

AsteOnline

Francesco Tarantino

Prof. Fraternali Piero

Anno Accademico 2022-23

Specifica

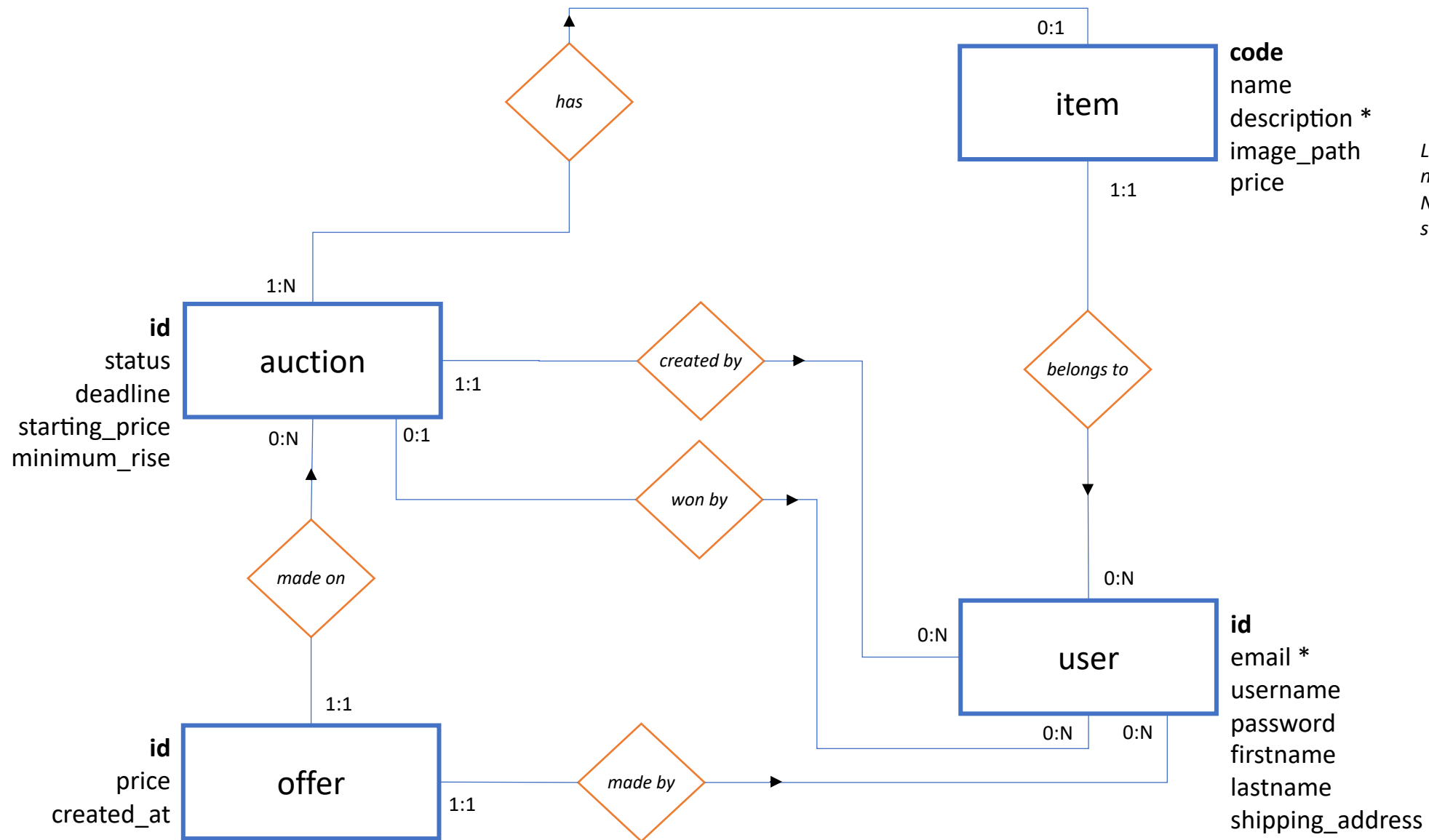
Un'applicazione web consente la gestione di aste online. Gli utenti accedono tramite login e possono vendere e acquistare all'asta. La HOME page contiene due link, uno per accedere alla pagina VENDO e uno per accedere alla pagina ACQUISTO. La pagina VENDO mostra una lista delle aste create dall'utente 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.

Analisi dei dati

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.

Entities, **attributes**, **relationships**

Diagramma E/R



Schema Database

```
CREATE TABLE user
(
    id                INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    email              VARCHAR(80) NULL,
    username            VARCHAR(80) NOT NULL,
    password            CHAR(128)    NOT NULL,
    firstname           VARCHAR(80)  NOT NULL,
    lastname            VARCHAR(80)  NOT NULL,
    shipping_address    TEXT          NOT NULL
);
```

- ❖ *L'inserimento di una email è facoltativo.*
- ❖ *La password non viene salvata in chiaro ma sotto forma di hash. L'algoritmo usato per l'hashing è **SHA-512**, che produce una sequenza fissa di 512 bit (128 caratteri esadecimali).*

Schema Database

```
CREATE TABLE auction
(
    id                INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    status             ENUM ('OPEN', 'CLOSED') DEFAULT 'OPEN' NOT NULL,
    deadline           DATETIME                                NOT NULL,
    starting_price     DECIMAL(20, 2) UNSIGNED                NOT NULL,
    minimum_rise       INT UNSIGNED                            NOT NULL,
    user_id            INT UNSIGNED                            NOT NULL,
    won_by             INT UNSIGNED                            NULL,
    final_price        DECIMAL(20, 2) UNSIGNED                NULL,

    CONSTRAINT auction_created_by
        FOREIGN KEY (user_id) REFERENCES user(id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT auction_won_by
        FOREIGN KEY (won_by) REFERENCES user(id)
        ON UPDATE CASCADE
);
```

Schema Database

```
CREATE TABLE item
(
    code          VARCHAR(20)          NOT NULL PRIMARY KEY,
    name           VARCHAR(255)         NOT NULL,
    description    TEXT                 NULL,
    image_path     TEXT                 NOT NULL,
    price          DECIMAL(10, 2) UNSIGNED NOT NULL,
    user_id        INT UNSIGNED         NOT NULL,
    auction_id     INT UNSIGNED         NULL,

    CONSTRAINT item_created_by
        FOREIGN KEY (user_id) REFERENCES user(id)
        ON UPDATE CASCADE ON DELETE CASCADE,
    CONSTRAINT item_in_auction
        FOREIGN KEY (auction_id) REFERENCES auction(id)
        ON UPDATE CASCADE ON DELETE SET NULL
);
```

Schema Database

```
CREATE TABLE offer
(
    id          INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    user_id      INT UNSIGNED                      NOT NULL,
    auction_id  INT UNSIGNED                      NOT NULL,
    price        DECIMAL(20, 2) UNSIGNED          NOT NULL,
    created_at  DATETIME DEFAULT CURRENT_TIMESTAMP NOT NULL,

    CONSTRAINT offer_made_by
        FOREIGN KEY (user_id) REFERENCES user(id)
            ON UPDATE CASCADE,
    CONSTRAINT offer_made_on
        FOREIGN KEY (auction_id) REFERENCES auction(id)
            ON UPDATE CASCADE
);
```


Schema Database

```
CREATE VIEW winning_offer AS
(
    SELECT offer.auction_id AS auction_id,
           offer.user_id   AS user_id,
           offer.price      AS price
    FROM offer
    WHERE (offer.price >= (SELECT MAX(o.price)
                           FROM offer AS o
                           WHERE (offer.auction_id = o.auction_id)
                          )
           )
);
```

❖ *La view **winning_offer** agevola la ricerca dell'offerta vincente per ogni asta.*

Analisi requisiti

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**.

Pages (views), view components

Analisi requisiti

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.

Events, actions

Versione Pure HTML

Componenti

- **Model objects** (JavaBeans)
 - User
 - Auction
 - Item
 - Offer
- **Data Access Objects**
 - UserDao
 - getUserByID(id)
 - authenticateUser(usernameOrEmail, password)
 - createUser(user, password)
 - isUsernameTaken(username)
 - ItemDAO
 - addItem(item)
 - getItemsAvailableForAuction(userID)
 - attachItemToAuction(itemCode, auctionID)
 - getItemsAttachedToAuction(auctionID)
 - OfferDAO
 - getAuctionOffers(auctionID)
 - addOffer(auctionID, userID, price)
 - AuctionDAO
 - newAuction(deadline, minimumRise, userID, itemsCodes[])
 - userAuctions(userID, status)
 - openUserAuctions(userID)
 - closedUserAuctions(userID)
 - getAuction(auctionID)
 - closeAuction(auctionID)
 - searchOpenAuctions(query, userID)
 - getWonAuctions(userID)

Componenti

- **Controllers** (*servlets*)

- *Authentication*
 - PerformLogin
 - PerformSignup
 - Logout
- *Buy*
 - Buy
 - NewOffer
 - Offer
- *Sell*
 - AddItem
 - AuctionDetails
 - CloseAuction
 - NewAuction
 - Sell
- GetImage
- Home

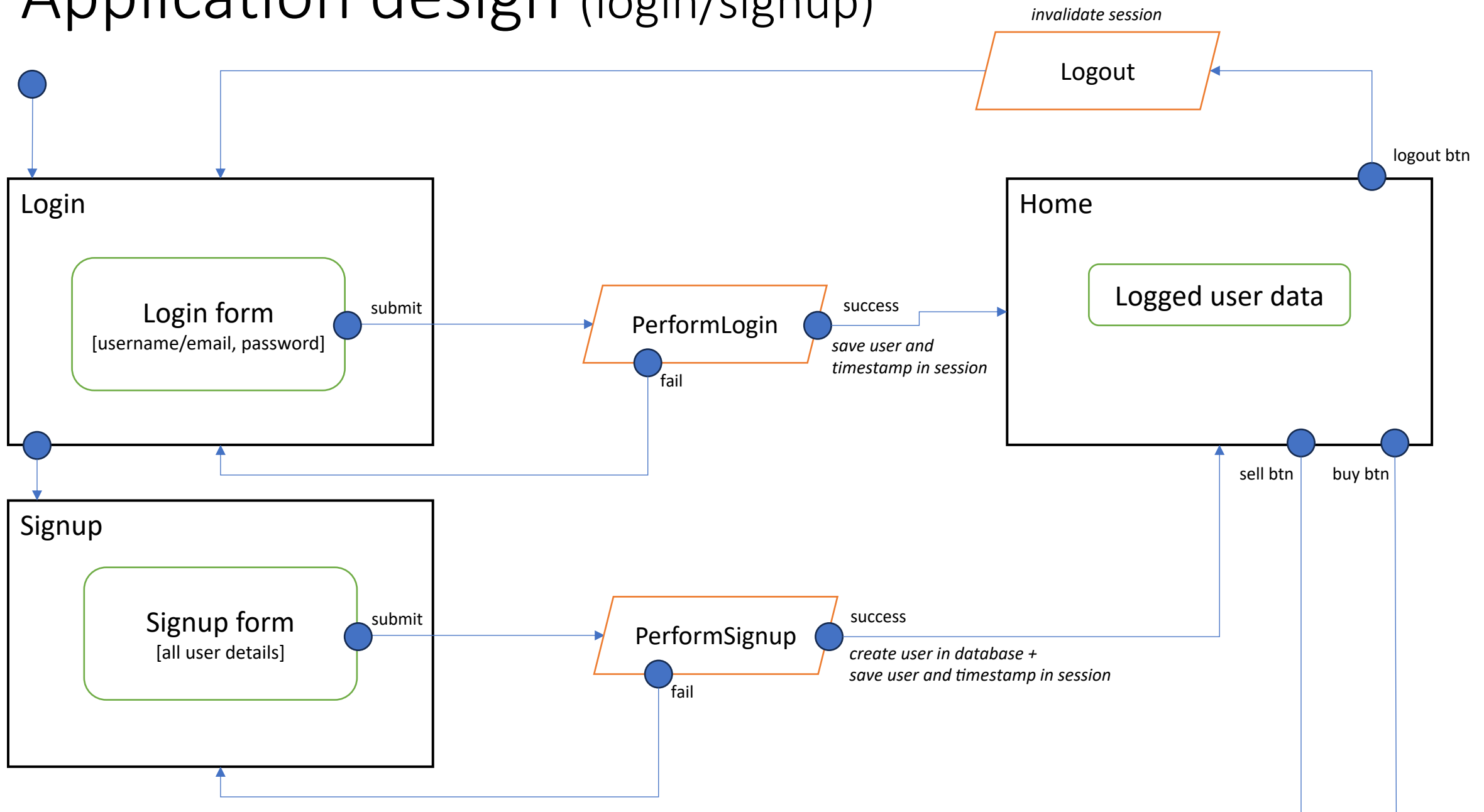
- **Views** (*templates*)

