

# Sito di aste Online

Versione THIN CLIENT  
(HTML Pure)

STUDENTE : DEAN NICOLA

Mat:911817

[nicola.dean@mail.polimi.it](mailto:nicola.dean@mail.polimi.it)

# Specifiche(1/2)

## HTML pure

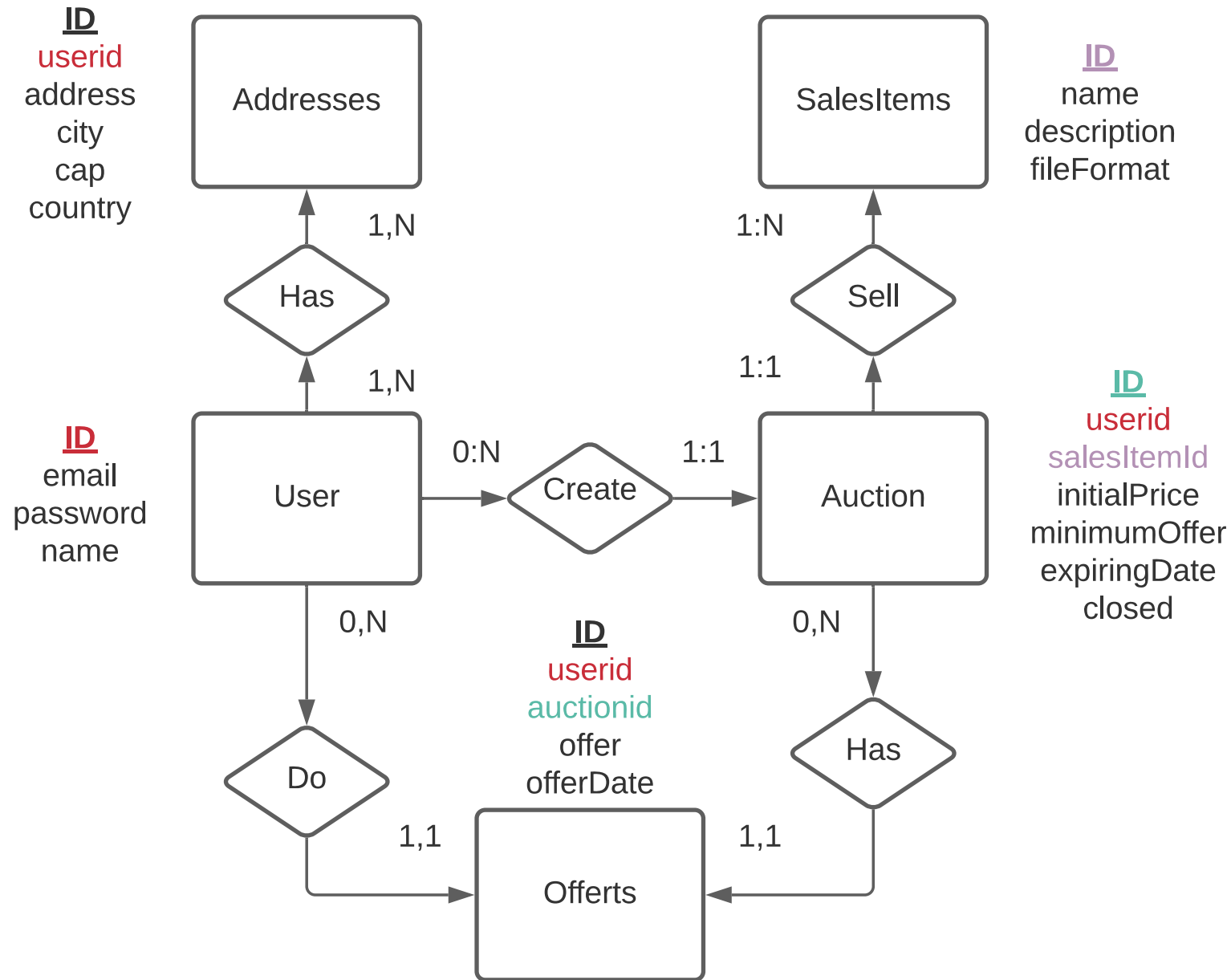
- Un'applicazione web consente la gestione di aste online. Gli utenti accedono tramite login e possono vendere e acquistare all'asta. La HOME page contiene due link, uno per accedere alla pagina VENDO e uno per accedere alla pagina ACQUISTO. La pagina VENDO mostra una lista delle aste create dall'utente e non ancora chiuse, una lista delle aste da lui create e chiuse e una form per creare un nuovo articolo e una nuova asta per venderlo. L'asta comprende l'articolo da mettere in vendita (codice, nome, descrizione, immagine), prezzo iniziale, 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). La lista delle aste è ordinata per data+ora crescente. L'elenco riporta: codice e nome dell'articolo, 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 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). Se l'asta è chiusa, la pagina riporta 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) il cui articolo contiene la parola chiave nel nome o nella descrizione. 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 dell'articolo, 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 dell'articolo e il prezzo finale.

