PROGETTO DI TECNOLOGIE INFORMATICHE PER IL WEB

Esercizio 1 – ASTE ONLINE

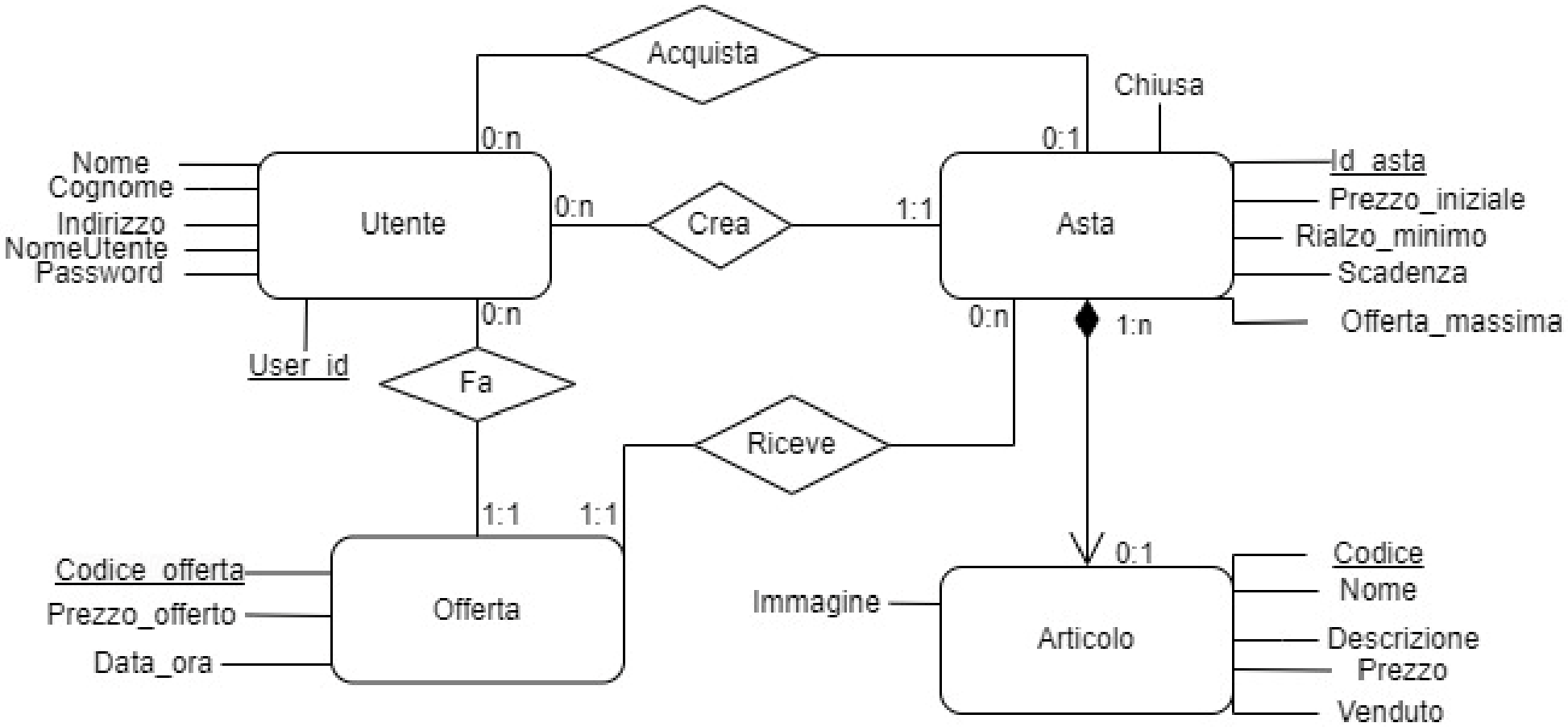
Martina Riva – Martina Quaregna

ANALISI DEI DATI

Entità, Relazioni, attributi

Un'applicazione web consente la gestione di aste online. Gli utenti accedono tramite login e possono vendere e acquistare all'asta. La HOME page contiene due link, uno per accedere alla pagina VENDO e uno per accedere alla pagina ACQUISTO. La pagina VENDO mostra una lista delle aste create dall'utente e non ancora chiuse, una lista delle aste da lui create e chiuse e due form, una per creare un nuovo articolo e una per creare una nuova asta per vendere gli articoli dell'utente. Il primo form inserisce nuovi articoli nel database e il secondo mostra l'elenco degli articoli disponibili nel database e dà la possibilità di selezionarne più di uno. Un articolo ha codice, nome, descrizione, immagine e prezzo. Un'asta comprende uno o più articoli messi in vendita, il prezzo iniziale dell'insieme di articoli, il rialzo minimo di ogni offerta (espresso come un numero intero di euro) e una scadenza (data e ora, es. 19-04-2021 alle 24:00). Il prezzo iniziale dell'asta è ottenuto come somma del prezzo degli articoli compresi nell'offerta. Lo stesso articolo non può essere incluso in aste diverse. Una volta venduto, un articolo non deve essere più disponibile per l'inserimento in ulteriori aste. La lista delle aste è ordinata per data+ora crescente. L'elenco riporta: codice e nome degli articoli compresi nell'asta, offerta massima, tempo mancante (numero di giorni e ore) tra il momento (data ora) del login e la data e ora di chiusura dell'asta. Cliccando su un'asta compare una pagina DETTAGLIO ASTA che riporta per un'asta aperta tutti i dati dell'asta e la lista delle offerte (nome utente, prezzo offerto, data e ora dell'offerta) ordinata per data+ora decrescente. Un bottone CHIUDI permette all'utente di chiudere l'asta se è giunta l'ora della scadenza (si ignori il caso di aste scadute ma non chiuse dall'utente e non ci si occupi della chiusura automatica di aste dopo la scadenza). Se l'asta è chiusa, la pagina riporta tutti i dati dell'asta, il nome dell'aggiudicatario, il prezzo finale e l'indirizzo (fisso) di spedizione dell'utente. La pagina ACQUISTO contiene una form di ricerca per parola chiave. Quando l'acquirente invia una parola chiave la pagina ACQUISTO è aggiornata e mostra un elenco di aste aperte (la cui scadenza è posteriore alla data e ora dell'invio) per cui la parola chiave compare nel nome o nella descrizione di almeno uno degli articoli dell'asta. La lista è ordinata in modo decrescente in base al tempo (numero di giorni e ore) mancante alla chiusura. Cliccando su un'asta aperta compare la pagina OFFERTA che mostra i dati degli articoli, l'elenco delle offerte pervenute in ordine di data+ora decrescente e un campo di input per inserire la propria offerta, che deve essere superiore all'offerta massima corrente di un importo pari almeno al rialzo minimo. Dopo l'invio dell'offerta la pagina OFFERTA mostra l'elenco delle offerte aggiornate. La pagina ACQUISTO contiene anche un elenco delle offerte aggiudicate all'utente con i dati degli articoli e il prezzo finale.

DATABASE


Acquista
Chiusa
0:n
0:1
Nome
Cognome
Indirizzo
NomeUtente
Password
Utente
0:n
Crea
1:1
Asta
Id_asta
Prezzo_iniziale
Rialzo_minimo
Scadenza
Offerta_massima
0:n
1:n
User_id
Fa
Riceve
1:1
1:1
0:n
Codice_offerta
Prezzo_offerto
Data_ora
Offerta
Immagine
Articolo
Codice
Nome
Descrizione
Prezzo
Venduto

DATABASE SCHEMA

```
CREATE TABLE `utente` (
  `Nome` varchar(45) NOT NULL,
  `Cognome` varchar(45) NOT NULL,
  `Nome_utente` varchar(45) NOT NULL,
  `Password` varchar(45) NOT NULL,
  `Indirizzo` varchar(100) NOT NULL,
  `User_id` int NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`User_id`),
  UNIQUE KEY `Nome_utente` (`Nome_utente`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```sql
CREATE TABLE `articolo` (
  `Codice` int NOT NULL AUTO_INCREMENT,
  `Nome` varchar(45) NOT NULL,
  `Descrizione` varchar(255) NOT NULL,
  `Immagine` longblob NOT NULL,
  `Prezzo` int NOT NULL,
  `Venduto` tinyint(1) NOT NULL DEFAULT '0',
  `User_id` int NOT NULL,
  PRIMARY KEY (`Codice`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```sql
CREATE TABLE `articoloasta` (
  `Id_asta` int NOT NULL,
  `Codice` int NOT NULL,
  PRIMARY KEY (`Id_asta`,`Codice`),
  KEY `Codice` (`Codice`),
  CONSTRAINT `articoloasta_ibfk_1` FOREIGN KEY (`Id_asta`) REFERENCES `asta` (`Id_asta`) ON UPDATE CASCADE,
  CONSTRAINT `articoloasta_ibfk_2` FOREIGN KEY (`Codice`) REFERENCES `articolo` (`Codice`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```sql
CREATE TABLE `asta` (
  `Id_asta` int NOT NULL AUTO_INCREMENT,
  `Prezzo_iniziale` int NOT NULL,
  `Rialzo_minimo` int NOT NULL,
  `Scadenza` datetime NOT NULL,
  `Offerta_massima` int NOT NULL,
  `Chiusa` tinyint(1) NOT NULL DEFAULT '0',
  `User_id` int NOT NULL,
  PRIMARY KEY (`Id_asta`),
  KEY `User_id` (`User_id`),
  CONSTRAINT `asta_ibfk_1` FOREIGN KEY (`User_id`) REFERENCES `utente` (`User_id`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `offerta` (
  `Id_asta` int NOT NULL,
  `User_id` int NOT NULL,
  `Prezzo_offerto` int NOT NULL,
  `Data_ora` datetime NOT NULL,
  `Codice_offerta` int NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`Codice_offerta`),
  KEY `Id_asta` (`Id_asta`),
  KEY `User_id` (`User_id`),
  CONSTRAINT `offerta_ibfk_1` FOREIGN KEY (`Id_asta`) REFERENCES `asta` (`Id_asta`) ON UPDATE CASCADE,
  CONSTRAINT `offerta_ibfk_2` FOREIGN KEY (`User_id`) REFERENCES `utente` (`User_id`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

ANALISI DEI REQUISITI

Pages, view components, events, actions

Un'applicazione web consente la gestione di aste online. Gli utenti accedono tramite login e possono vendere e acquistare all'asta. La HOME page contiene due link, uno per accedere alla pagina VENDO e uno per accedere alla pagina ACQUISTO. La pagina VENDO mostra una lista delle aste create dall'utente e non ancora chiuse, una lista delle aste da lui create e chiuse e due form, una per creare un nuovo articolo e una per creare una nuova asta per vendere gli articoli dell'utente. Il primo form inserisce nuovi articoli nel database e il secondo mostra l'elenco degli articoli disponibili nel database e dà la possibilità di selezionarne più di uno. Un articolo ha codice, nome, descrizione, immagine e prezzo. Un'asta comprende uno o più articoli messi in vendita, il prezzo iniziale dell'insieme di articoli, il rialzo minimo di ogni offerta (espresso come un numero intero di euro) e una scadenza (data e ora, es. 19-04-2021 alle 24:00). Il prezzo iniziale dell'asta è ottenuto come somma del prezzo degli articoli compresi nell'offerta. Lo stesso articolo non può essere incluso in aste diverse. Una volta venduto, un articolo non deve essere più disponibile per l'inserimento in ulteriori aste. La lista delle aste è ordinata per data+ora crescente. L'elenco riporta: codice e nome degli articoli compresi nell'asta, offerta massima, tempo mancante (numero di giorni e ore) tra il momento (data ora) del login e la data e ora di chiusura dell'asta. Cliccando su un'asta compare una pagina DETTAGLIO ASTA che riporta per un'asta aperta tutti i dati dell'asta e la lista delle offerte (nome utente, prezzo offerto, data e ora dell'offerta) ordinata per data+ora decrescente. Un bottone CHIUDI permette all'utente di chiudere l'asta se è giunta l'ora della scadenza (si ignori il caso di aste scadute ma non chiuse dall'utente e non ci si occupi della chiusura automatica di aste dopo la scadenza). Se l'asta è chiusa, la pagina riporta tutti i dati dell'asta, il nome dell'aggiudicatario, il prezzo finale e l'indirizzo (fisso) di spedizione dell'utente. La pagina ACQUISTO contiene una form di ricerca per parola chiave. Quando l'acquirente invia una parola chiave la pagina ACQUISTO è aggiornata e mostra un elenco di aste aperte (la cui scadenza è posteriore alla data e ora dell'invio) per cui la parola chiave compare nel nome o nella descrizione di almeno uno degli articoli dell'asta. La lista è ordinata in modo decrescente in base al tempo (numero di giorni e ore) mancante alla chiusura. Cliccando su un'asta aperta compare la pagina OFFERTA che mostra i dati degli articoli, l'elenco delle offerte pervenute in ordine di data+ora decrescente e un campo di input per inserire la propria offerta, che deve essere superiore all'offerta massima corrente di un importo pari almeno al rialzo minimo. Dopo l'invio dell'offerta la pagina OFFERTA mostra l'elenco delle offerte aggiornate. La pagina ACQUISTO contiene anche un elenco delle offerte aggiudicate all'utente con i dati degli articoli e il prezzo finale.

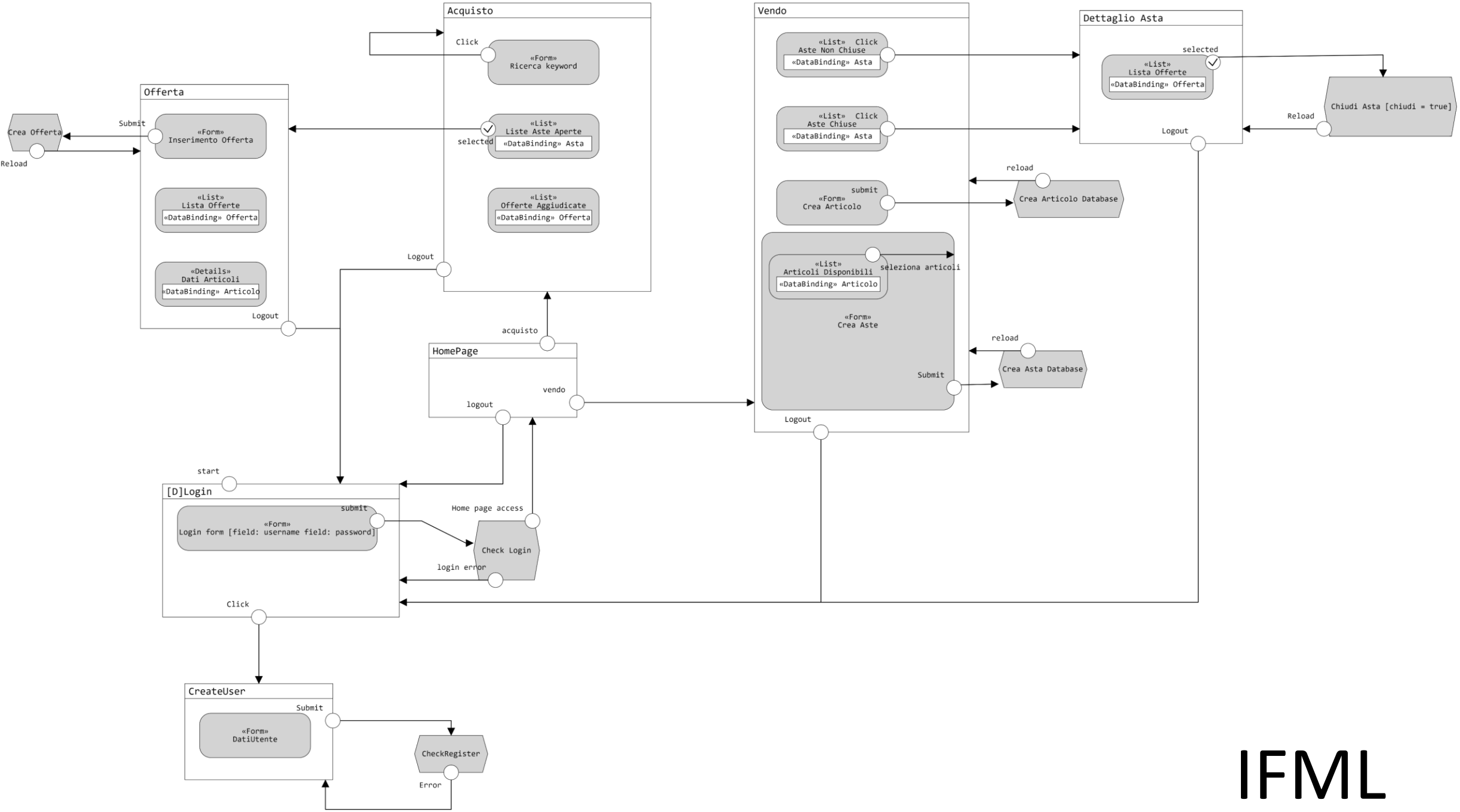
HTML pure Asta

COMPLETAMENTO SPECIFICHE

- Form di registrazione aggiunto nella pagina «Registrazione»
- Bottone di LOGOUT in tutte le pagine
- Dalla pagina acquisto è possibile accedere alla pagina vendo tramite un pulsante, e viceversa. Dalla pagina offerta si può tornare alla Home o Acquisto, e dalla pagina dettaglio asta si può tornare alla Home o alla pagina Vendo, tramite dei bottoni.