- Login
- Signup
- Home
- Buy
- Sell
- AuctionDetails
- Offer

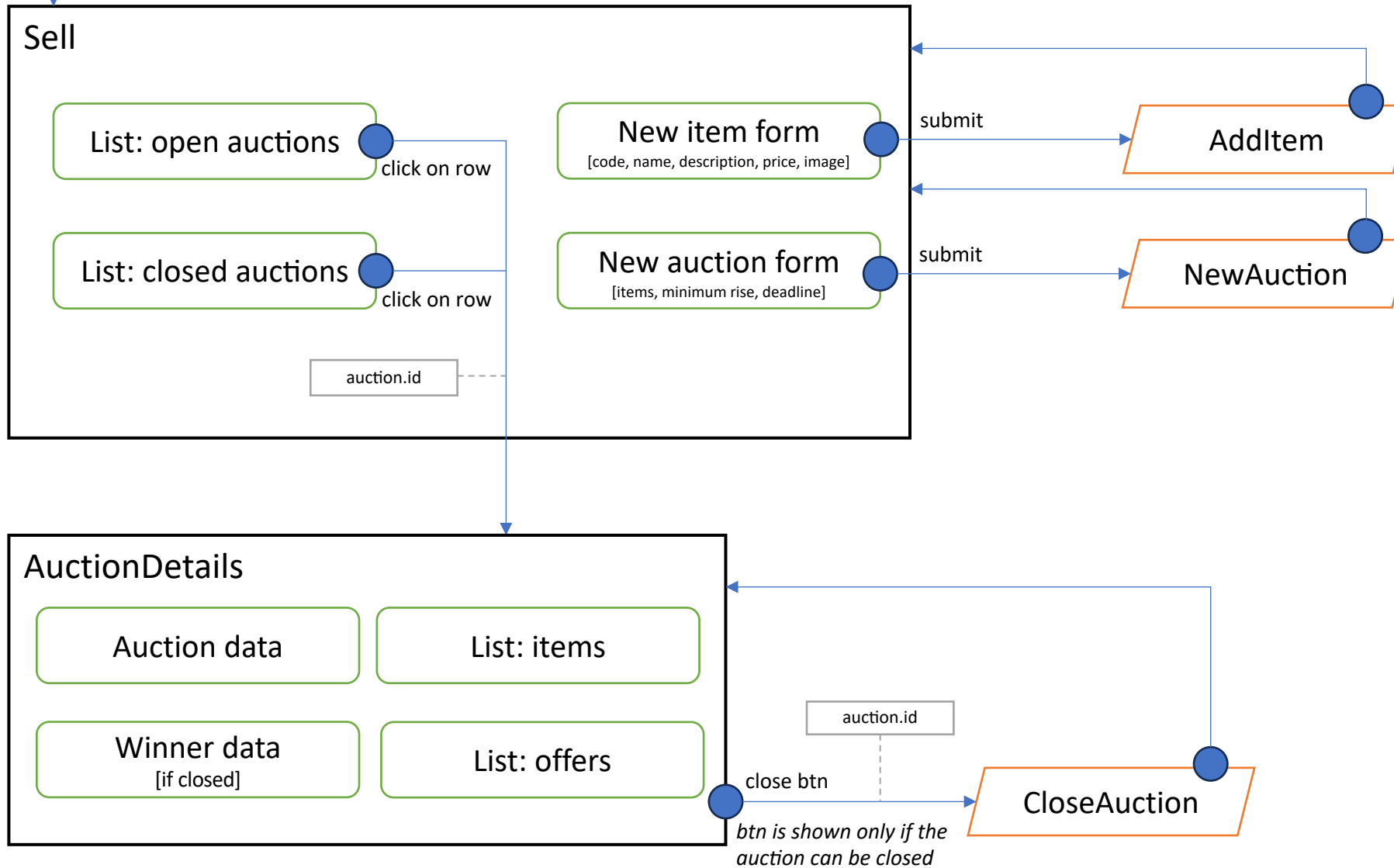
- *Classi di utilità*

- ConnectionHandler
- Constants
- Dates
- FileManager
- Hash

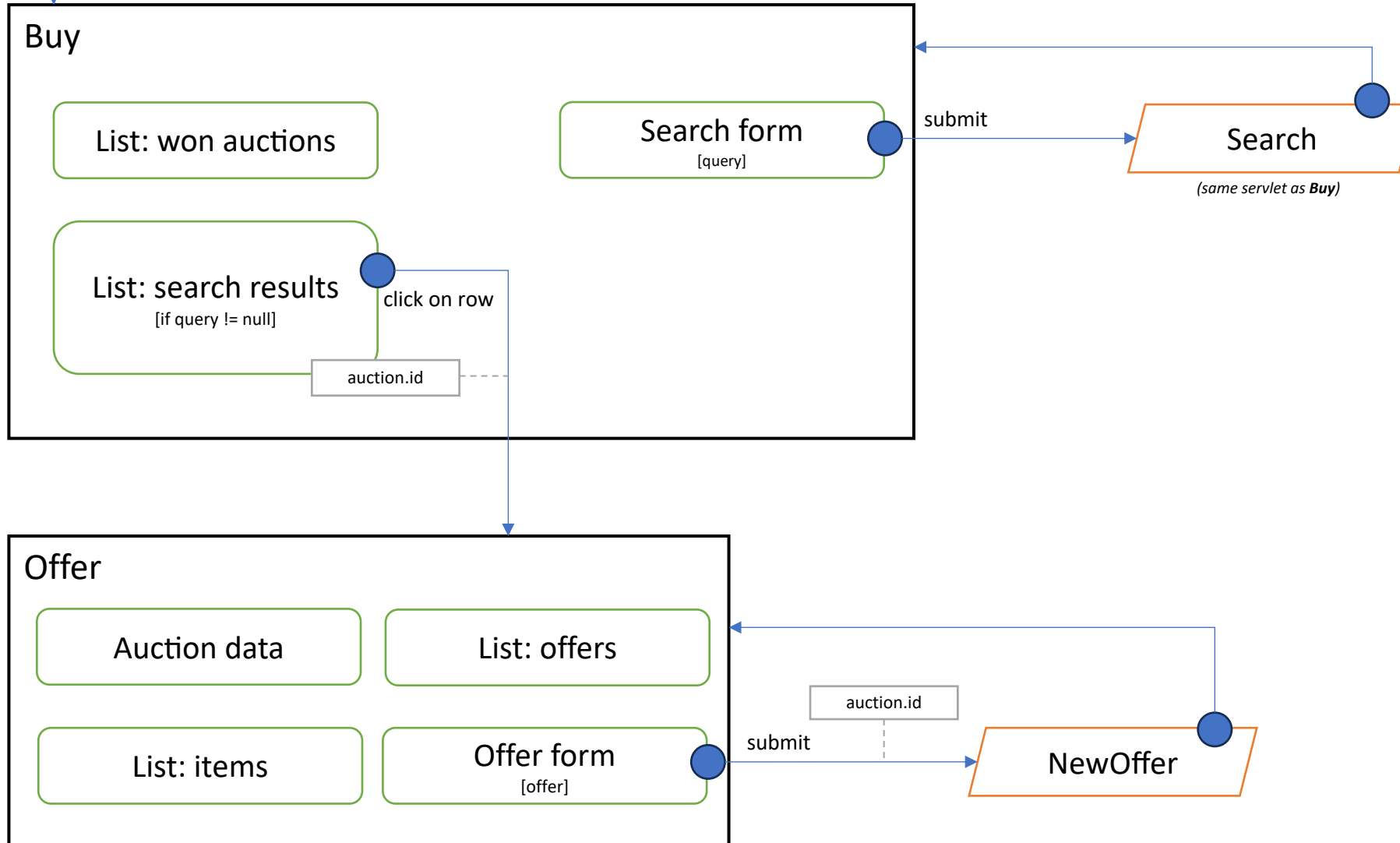
Application design (login/signup)



Application design (sell)

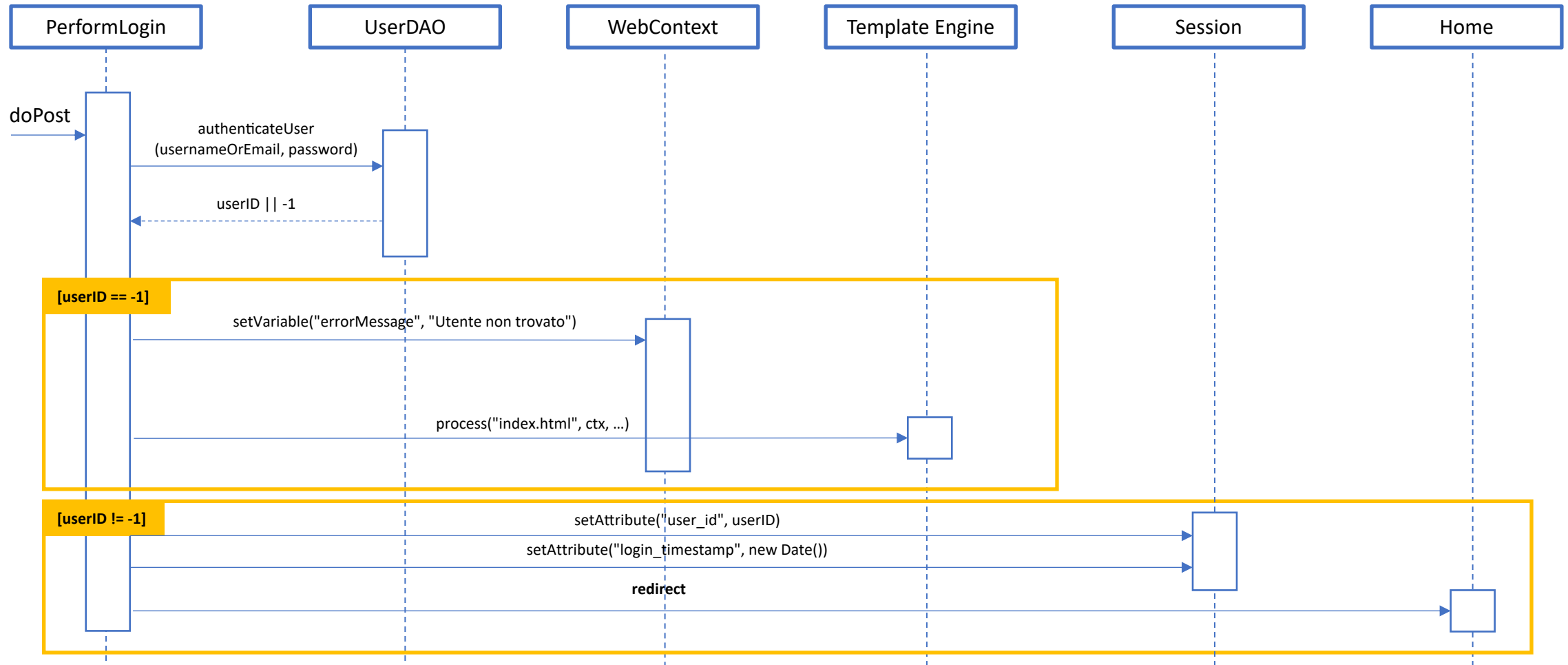


Application design (buy)



Event: login

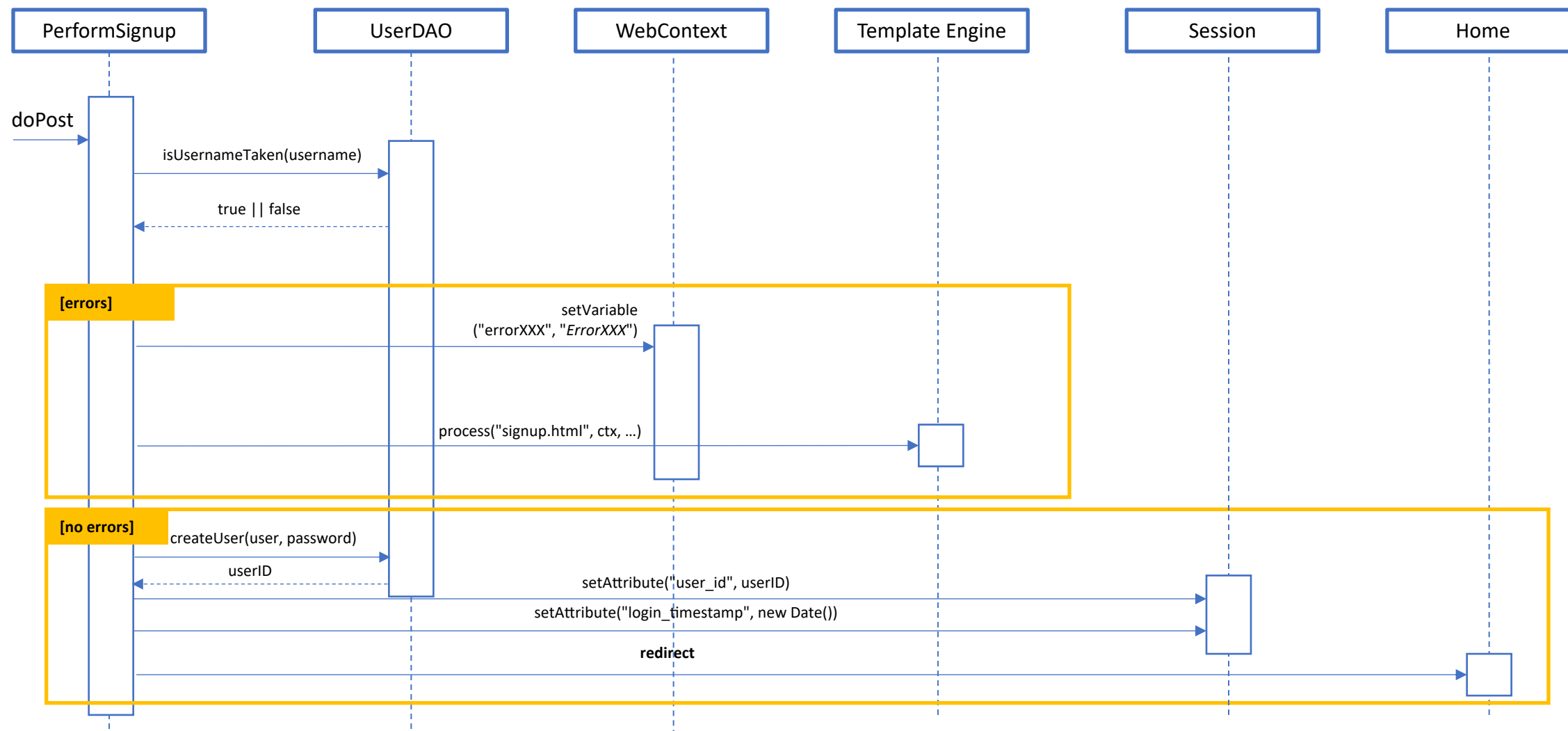
POST request from **index.html**
Params: [usernameOrEmail, password]