# Analisi Dati Per Database

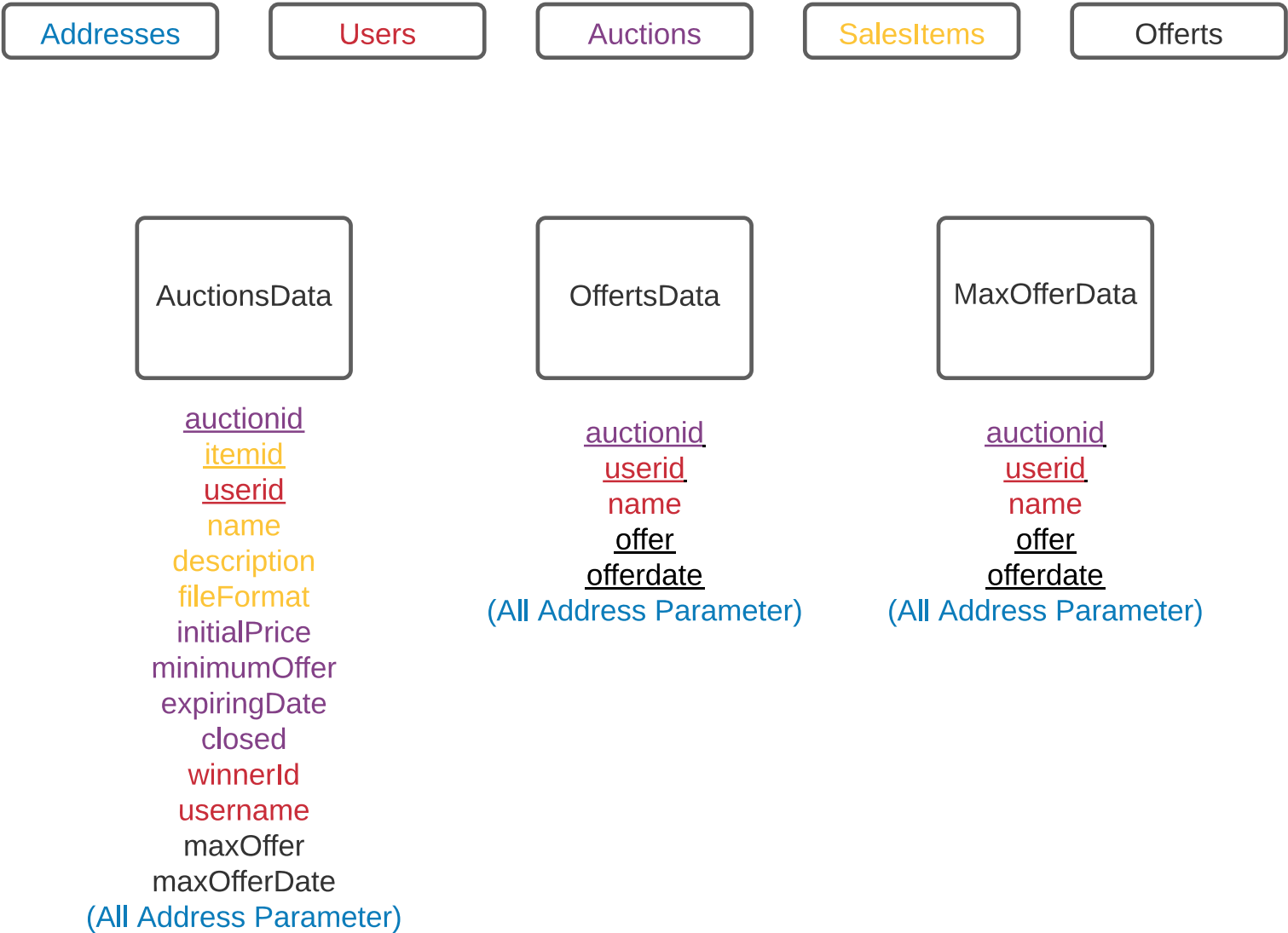
- Un'applicazione web consente la gestione di **aste** online. Gli **utenti** accedono tramite login e possono vendere e acquistare all'asta. La HOME page contiene due link, uno per accedere alla pagina VENDO e uno per accedere alla pagina ACQUISTO. La pagina VENDO mostra una lista delle **aste create** dall'utente e non ancora chiuse, una lista delle aste da lui create e chiuse e una form per creare un nuovo articolo e una nuova asta per venderlo. L'asta comprende **l'articolo** da mettere in vendita (**codice, nome, descrizione, immagine**), **prezzo iniziale, 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)**. La lista delle aste è ordinata per data+ora crescente. L'elenco riporta: codice e nome dell'articolo, 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 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). Se l'asta è chiusa, la pagina riporta 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) il cui articolo contiene la parola chiave nel nome o nella descrizione. 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 dell'articolo, 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 dell'articolo e il prezzo finale.

