

Carrello con più fornitori

pure HTML version

Project TIW 2022/2023

Yang Hao Mao

Ken Chen

Group 100

Specifications [pureHTML] (1/2)

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).

Specifications [pureHTML] (2/2)

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

Data Base Analysis [pureHTML]

Entities, attributes, relationships

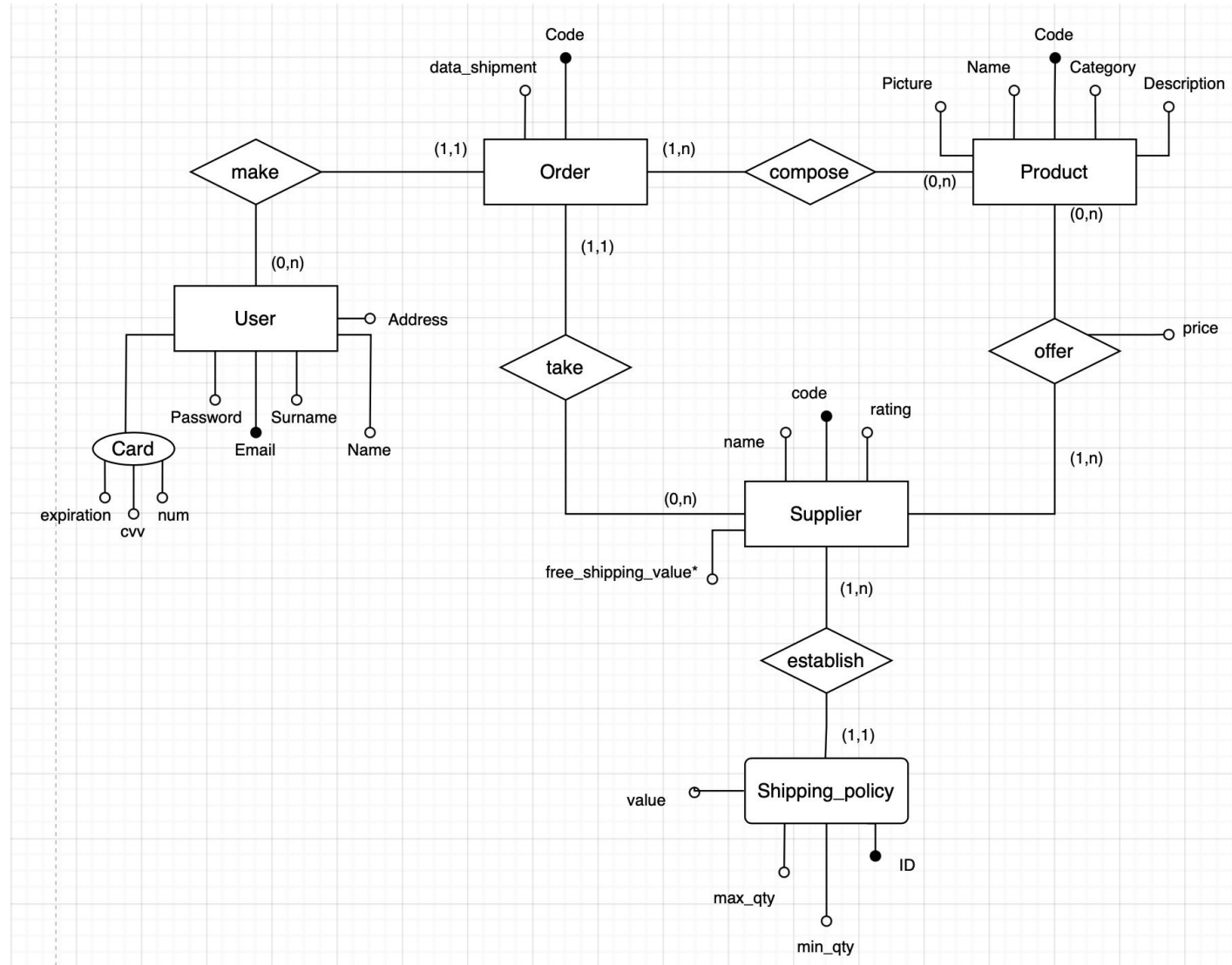
Un'applicazione di commercio elettronico consente all'**utente** (acquirente) di **visualizzare** un catalogo di **prodotti venduti** da diversi **fornitori**, inserire prodotti in un carrello della spesa e **creare** un **ordine** di acquisto a partire dal contenuto del carrello. Un prodotto ha un **codice** (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.

.....

.....

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.

Data Base design



Data Base design

Logical scheme:

USER(**Email** , Name , Surname, Password, Address, Card_number, Card_expiration, Card_Cvv)

ORDER(**Code** , Code_Supplier , Data_Shipment)

PRODUCT(**Code** , Name , Category , Description , Photo)

compose(**Code_Order** , **Code_Product**)

SUPPLIER(**Code** , name, rating, free_shipping*)

offer(**Code_Product** , **Code_Supplier** , price)

SHIPPING_POLICY(**ID** , **Code_Supplier** , max_qty, min_qty, value)

Foreign keys:

ORDER.Code_Supplier , offer.Code_Supplier, SHIPPING_POLICY.Code_Supplier->SUPPLIER.Code

compose.Code_Order-> Order.Code,

compose.Code_Product, offer.Code_Product-> PRODUCT.Code

Local Database schema

```
CREATE TABLE `user` (  
  `name` varchar(45) NOT NULL,  
  `surname` varchar(45) NOT NULL,  
  `email` varchar(45) NOT NULL,  
  `password` varchar(45) NOT NULL,  
  `address` varchar(45) NOT NULL,  
  `card_number` varchar(16) NOT NULL,  
  `card_expiration` date NOT NULL,  
  `card_cvv` int NOT NULL,  
  PRIMARY KEY (`email`),  
  UNIQUE KEY `email_UNIQUE` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `order` (  
  `code` int NOT NULL AUTO_INCREMENT,  
  `suplier_code` varchar(20) NOT NULL,  
  `user_email` varchar(45) NOT NULL,  
  `total_value` float NOT NULL,  
  `date_of_shipment` date NOT NULL,  
  PRIMARY KEY (`code`),  
  KEY `user_email_idx` (`user_email`),  
  KEY `supllier_code_idx` (`suplier_code`),  
  CONSTRAINT `supllier_code` FOREIGN KEY (`suplier_code`) REFERENCES `supplier` (`code`),  
  CONSTRAINT `user_email` FOREIGN KEY (`user_email`) REFERENCES `user` (`email`)  
) ENGINE=InnoDB AUTO_INCREMENT=24 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Local Database schema

```
CREATE TABLE `order_composition` (  
  `code_order` int NOT NULL,  
  `code_product` varchar(20) NOT NULL,  
  PRIMARY KEY (`code_product`, `code_order`),  
  KEY `order_code_idx` (`code_order`),  
  KEY `product_code_idx` (`code_product`),  
  CONSTRAINT `order_code` FOREIGN KEY (`code_order`) REFERENCES `order` (`code`),  
  CONSTRAINT `product_code` FOREIGN KEY (`code_product`) REFERENCES `product` (`code`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `product` (  
  `code` varchar(20) NOT NULL,  
  `name` varchar(100) NOT NULL,  
  `category` varchar(45) NOT NULL,  
  `description` varchar(500) DEFAULT NULL,  
  `photo` varchar(100) DEFAULT NULL,  
  PRIMARY KEY (`code`),  
  UNIQUE KEY `code_UNIQUE` (`code`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```