- Quando un utente in sessione commette un errore (come ad esempio inserire parametri non validi nei form) viene reindirizzato alla pagina di errore, che mostra un messaggio. Da qui può tornare alla pagina Home o fare il logout.
- Se le liste sono vuote viene mostrato un messaggio, che avverte dell'assenza di valori.
- Quando l'utente accede alla pagina acquisto vengono mostrate tutte le aste aperte.
- Si assume che in assenza di offerte per una data asta, l'utente possa inserire un'offerta pari al prezzo iniziale più il rialzo minimo. Per comodità viene mostrato il prezzo minimo che l'utente può offrire.
- Un utente può fare più offerte (anche rialzare la propria offerta)
- Un utente non può visualizzare la pagina di dettaglio di un'asta di un altro utente.
- Un utente non può visualizzare una pagina di offerta di un'asta propria.

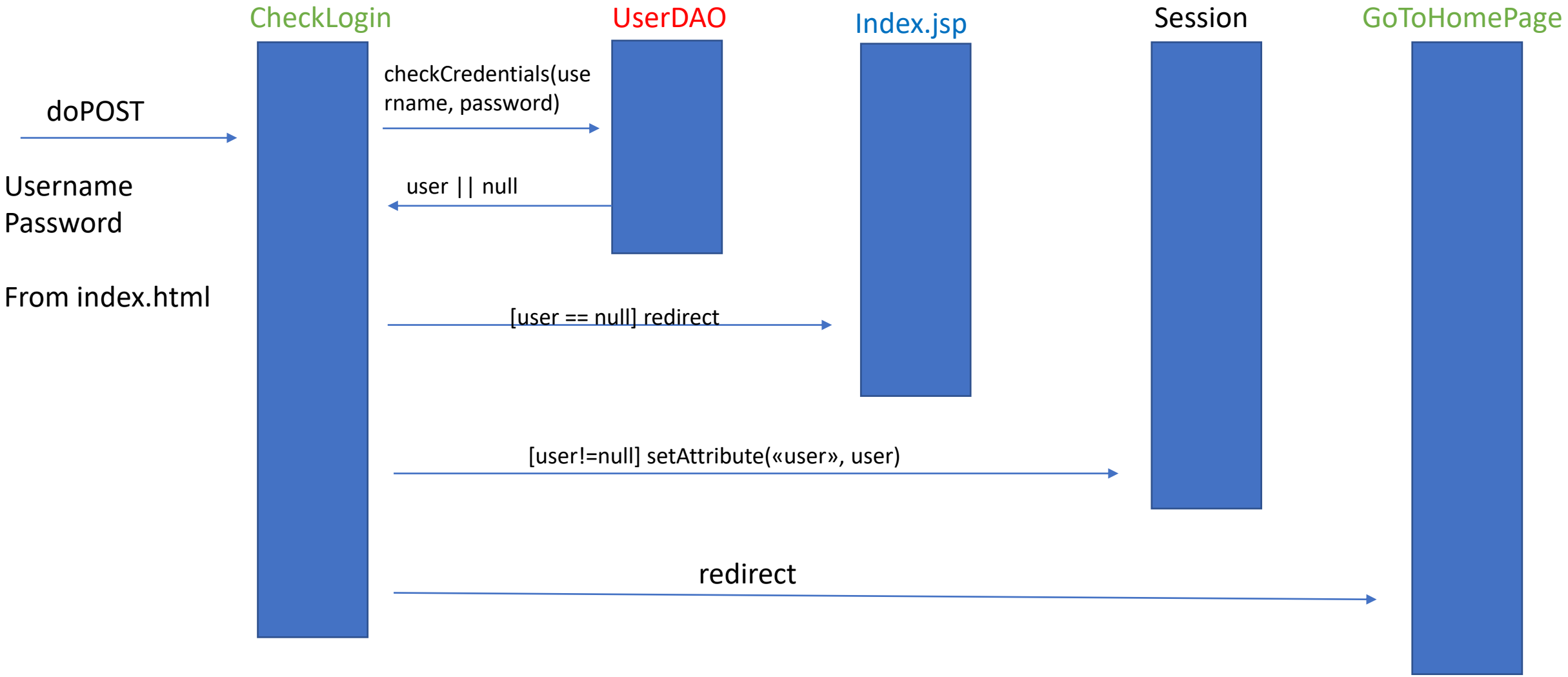
COMPONENTI

- **DAO**:
  - UserDAO
    - checkCredentials(String username, String password)
    - checkRegister(String Nome, String Cognome, String username, String Password, String Indirizzo)
    - createUser(String Nome, String Cognome, String Nome_utente, String Password, String Indirizzo)
    - datiUtente(int userId)
  - AstaDAO
    - findAsteAperteNotUser(Utente utente)
    - findAsteAperteByKeyword(String keyword, int utenteId)
    - findAsteChiuse(int idUser)
    - findAsteNonChiuse(int idUser)
    - createAsta(int rialzoMinimo, Timestamp scadenza, int idUser, ArrayList<Articolo> articoliSelezionati)
    - maxAstaId()
    - checkAstaScaduta(int idAsta) -> protected
    - chiudiAstaSeScaduta(int idAsta)
    - prezzoMinimo(int idAsta, boolean primaOfferta)
    - datiAsta(int idAsta)
    - updateOffertaMAX(int offertaMax, int idAsta)
  - ArticoloDAO
    - findArticoliAsta(int idAsta)
    - findArticoliDisponibili(int idUser)
    - selectArticolo(int codice, int idUser)
    - creaArticolo(String nome, String descrizione, int prezzo, boolean venduto, int UserId, inputStream input)
    - aggiungiArticoloAdAsta(int codice, int idAsta)
  - OffertaDAO
    - findOfferteAggiudicateUser(int utente)
    - findOfferteAsta(int idAsta)
    - creaOfferta(int idAsta, int idUser, int prezzoOfferta, Timestamp dataEOra)
    - checkOfferta(int idAsta, int prezzoOfferta)
    - maxOffertaAsta(int idAsta)
- **BEANS**:
  - Articolo
  - Articoloasta
  - Asta
  - Offerta
  - Utente
- **VIEWS**
  - Acquisto.jsp
  - DettaglioAsta.jsp
  - ErrorPage.jsp
  - Homepage.html
  - Offerta.jsp
  - Registrazione.html
  - Vendo.jsp
  - Index.jsp
- **SERVLET (CONTROLLERS)**
  - CheckLogin
  - ChiudiAsta
  - CreaArticolo
