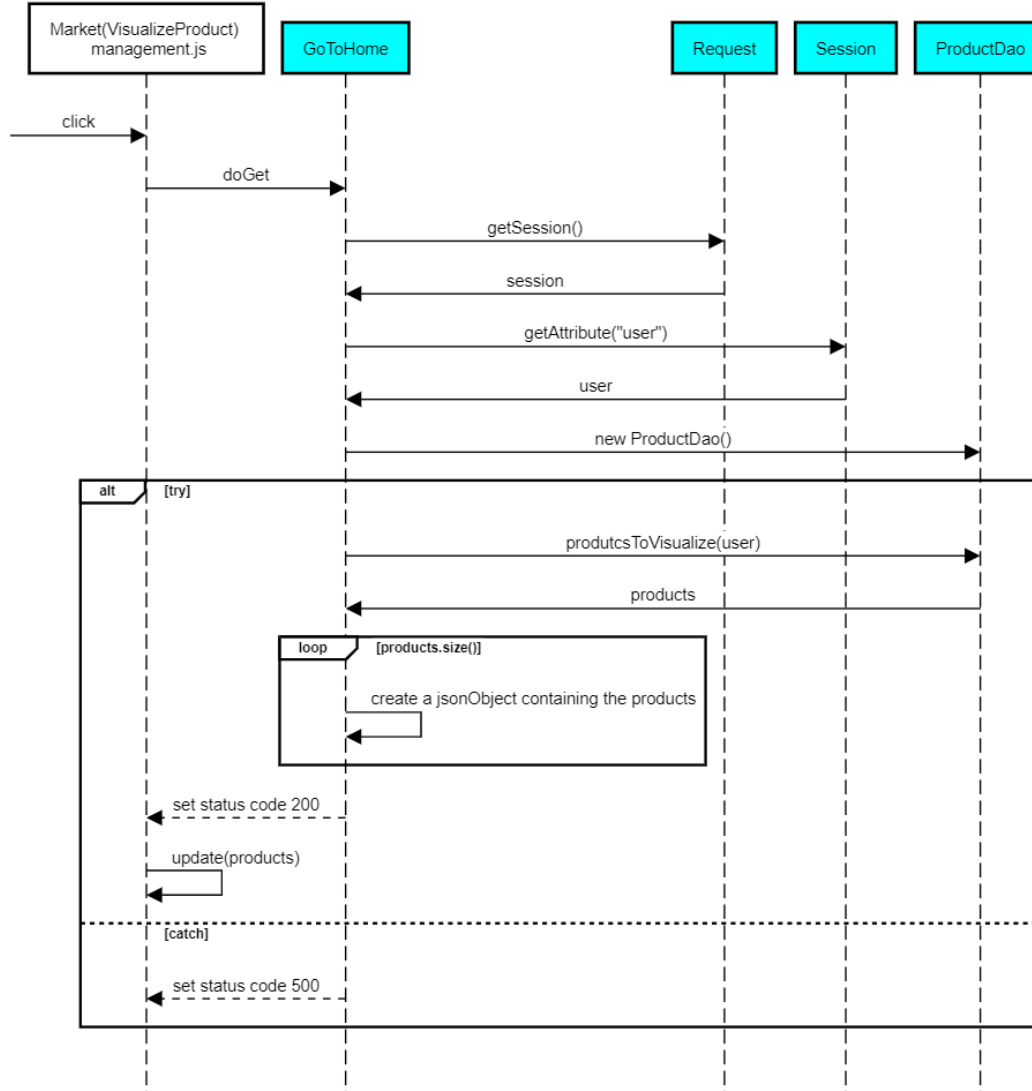


LOGIN



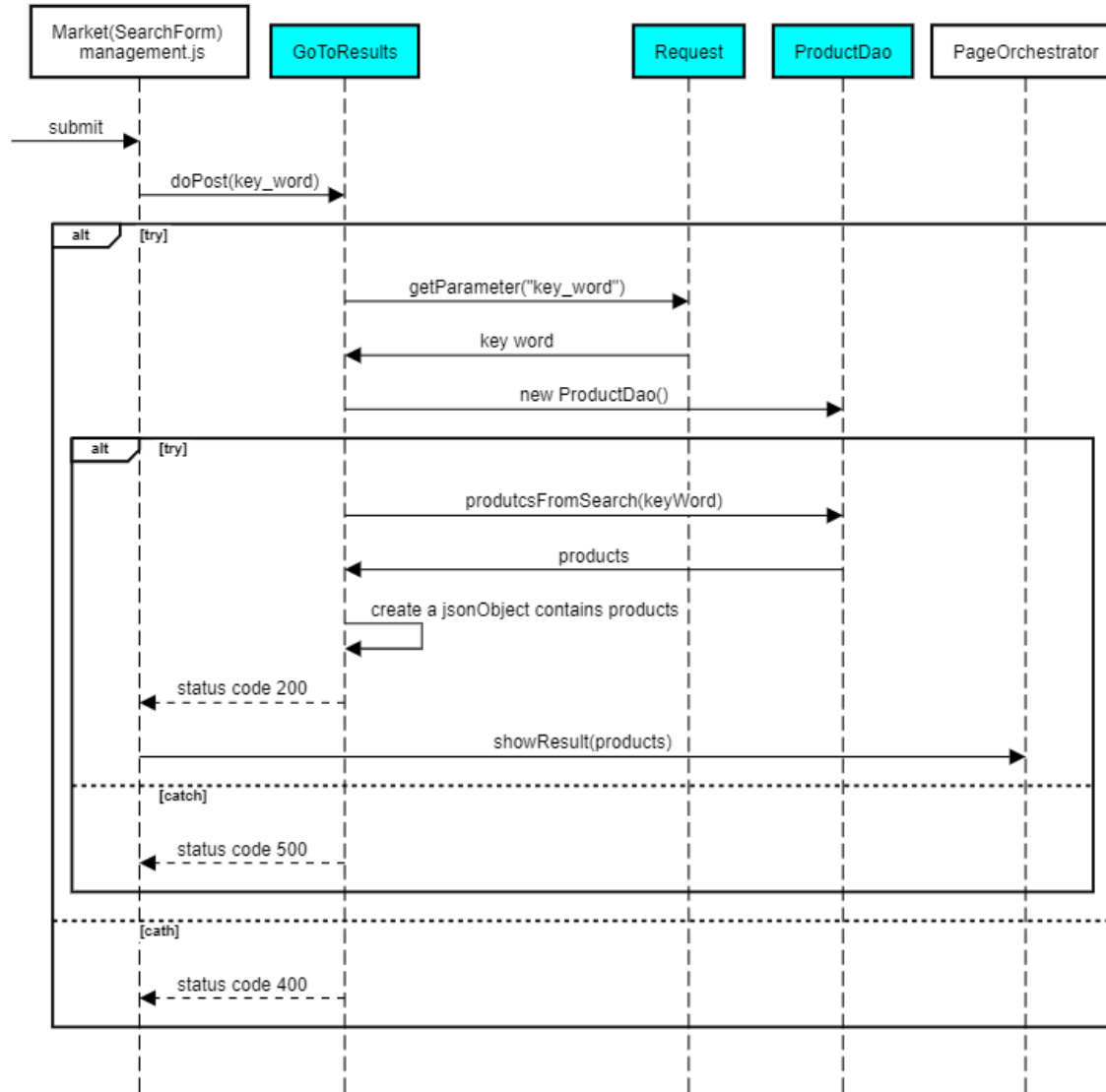