Local Database schema

```
CREATE TABLE `supplier` (  
  `code` varchar(20) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `rating` int NOT NULL,  
  `free_shipping` float DEFAULT NULL,  
  PRIMARY KEY (`code`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `offer` (  
  `product_code` varchar(20) NOT NULL,  
  `supplier_code` varchar(20) NOT NULL,  
  `price` float NOT NULL,  
  PRIMARY KEY (`product_code`, `supplier_code`),  
  KEY `supplier_code_idx` (`supplier_code`),  
  CONSTRAINT `product_id` FOREIGN KEY (`product_code`) REFERENCES `product` (`code`),  
  CONSTRAINT `supplier_id` FOREIGN KEY (`supplier_code`) REFERENCES `supplier` (`code`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `spending_range` (  
  `id` varchar(20) NOT NULL,  
  `supplier_code` varchar(20) NOT NULL,  
  `min_qty` int NOT NULL,  
  `max_qty` int NOT NULL,  
  `price` float NOT NULL,  
  PRIMARY KEY (`id`, `supplier_code`),  
  KEY `supplier_code_idx` (`supplier_code`),  
  CONSTRAINT `supplier_code` FOREIGN KEY (`supplier_code`) REFERENCES `supplier` (`code`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Application Requirements Analysis [pureHTML] (1/2)

Pages (views), view components, events, actions

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).

Application Requirements Analysis [pureHTML] (2/2)

Pages (views), view components, events, actions

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**.

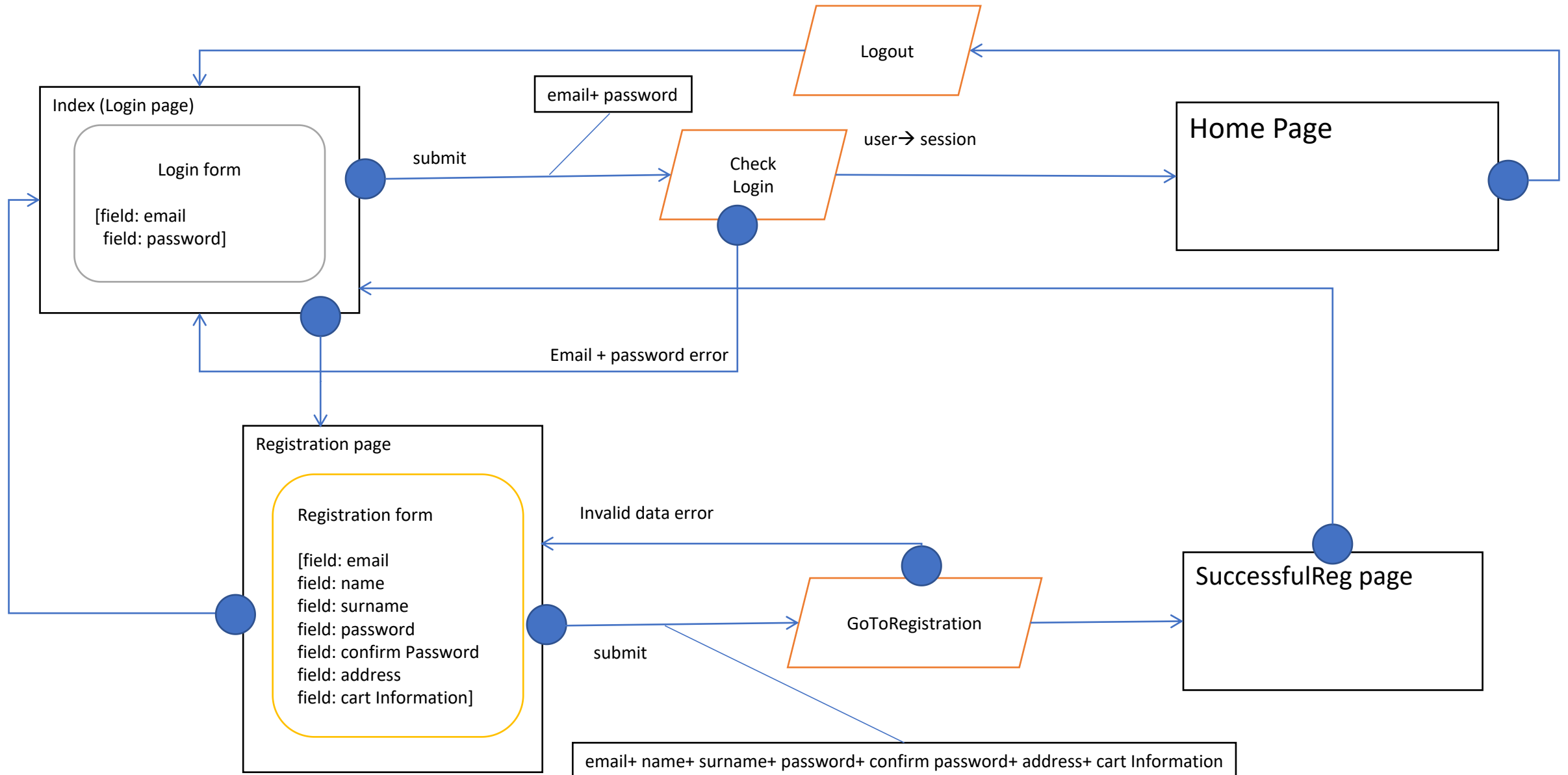
Premere il bottone comporta **l'eliminazione dei prodotti del fornitore dal carrello e la creazione di un ordine corrispondente**. Un