  - CreaAsta
  - CreateOfferta
  - CreateUser
  - GoToAcquistoPage
  - GoToDettaglioAsta
  - GoToHomePage
  - GoToOffertaPage
  - GoToVendoPage
  - Logout

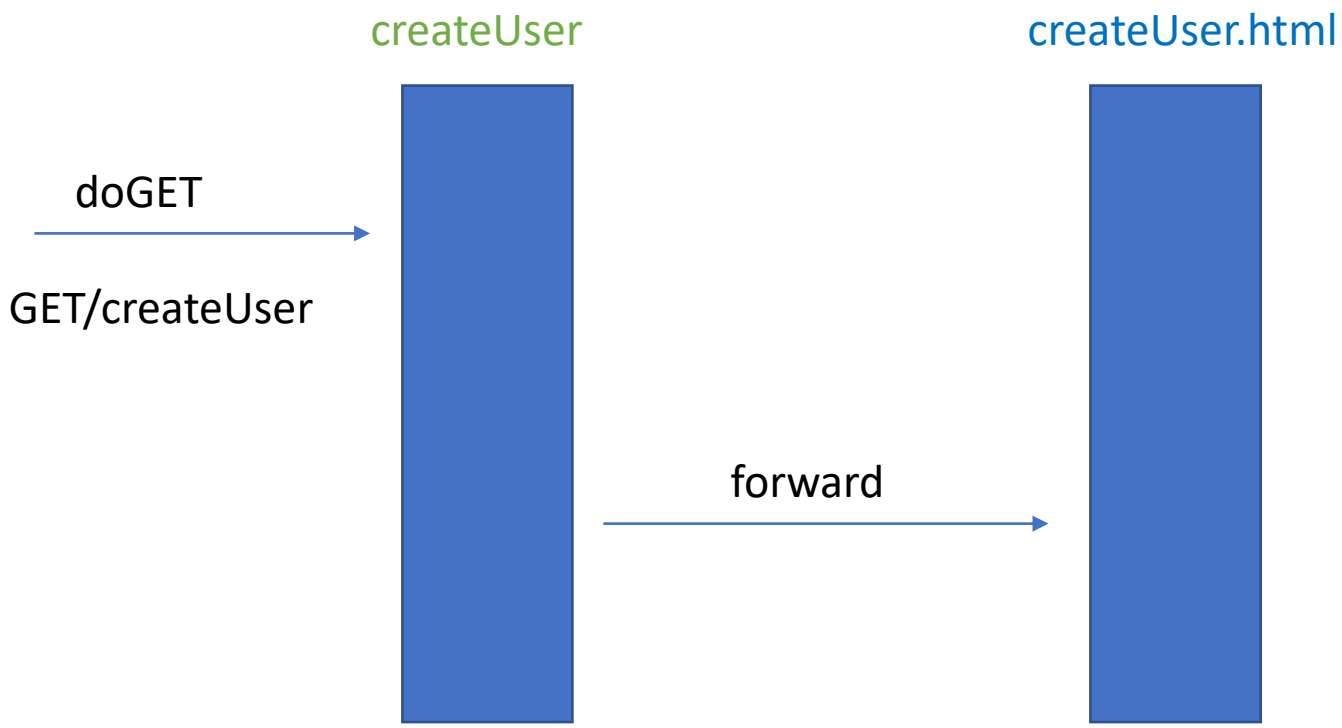

Acquisto
Click
«Form»
Ricerca keyword
«List»
Liste Aste Aperte
«DataBinding» Asta
selected
«List»
Offerte Aggiudicate
«DataBinding» Offerta
Logout
Offerta
Crea Offerta
Submit
«Form»
Inserimento Offerta
Reload
«List»
Lista Offerte
«DataBinding» Offerta
«Details»
Dati Articoli
«DataBinding» Articolo
Logout
acquisto
HomePage
vendo
logout
Vendo
«List» Click
Aste Non Chiuse
«DataBinding» Asta
«List» Click
Aste Chiuse
«DataBinding» Asta
submit
«Form»
Crea Articolo
reload
Crea Articolo Database
«List»
Articoli Disponibili
«DataBinding» Articolo
seleziona articoli
«Form»
Crea Aste
Submit
reload
Crea Asta Database
Logout
Dettaglio Asta
selected
«List»
Lista Offerte
«DataBinding» Offerta
Reload
Chiudi Asta [chiudi = true]
Logout
start
[D]Login
submit
«Form»
Login form [field: username field: password]
Home page access
Check Login
login error
Click
CreateUser
Submit
«Form»
DatiUtente
CheckRegister
Error

IFML

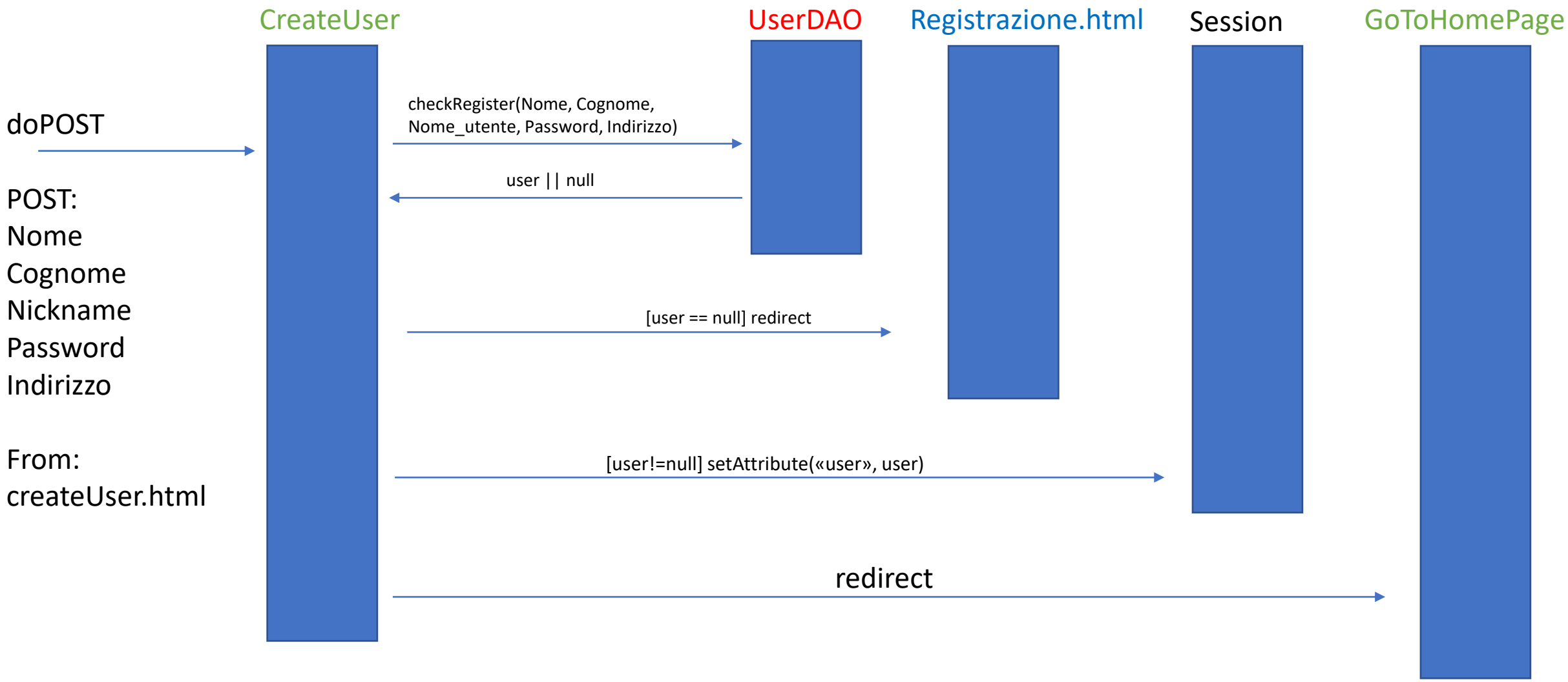
Event: LOGIN


CheckLogin
UserDAO
Index.jsp
Session
GoToHomePage
doPOST
Username
Password
From index.html
checkCredentials(username, password)
user || null
[user == null] redirect
[user!=null] setAttribute(«user», user)
redirect

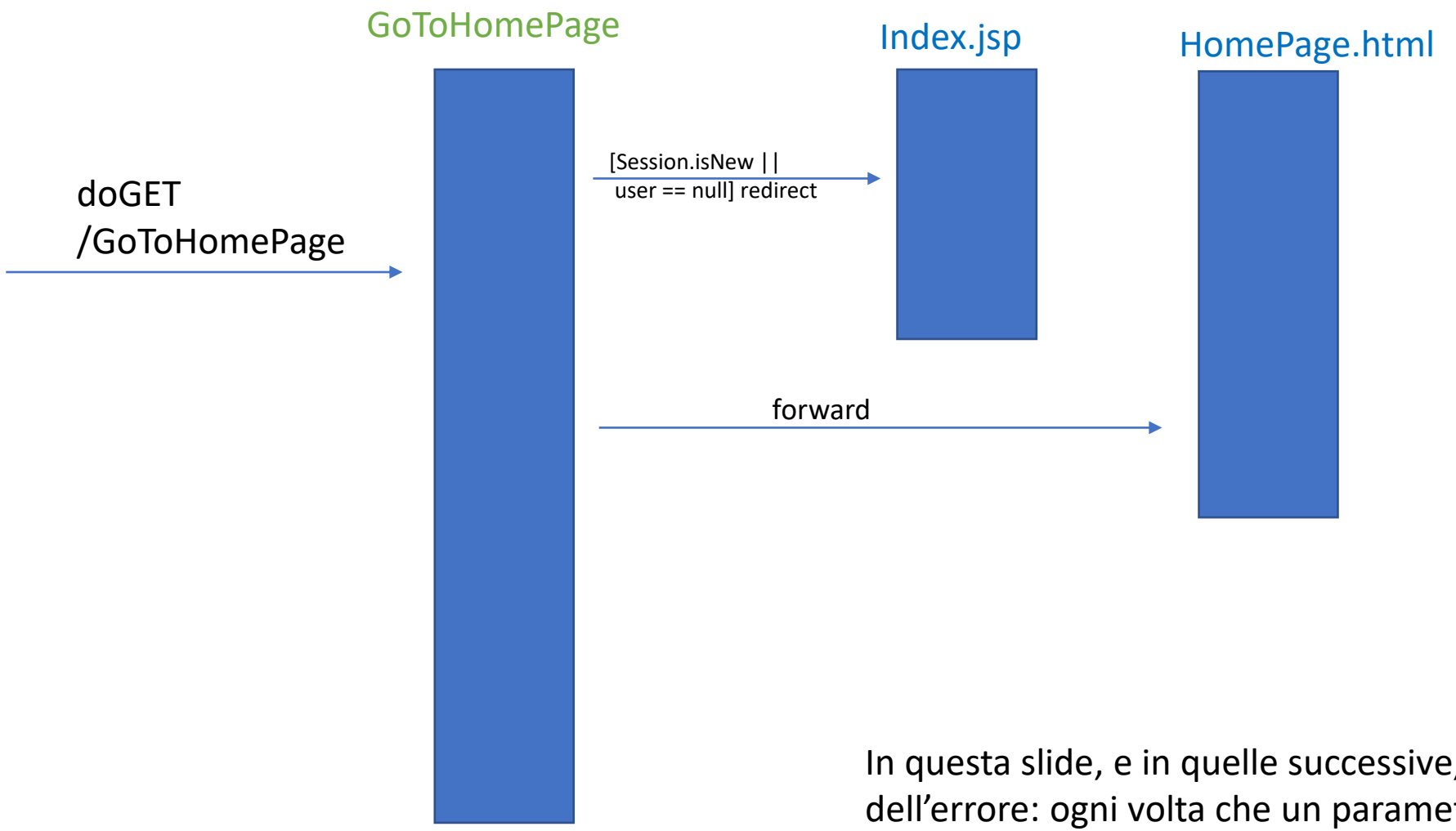
Event: accesso a createUser.html



Event: CreateUser

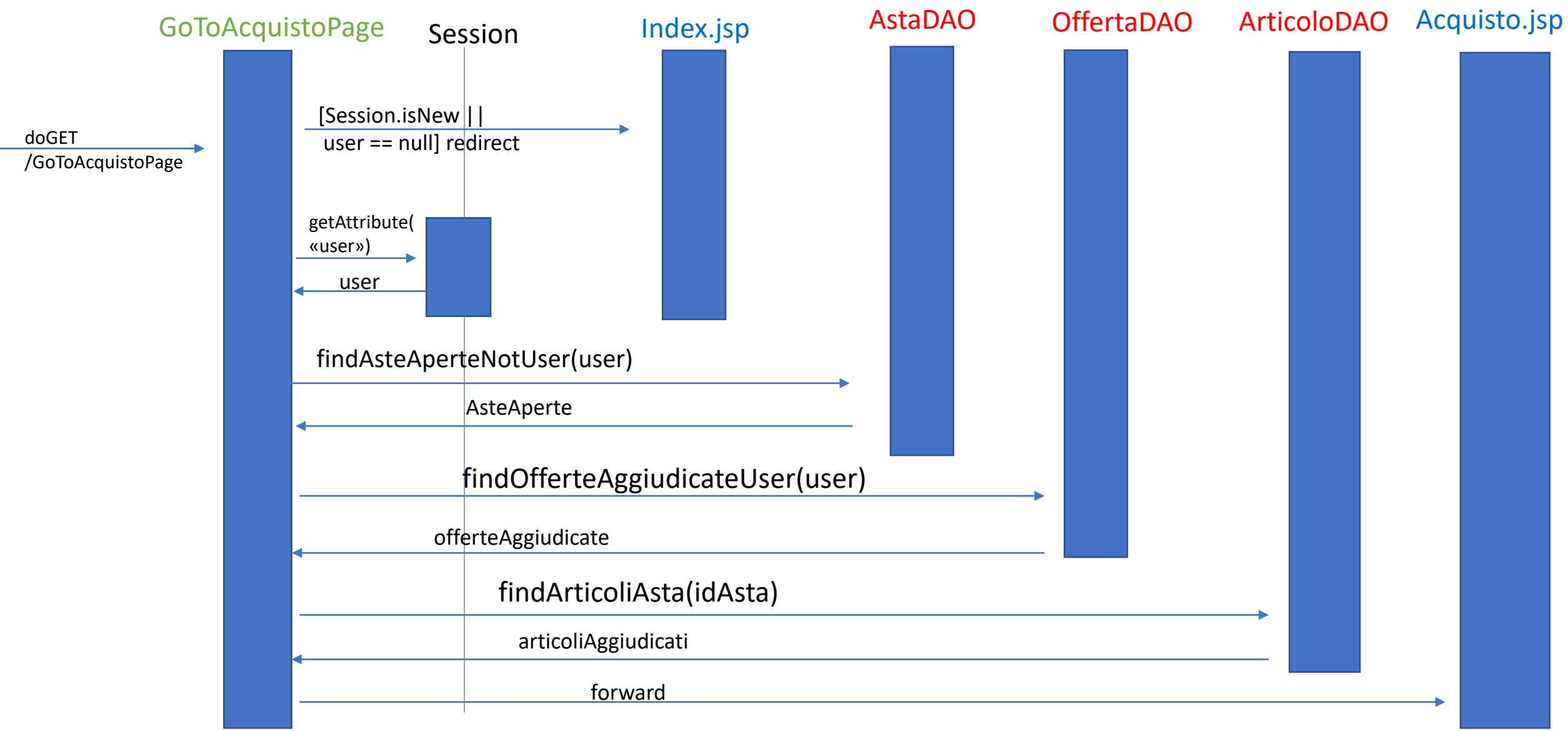


Event: Accesso pagina home

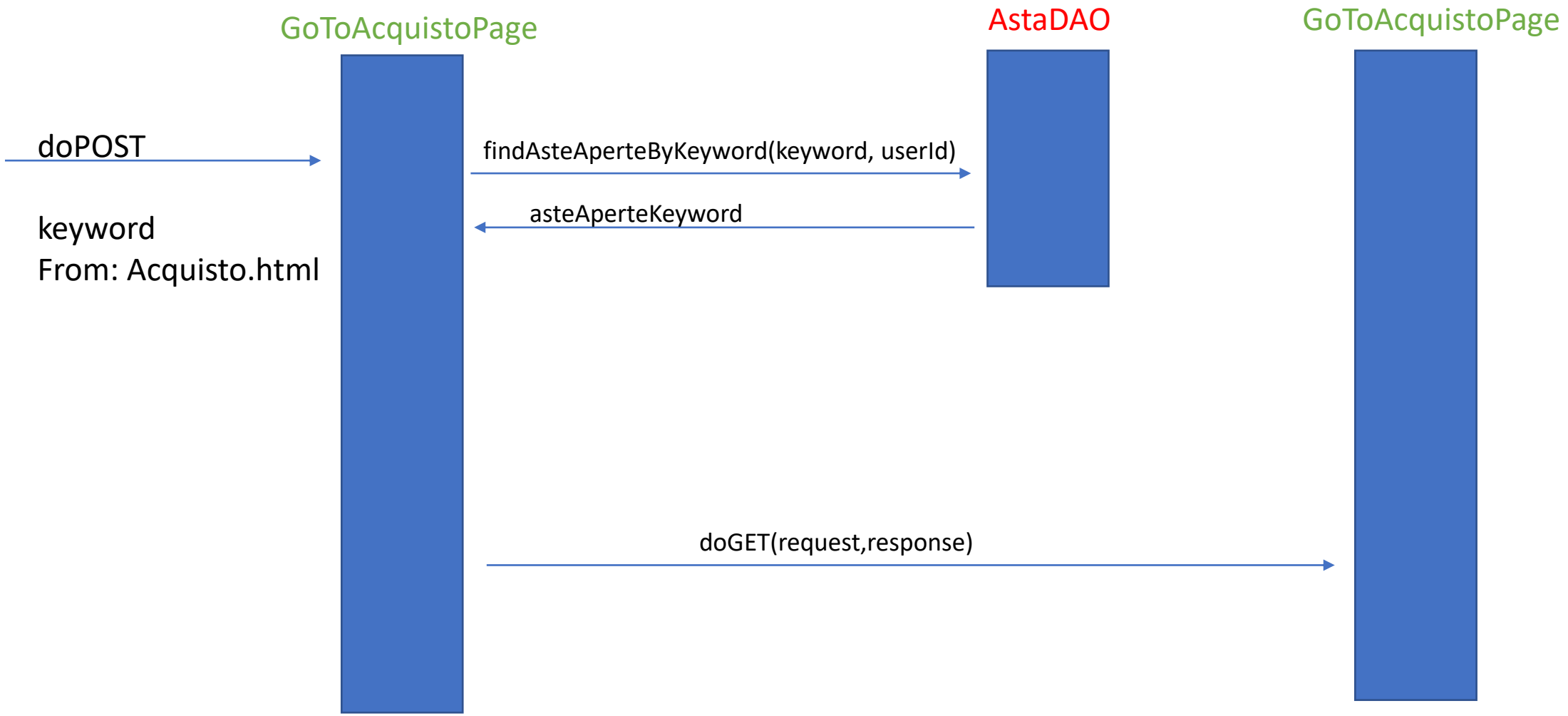


In questa slide, e in quelle successive, è omessa la gestione dell'errore: ogni volta che un parametro è errato o non rispetta determinate condizioni, o vengono lanciate eccezioni, viene mostrata la ErrorPage col messaggio d'errore.

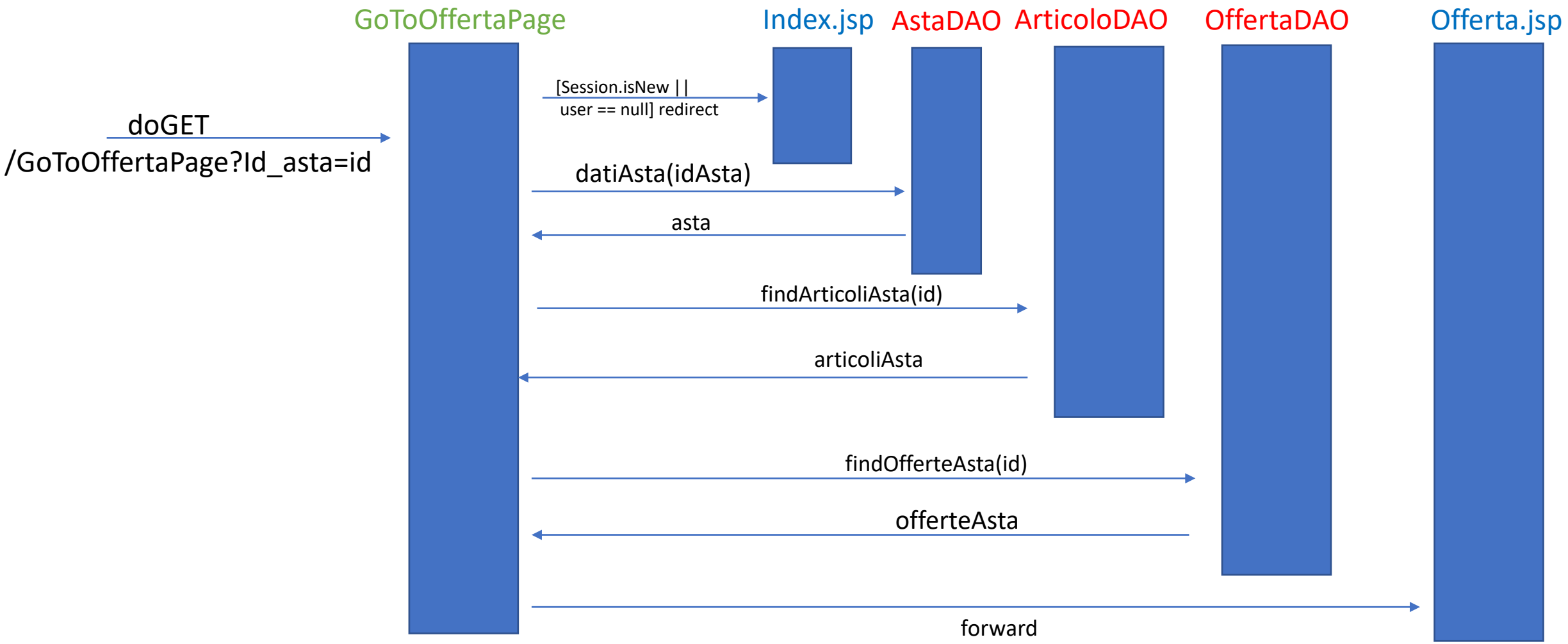
Event: Click link Acquisto page



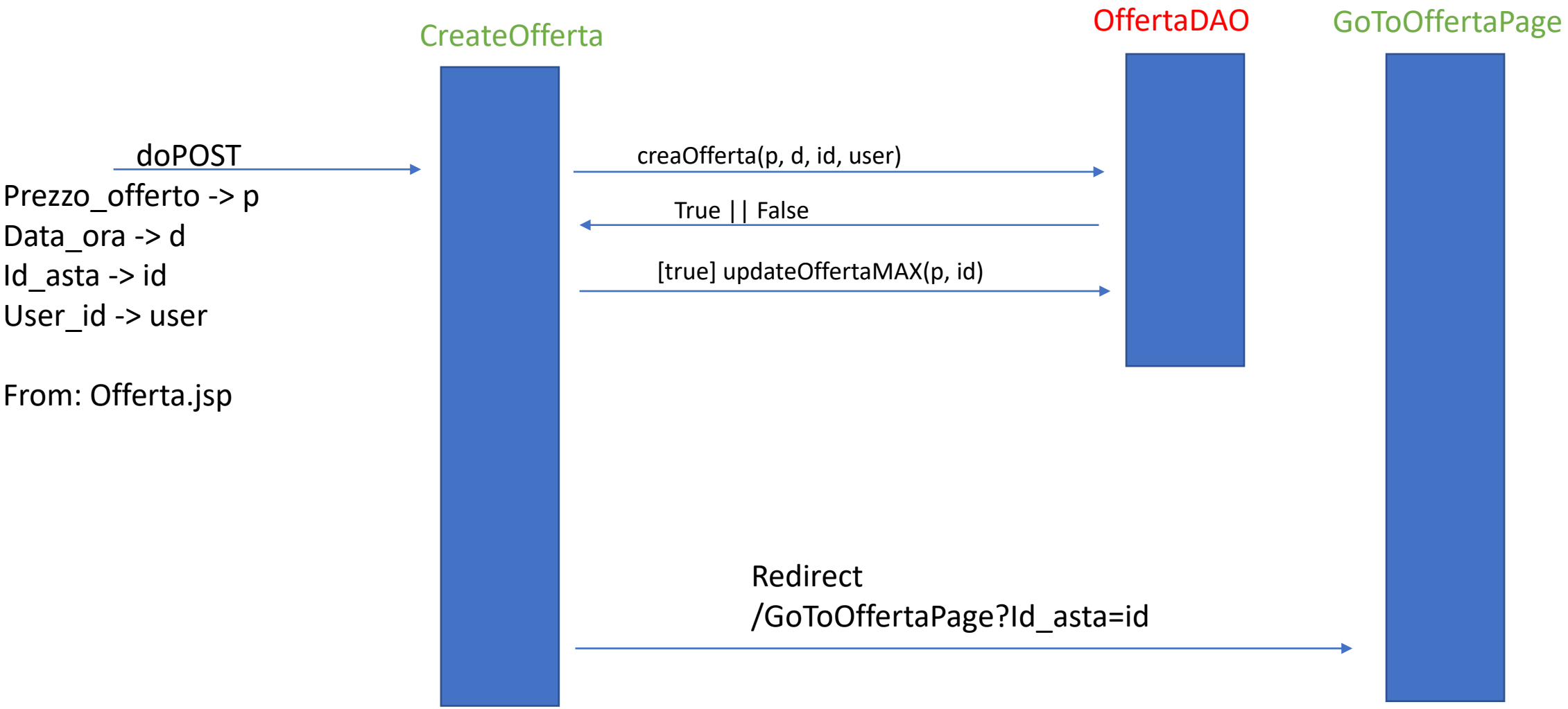
Event: ricerca per parola chiave



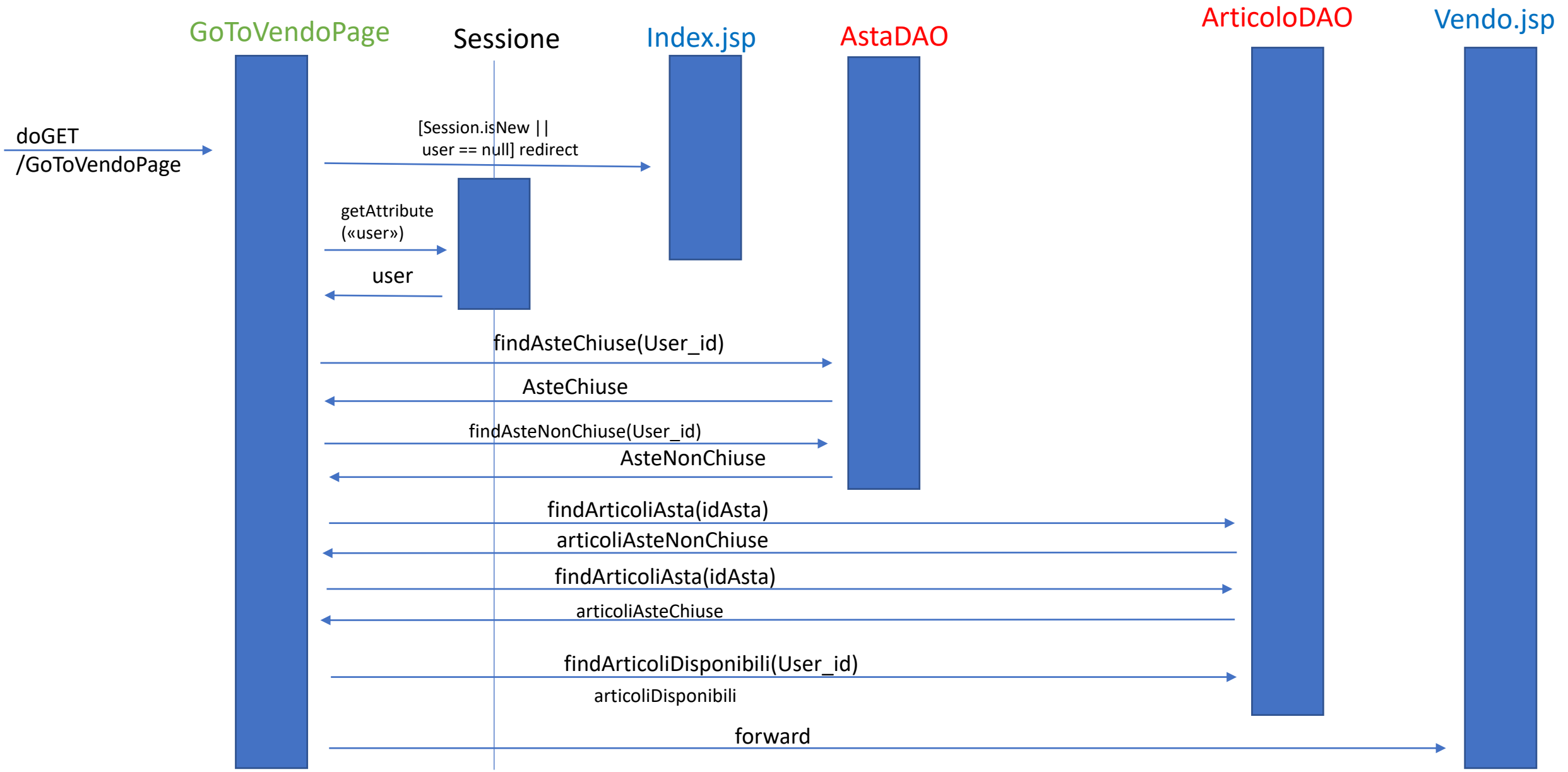
Event: click su asta aperta



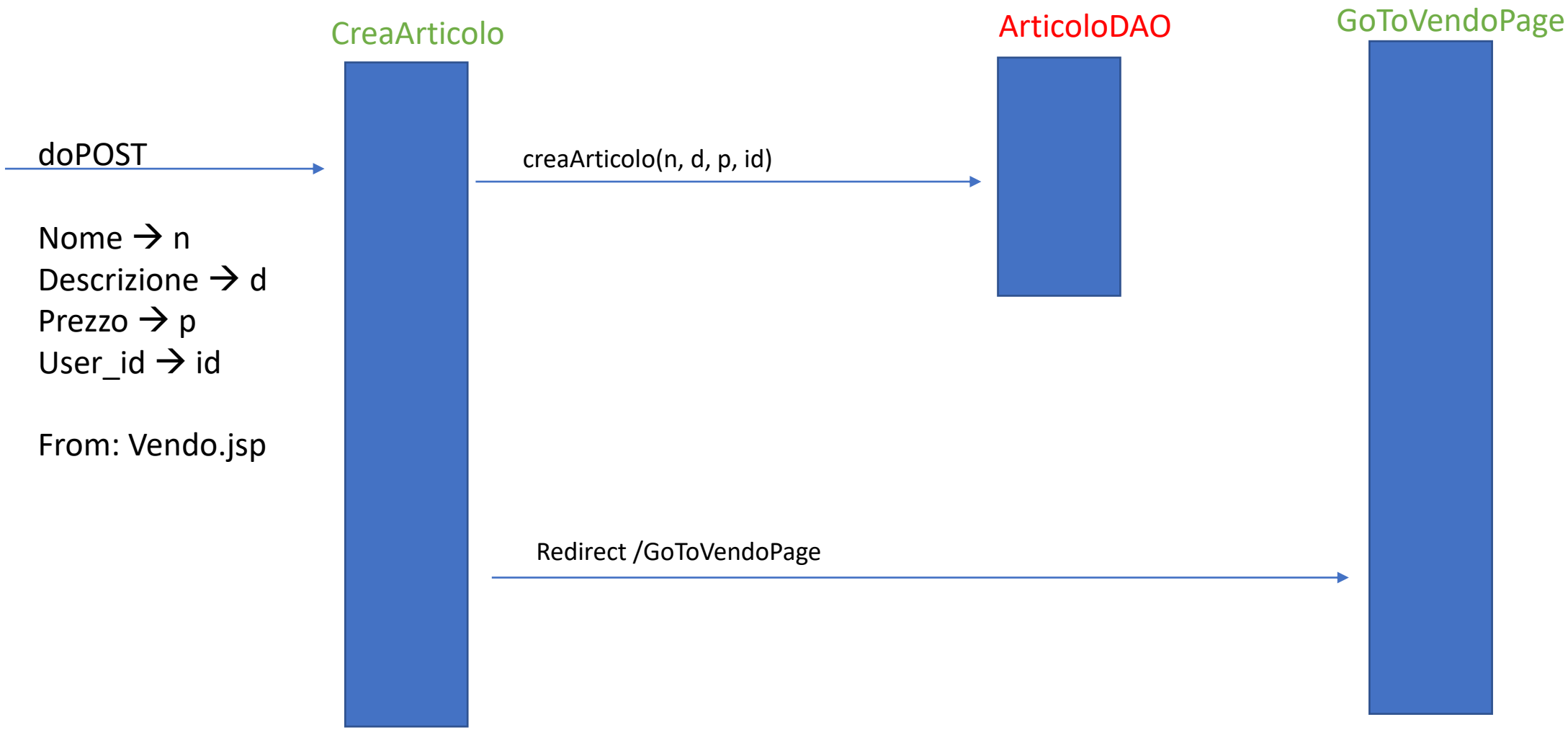
Event: inserisci offerta



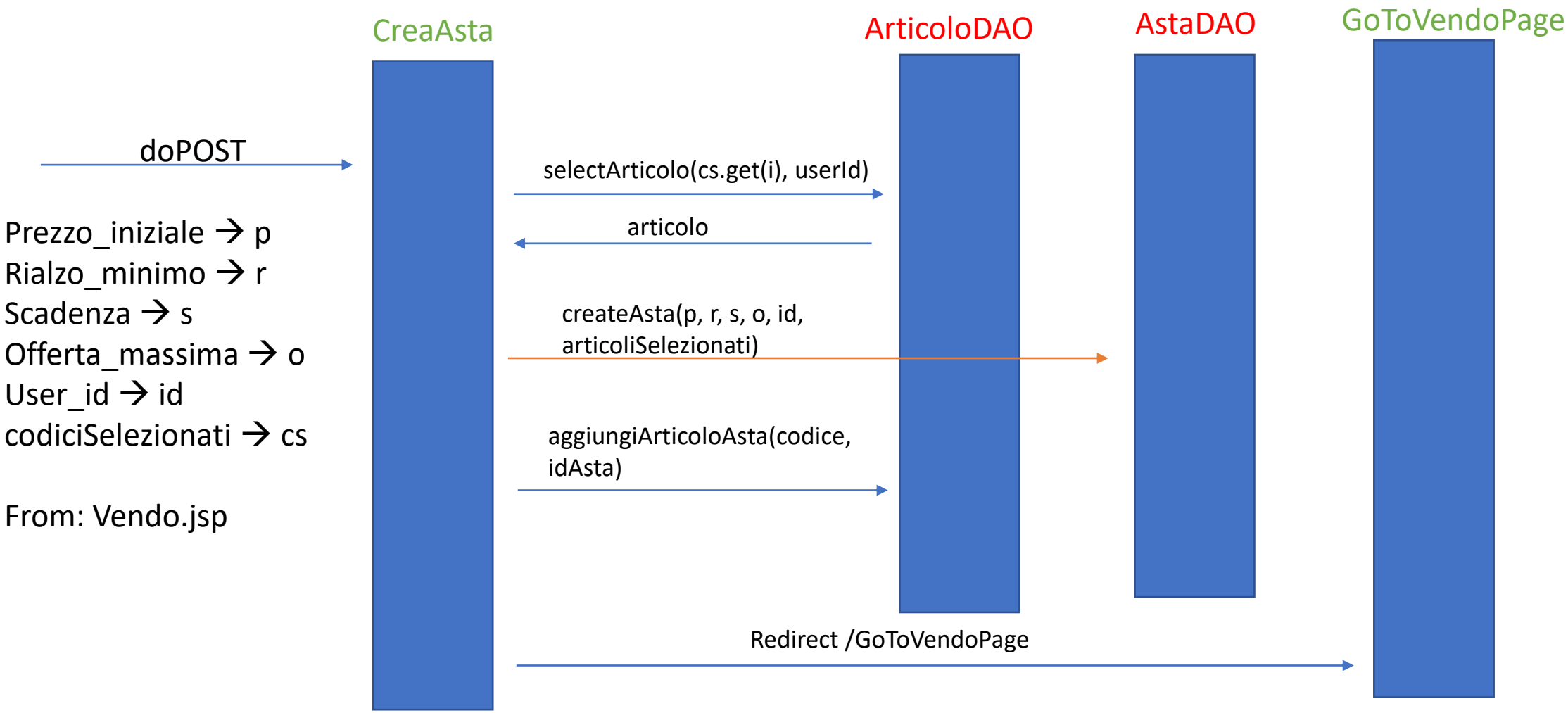
Event: Accesso VENDO page



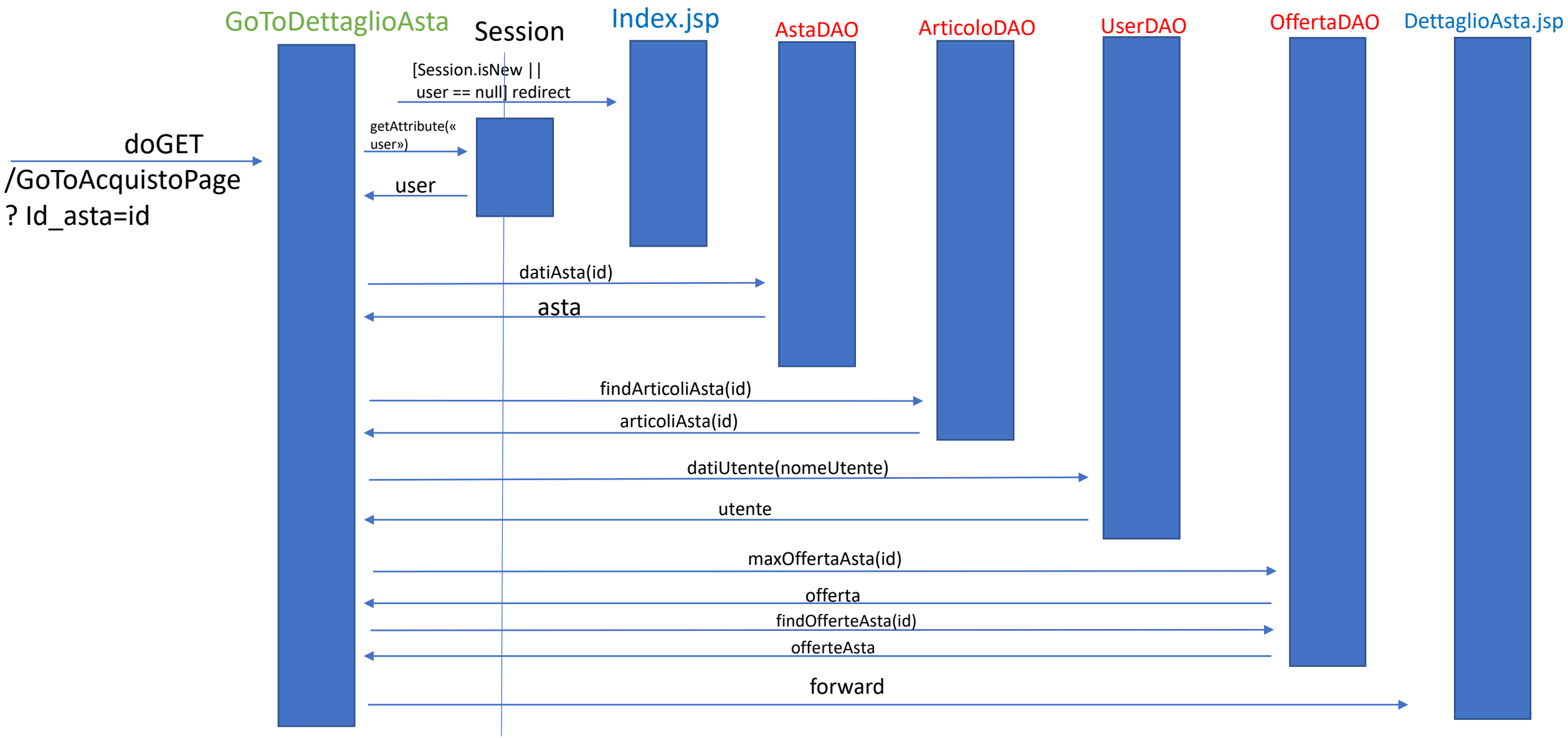
Event: Crea articolo



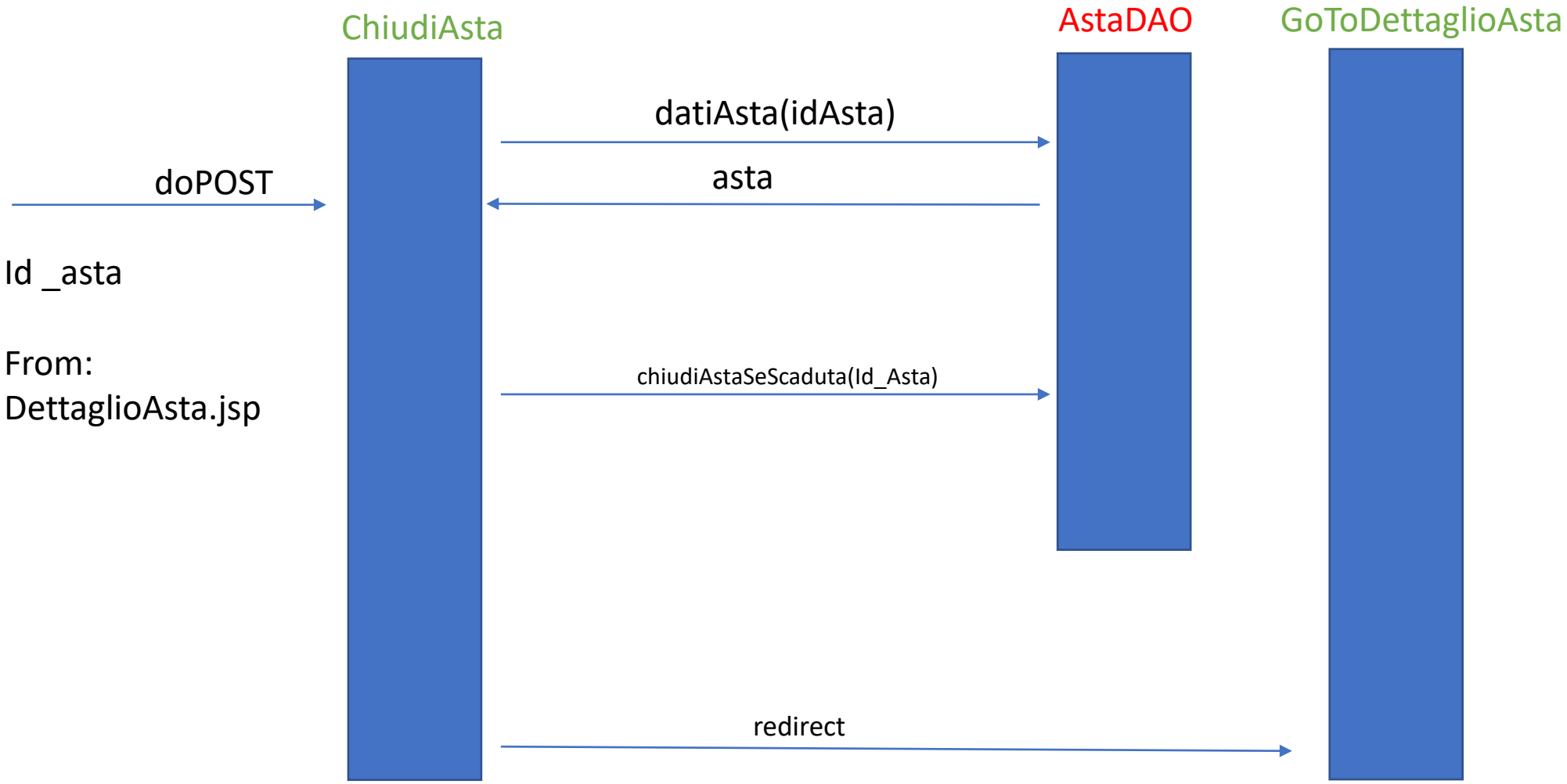
Event: Crea asta



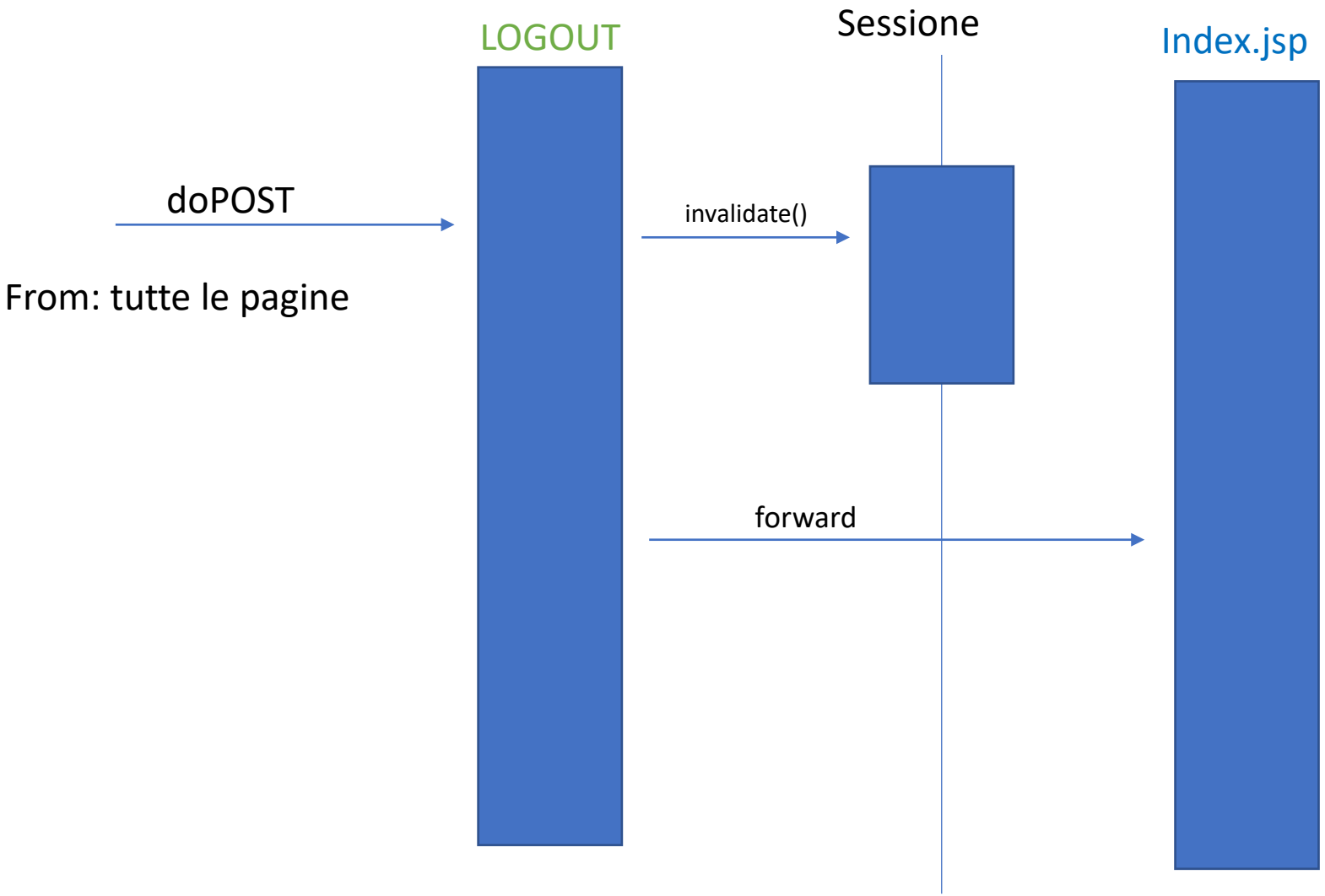
Event: Accesso Dettaglio Asta



Event: Chiudi Asta



Evento: LOGOUT



JavaScript RIA Asta

Specifiche:

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

- Dopo il login, l'intera applicazione è realizzata con un'unica pagina.