PAGEORCHESTRATOR

GotoHome

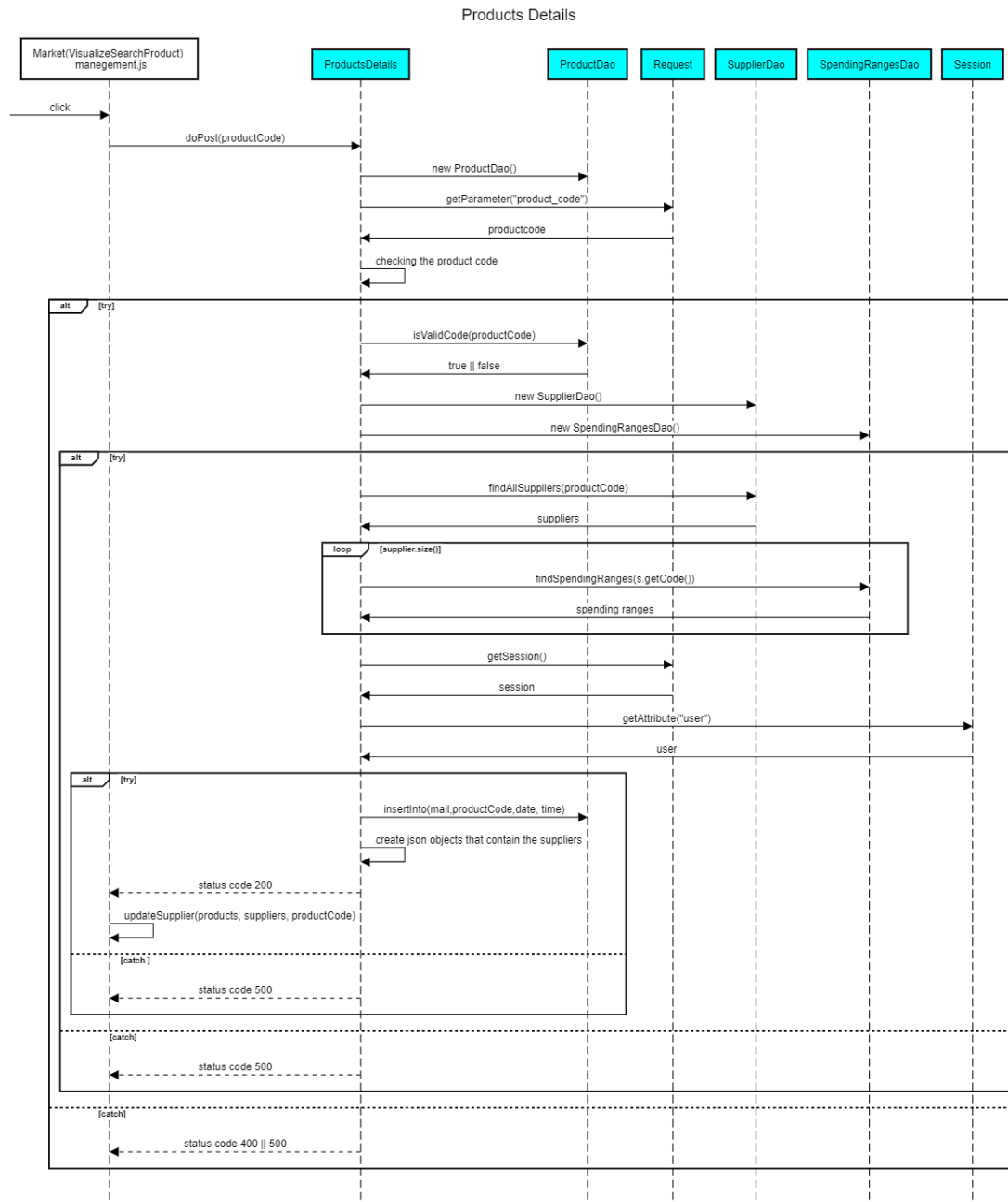