- **Entità,Attributi,Relazioni**

# Diagramma Tabelle Database



# Diagramma View Database



```
CREATE TABLE users(  
    id int NOT NULL AUTO_INCREMENT,  
    email      varchar(20),  
    password   varchar(20),  
    name       varchar(20),  
    PRIMARY KEY(id)  
);
```

```
CREATE TABLE salesItems(  
    id int NOT NULL AUTO_INCREMENT,  
    name      varchar(20),  
    description varchar(100),  
    fileFormat varchar(10),  
    PRIMARY KEY(id)  
);
```

```
CREATE TABLE auctions(  
    id int NOT NULL AUTO_INCREMENT,  
    userid      int,  
    salesItemid int,  
    initialPrize int,  
    minimumOffer int,  
    expiringDate datetime,  
    closed      boolean,  
    PRIMARY KEY(id),  
    FOREIGN KEY (userid) REFERENCES users(id)  
    ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (salesItemid) REFERENCES salesitems(id)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE offerts(  
    id int NOT NULL AUTO_INCREMENT,  
    userid          int,  
    auctionsid      int,  
    offer           int,  
    offerDate       TIMESTAMP,  
    PRIMARY KEY(id),  
    FOREIGN KEY (userid) REFERENCES users(id)  
ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (auctionsid) REFERENCES auctions(id)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
create table addresses(  
    id int NOT NULL AUTO_INCREMENT,  
    userid          int,  
    address         varchar(20),  
    city            varchar(20),  
    cap             int,  
    country         varchar(3),  
    PRIMARY KEY(id),  
    FOREIGN KEY (userid) REFERENCES users(id)  
ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```

create view offertsData as
(
    select auctionsid,offers.userid,name,offer,offerDate,addresses.address,addresses.city,addresses.cap,addresses.country
    from (offers join users
    | | | | on offers.userid = users.id) join addresses
    on users.id = addresses.userid
);

create view maxOfferData as |
(
    select *
    from offertsData
    where offer = (select max(offer) from offertsData as 0 where 0.auctionsid = offertsData.auctionsid)
);

drop view if exists auctionsdata;
create view auctionsData as
(
    select auctions.id as auctionid,salesitems.id as itemId,auctions.userid,salesitems.name as name,description,initialPrize,minimumOffer,maxoffe
    from (auctions join salesitems on auctions.salesItemId = salesitems.id) left join maxOfferData
    on auctions.id = maxOfferData.auctionsid
);

```



# Analisi Requisiti Applicazione

- 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 una form per creare un nuovo articolo e una nuova asta per venderlo. L'asta comprende l'articolo da mettere in vendita (codice, nome, descrizione, immagine), prezzo iniziale, 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). La lista delle aste è ordinata per data+ora crescente. L'elenco riporta: codice e nome dell'articolo, 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 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). Se l'asta è chiusa, la pagina riporta 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) il cui articolo contiene la parola chiave nel nome o nella descrizione. 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 dell'articolo, 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 dell'articolo e il prezzo finale.
- **Pagine**, Componenti visivi, Eventi/Azioni

# Completamento Specifiche:

- La pagina di default è index.html e contiene il form di Login
- Tutti i dati del form di creazione di un asta sono obbligatori e se nulli verrà stampato un errore adeguato
- In caso l'utente capiti per sbaglio o malintenzionatamente sulla pagina di un asta chiusa non da lui posseduta gli sarà impedito di CHIUDERLA
- Una volta inviata un offerta la pagina di dettaglio si aggiorna in automatico
- Se l'utente è loggato e apre index.html / Login verrà reindirizzato alla home
- Se l'utente non è loggato e apre home.html (o derivati) verrà reindirizzato al Login

# Note Aggiunte/Aggiunte

- Per migliorare la visualizzazione/esperienza utente le aste vinte vengono visualizzate nella «UserPage» anziché nella pagina «Compro»
- La tabella Addresses è stata aggiunta per potenziali aggiornamenti futuri che permettano all'utente di avere più indirizzi (come opzione di consegna)
- La tabella ItemSales è separata dalla tabella Auctions per potenziali aggiornamenti futuri che permettano ad un utente di creare più aste con lo stesso oggetto (eg un artigiano produce in serie 100 statuine e vuole venderne 10 all'asta)

## Models

- Address
- User
- Auction
- ItemSales
- Offer
- UserProfileData

## Thymeleaf Template

- Index.html
- Header.html
- errorTemplate.html
- auctionCreation.html
- auctionDetails.html
- auctions.html
- sellerPage.html
- home.html

## Filters

- LoggedFilter
- NoLoggedFilter

# Server Side Components

## Database Access OBJ (DAO)

- UserDao
  - checkUserLogin(email,psw) : user
- AuctionDao (all function use auction select)
  - auctionSelect(QUERY string) : list<Auction>
  - getAuctionById(id) Auction
  - getOpenAuction(userID) list<Auction>
  - getClosedAuction(userID) list<Auction>
  - getWincedAuction(userID) list<Auction>
  - getAuctions() list<Auction>
  - createAuctions(name,desce...) void
- OfferDao
  - OfferSelect(query string) List<Offert>
  - createOffer(userID,auctionId,offer) ) List<Offert>
  - getOffertsById(auctionId) ) List<Offert>
  - getMaxOffert(auctionId) ) List<Offert>
- ItemSalesDao
  - createSalesItem(name,desc...)
  - getSalesItem(id) ItemSales