- Se l'utente accede per la prima volta l'applicazione mostra il contenuto della pagina ACQUISTO. Se l'utente ha già usato l'applicazione, questa mostra il contenuto della pagina VENDO se l'ultima azione dell'utente è stata la creazione di un'asta; altrimenti mostra il contenuto della pagina ACQUISTO con l'elenco (eventualmente vuoto) delle aste su cui l'utente ha cliccato in precedenza e che sono ancora aperte. L'informazione dell'ultima azione compiuta e delle aste visitate è memorizzata a lato client per la durata di un mese.
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica solo del contenuto da aggiornare a seguito dell'evento.

COMPLETAMENTO DELLE SPECIFICHE

- In caso di errore (parametri mancanti, richiesta errata,…) vengono mostrate dei messaggi sulla pagina.
- Una volta fatto il login, l'utente può vedere la pagina vendo/acquisto a seconda dell'ultima azione compiuta (come il click su un'asta), e da qui può, con un pulsante, fare logout o accedere ad acquisto/vendo.
- Le componenti corrispondenti alla pagina dettaglio asta/offerta vengono mostrate solo in caso di click su un'asta.
- Non si accettano prezzi negativi o date antecedenti a quella attuale quando si crea un articolo, un'asta o un'offerta.
- Messaggio di "Benvenuto nomeUtente" quando l'utente accede.
- Nella sezione "offerta" viene mostrata la minima offerta che l'utente può fare.

COMPONENTI

- **DAO**:
  - UserDAO
    - checkCredentials(String username, String password)
    - checkRegister(String Nome, String Cognome, String username, String Password, String Indirizzo)
    - datiUtente(int userId)
  - AstaDAO
    - findAsteAperteNotUser(Utente utente)
    - findAsteAperteByKeyword(String keyword, int utenteId)
    - findAsteChiuse(int idUser)
    - findAsteNonChiuse(int idUser)
    - createAsta(int rialzoMinimo, Timestamp scadenza, int idUser, ArrayList<Articolo> articoliSelezionati)
    - maxAstaId()
    - checkAstaScaduta(int idAsta) -> protected
    - chiudiAstaSeScaduta(int idAsta)
    - prezzoMinimo(int idAsta, boolean primaOfferta)
    - datiAsta(int idAsta)
    - updateOffertaMAX(int offertaMax, int idAsta)
  - ArticoloDAO
    - findArticoliAsta(int idAsta)
    - findArticoliDisponibili(int idUser)
    - selectArticolo(int codice, int idUser)
    - creaArticolo(String nome, String descrizione, int prezzo, boolean venduto, int UserId, inputStream input)
    - aggiungiArticoloAdAsta(int codice, int idAsta)
  - OffertaDAO
    - findOfferteAggiudicateUser(int utente)
    - findOfferteAsta(int idAsta)
    - creaOfferta(int idAsta, int idUser, int prezzoOfferta, Timestamp dataEOra)