Event: signup

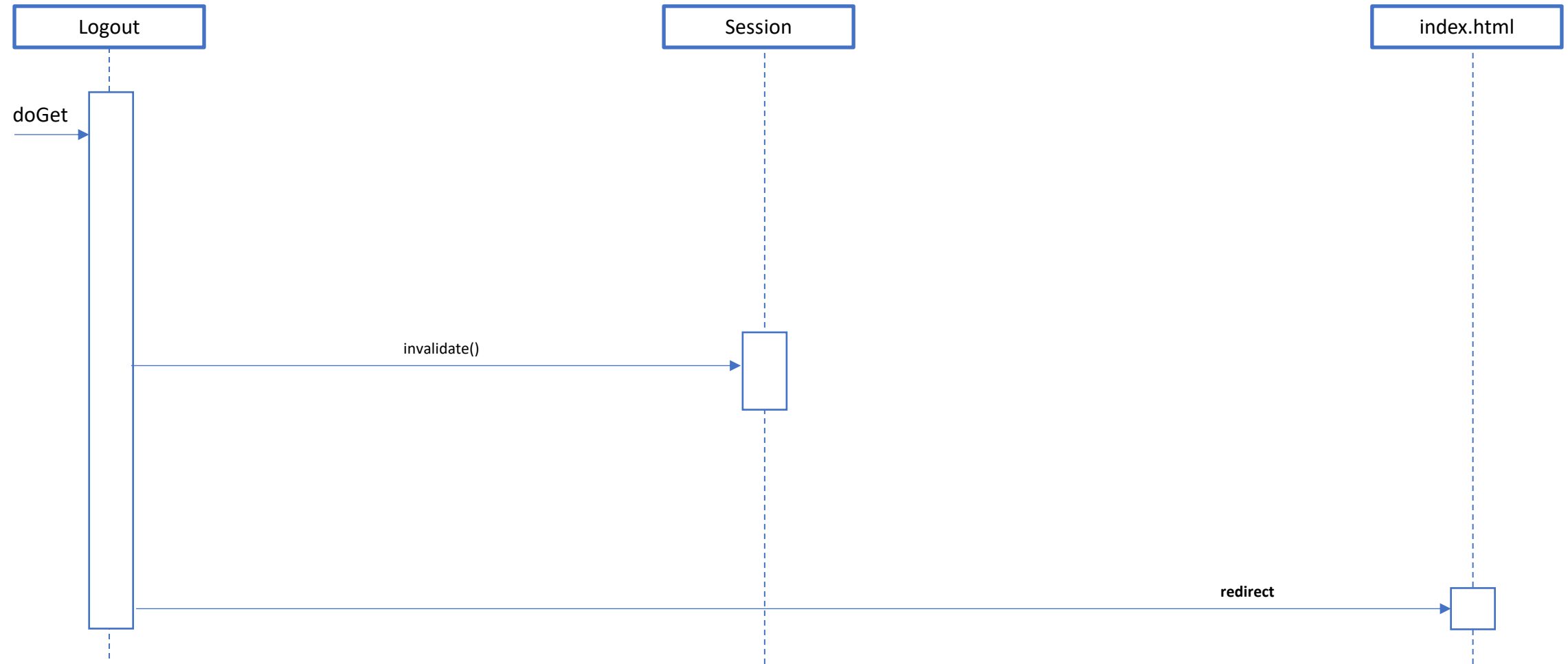
POST request from **signup.html**

Params: [firstname, lastname, shipping_address, email *, username, password, password_confirm]



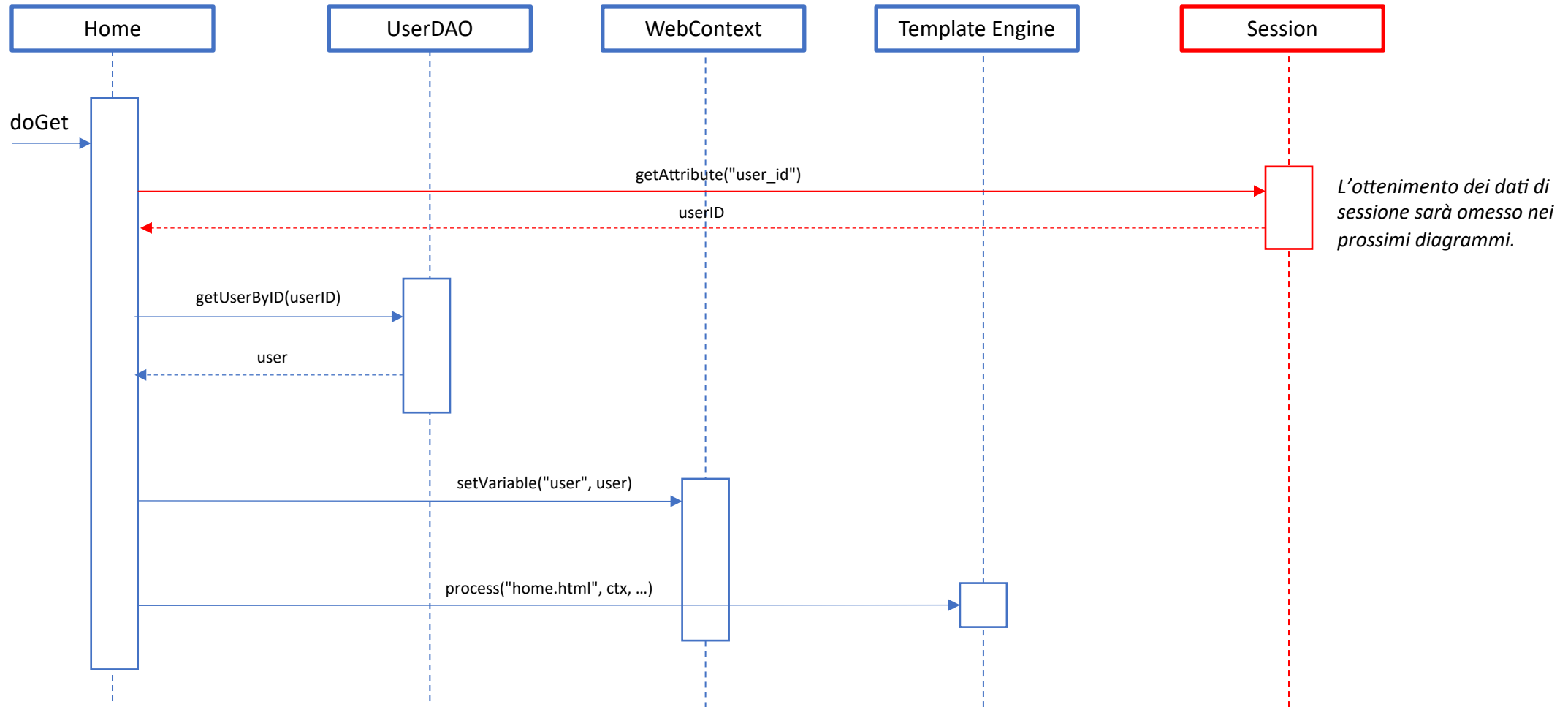
Event: logout

GET request



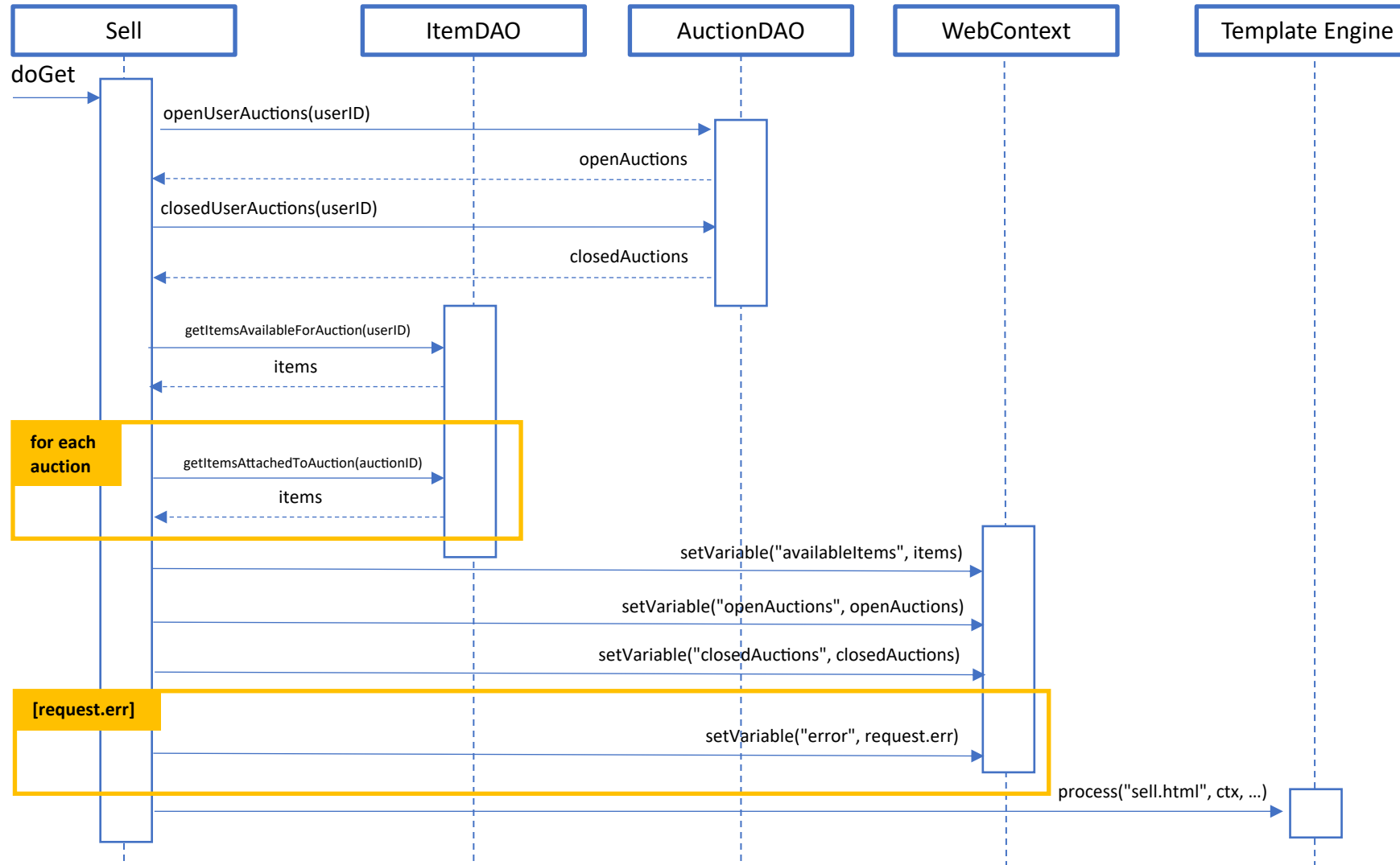
View: Home

GET request



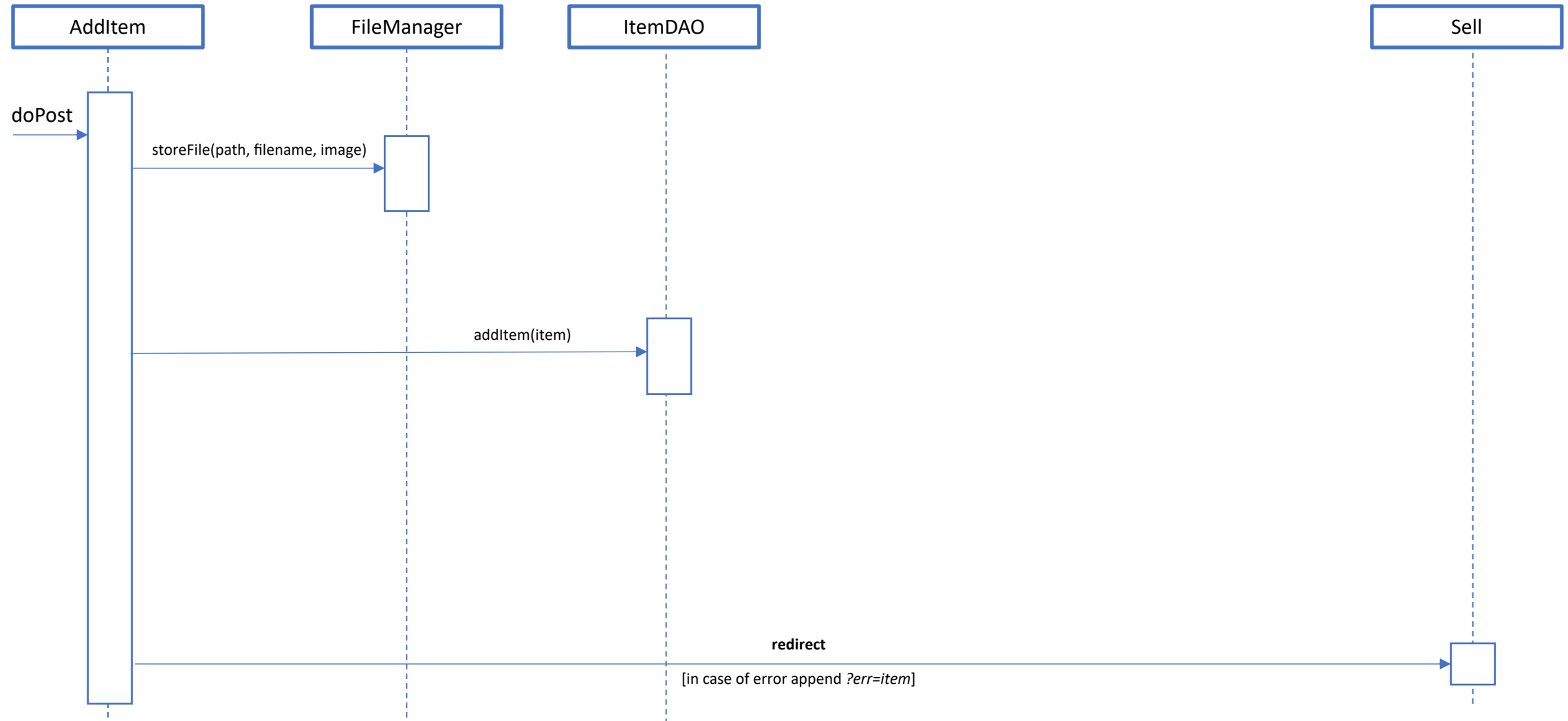
View: Sell

GET request
Params: [err *]