## Controllers (not Logged Filter)

- CheckLoginData

## Controllers (Logged Filter)

- CreateAuction
- CreateOffer
- ShowAuctions
- ShowAuctionDetails
- GetUserProfileData
- ImageGetter
- LogoutUser
- CloseAuction
- GoToSellerPage
- GoToHomePage
- GoToCreationPage

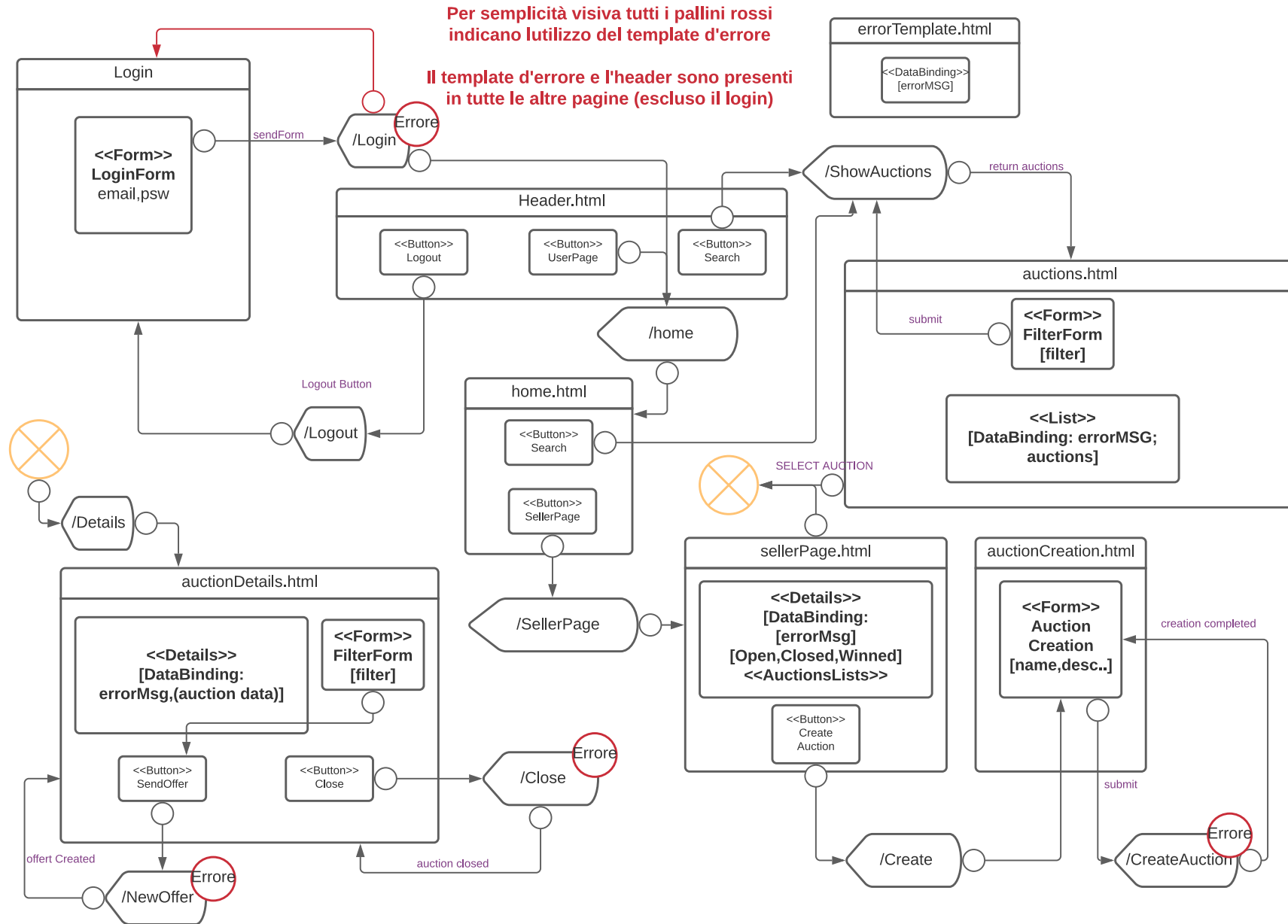
# Architettura

- Ogni controller all'interno dell'App estende la classe «BasicThymeleafServerlet» o «BasicThymeleafServerletSQL»
- Entrambe le classi hanno implementato un metodo chiamato `templateRenderer(...)` e uno `setError (...)` che permettono di generare comodamente **e senza codice ripetuto** un template thymeleaf (o un eventuale errore)
- `BasicThymeleafServerletSQL` è uguale alla prima solamente che possiede pure un'inizializzazione autonoma del database (**anche qui evita ripetizioni di codice**)

# Life Cycle

- Tutti le pagine vengono generate da Thymeleaf Template Engine
- La pagina è dinamicamente caricata utilizzando gli attributi settati grazie a Thymeleaf
- Anche gli errori vengono caricati con la stessa tecnica usando `ErrorTemplate`
- `Tutte le pagine` contengono `Header.html` e `ErrorTemplate.html`
- `ErrorTemplate.html` ha un attributo `thymeleaf` chiamato «`errorMsg`»