    - checkOfferta(int idAsta, int prezzoOfferta)
    - maxOffertaAsta(int idAsta)
- **BEANS**:
  - Articolo
  - Articoloasta
  - Asta
  - Offerta
  - Utente
- **VIEWS**
  - index.html
  - Homepage.html
- **SERVLET (CONTROLLERS)**
  - CheckLogin
  - ChiudiAsta
  - CreaArticolo
  - CreaAsta
  - CreateOfferta
  - GetArticoliAsta
  - GetAstaAperta
  - GetAstaChiusa
  - GetDettaglioAsta
  - GetOfferteAsta
  - GoToAcquistoPage
  - GoToHomePage
  - GoToVendoPage
  - Logout
  - RicercaKeyword

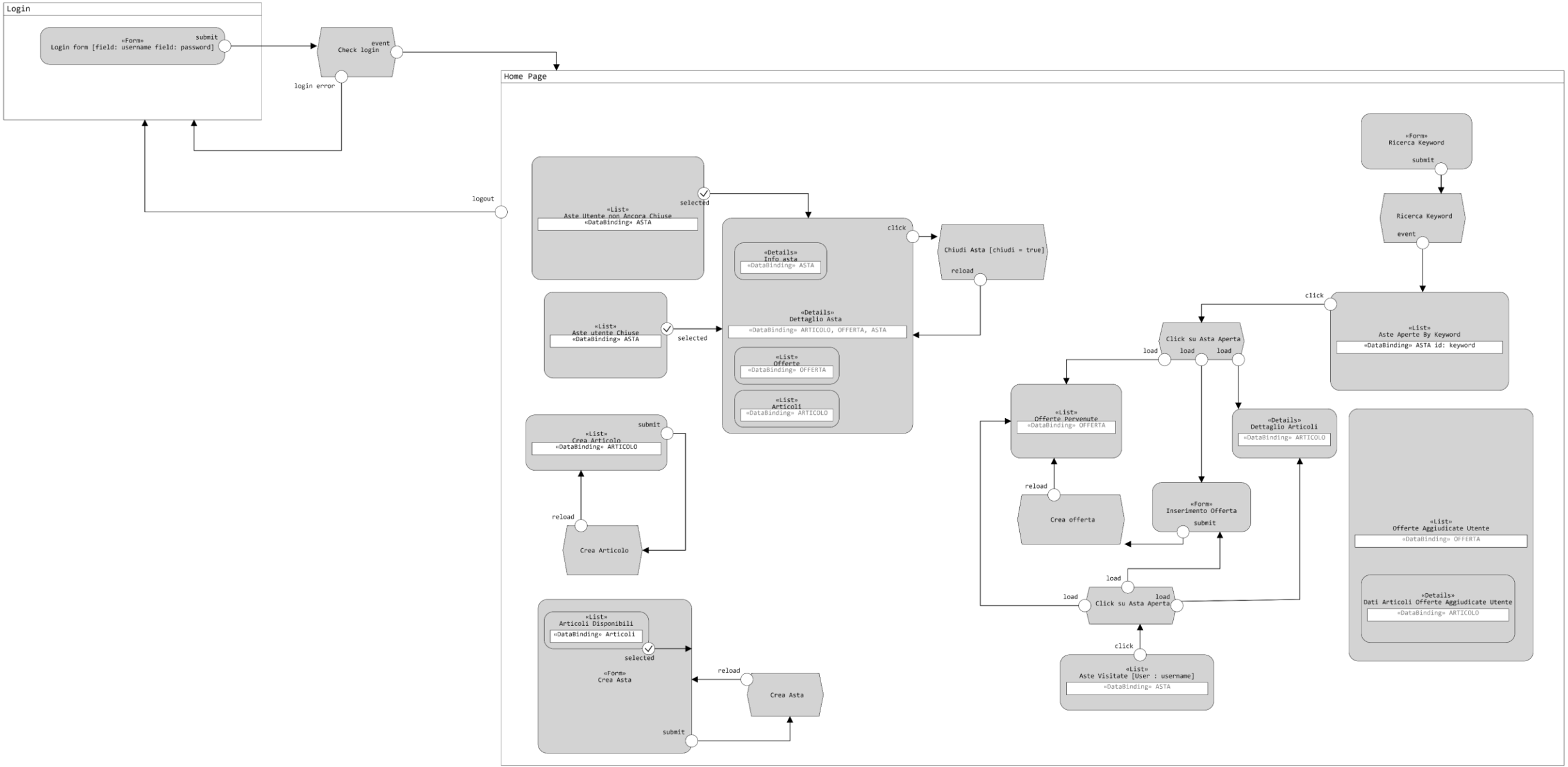
Client side: view component

- checkLogin.js
- utils.js
- aste.js:
  - PageOrchestrator
    - Start()
    - Refresh()
  - PersonalMessage
    - Show()
  - PaginaAcquisto
    - Show()
    - Update(offerteList, articoliList) → aggiorna la lista di offerte aggiudicate
    - ShowAsteVisitate(asteAperte) → mostra la lista di aste aperte che l'utente ha visitato
    - registerEvents(orchestrator) → bottone vendo
  - RicercaKeyword
    - registerEvents() → form di ricerca parola chiave
    - Update(asteAperteKeyword) → mostra le aste aperte contenti la keyword
  - PaginaOfferta
    - Show(idAsta)
    - ShowArticoli(idAsta) → mostra gli articoli dell'asta
    - ShowOfferte(idAsta) → mostra le offerte dell'asta
    - Reset() → nasconde la pagina offerta
    - registerEvents() → form per creare l'offerta

- PaginaVendo
  - registerEvents(orchestrator) → bottone per passare ad acquisto
  - show()
  - showAsteChiuse(asteChiuse, articoliAsteChiuse) → mostra la tabella delle aste chiuse
  - showAsteNonChiuse(asteNonChiuse, tempoMancanteNonChiuse, articoliAsteNonChiuse) → mostra aste non chiuse
  - updateArticoliDisponibili(articoliDisponibiliList) → aggiorna gli articoli disponibili nel crea aste quando invocato
- CreaArticolo
  - registerEvents(pagVendo) → form per la creazione dell'articolo
  - uploadFile() → caricamento dell'immagine
- CreaAsta
  - registerEvents(pagVendo) → form di creazione dell'asta
- DettaglioAsta
  - registerEvents() → bottone chiudi asta
  - Reset() → nasconde la pagina di dettaglio asta
  - Show(idAsta, chiusa) → mostra articoli dell'asta e invoca showAperta o showChiusa a seconda dello stato dell'asta
  - showAperta(idAsta) → mostra dati dell'asta e tabella offerte
  - showChiusa(idAsta) → mostra dati asta e aggiudicatario se presente.

- CreateStorage(nome)
- Checkstorage(nome)
- UpdateStorageList(nome, id)
- setStorageEvent(nome, evento)
- showAcquisto(orchestrator)
- showVendo(orchestrator)

Le funzioni in questa slide servono per le specifiche di javascript: memorizzare l’ultima azione e le aste visitate (viene utilizzato localStorage).