GO TO HOME

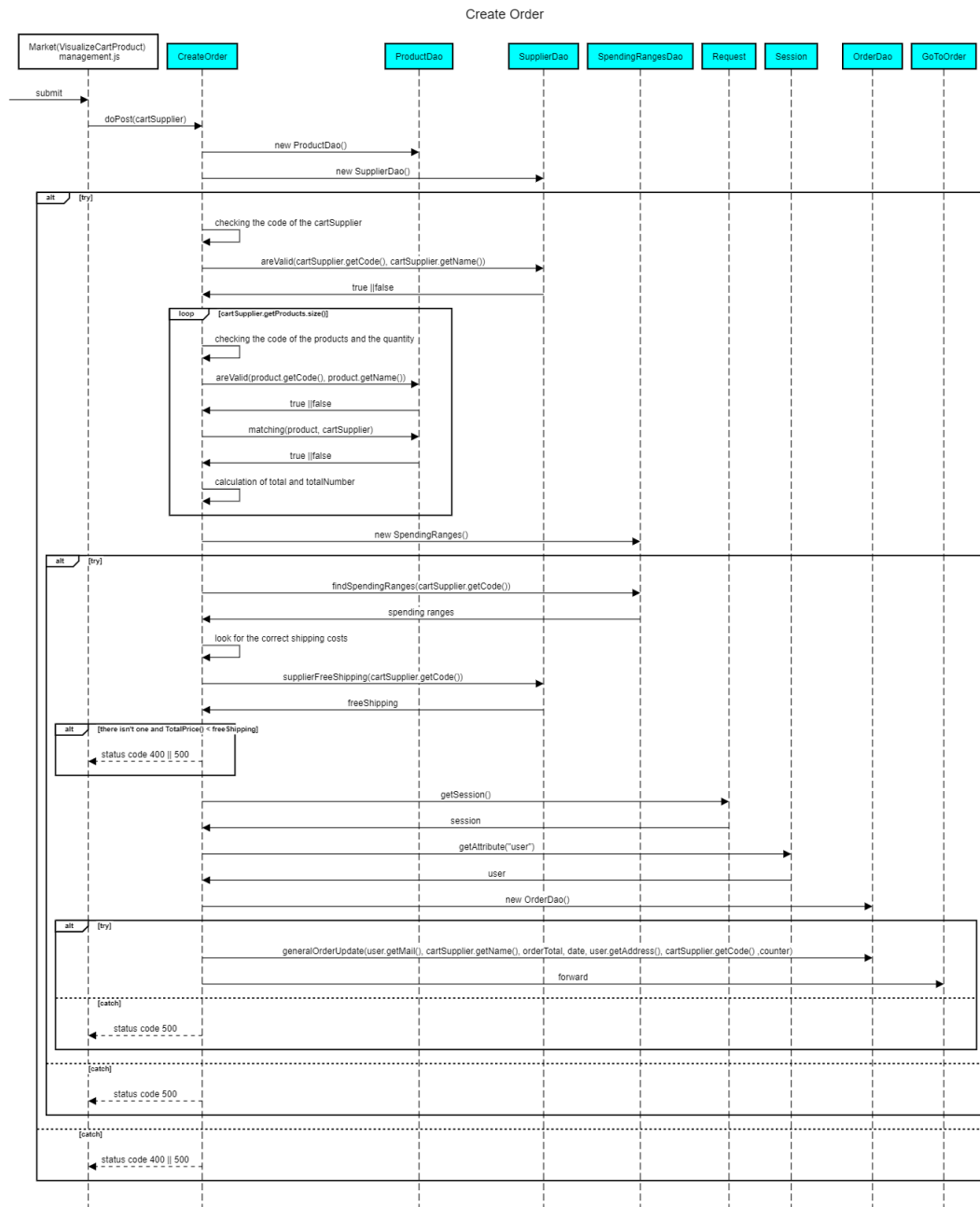
GoToResults