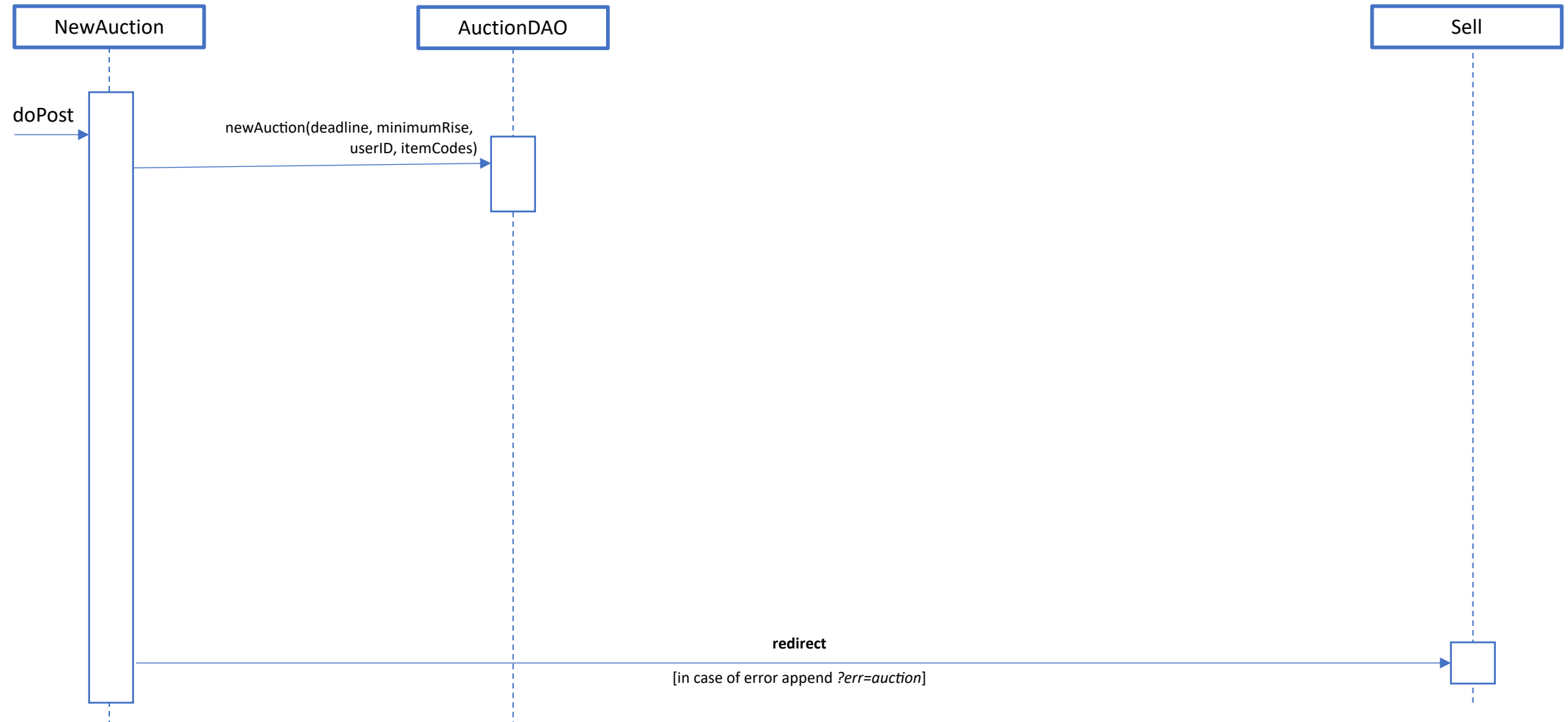
Event: new item

POST request from **/Sell**
Params: [code, name, description *, price, image]



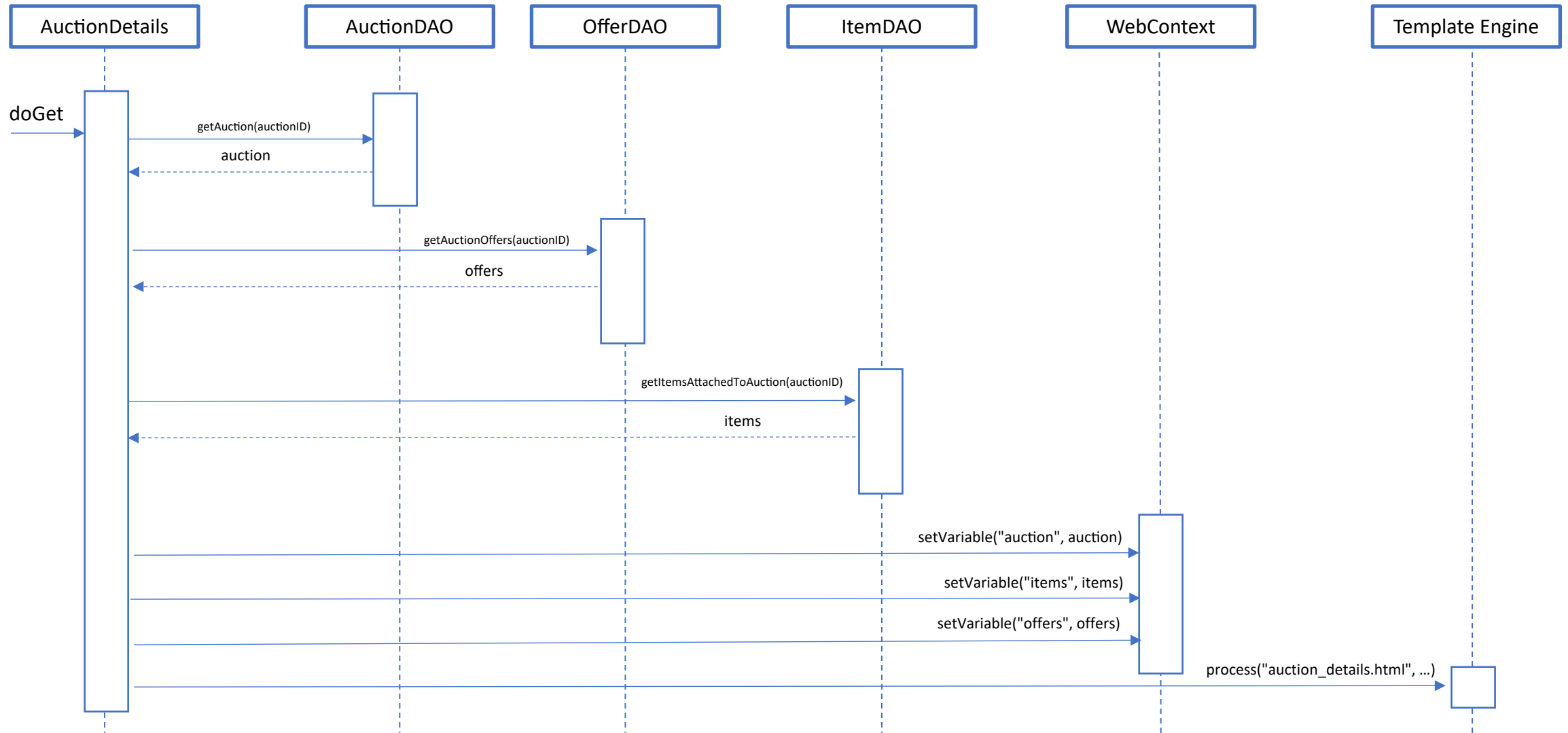
Event: new auction

POST request from /Sell
Params: [items[], minimum_rise, deadline]



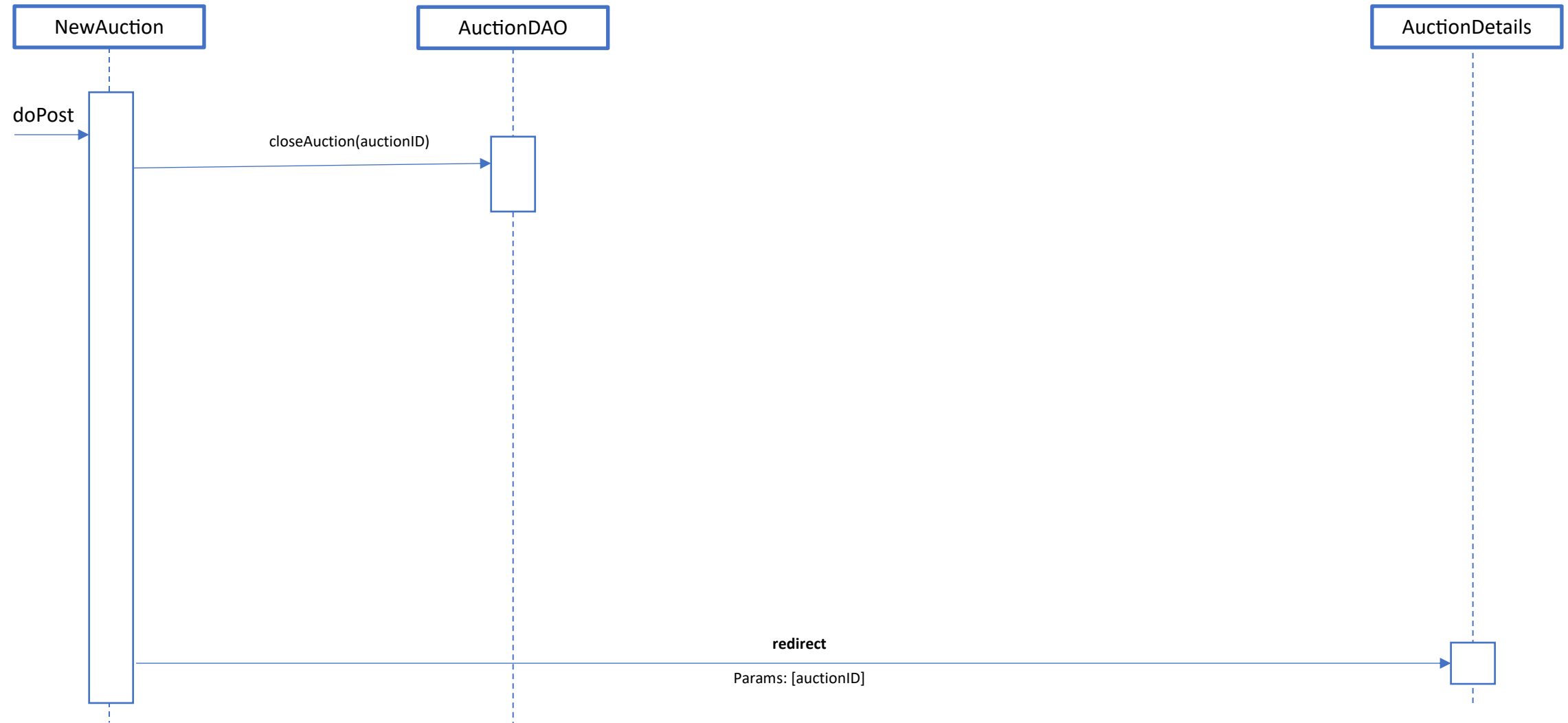
View: AuctionDetails

GET request from /Sell
Params: [auctionID]



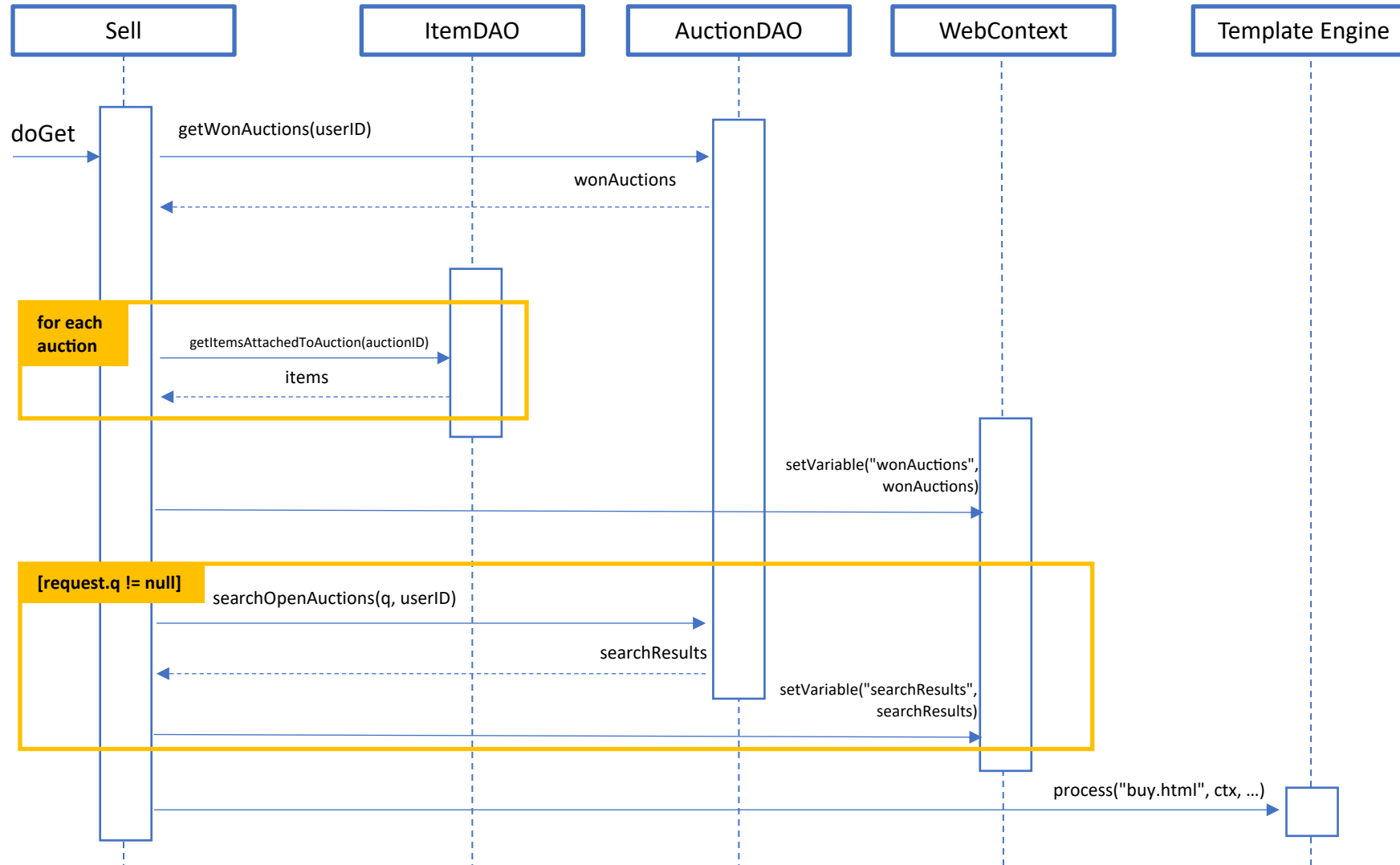
Event: close auction

POST request from **/AuctionDetails**
Params: [auctionID]