Login
«Form»
Login form [field: username field: password]
submit
event
Check login
login error
logout
Home Page
«List»
Aste Utente non Ancora Chiuse
«DataBinding» ASTA
selected
«Details»
Info asta
«DataBinding» ASTA
click
Chiudi Asta [chiudi = true]
reload
«List»
Aste utente Chiuse
«DataBinding» ASTA
selected
«Details»
Dettaglio Asta
«DataBinding» ARTICOLO, OFFERTA, ASTA
«List»
Offerte
«DataBinding» OFFERTA
«List»
Articoli
«DataBinding» ARTICOLO
«Form»
Ricerca Keyword
submit
Ricerca Keyword
event
click
«List»
Aste Aperte By Keyword
«DataBinding» ASTA id: keyword
Click su Asta Aperta
load
load
load
«List»
Offerte Pervenute
«DataBinding» OFFERTA
«Details»
Dettaglio Articoli
«DataBinding» ARTICOLO
reload
Crea offerta
«Form»
Inserimento Offerta
submit
«List»
Offerte Aggiudicate Utente
«DataBinding» OFFERTA
«Details»
Dati Articoli Offerte Aggiudicate Utente
«DataBinding» ARTICOLO
load
load
load
Click su Asta Aperta
click
«List»
Aste Visitate [User : username]
«DataBinding» ASTA
submit
«List»
Crea Articolo
«DataBinding» ARTICOLO
reload
Crea Articolo
«List»
Articoli Disponibili
«DataBinding» Articoli
selected
«Form»
Crea Asta
reload
Crea Asta
submit

Eventi & Azioni

Client side		Server side	
EVENTO	**AZIONE**	**EVENTO**	**AZIONE**
Index.html → form di login → invio	Controllo dati	POST username, password	Controllo credenziali utente
HomePage → load	Verifica se l'evento è acquisto o vendo e carica la pagina		
PaginaAcquisto → load	Aggiorna la view con offerte aggiudicate e aste aperte visitate	GET	Estrazione offerte aggiudicate con relativi articoli Estrazione aste aperte visitate
RicercaKeyword → submit form di ricerca keyword	Aggiorna la tabella aste aperte (con keyword)	POST keyword	Estrazione aste aperte che contengono la keyword nel nome/descrizione di almeno un articolo
Aste aperte → click «vedi dettaglio offerta»	Aggiorna la view mostrando la pagina offerta	GET → GetArticoliAsta (idAsta) GET → GetOfferteAsta (idAsta)	Estrazione articoliAsta Estrazione offerteAsta
PaginaOfferta → submit form di crea offerta	Aggiornare la view	POST prezzoOfferto, idAsta	Inserimento offerta