# Diagramma Applicativo IFML



# Sequence Diagram

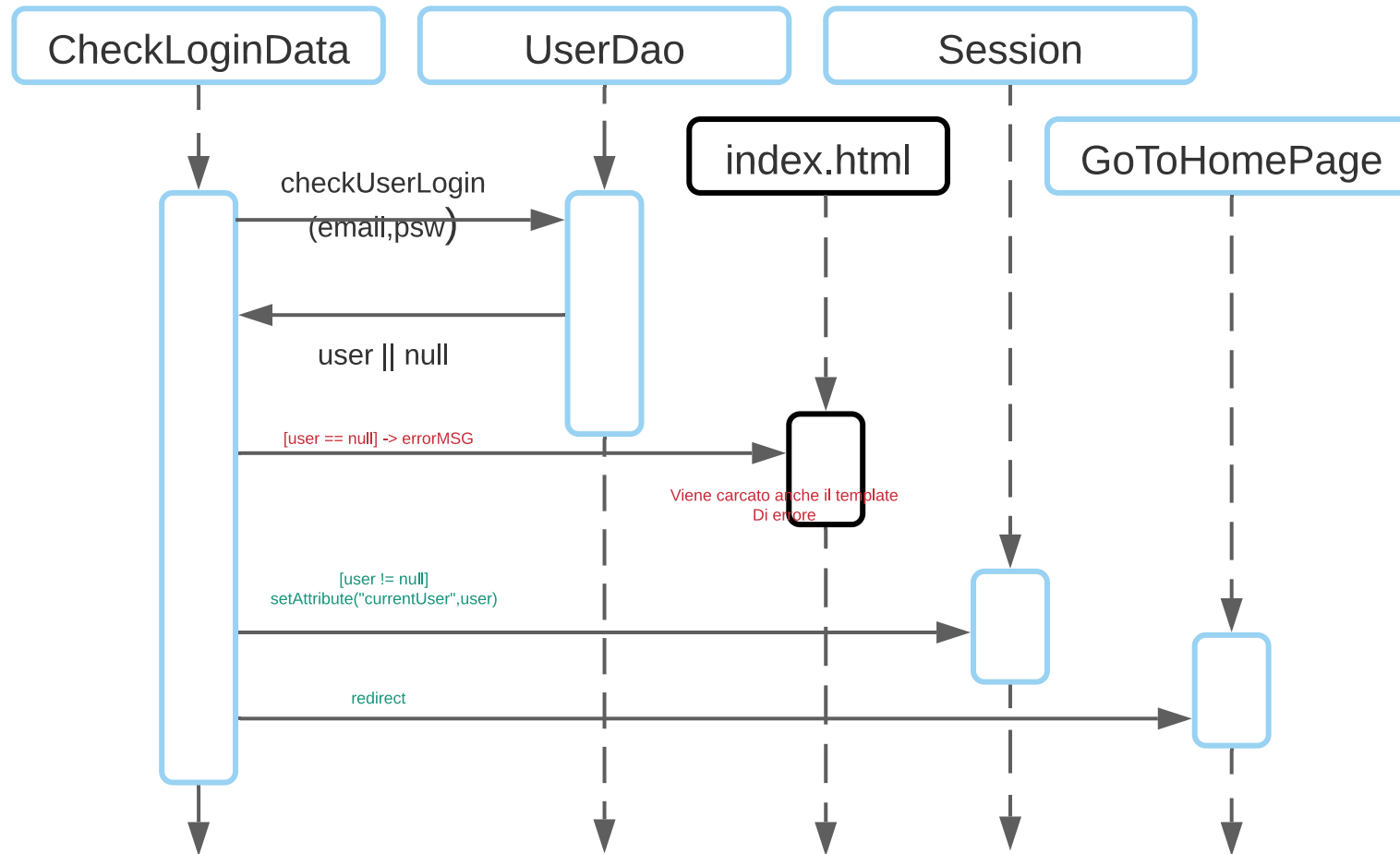
Nelle prossime pagine sono presenti i sequence diagram delle diverse possibili azioni percorribili dall'utente.

Alcuni dettagli minori quali la visualizzazione di alcuni errori o il caricamento di alcuni dati dalla sessione sono omessi per semplicità rappresentativa (sono comunque presenti delle note in viola che evidenziano tali omissioni).