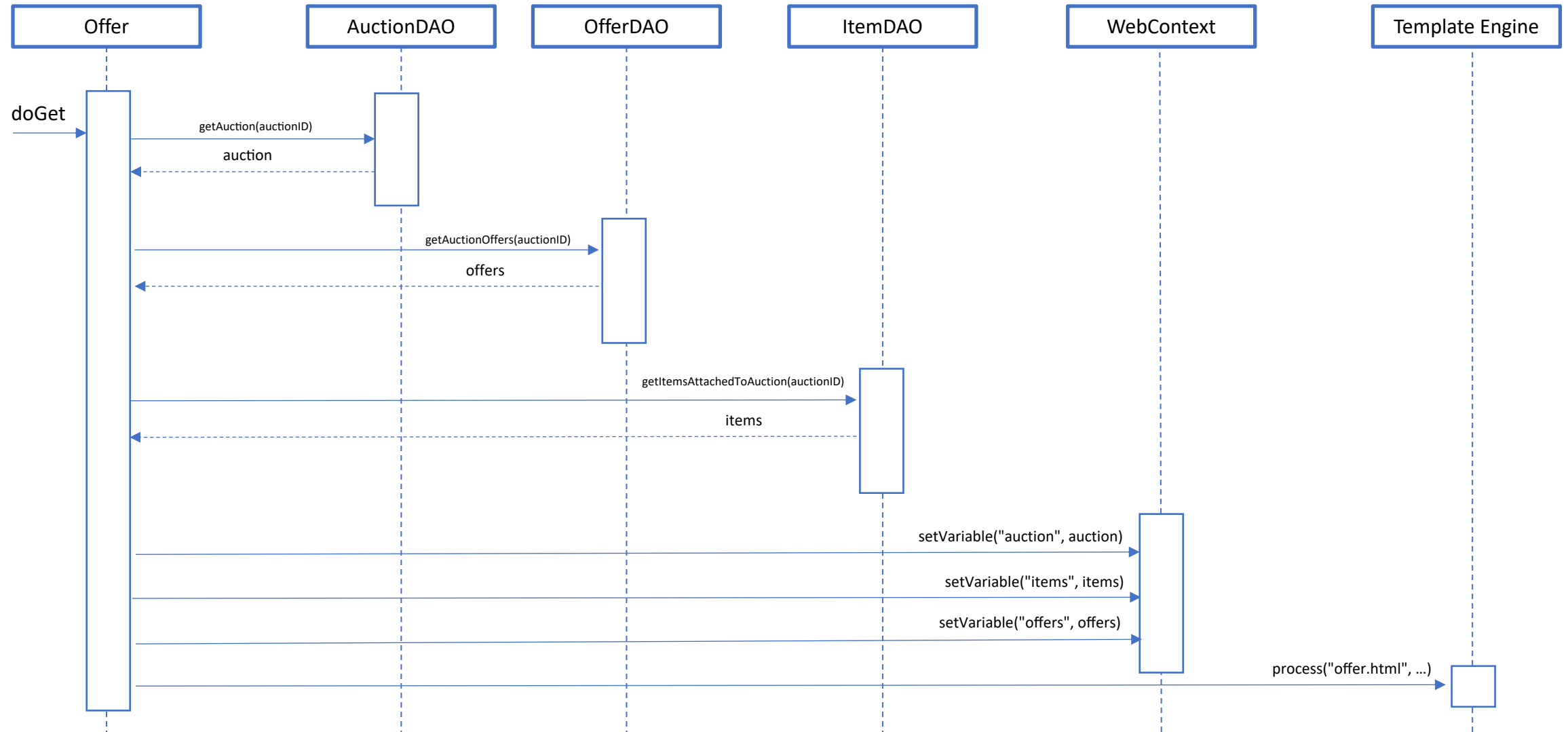
View: Buy

GET request
Params: [q * (search query)]



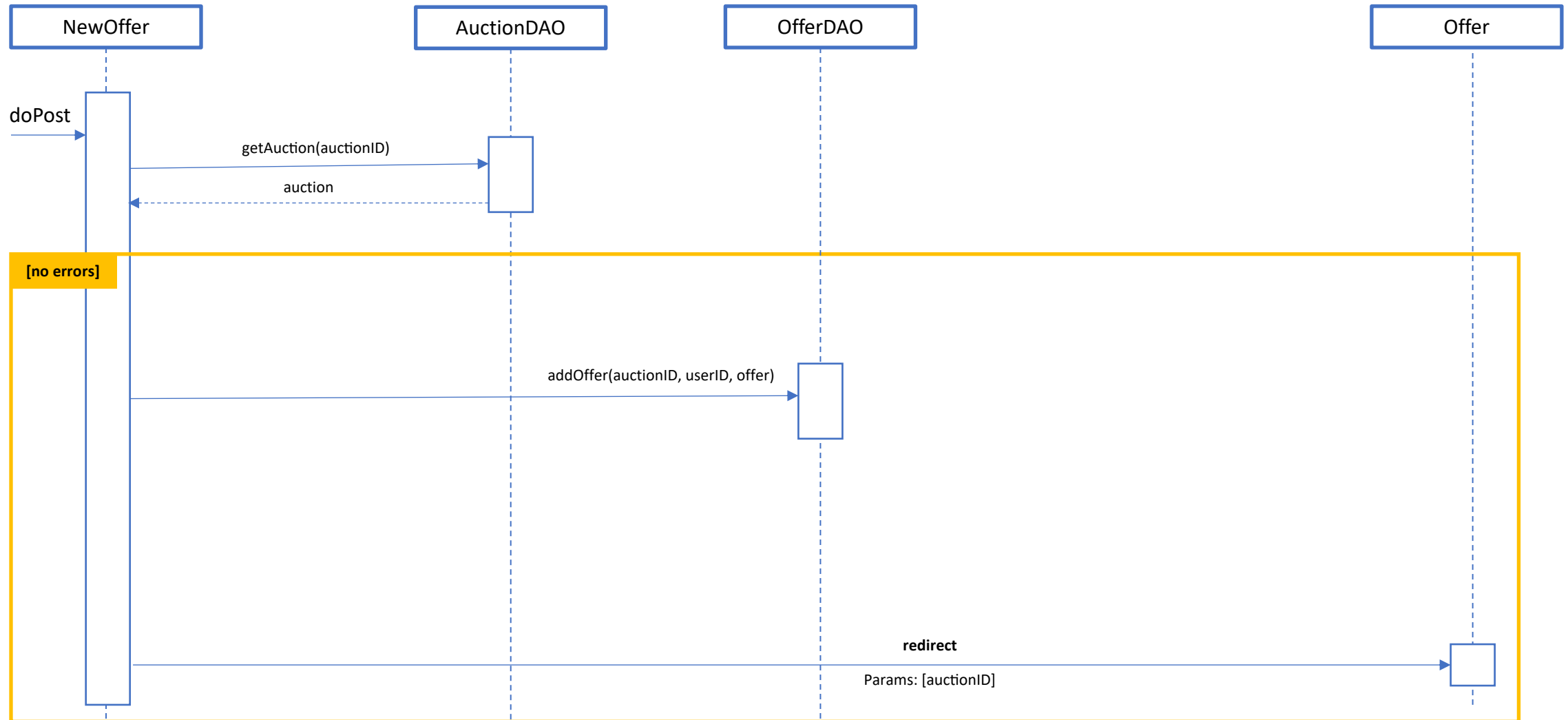
View: Offer

GET request from /Buy
Params: [auctionID]



Event: new offer

POST request from **/Offer**
Params: [auctionID, offer]



Versione Javascript

Specifiche aggiuntive

- 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.

❖ *La struttura della base di dati della versione Javascript è invariata.*

Componenti

- **Model objects** (JavaBeans)
 - User
 - Auction
 - Item
 - Offer
 - Error
- **Data Access Objects**
 - UserDao
 - getUserByID(id)
 - authenticateUser(usernameOrEmail, password)
 - createUser(user, password)
 - isUsernameTaken(username)
 - ItemDAO
 - addItem(item)
 - getItemsAvailableForAuction(userID)
 - attachItemToAuction(itemCode, auctionID)
 - getItemsAttachedToAuction(auctionID)
 - OfferDAO
 - getAuctionOffers(auctionID)
 - addOffer(auctionID, userID, price)
- AuctionDAO
 - newAuction(deadline, minimumRise, userID, itemsCodes[])
 - getUserAuctions(userID)
 - getAuction(auctionID)
 - closeAuction(auctionID)
 - searchOpenAuctions(query, userID)
 - getWonAuctions(userID)

Componenti

- **Controllers** (*servlets*)

- *Authentication*
 - PerformLogin
 - PerformSignup
 - GetUserDetails
 - Logout
- *Buy*
 - Search
 - MakeOffer
 - GetWonAuctions
 - GetAuctionDetails
- *Sell*
 - AddItem
 - GetAvailableItems
 - GetUserAuctions
 - NewAuction
 - CloseAuction
- GetImage

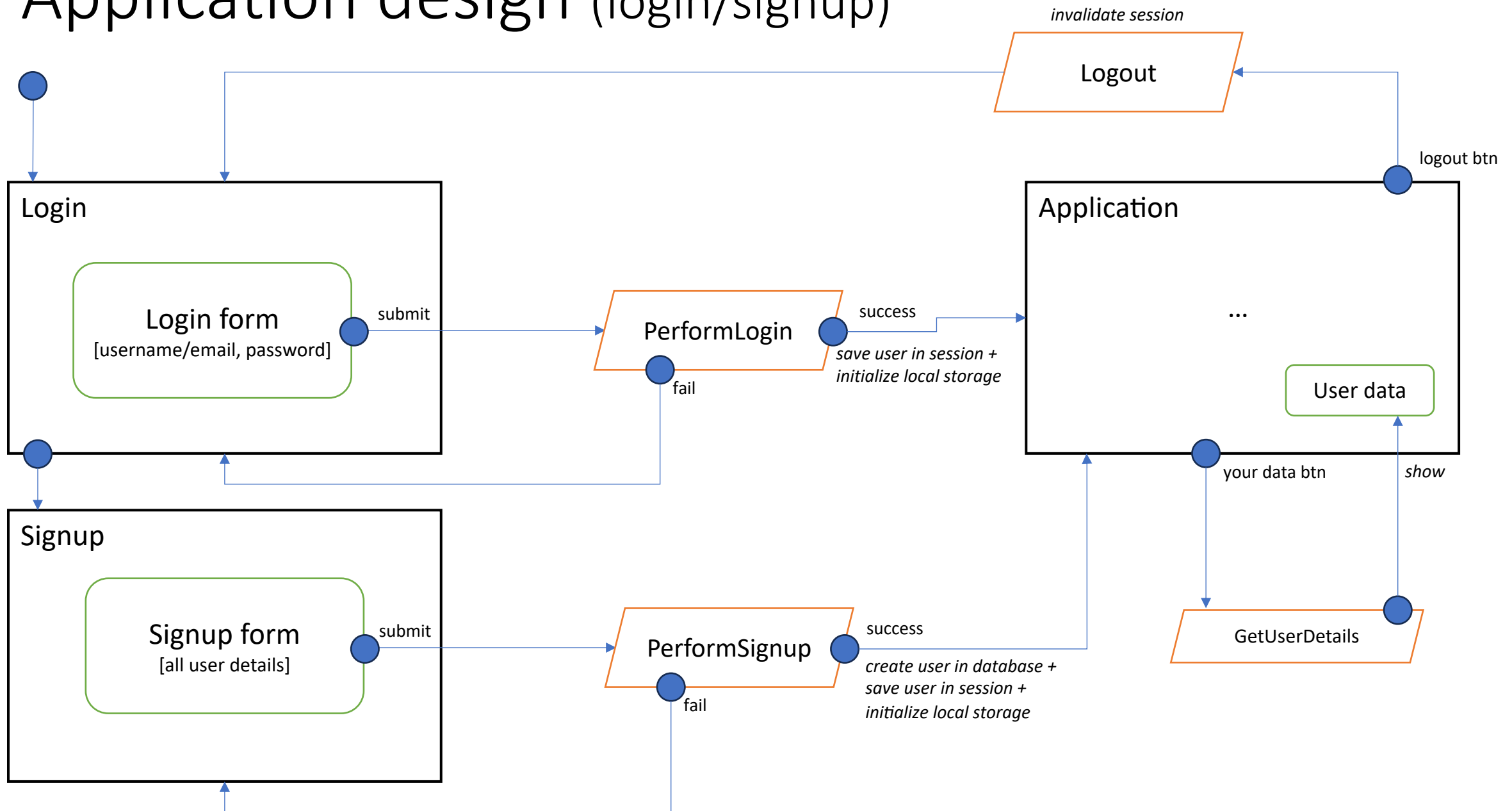
- **Views**

- Login (*index.html*)
- Signup (*signup.html*)
- Application (*app.html*)
 - UserDetails
 - Search
 - AuctionDetails
 - WonAuctions
 - **RecentlyViewed**
 - YourAuctions
 - AddItem
 - NewAuction