PaginaAcquisto → click bottone vendo	Aggiorna la view nascondendo la pagina acquisto e mostrando vendo		

Client side		Server side	
EVENTO	**AZIONE**	**EVENTO**	**AZIONE**
PaginaVendo → submit form crea articolo	Aggiornare la view (articoli disponibili)	POST nome, descrizione, prezzo, immagine	Inserimento articolo
PaginaVendo → submit form crea asta	Aggiorna la view (aste non chiuse)	POST rialzoMinimo, scadenza, articoliSelezionati	Inserimento asta
PaginaVendo → load	Aggiorna la view con aste non chiuse e aste chiuse	GET	Estrazione aste chiuse con articoli Estrazione aste non chiuse con articoli
Asta → click	Aggiorna la view mostrando pagina dettaglio asta	GET idAsta	Estrazione articoli asta
Dettaglio asta CHIUSA → load	Aggiorna la view mostrando aggiudicatario, indirizzo e prezzo offerto se presenti e i dati dell'asta	GET idAsta	Estrazione aggiudicatario Estrazione offerta Estrazione asta
Dettaglio asta APERTA → load	Aggiornare la view mostrando i dati dell'asta, le offerte (con il nome dell'utente)	GET idAsta	Estrazione asta Estrazione offerteAsta Estrazione nomeUtente
Dettaglio asta APERTA → click bottone chiudi	Aggiorna la view mostrando dettaglio asta chiusa e aggiornando le liste delle aste	POST idAsta	Cambio valore chiusa a true

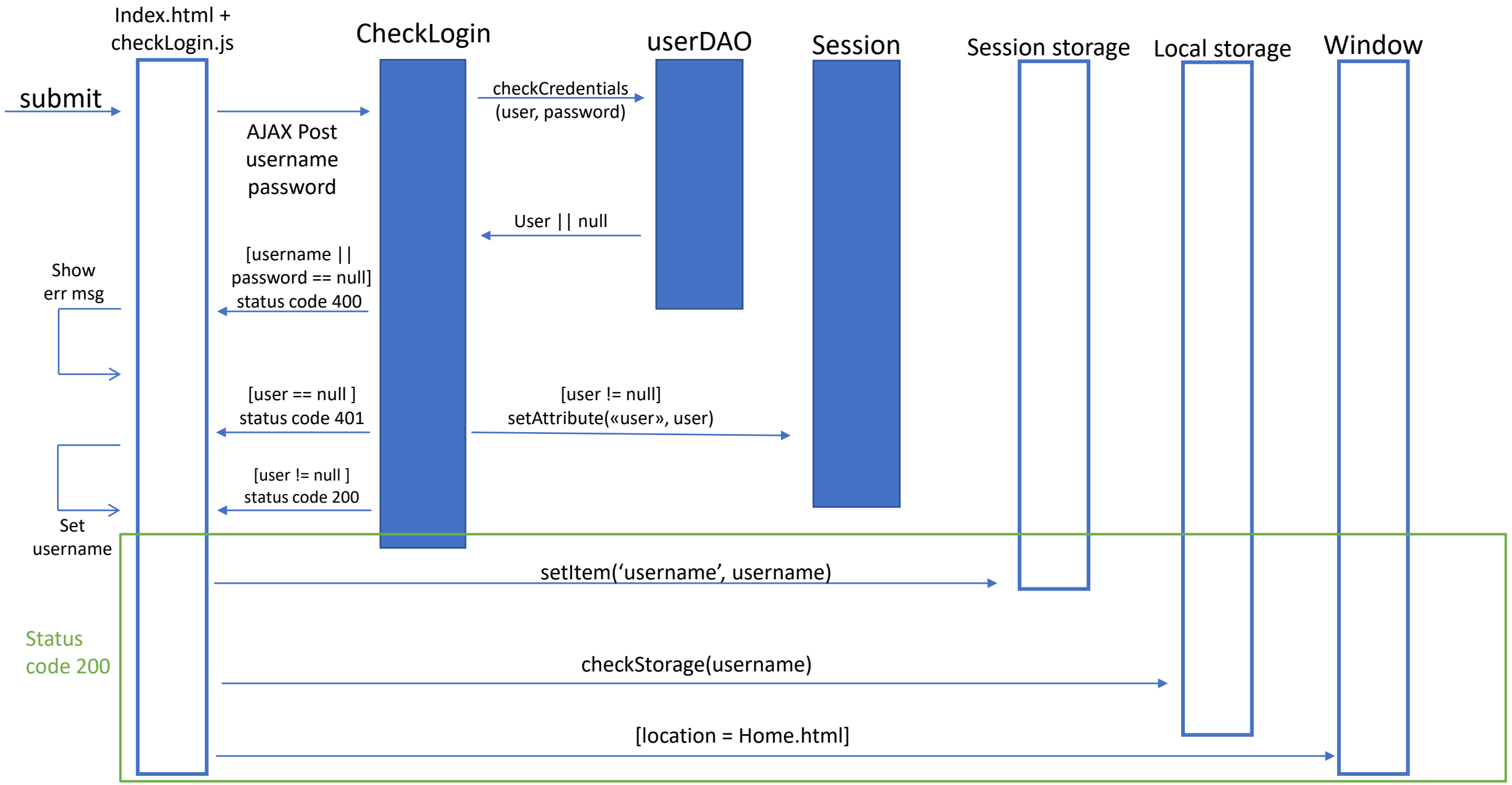
Controller & event handler

Client side		Server side	
EVENTO	**CONTROLLORE**	**EVENTO**	**CONTROLLORE**
Index.html → form di login → invio	makeCall	POST username, password	CheckLogin
HomePage → load	PageOrchestrator (sceglie tra acquisto e vendo)		
PaginaAcquisto → load	paginaAcquisto.show() → makeCall → update(offerte, articoli) + showAsteVisitate(aste)	GET	GoToAcquistoPage
RicercaKeyword → submit form di ricerca keyword	makeCall → ricercaKeyword.update(asteAperteKeyword)	POST keyword	RicercaKeyword
Aste aperte → click «vedi dettaglio offerta»	paginaOfferta.show(idAsta) → showOfferte(idAsta) + showArticoli(idAsta)	GET (idAsta) NB: sono due get	GetOfferteAsta GetArticoliAsta
PaginaOfferta → submit form di crea offerta	makeCall → paginaAcquisto.show() + showOfferte(idAsta)	POST prezzoOfferto, idAsta	CreateOfferta
PaginaAcquisto → click bottone vendo	showVendo(orchestrator)		