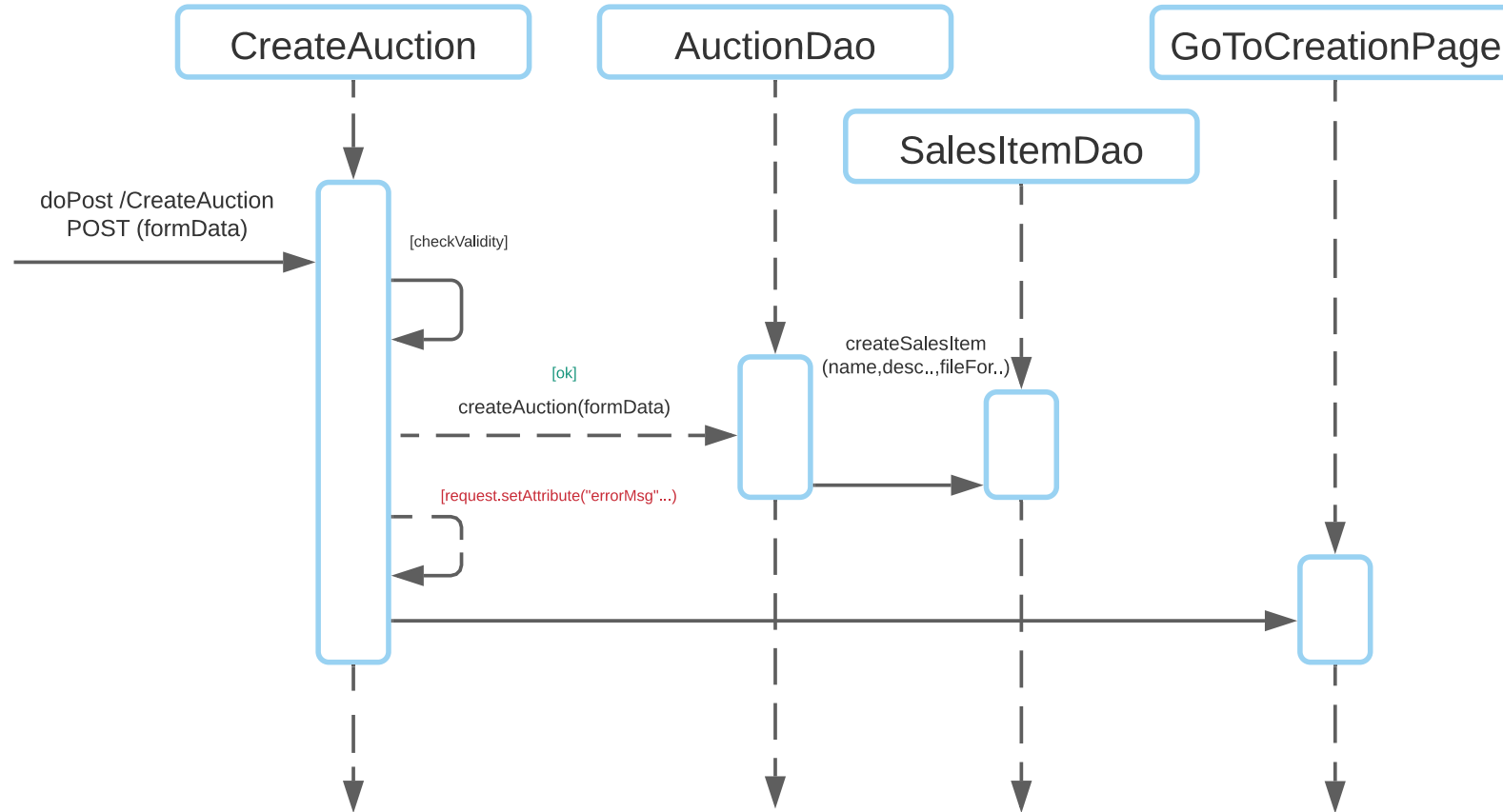
In Nero si troveranno le azioni ClientSide mentre in azzurrino quelle ServerSide