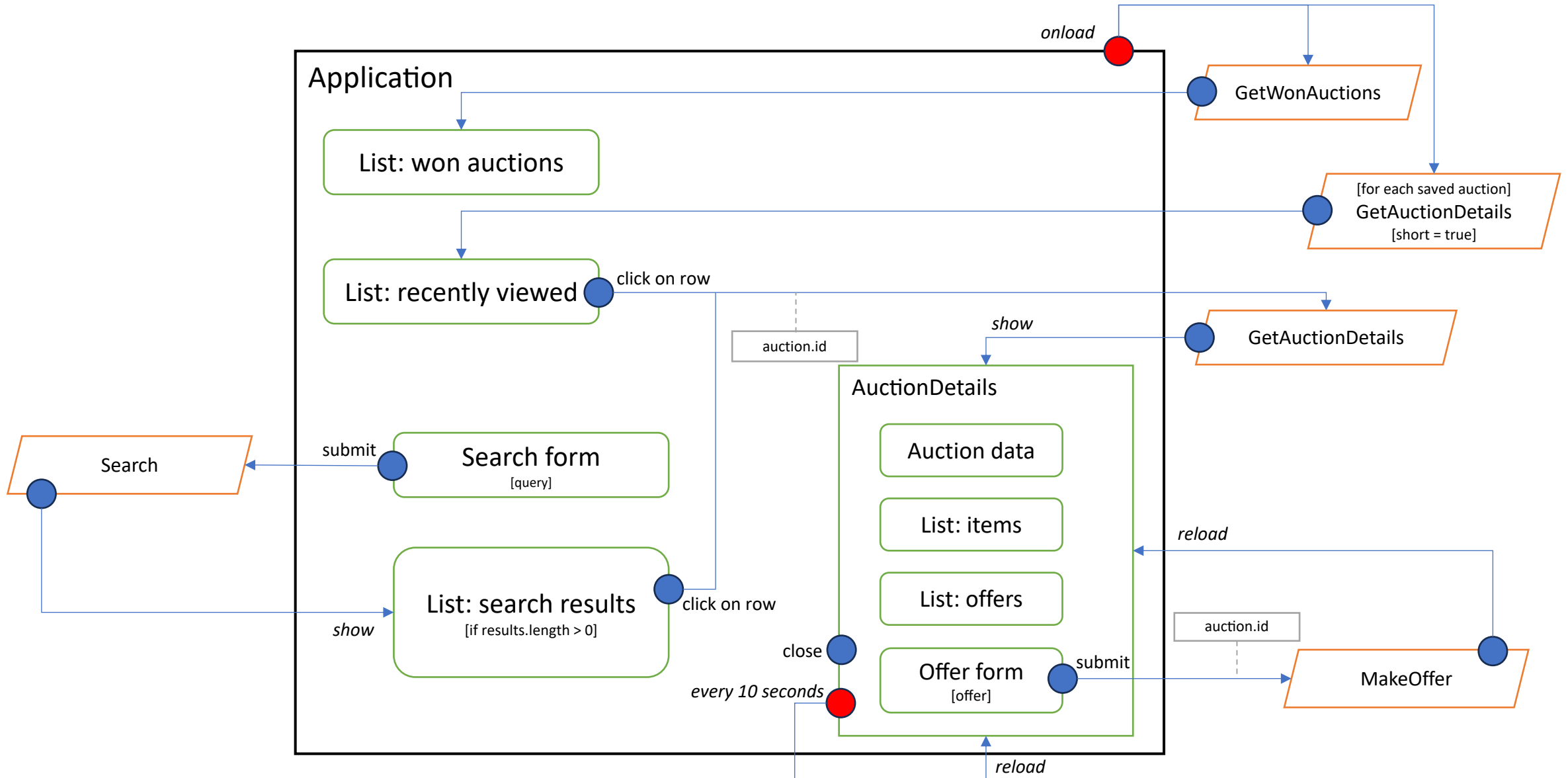
- *Classi di utilità*

- ConnectionHandler
- Constants
- FileManager
- Hash

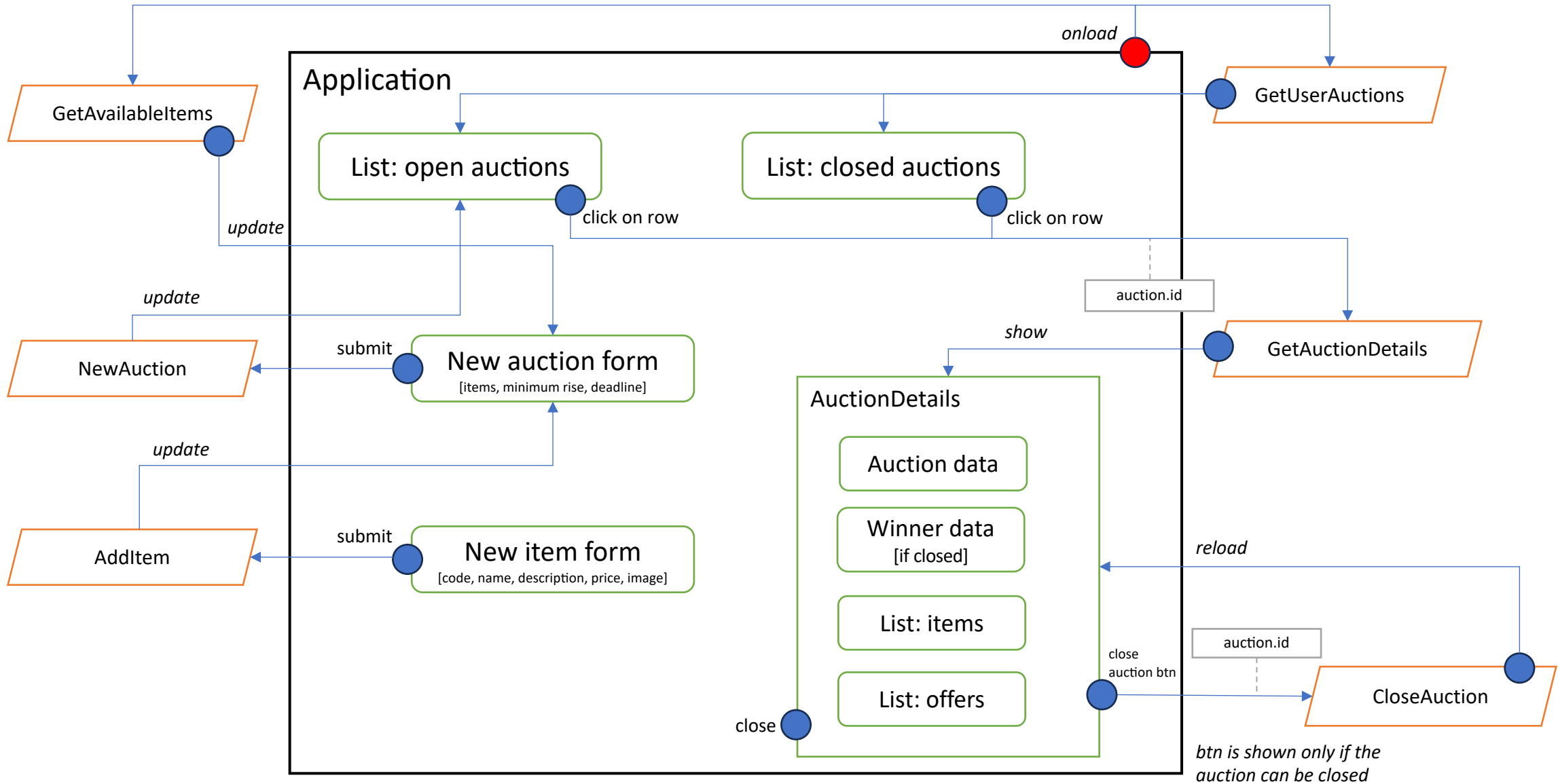
Application design (login/signup)



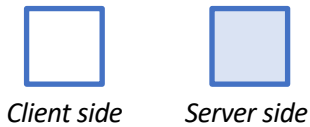
Application design (buy)



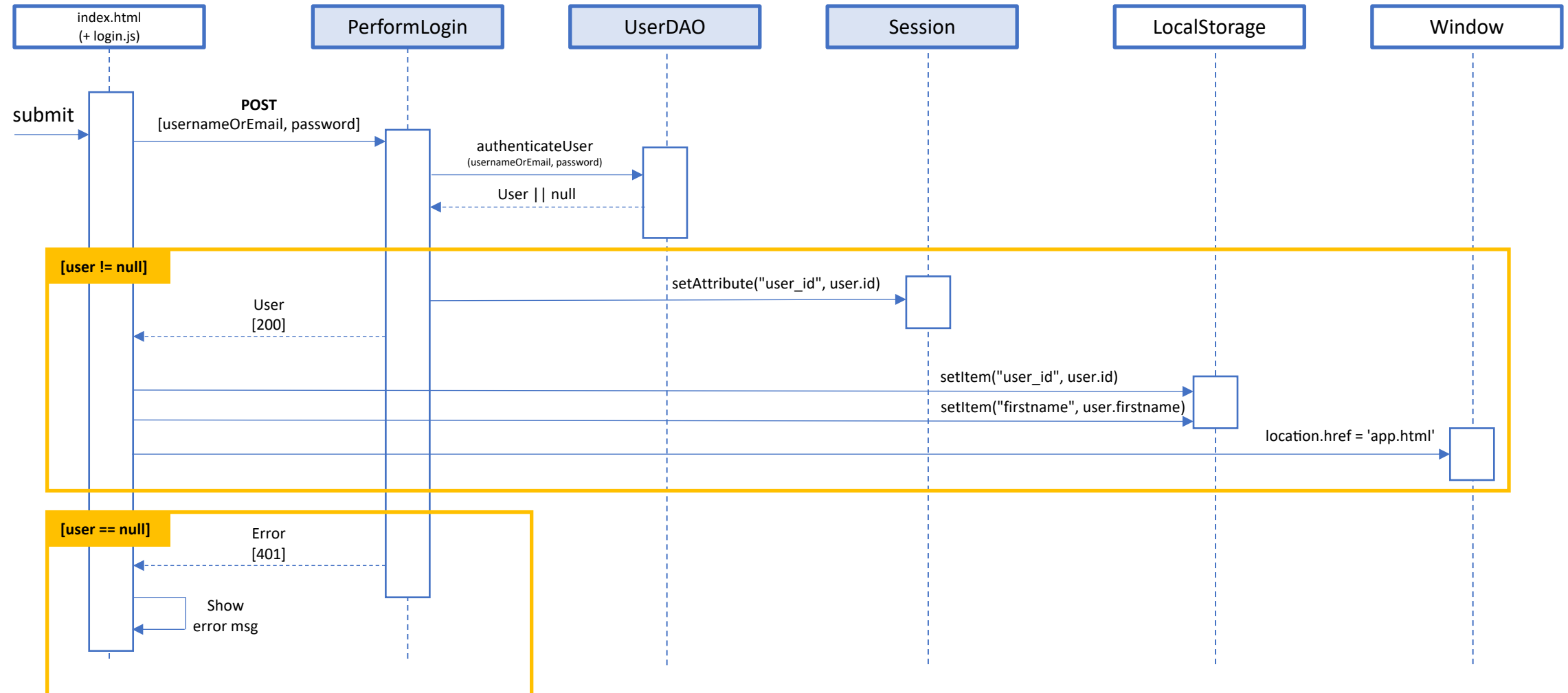
Application design (sell)



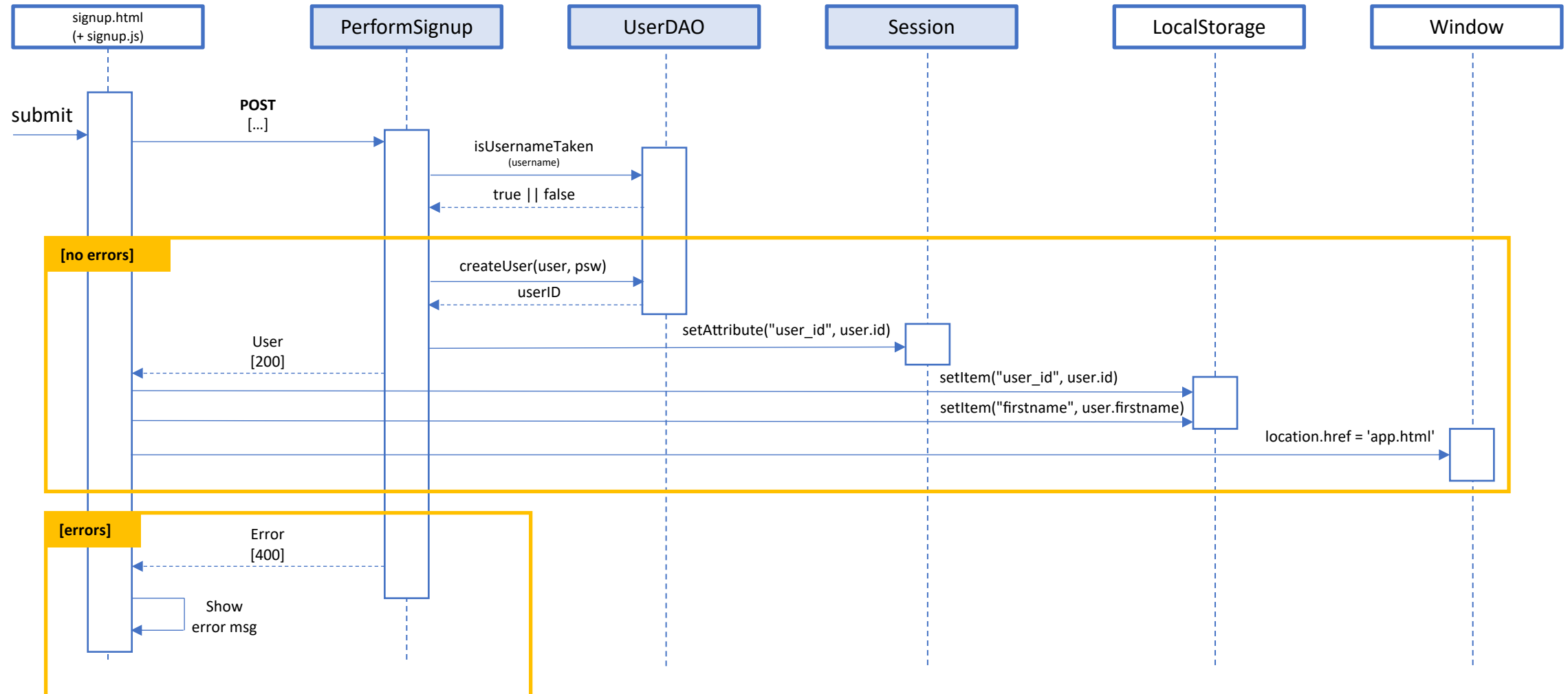
Event: login



Prima che venga aggiornato il *localStorage* a seguito di un login è necessario controllare che i dati salvati nel browser (ad es. aste recenti), se presenti, siano riferiti allo stesso utente che ha effettuato il login, e che non siano passati più di 30 giorni dall'ultimo aggiornamento dei dati. Se queste condizioni non sono verificate, allora è necessario fare *clear* del *localStorage* prima di proseguire.