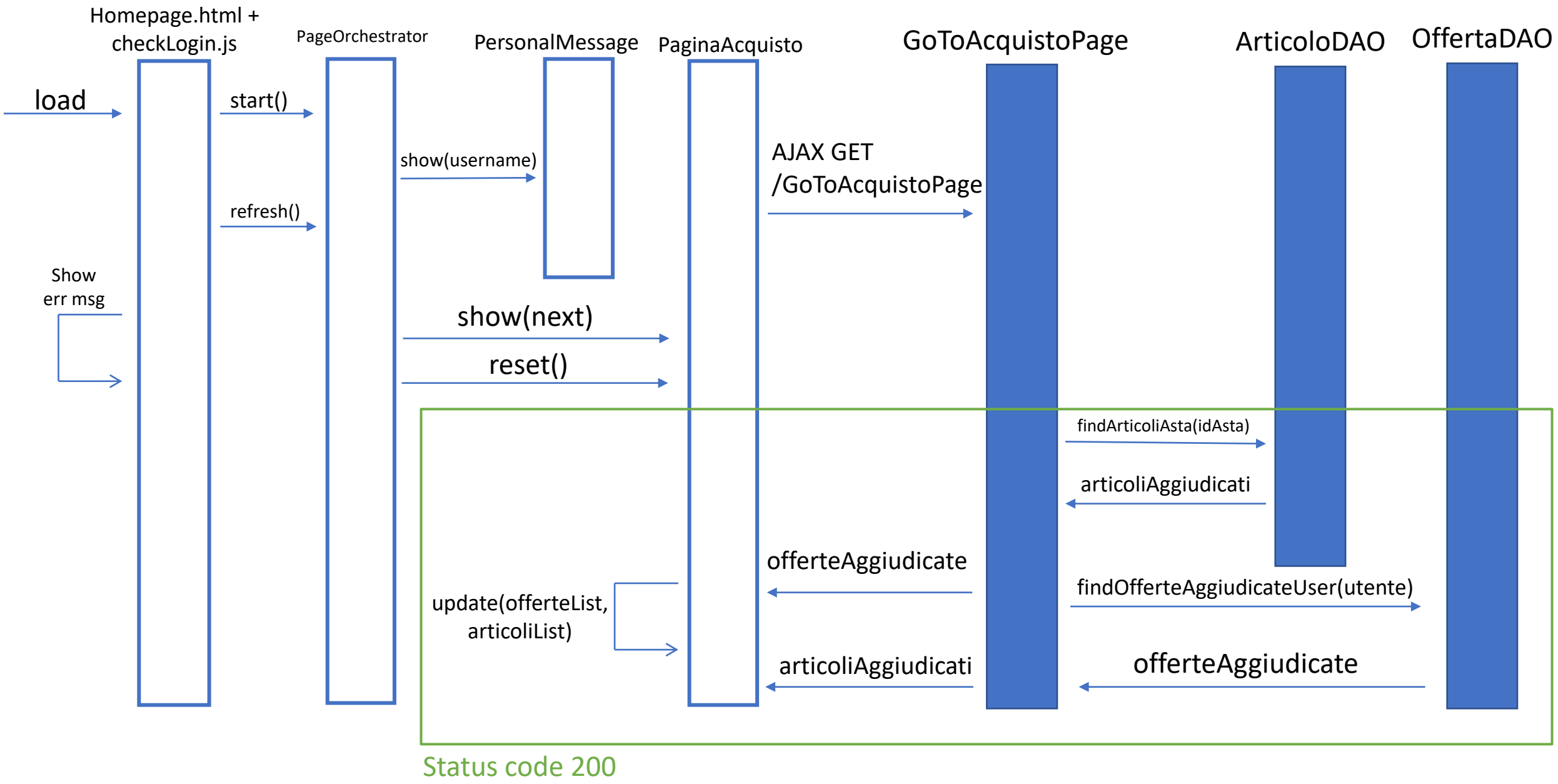
Client side		Server side	
EVENTO	**CONTROLLORE**	**EVENTO**	**CONTROLLORE**
PaginaVendo → submit form crea articolo	makeCall	POST nome, descrizione, prezzo, immagine	CreaArticolo
PaginaVendo → submit form crea asta	makeCall	POST rialzoMinimo, scadenza, articoliSelezionati	CreaAsta
PaginaVendo → load	PaginaVendo.show(): makeCall → showAsteNonChiuse(asteNonChiuse, tempoMancanteNonChiuse, articoliAsteNonChiuse) + showAsteChiuse(asteChiuse, articoliAsteChiuse) + updateArticoliDisponibili(articoliDisponibili)	GET	GoToVendoPage
PaginaVendo → click bottone acquisto	paginaAcquisto.show()	GET	GoToAcquistoPage
Asta → click	DettaglioAsta.show(idAsta, boolean chiusa): makeCall	GET idAsta	GetDettaglioAsta

Client side		Server side	
EVENTO	**CONTROLLORE**	**EVENTO**	**CONTROLLORE**
Dettaglio asta CHIUSA → load	DettaglioAsta.showChiusa(idAsta): makeCall	GET idAsta	GetAstaChiusa
Dettaglio asta APERTA → load	DettaglioAsta.showAperta(idAsta): makeCall	GET idAsta	GestAstaAperta
Dettaglio asta APERTA → click bottone chiudi	makeCall	POST idAsta	ChiudiAsta
LOGOUT		GET	Logout

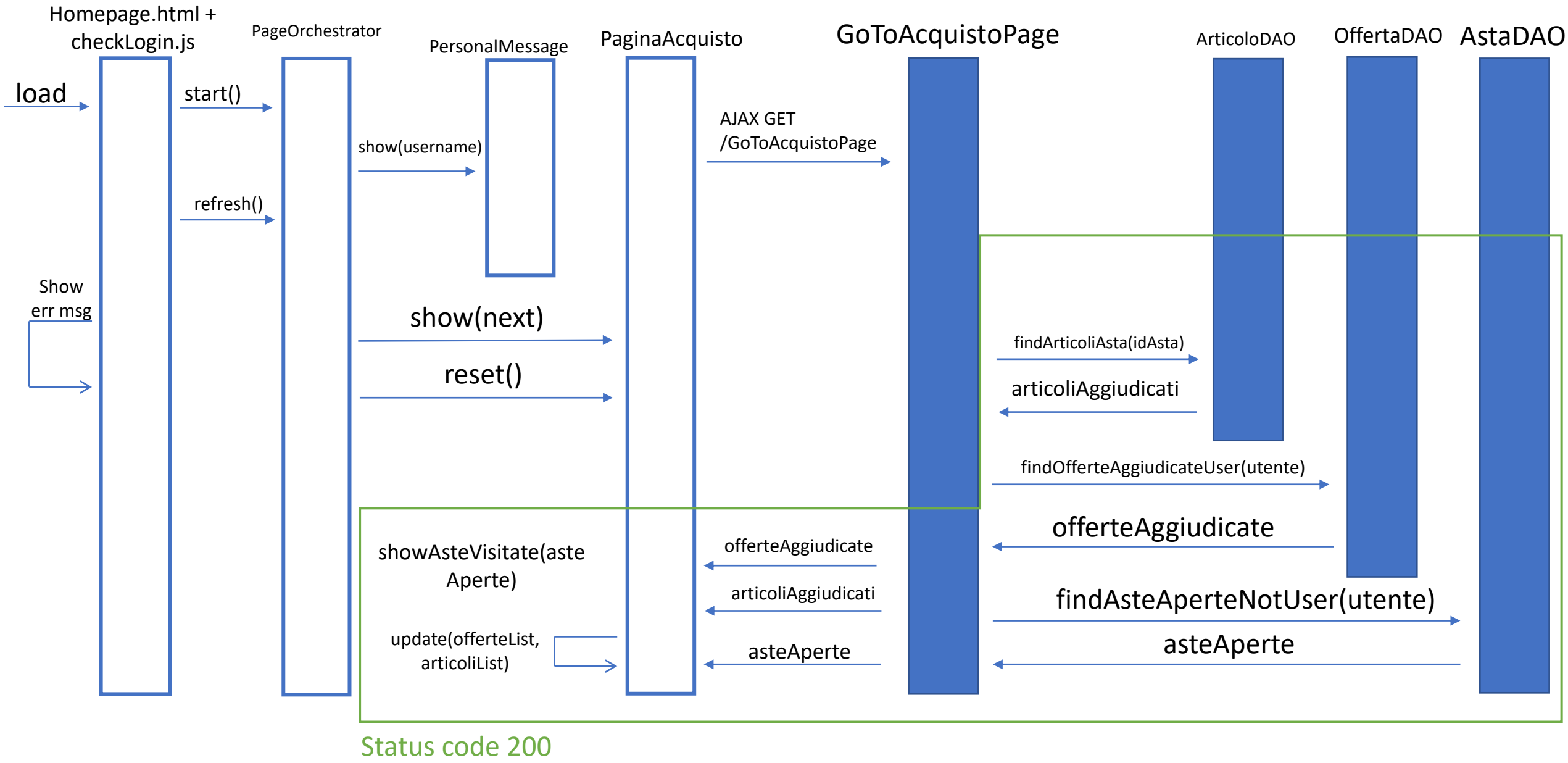
Check Login – evento



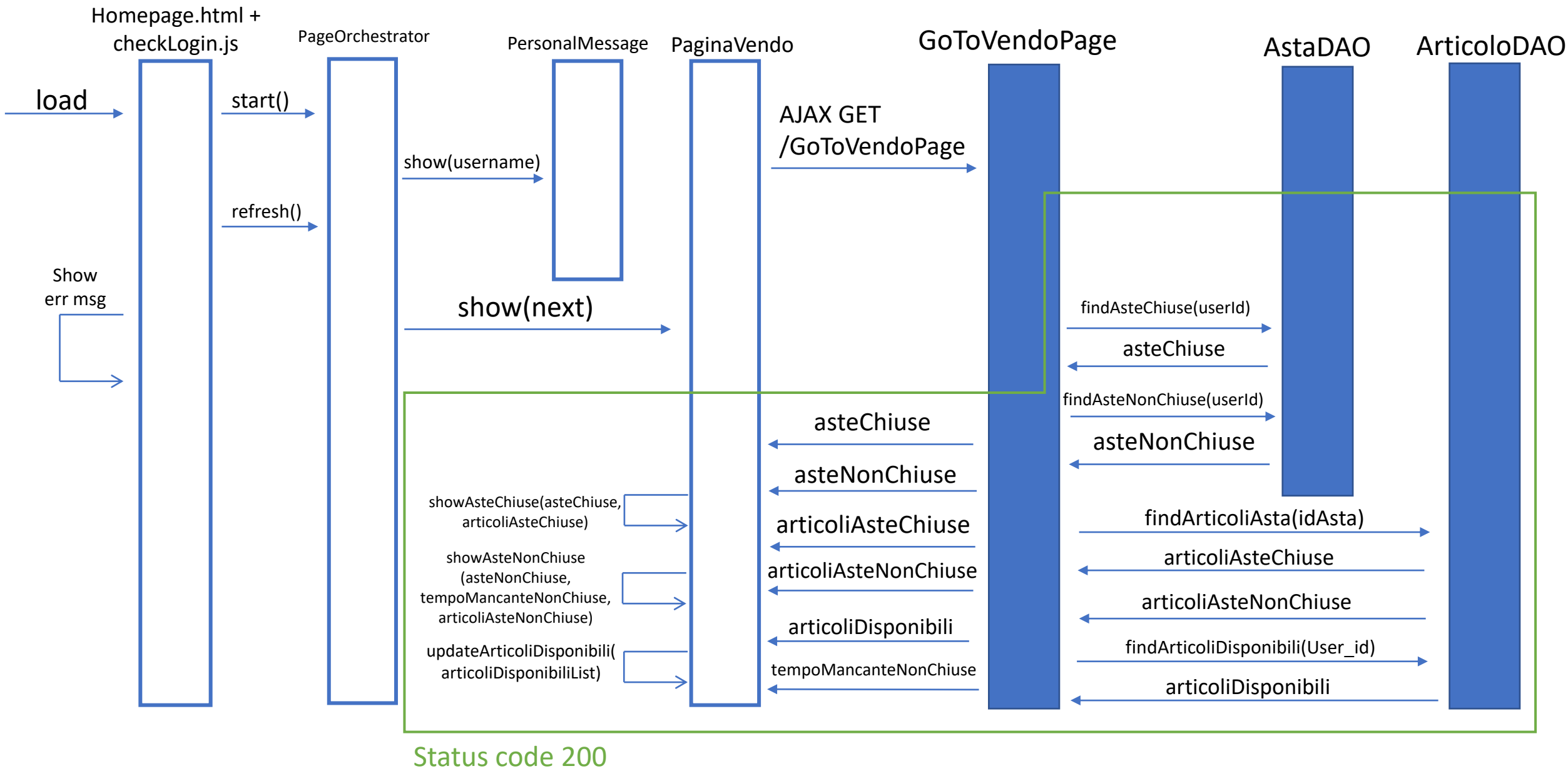
Caricamento HomePage **primo accesso** – evento



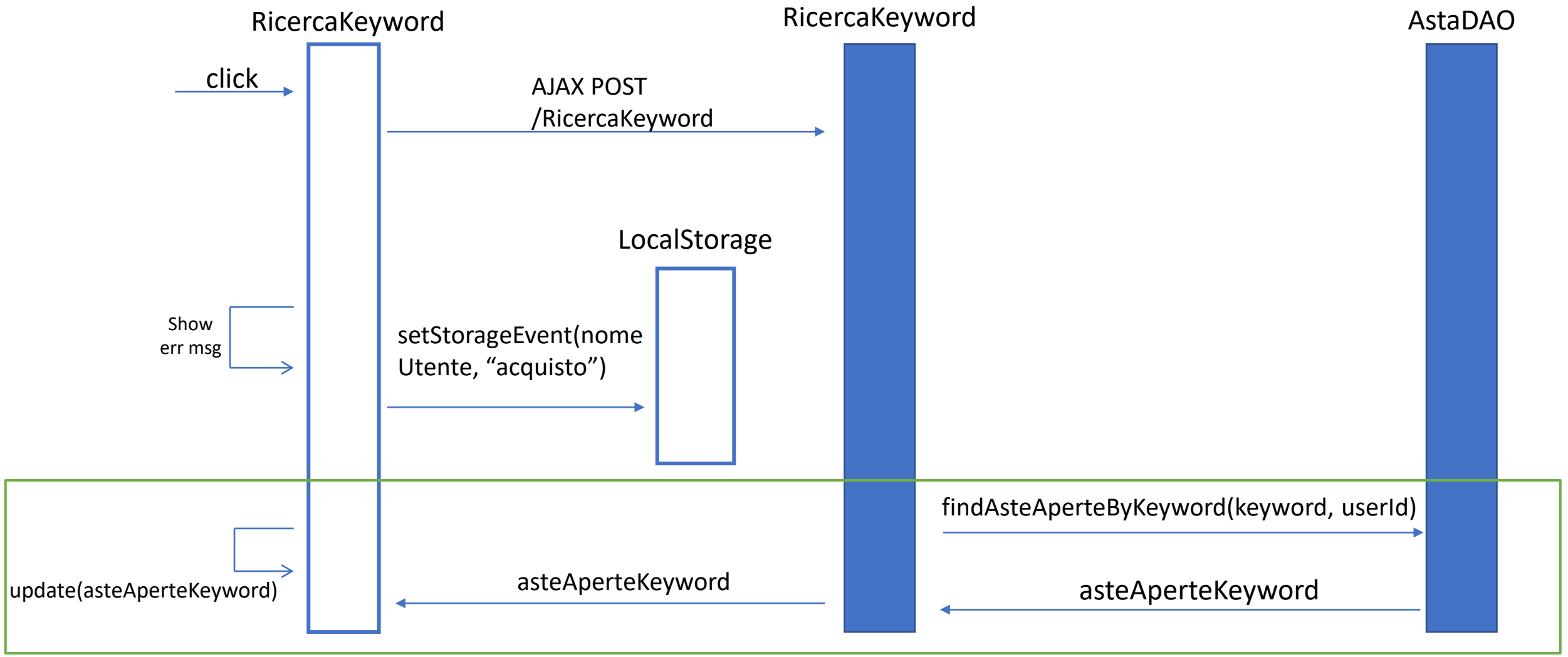
Caricamento HomePage **accesso successivo al primo dopo aver visitato aste** – evento = «ACQUISTO»



Caricamento HomePage **accesso successivo al primo dopo aver creato un'asta** – evento = «VENDO»

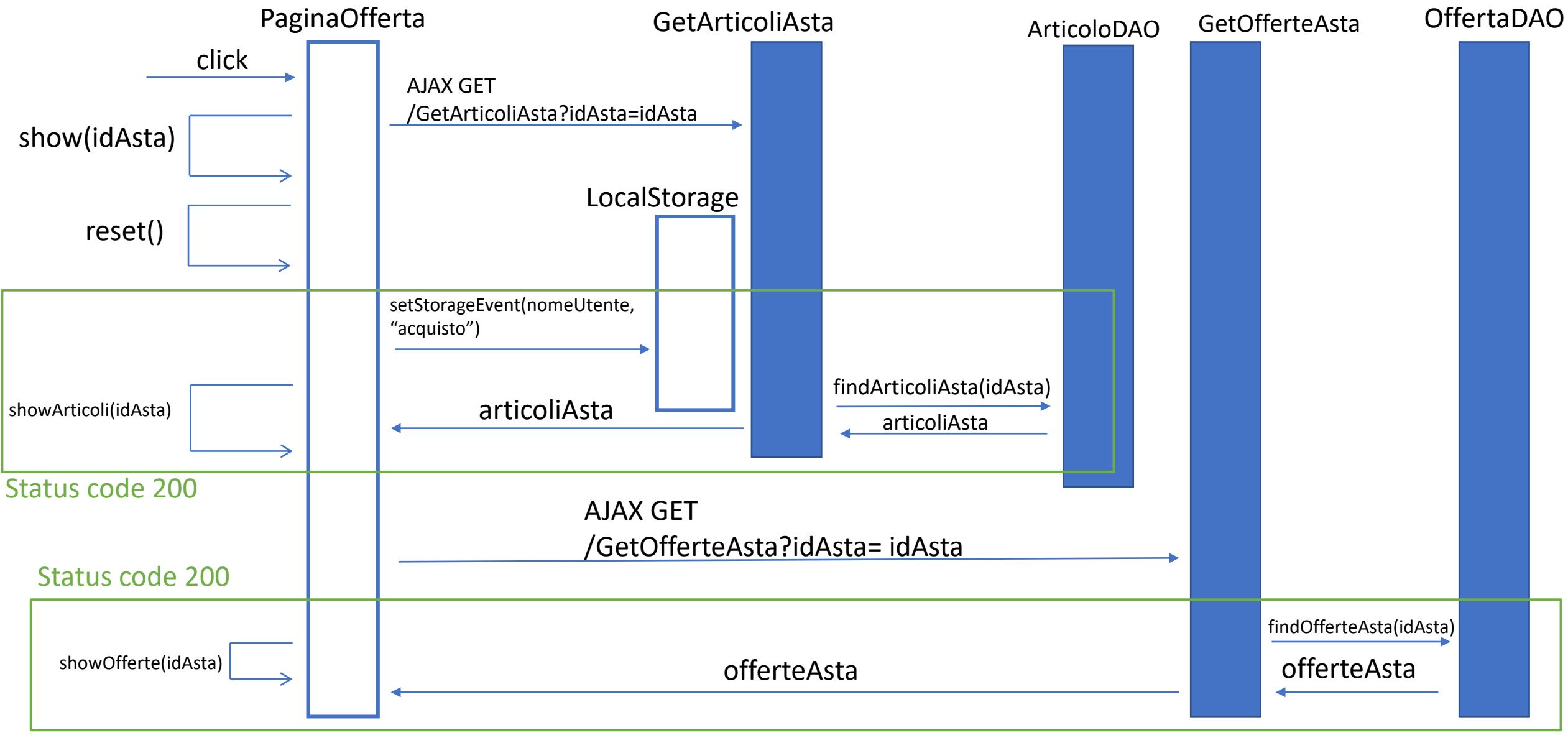


Ricerca keyword – evento

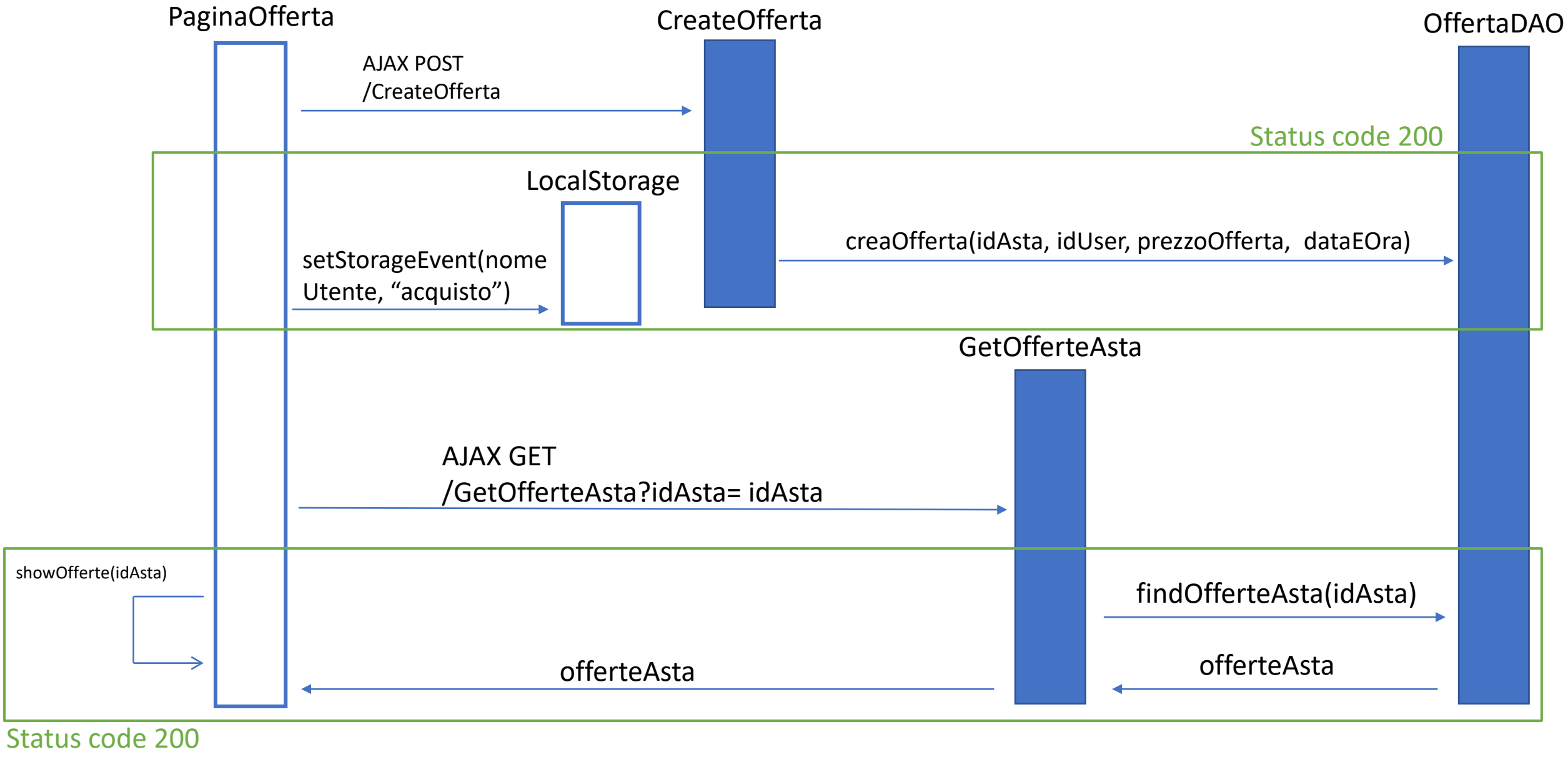


Status code 200

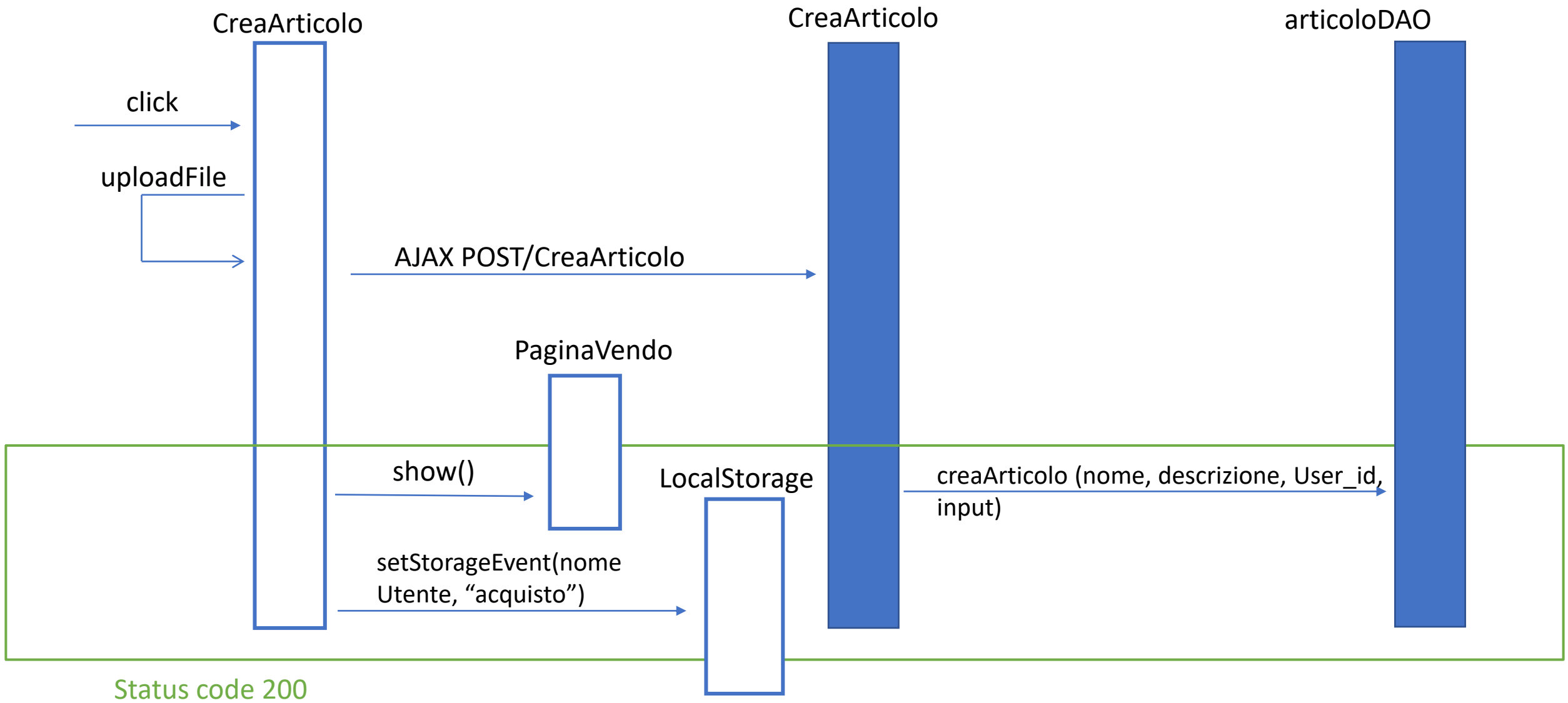
Click su asta aperta - evento



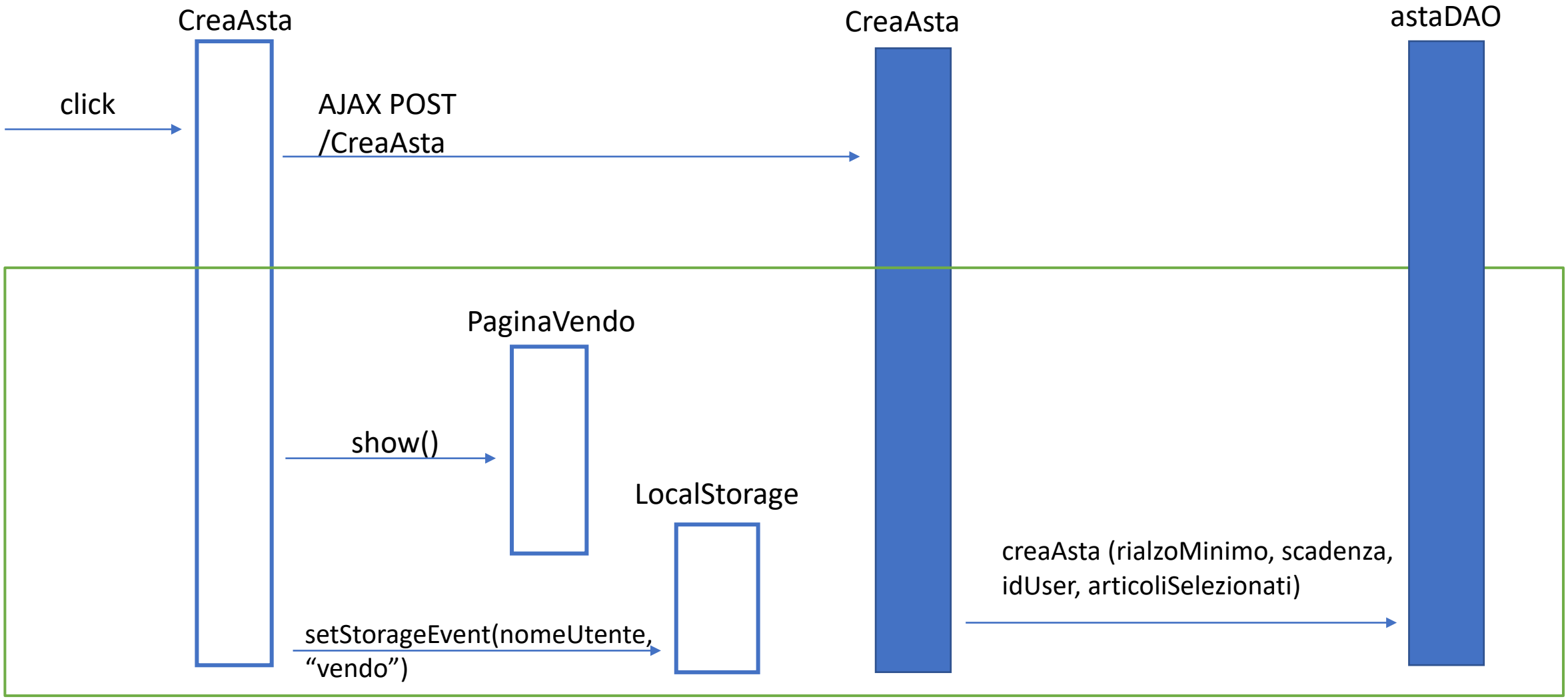
Inserimento offerta - evento



Creazione articoli - evento

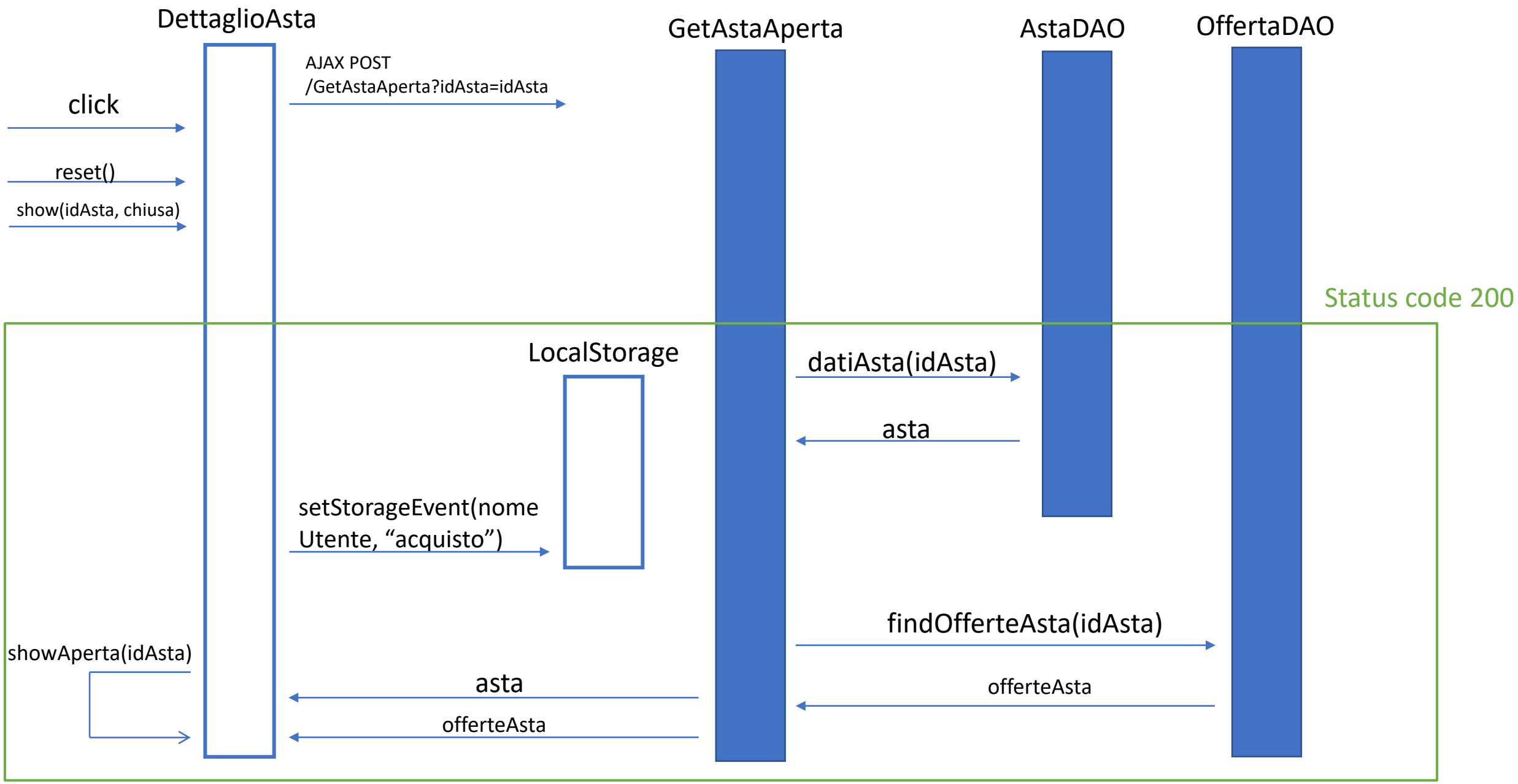


Creazione asta - evento

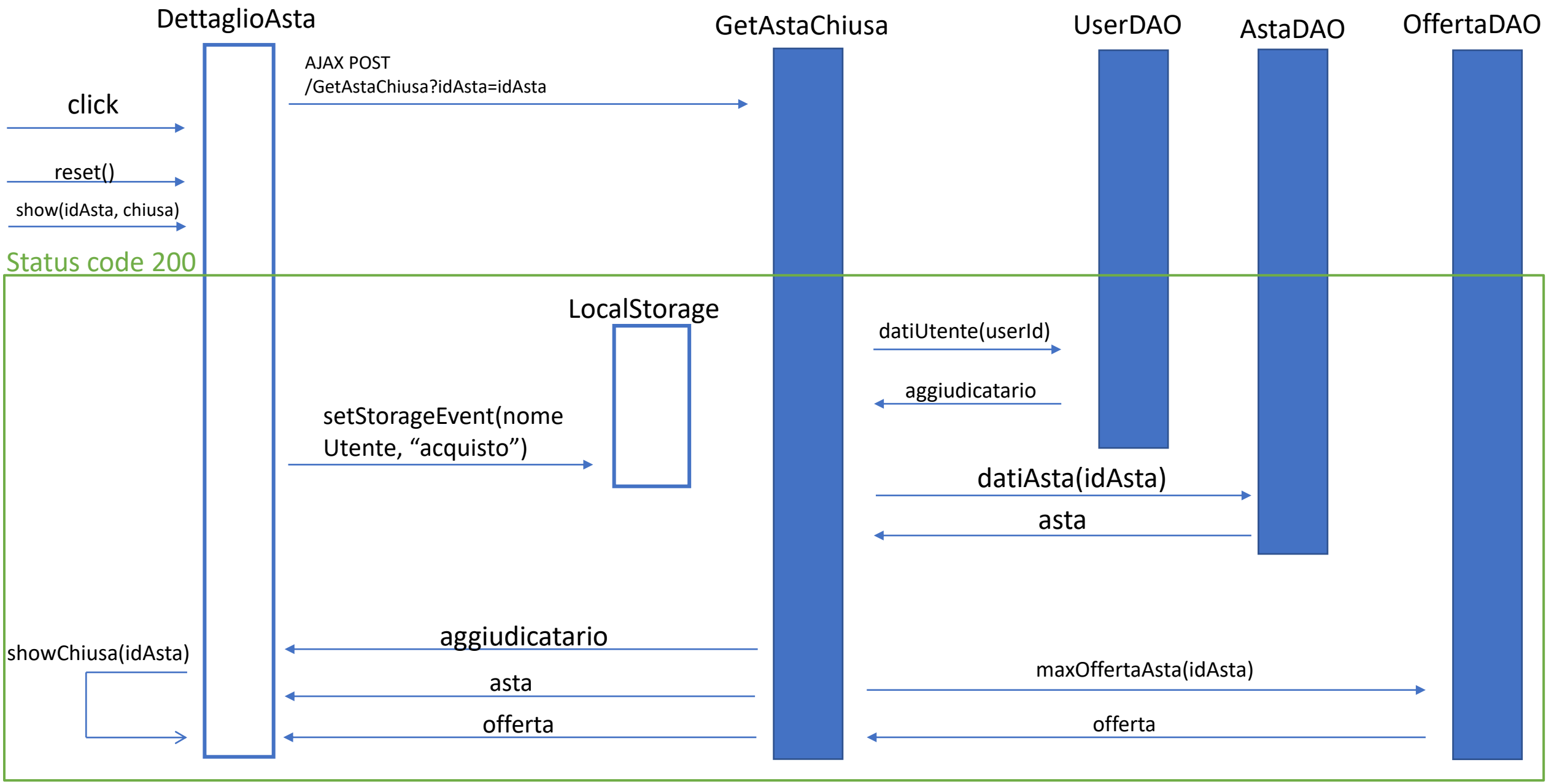


Status code 200

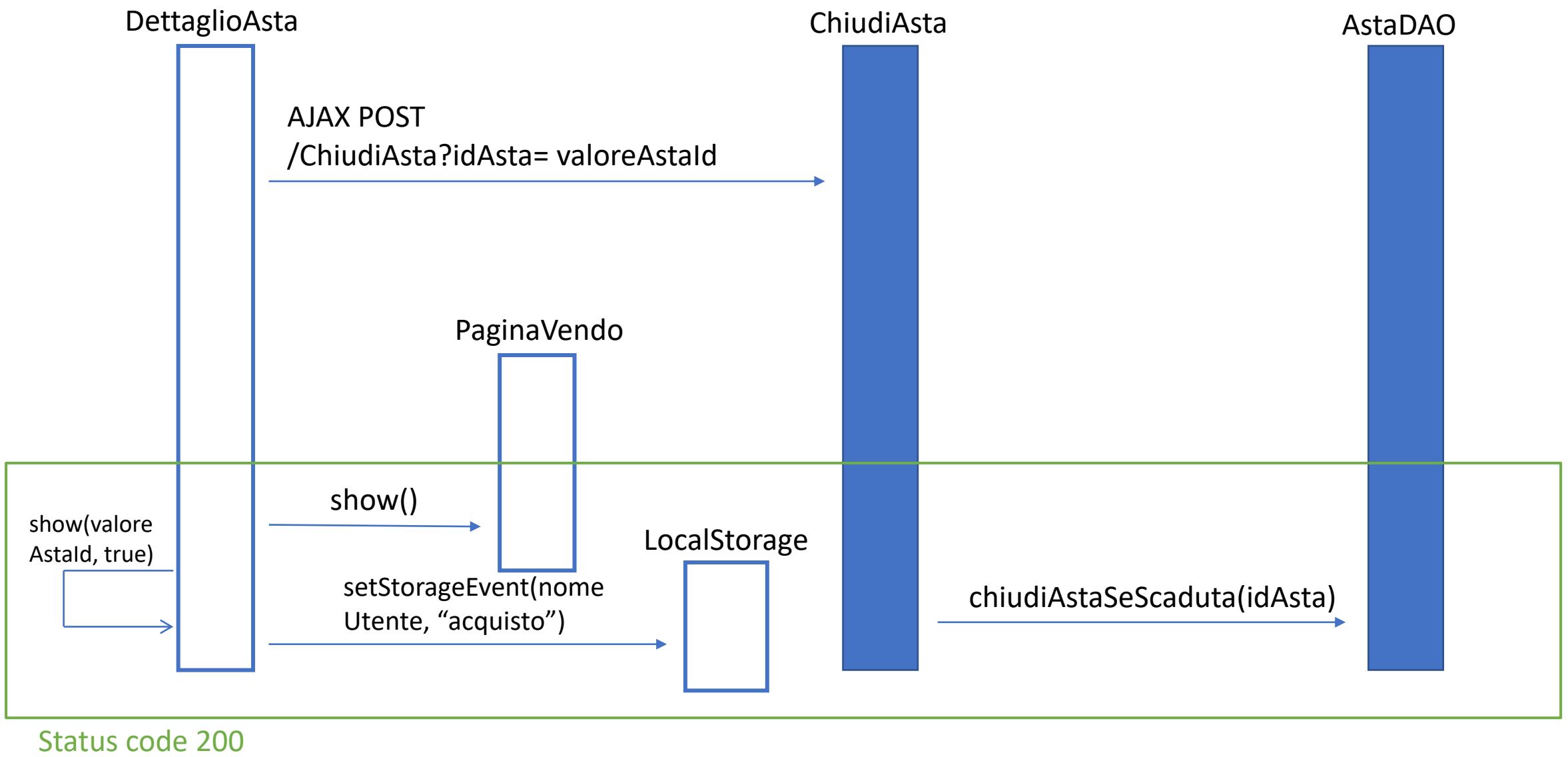
Click su asta aperta - evento



Click su asta chiusa - evento



Chiudi asta - evento



Status code 200

Logout - evento