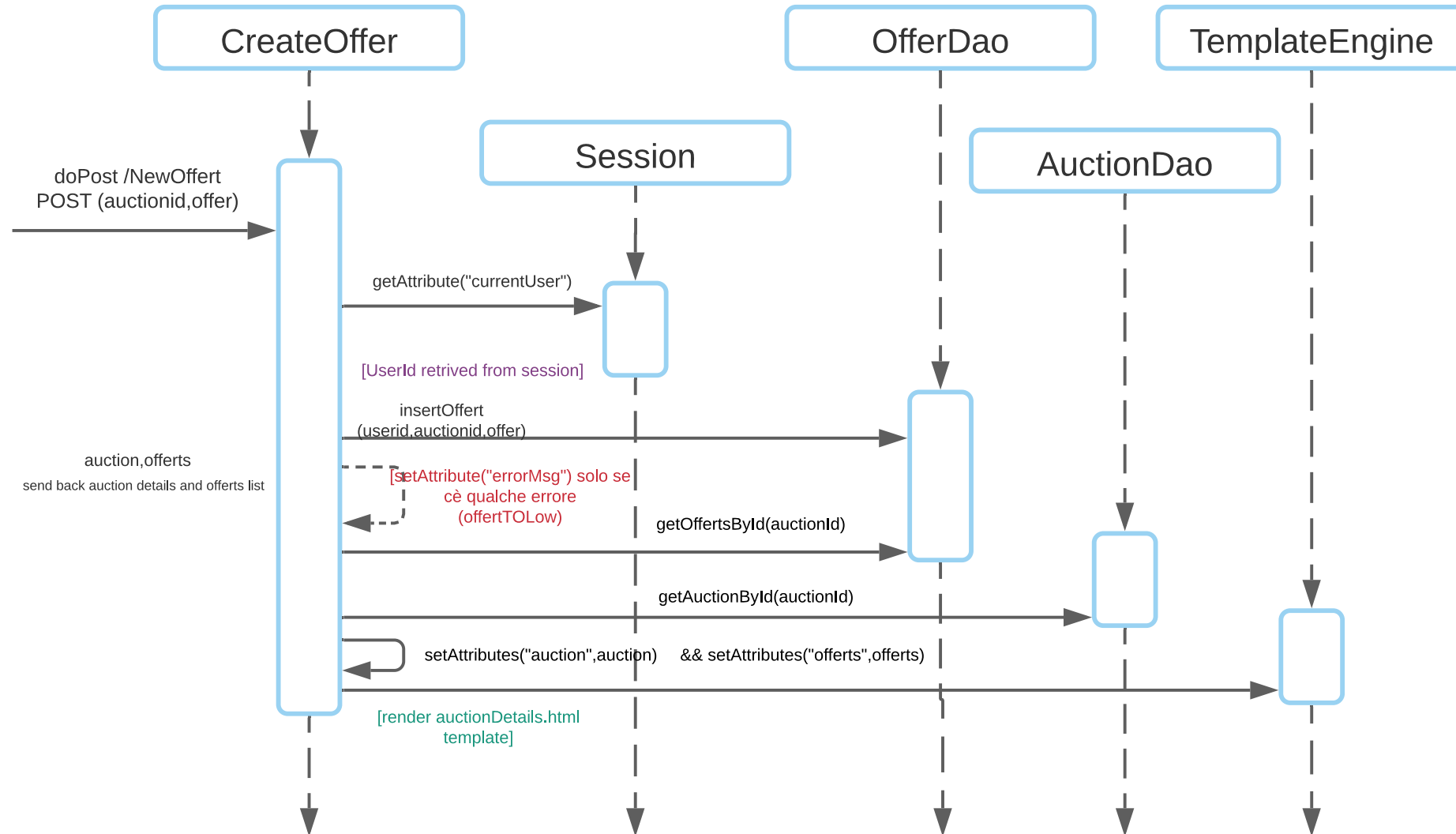
# Evento Di Login



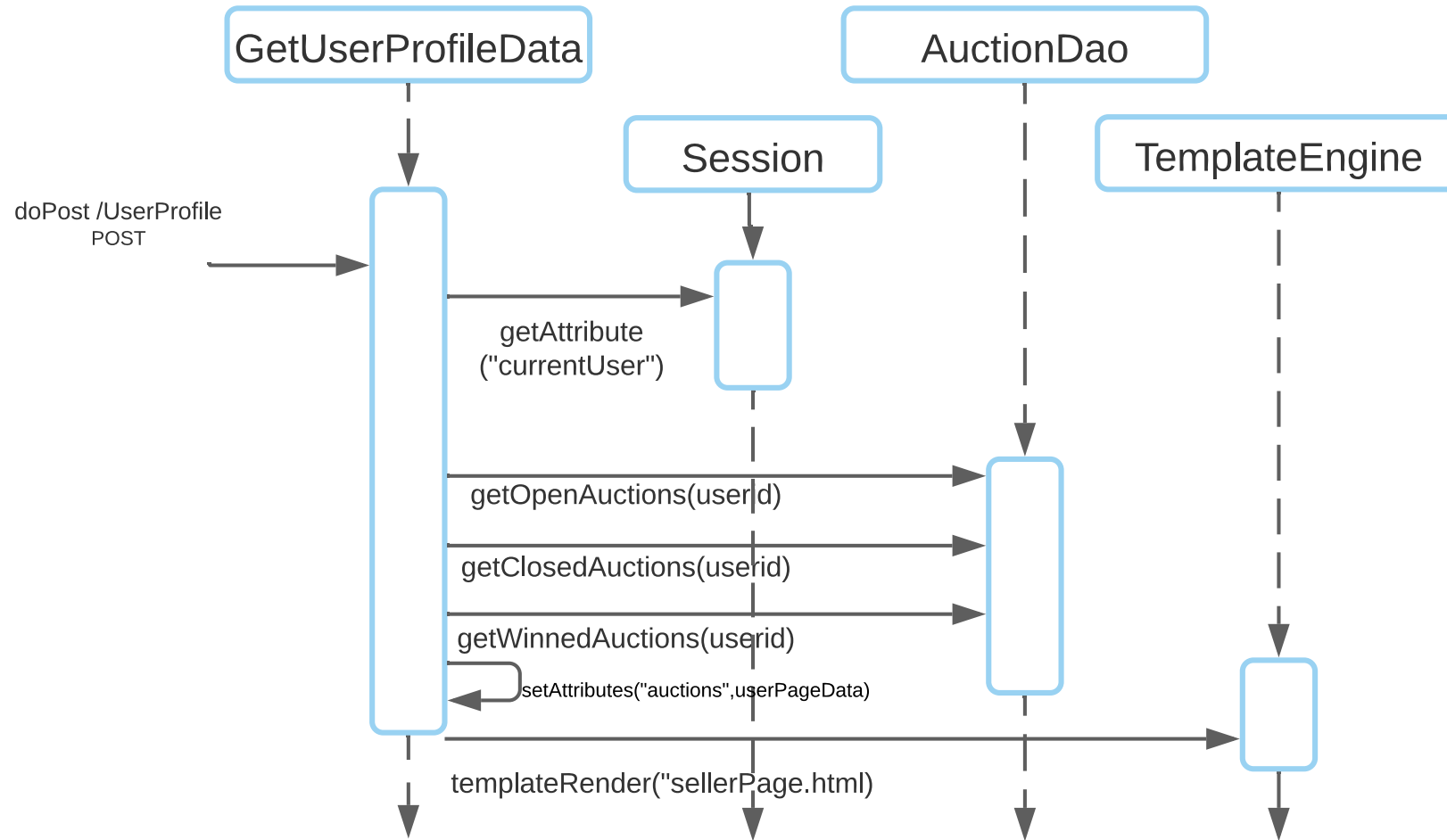