ordine ha un codice, il nome del fornitore, l'elenco dei prodotti, un valore totale composto dalla somma del valore dei prodotti e delle spese di spedizione, una data 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

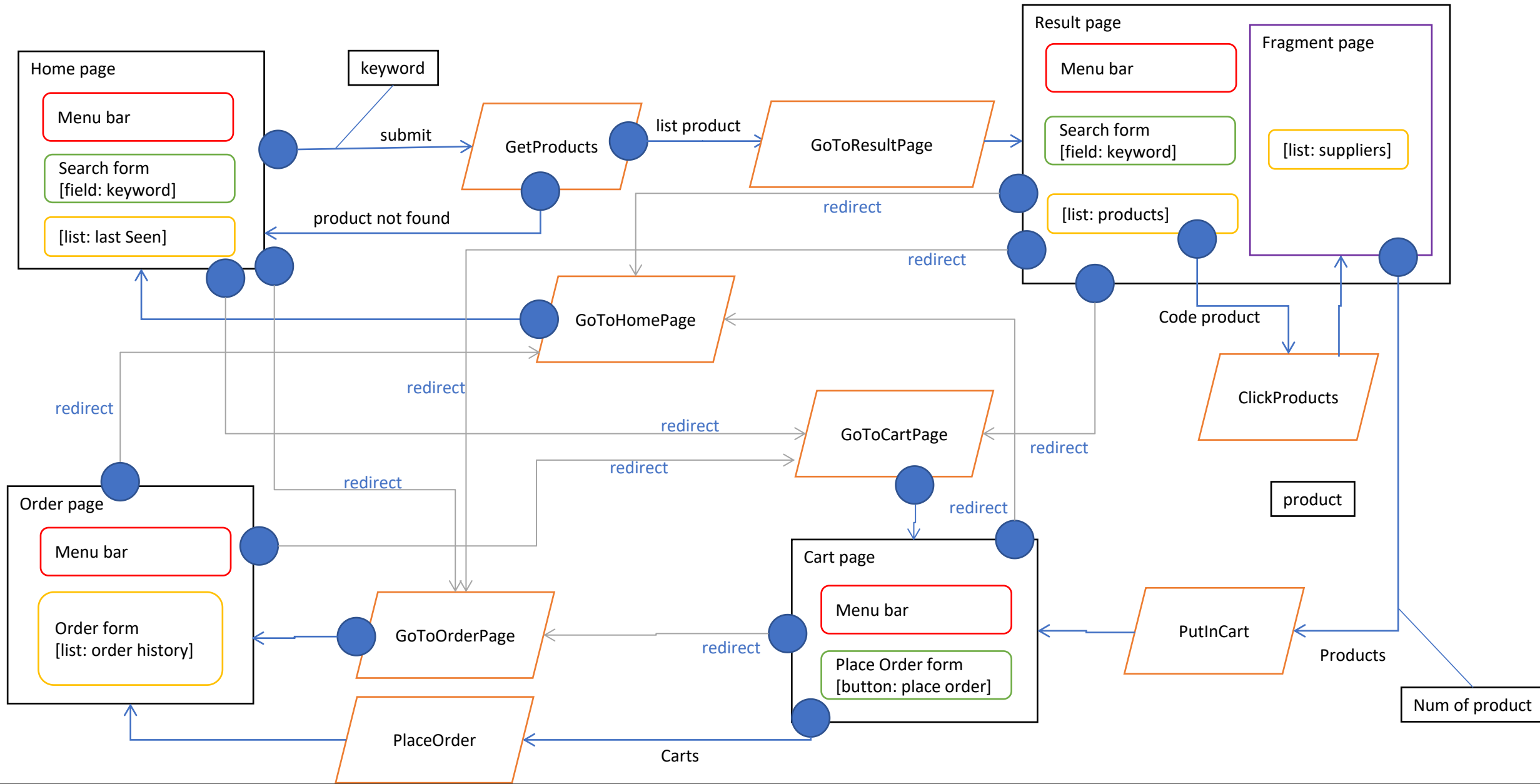
Completamento delle specifiche

- L'applicazione gestisce anche la registrazione di un nuovo account, dove i dati vengono inseriti tramite un form e poi caricati nel data base.
- La **pagina di default** contiene il **form di login** visibile a tutti.
- Le **pagine Home, Orders, Cart, Results** presentano un menu, che contiene i **link alle pagine Home, Orders, Cart** e un **pulsante per il Logout**, il quale **riporta** l'utente alla **pagina di default**.