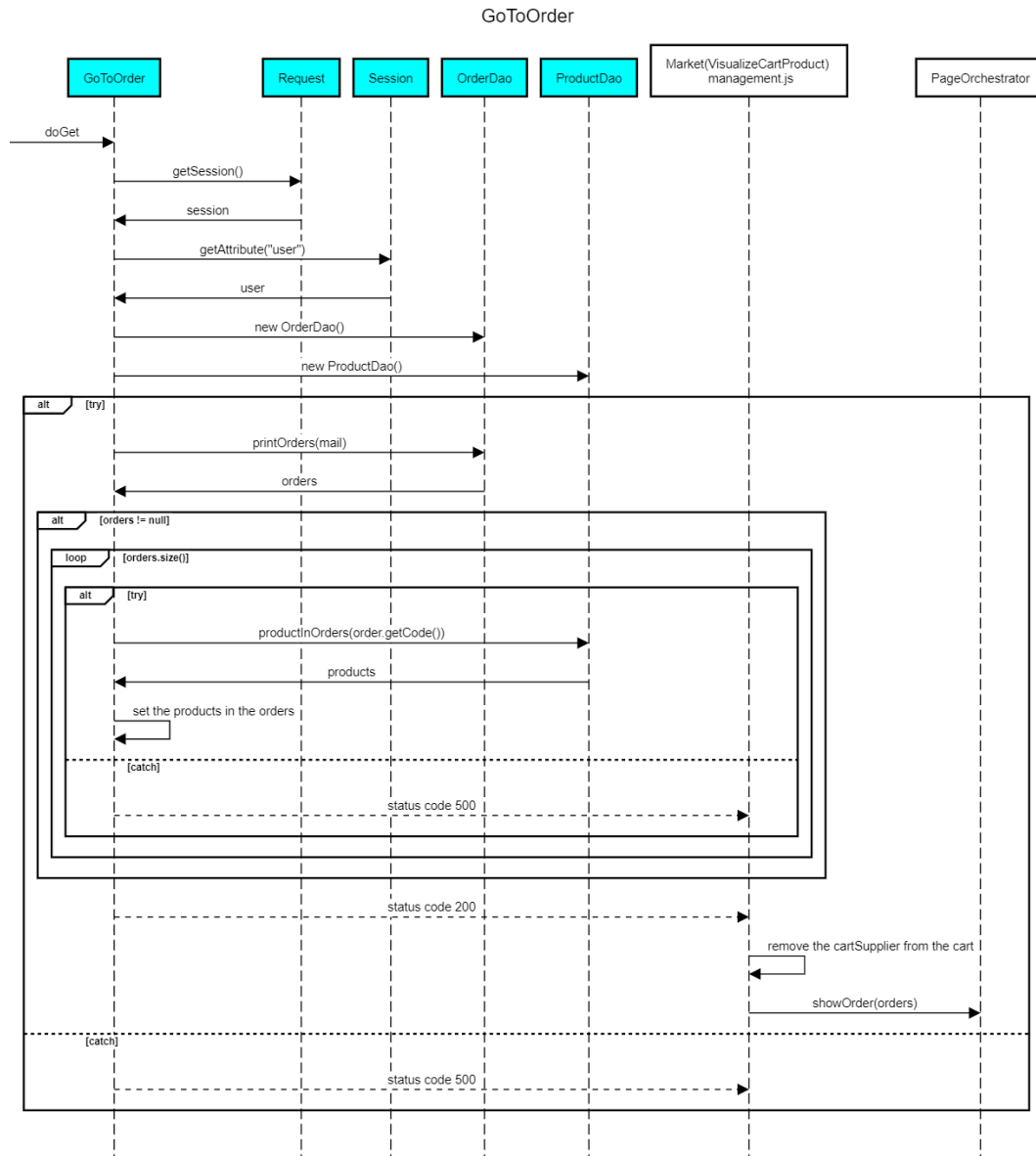
GO TO RESULTS



PRODUCT DETAILS