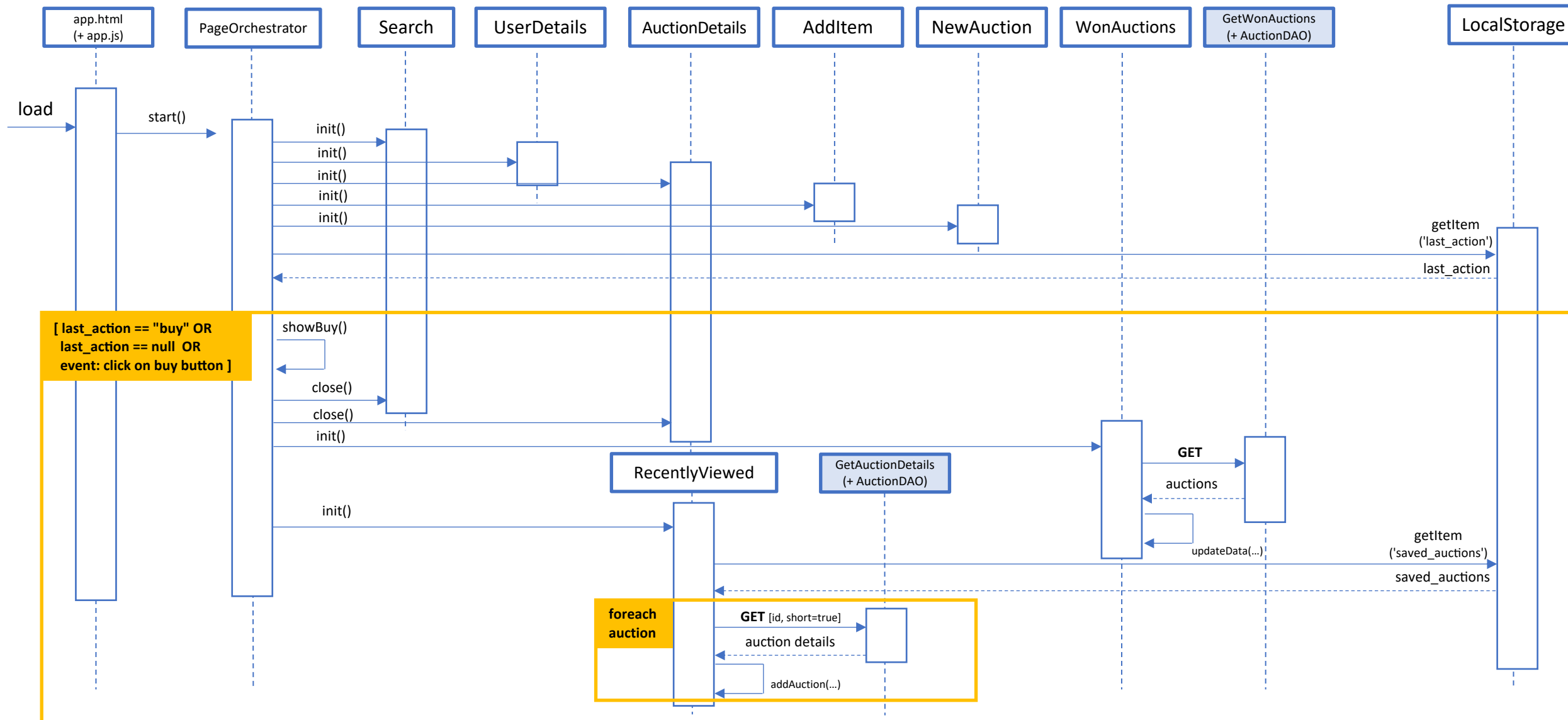


Event: signup

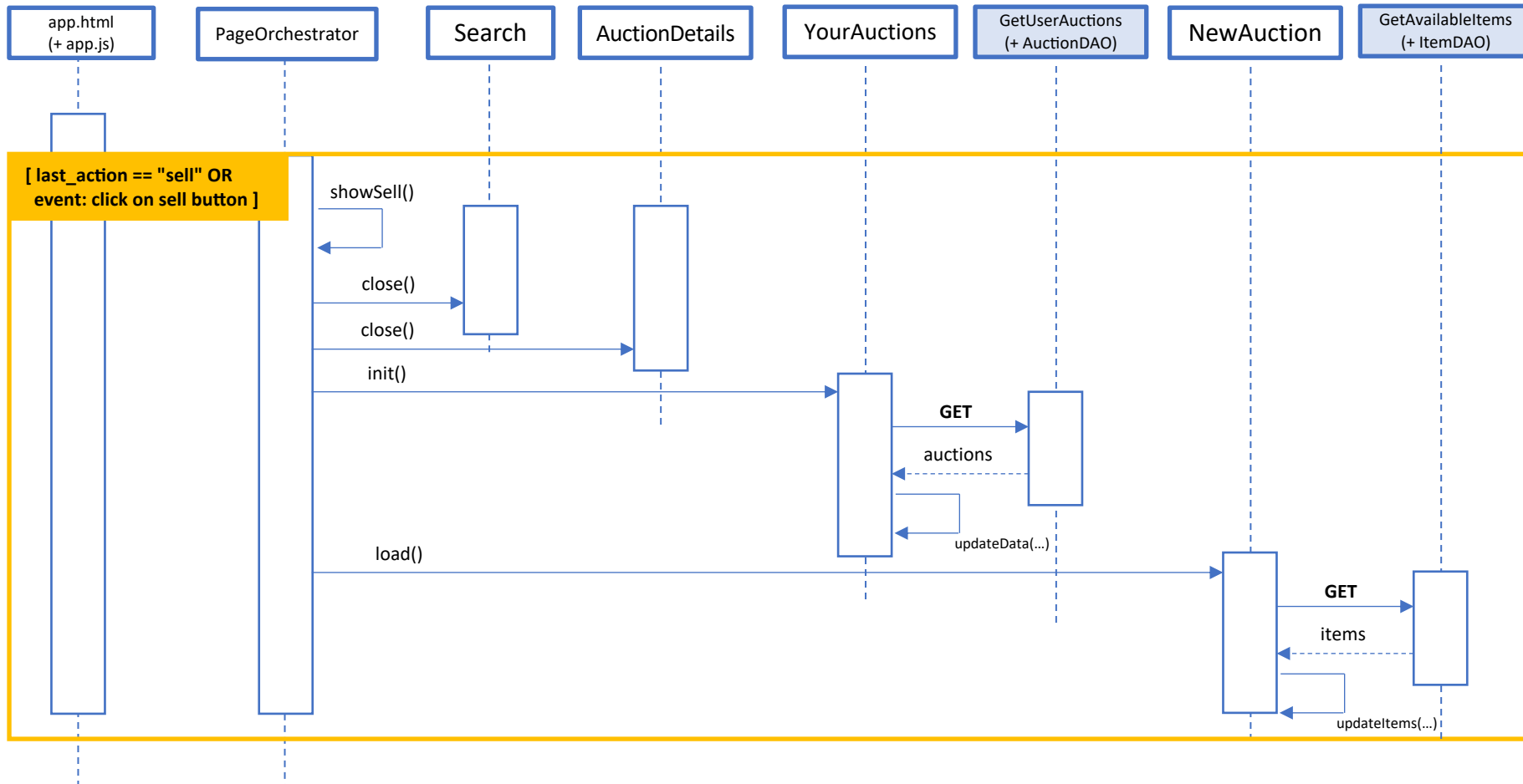


La funzione *init()* ha il compito di inizializzare il componente su cui viene chiamata, registrando gli opportuni *eventListener*.

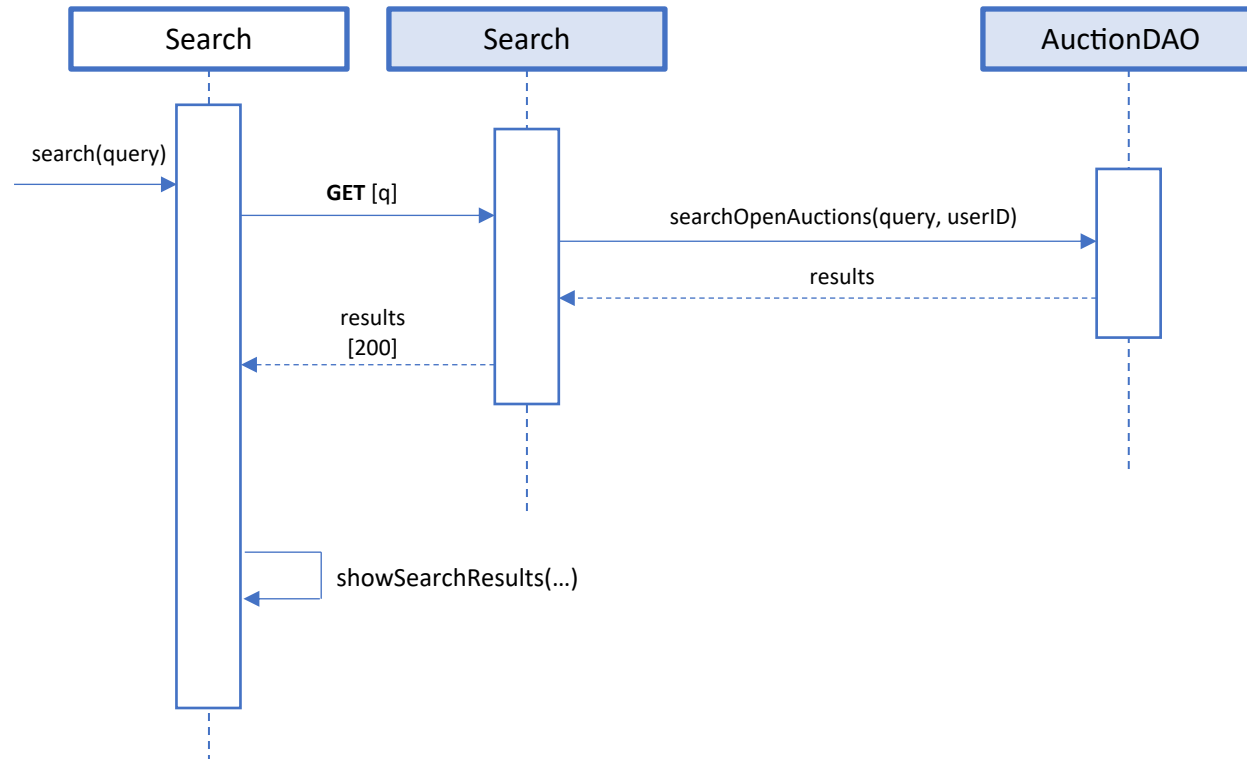
View: Application (*onload*, 1/2)



View: Application (*onload*, 2/2)