- La **pagina Home** contiene la **lista degli ultimi 5 prodotti visualizzati** dall'utente. Inoltre, la pagina contiene un **form che permette la ricerca di prodotti tramite una parola chiave**. La **sottomissione del form** avviene tramite un **pulsante Search**, con il quale **viene aperta** la **pagina Results**.
- La **pagina Results** contiene la **lista dei prodotti** trovati tramite la parola chiave. Per **visualizzare un prodotto** e, quindi, vederne i **dettagli**, **si preme il pulsante Details**. I **dettagli sono visualizzabili** nella stessa pagina.
- I **dettagli**, del prodotto visualizzato, mostrano una **lista di fornitori** che offrono il prodotto ad un prezzo differente, per ogni fornitore viene mostrato la **lista delle fasce di spesa** e un **form che permette di scegliere la quantità** di prodotto da inserire nel carrello. **L'azione di inserimento** avviene tramite un **pulsante PutInCart**, il quale **porta** alla **pagina Cart**.
- La **pagina Cart** contiene la **lista dei fornitori**, dove per ogni fornitore è mostrato **la lista di prodotti nel carrello** del medesimo fornitore. Per ogni fornitore è possibile **effettuare l'ordine** tramite il **click** di un **pulsante PlaceOrder**. L'ordine sarà visibile nella **pagina Orders** **cliccando il menu di navigazione**.
- La **pagina Orders** contiene la **lista degli ordini** effettuati dall'utente.

Application Design - (1/2)



Application Design - (2/2)

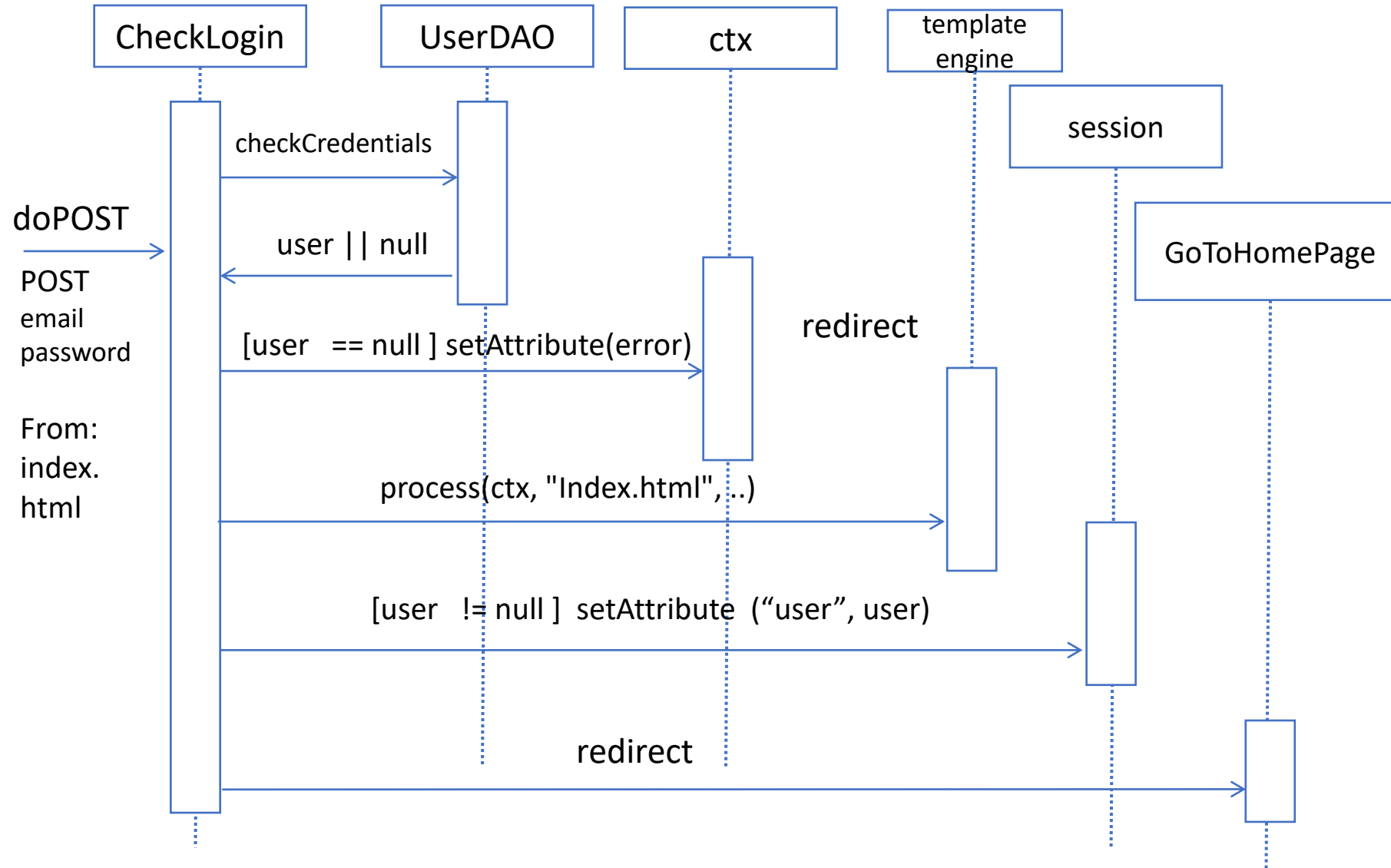


Components

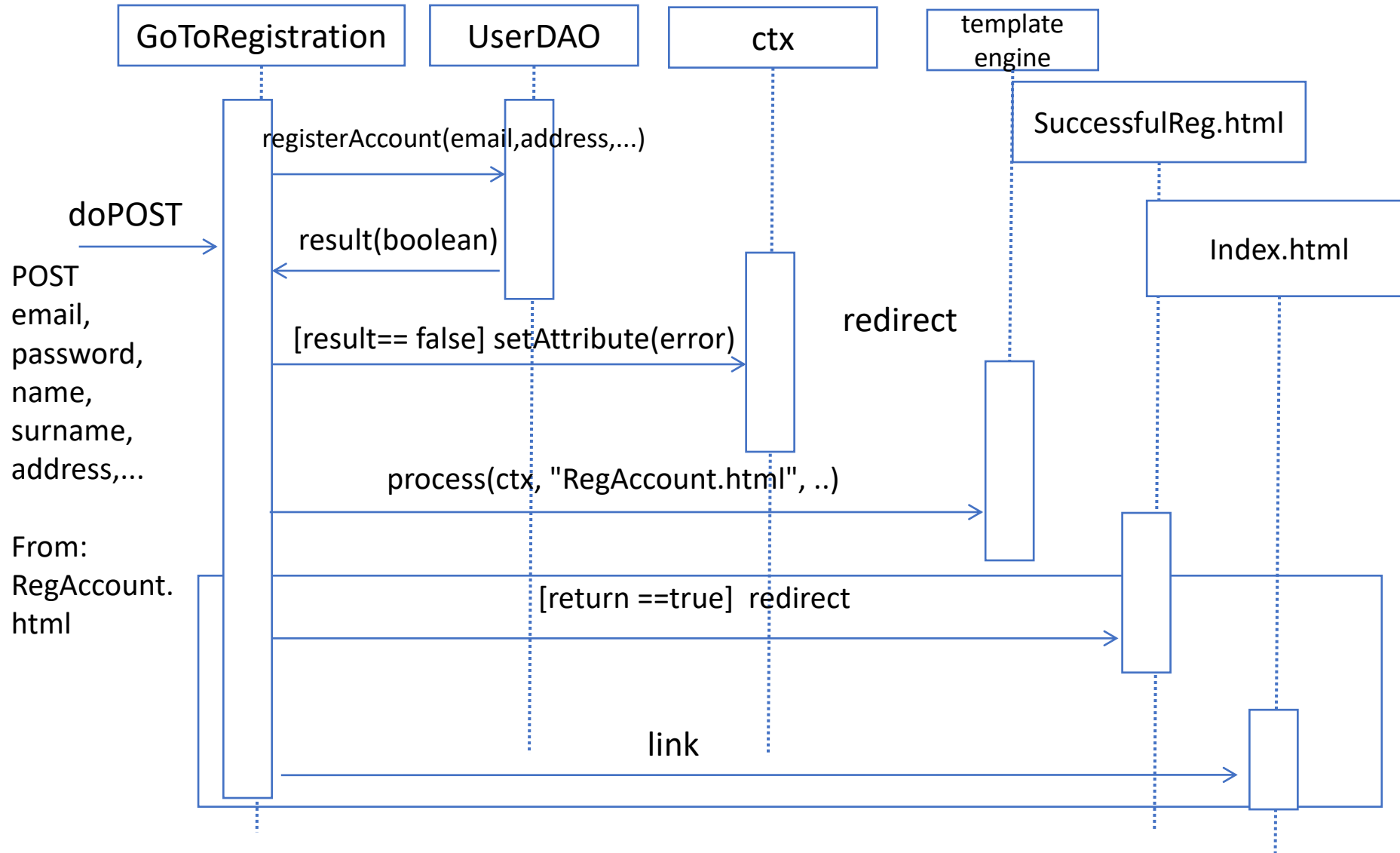
- **Model Objects (Beans)**
 - User
 - Supplier
 - SupplierOffer
 - Offer
 - SpendingRange
 - Product
 - PreOrder
 - Order
 - CartManager