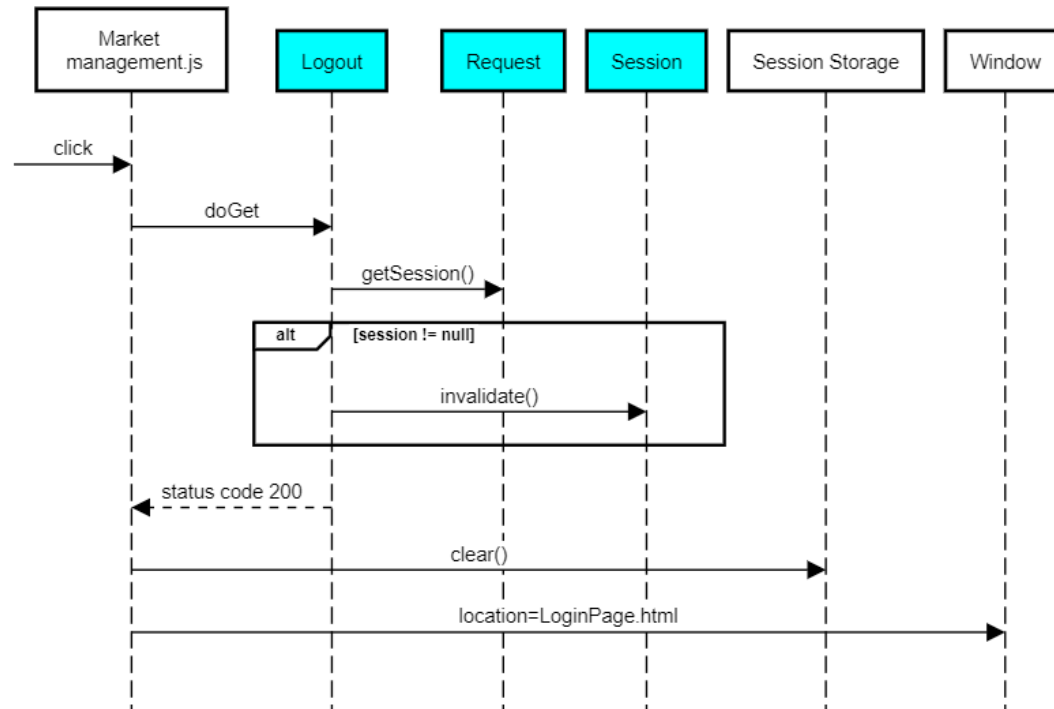


CREATE ORDER



GO TO ORDER

Logout



LOGOUT