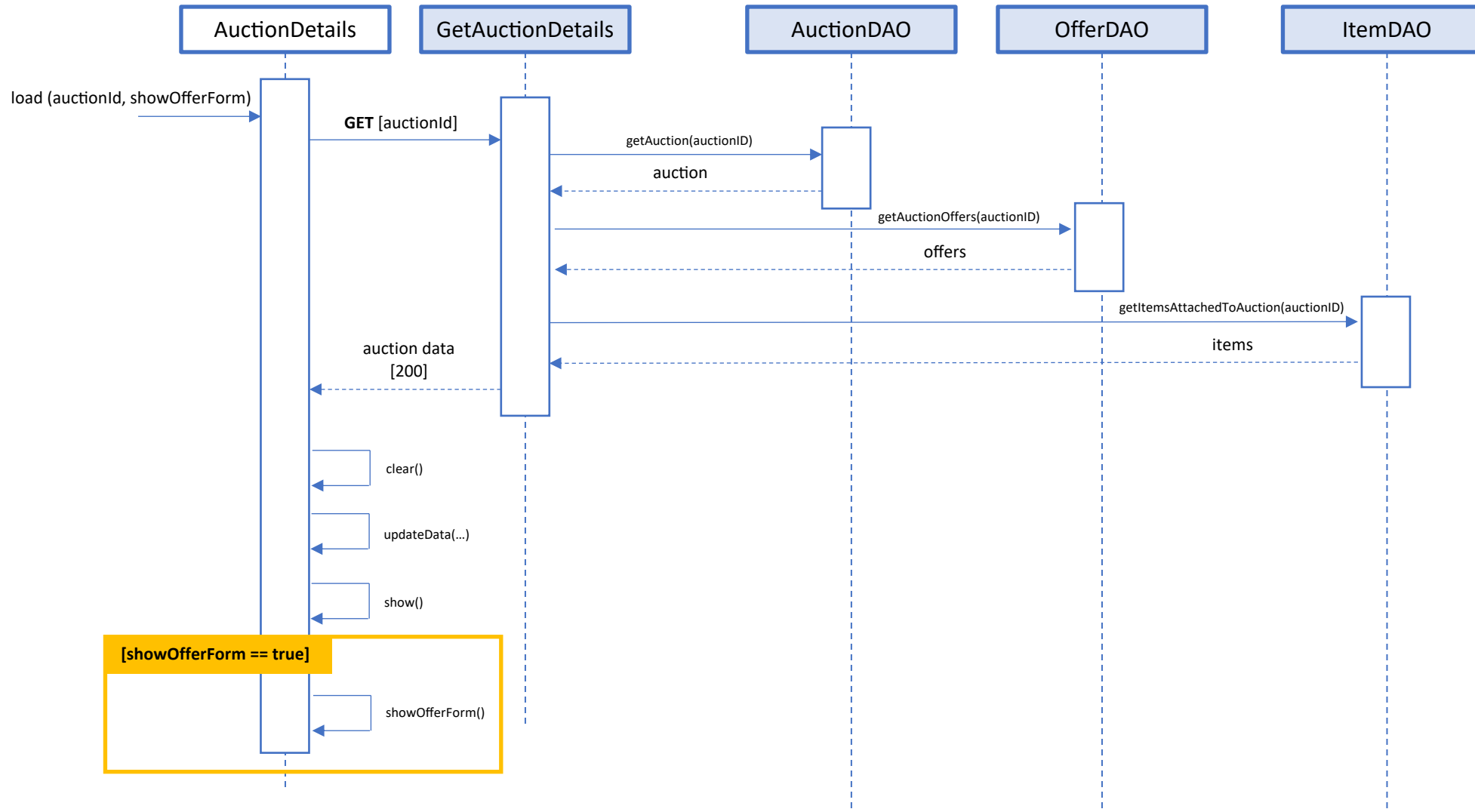


Event: search

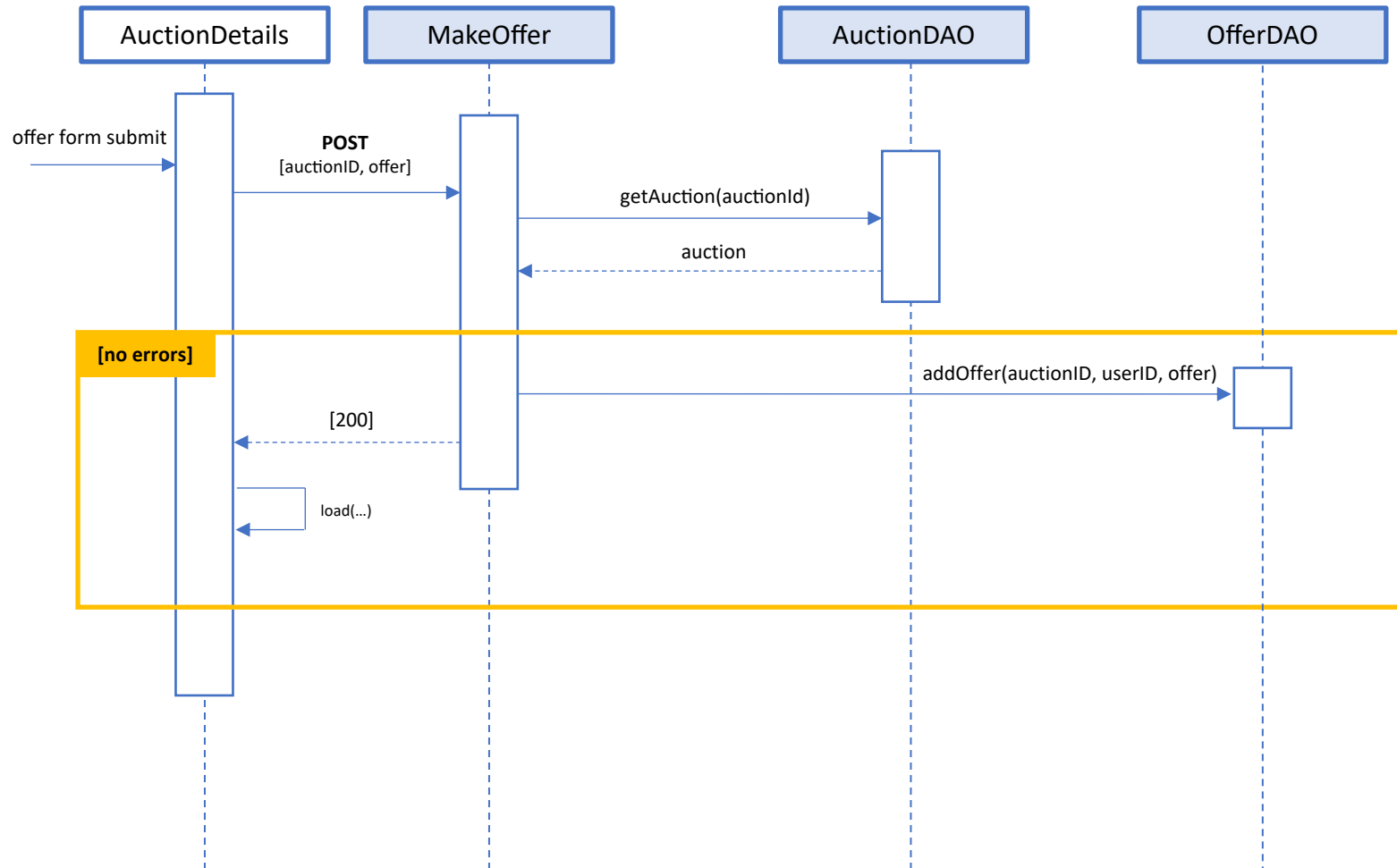


Event: click on auction

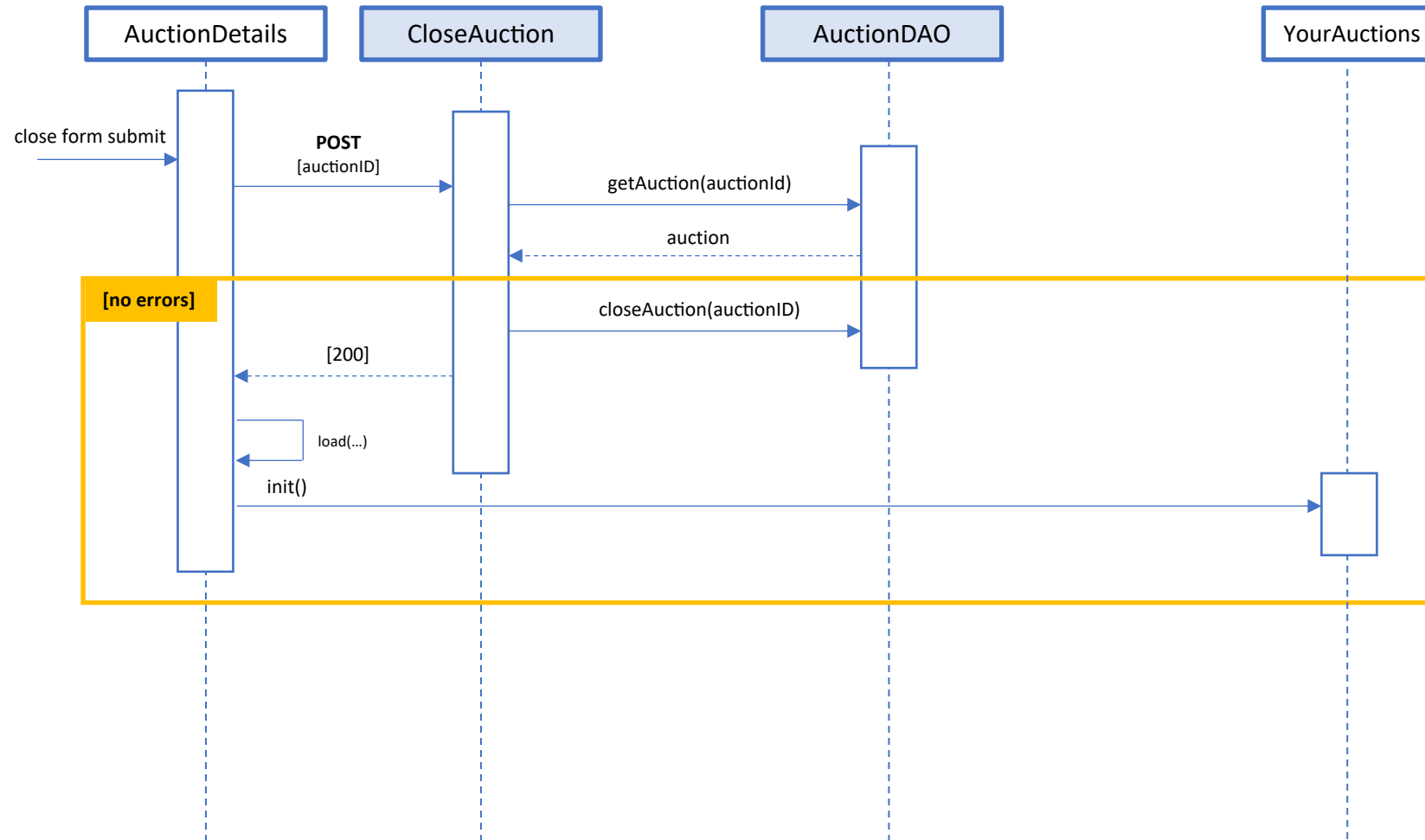
L'evento di *click* su un'asta in una qualsiasi lista è gestito dal componente **AuctionDetails**.
La *servlet* **GetAuctionDetails** si occuperà di verificare se l'utente è autorizzato a visualizzare i dati dell'asta richiesta.



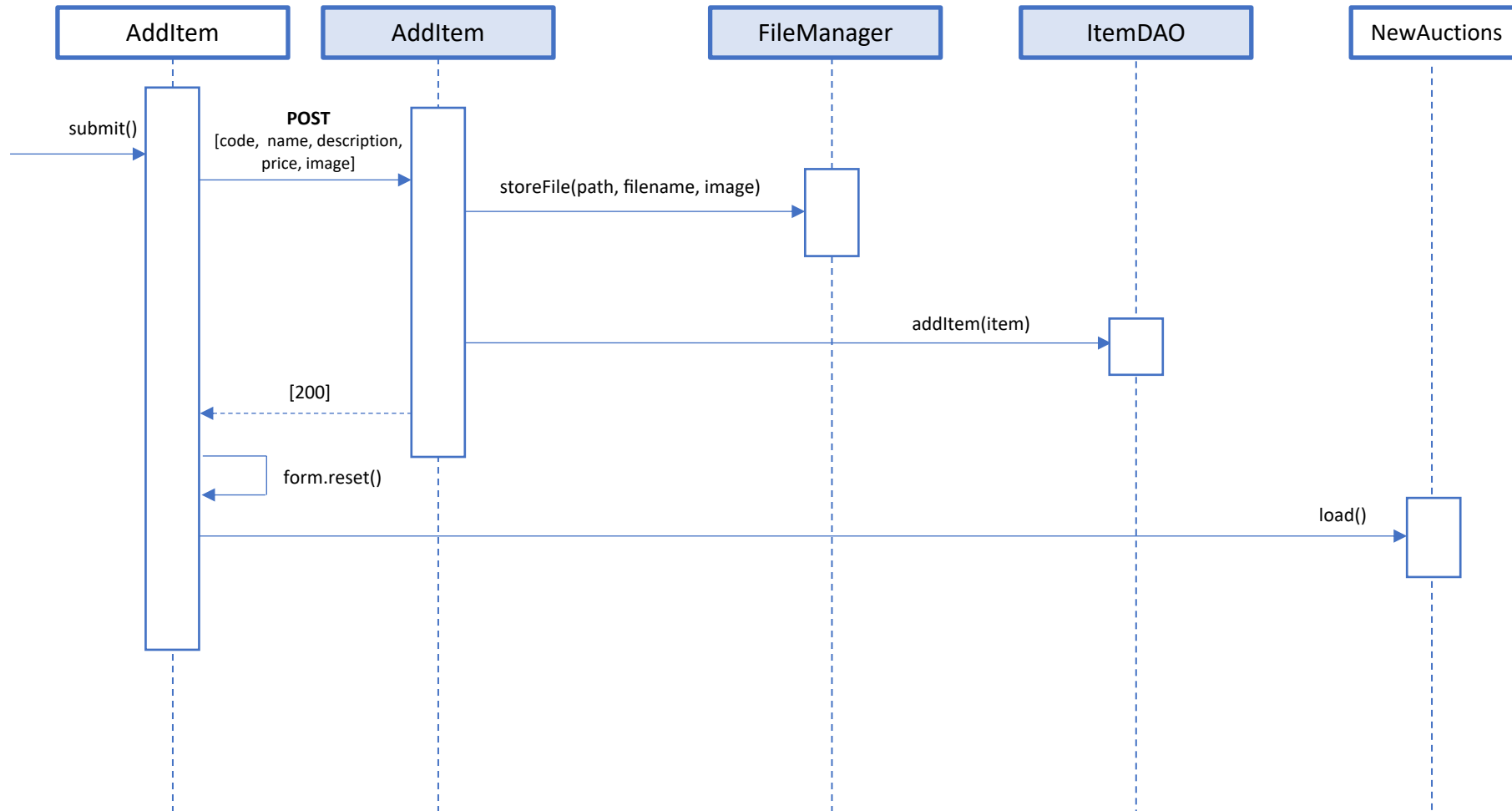
Event: new offer



Event: close auction



Event: new item



Event: new auction