 - ChronologyManager
- **Data Access Objects (Classes)**
 - **UserDAO**
 - ❑ existsAccount(email)
 - ❑ checkCredentials(email , password)
 - ❑ registerAccount(email , surname, name, password...)
 - **SupplierDAO**
 - ❑ getProductSupplierList(product_code)
 - ❑ getSupplier(supplier_code)
 - **ShopDAO**
 - ❑ searchProduct(search)
 - ❑ getProduct(product_code)
 - ❑ getProductWithPrice(product_code)
 - ❑ getCategoryProducts(default_category)
 - ❑ getRandomCategory()
 - **OrderDAO**
 - ❑ uploadOrder(order)
 - ❑ getOrder(user)
 - **RangeDAO**
 - ❑ getRanges(supplier_code)
- **Controllers (Servlets)**
 - CheckLogin
 - GoToRegistration
 - GoToHomePage
 - ClickProduct
 - GetProducts
 - PlaceOrder
 - PutInCart
 - GotToCart
 - GoToOrder
 - GoToResultsPage
 - Logout
- **Utils (Classes)**
 - Chronology
 - PriceQuantity
 - UserCart
 - OrdersManager
- **Views (Templates)**
 - Index.html
 - RegAccount.html
 - Home.html
 - Cart.html
 - Order.html
 - Results.html
 - Supplier_Fragment.html
 - SuccessfulReg.html

Sequence diagrams

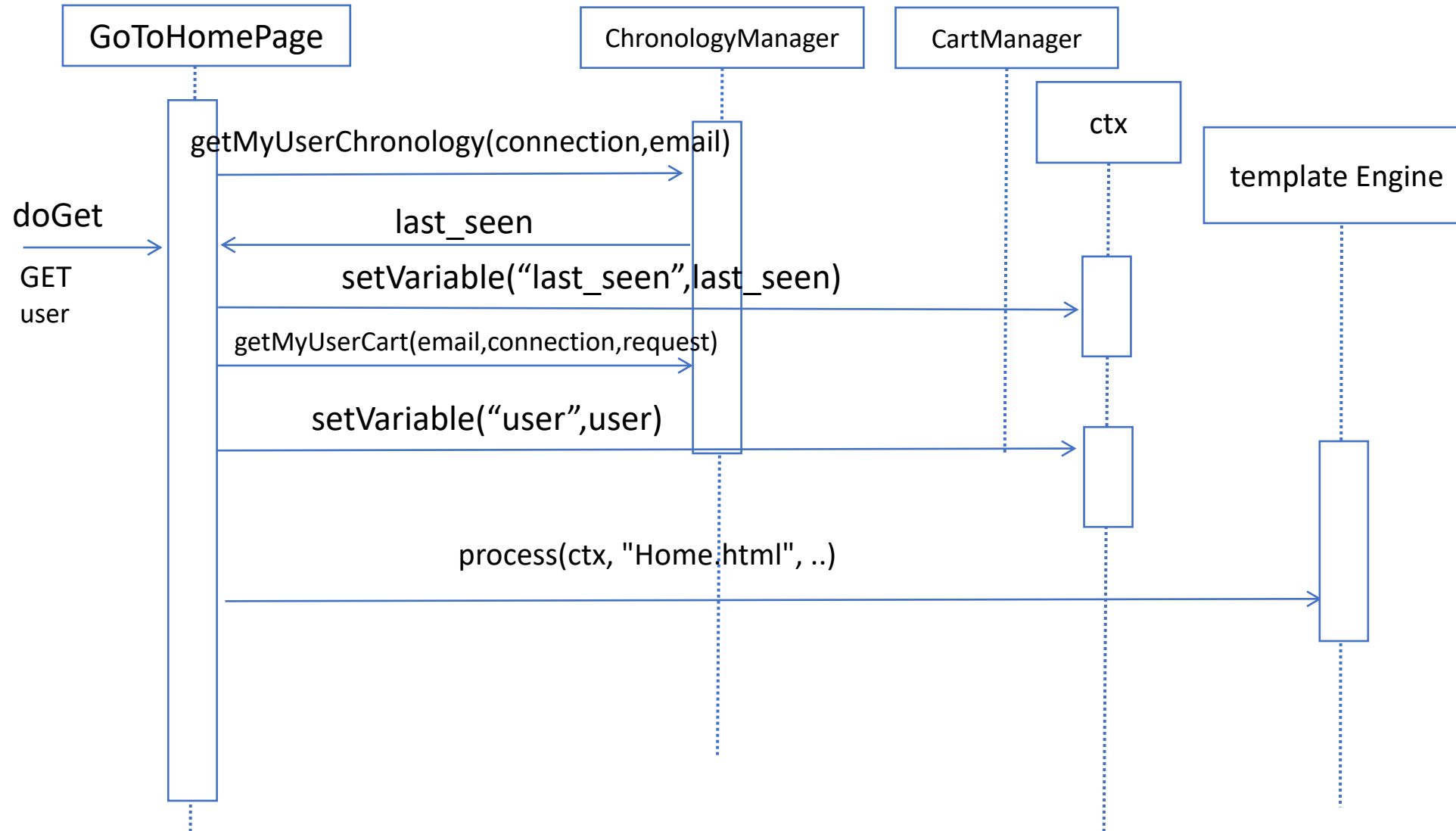
Event: Check Login



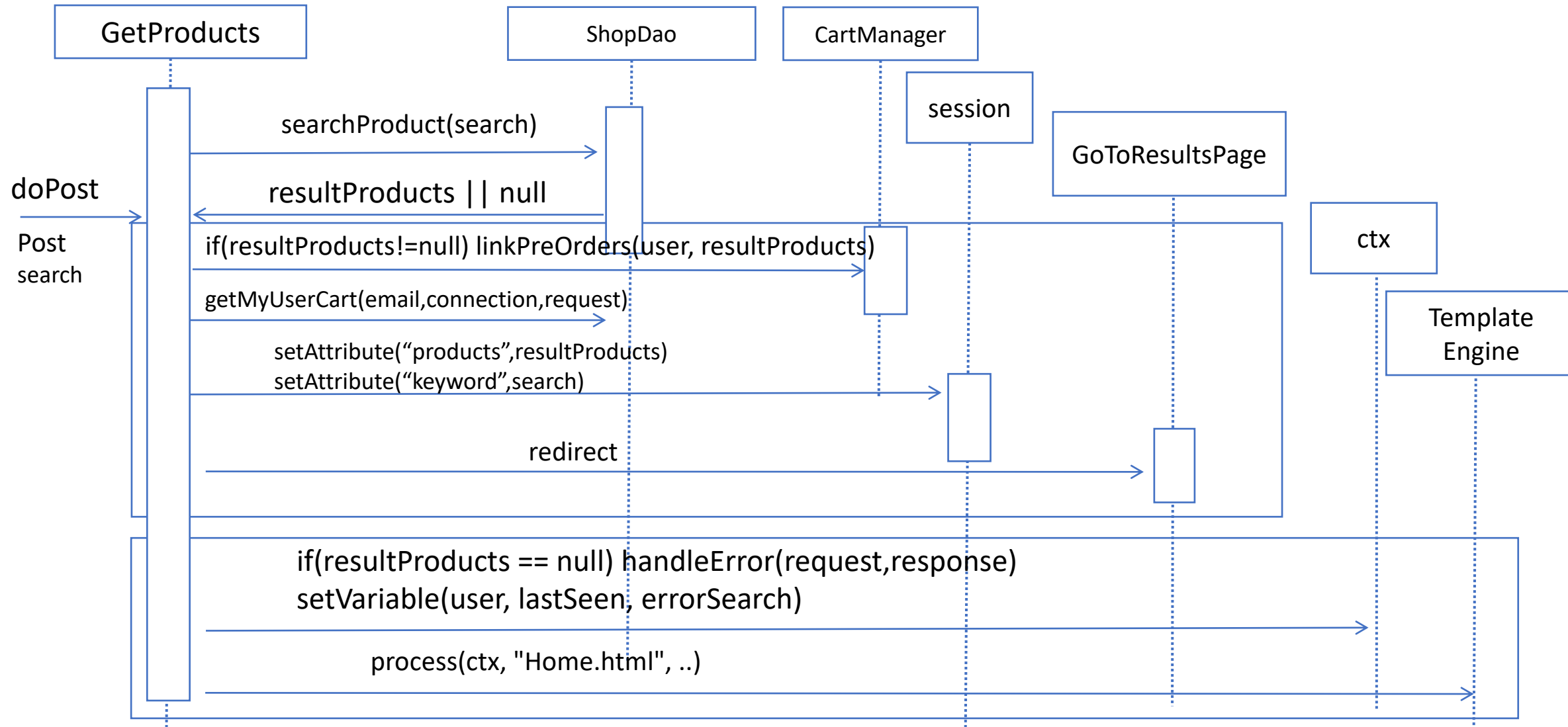
Event: Registration



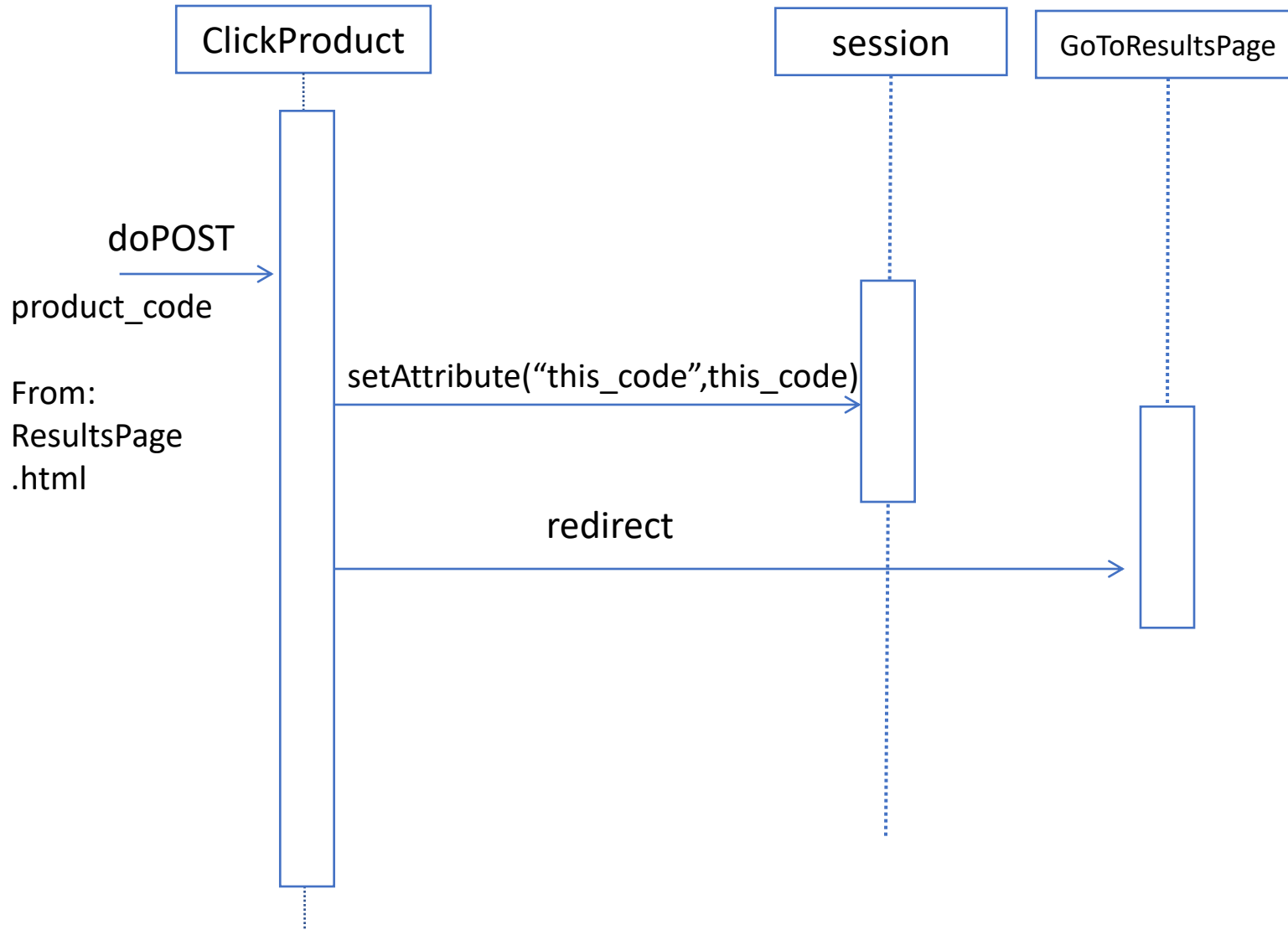
Event: Go To Home Page



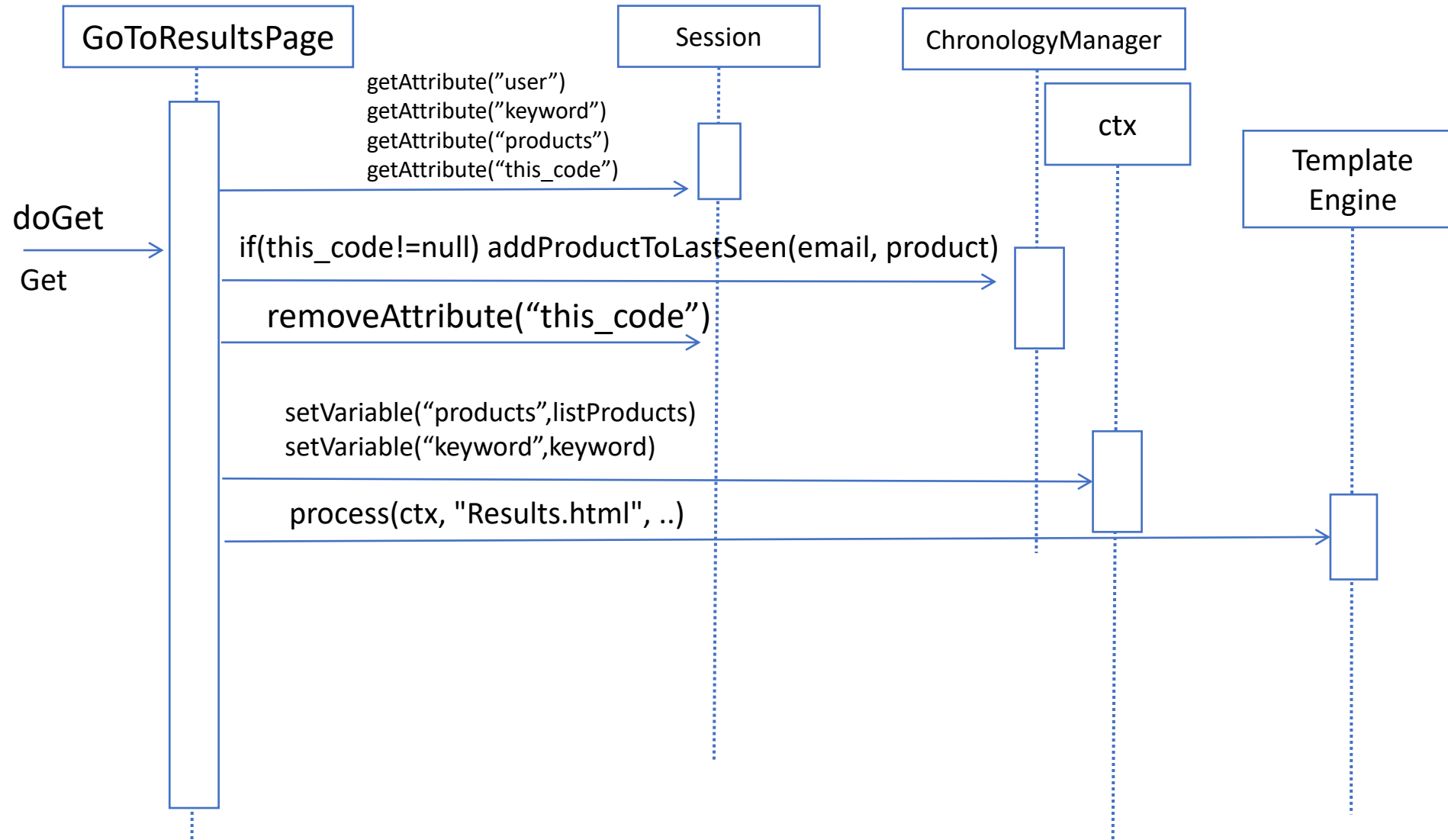
Event: Search Products



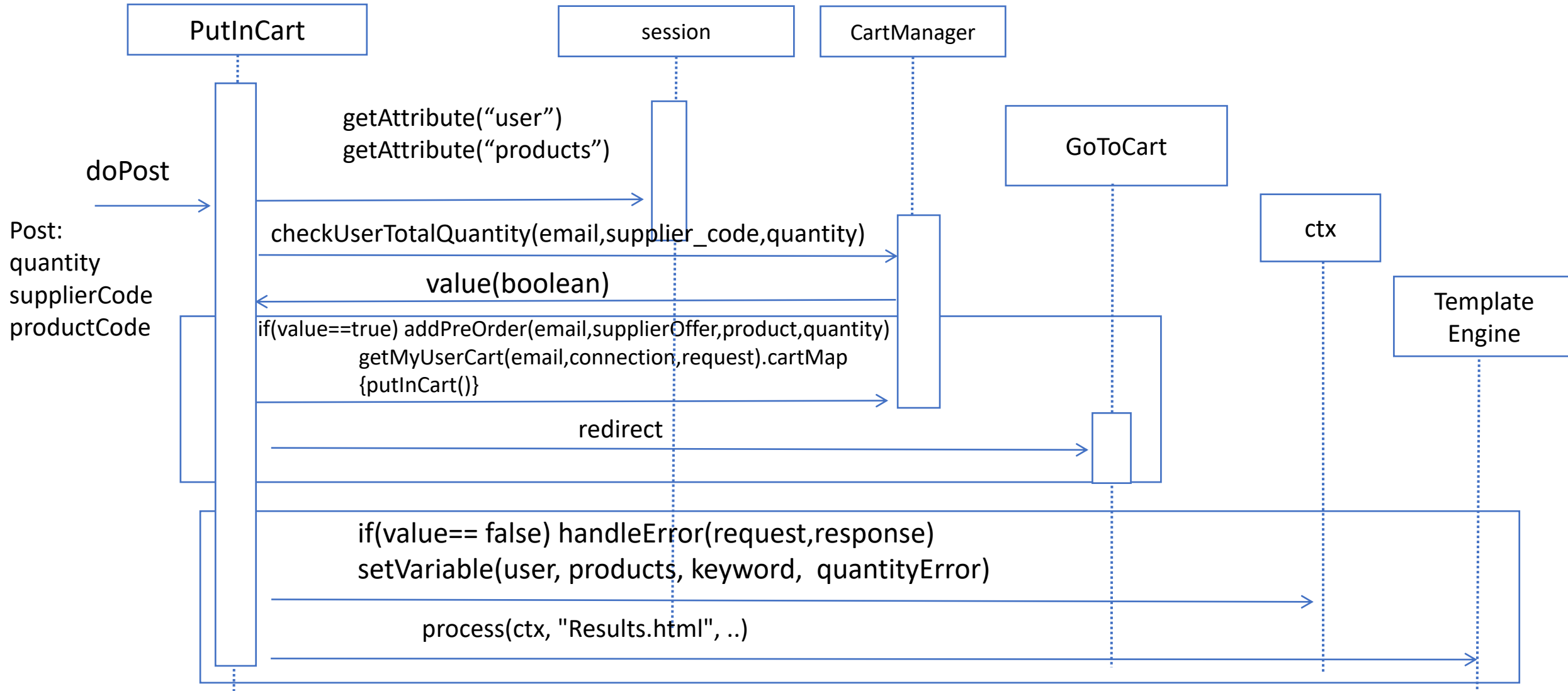
Event: Click Product



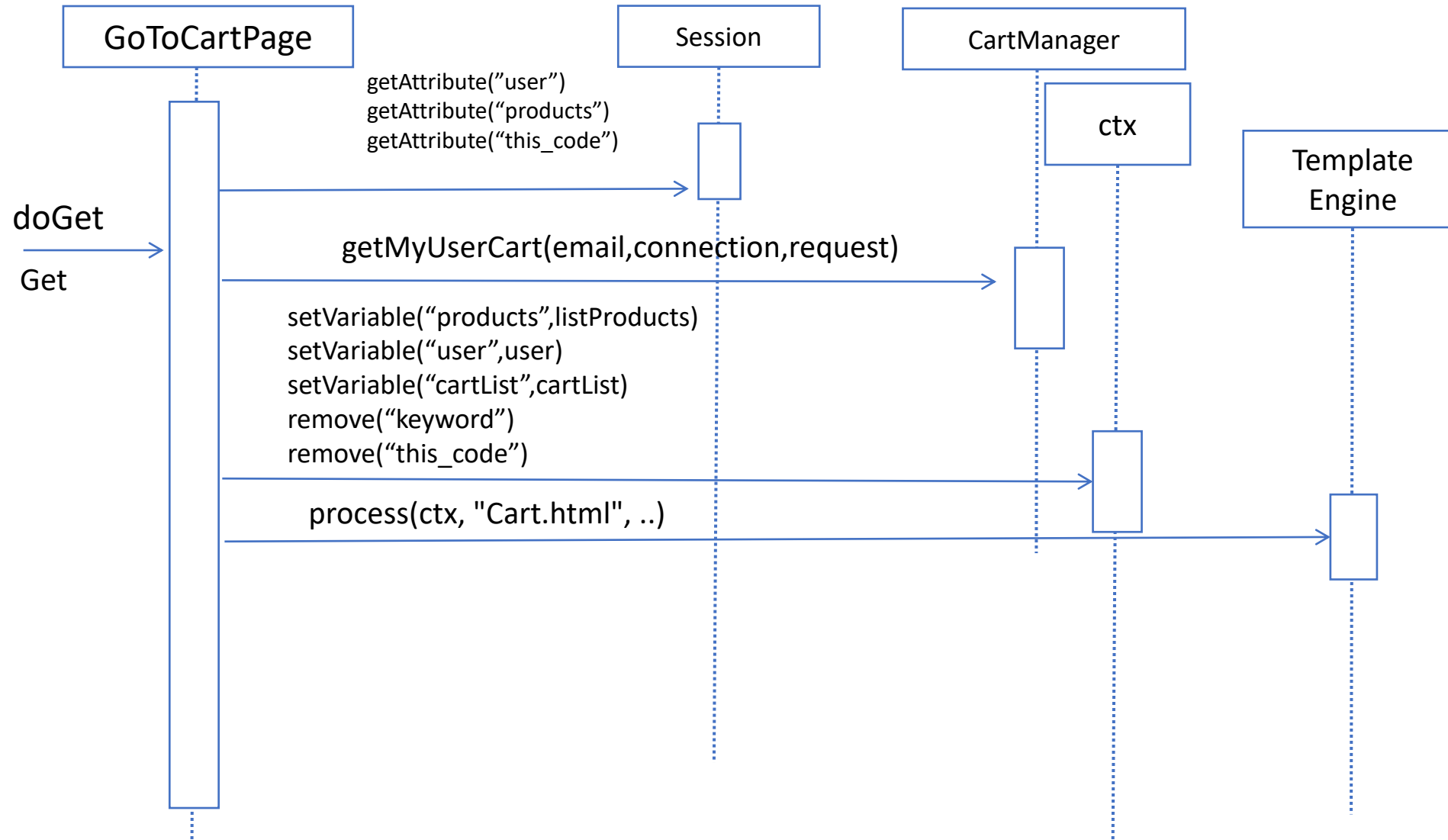
Event: GoToResultsPage



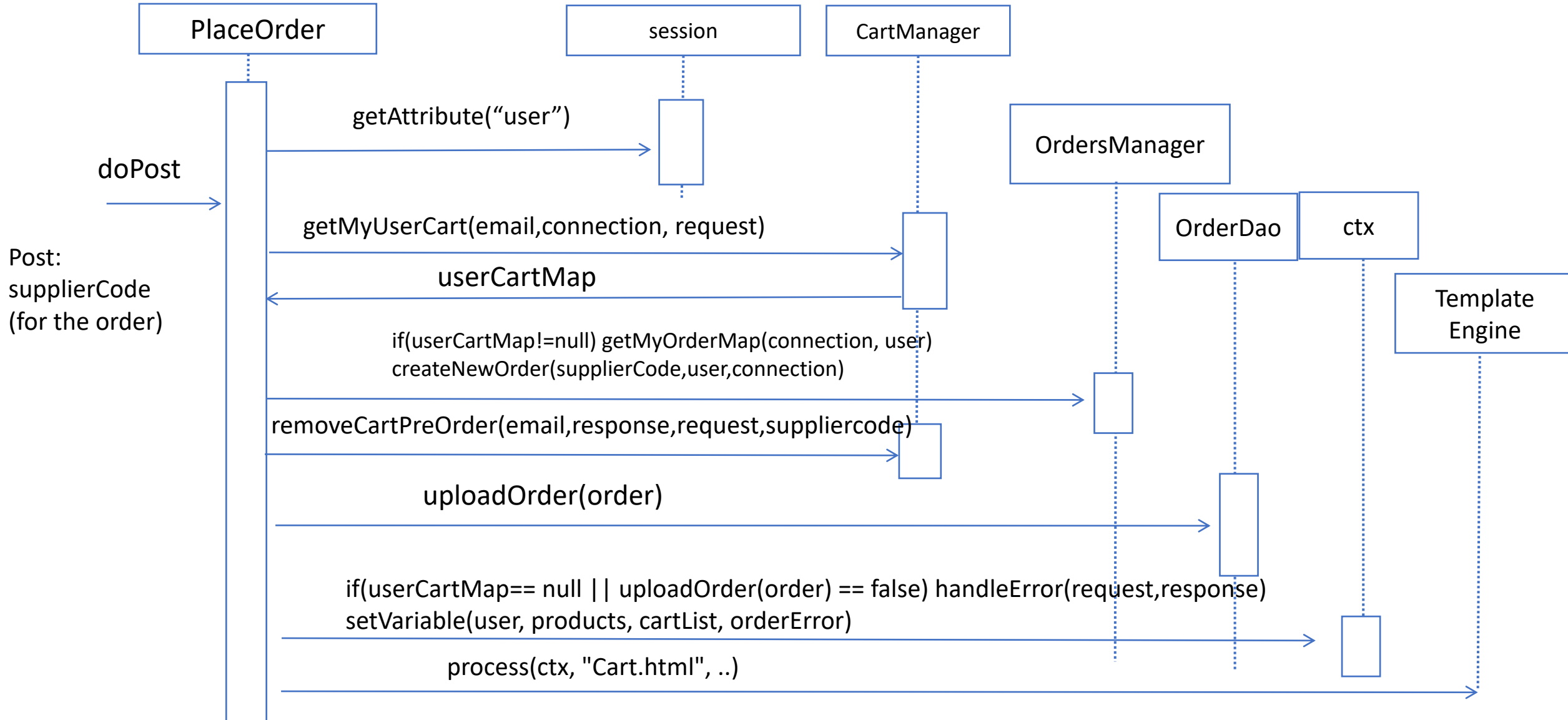
Event: Put In Cart



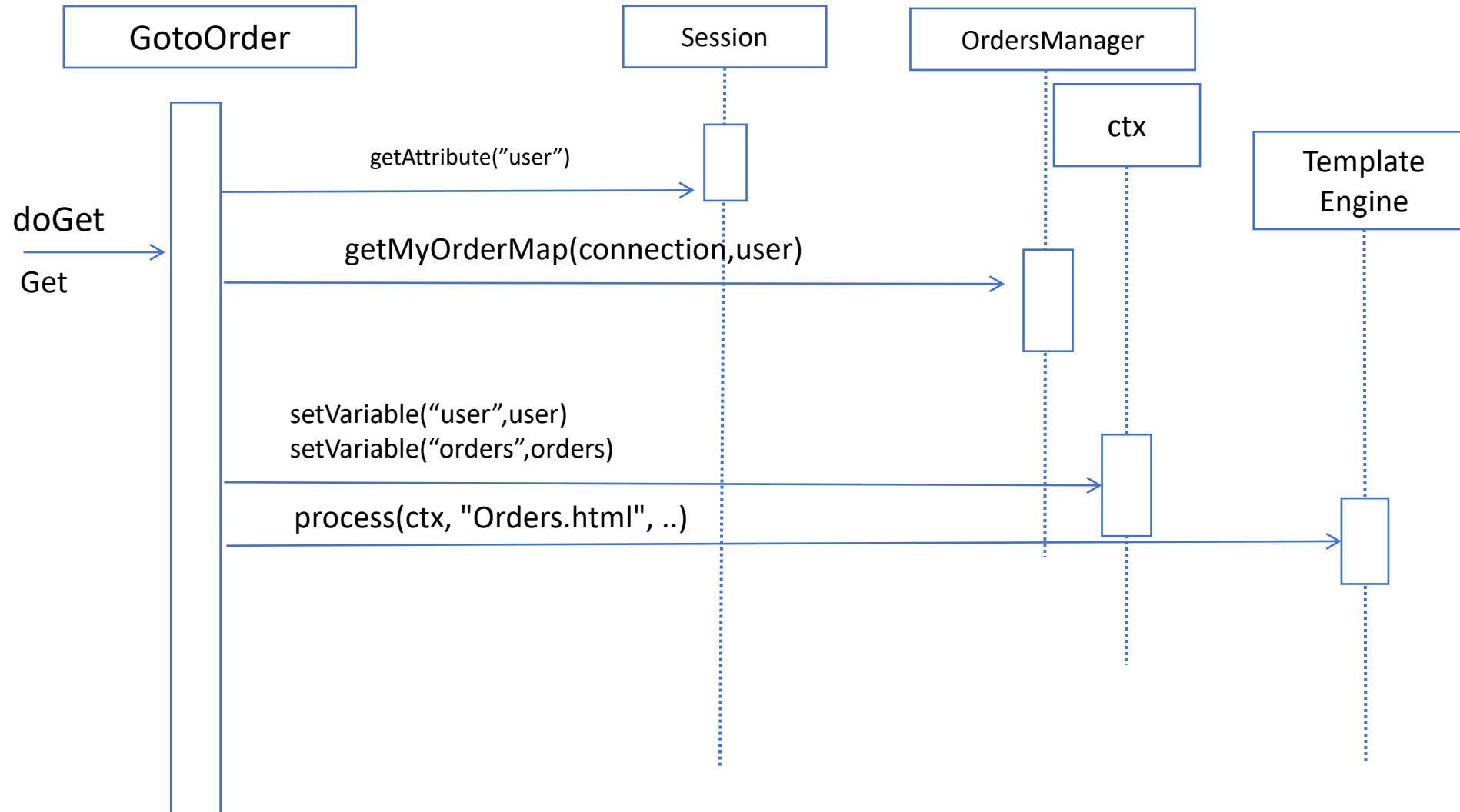
Event: Go To Cart Page



Event: Place Order



Event: Go To Order



Logout