# Evento Di Creazione Asta



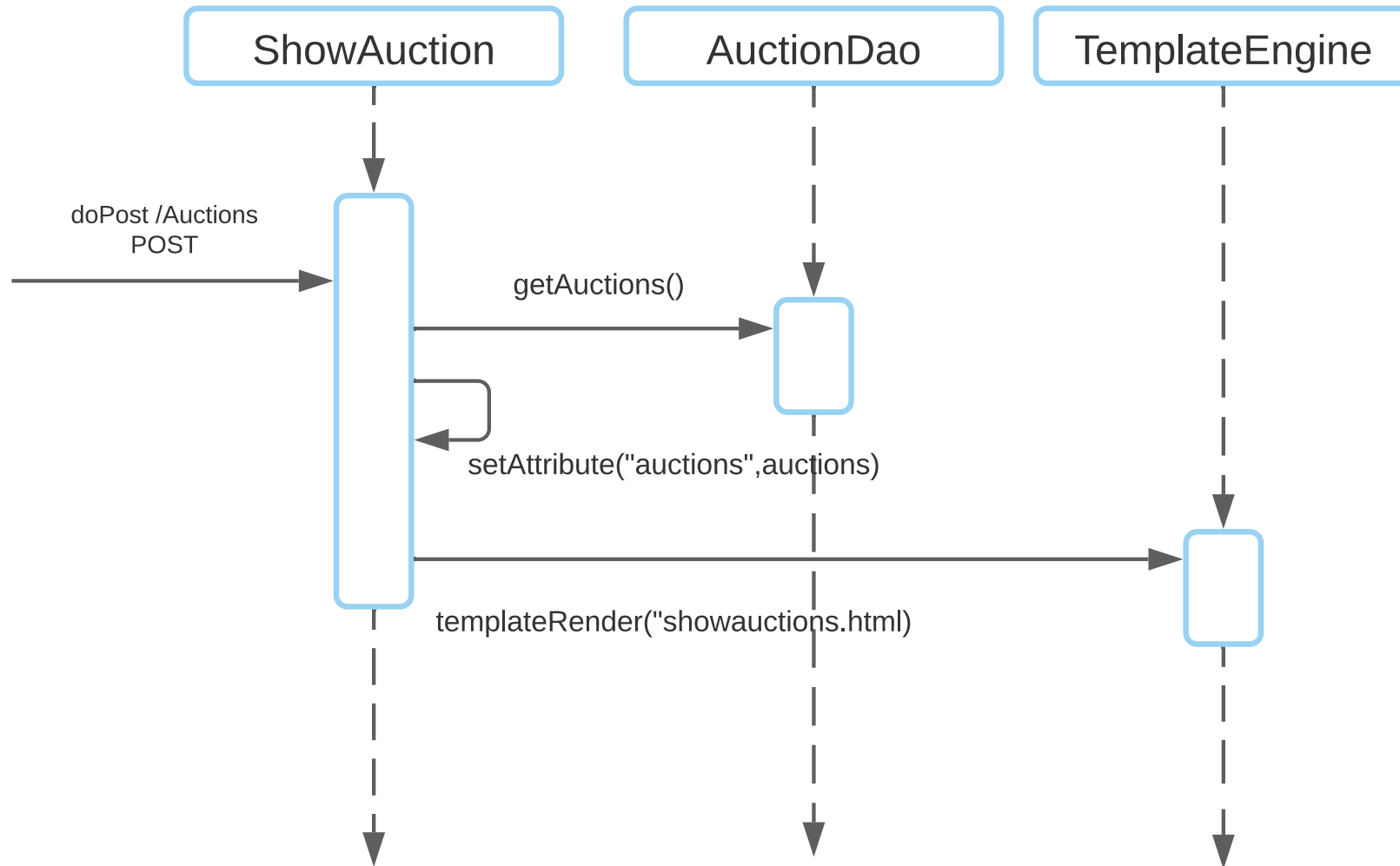
# Evento Di Creazione Offerta



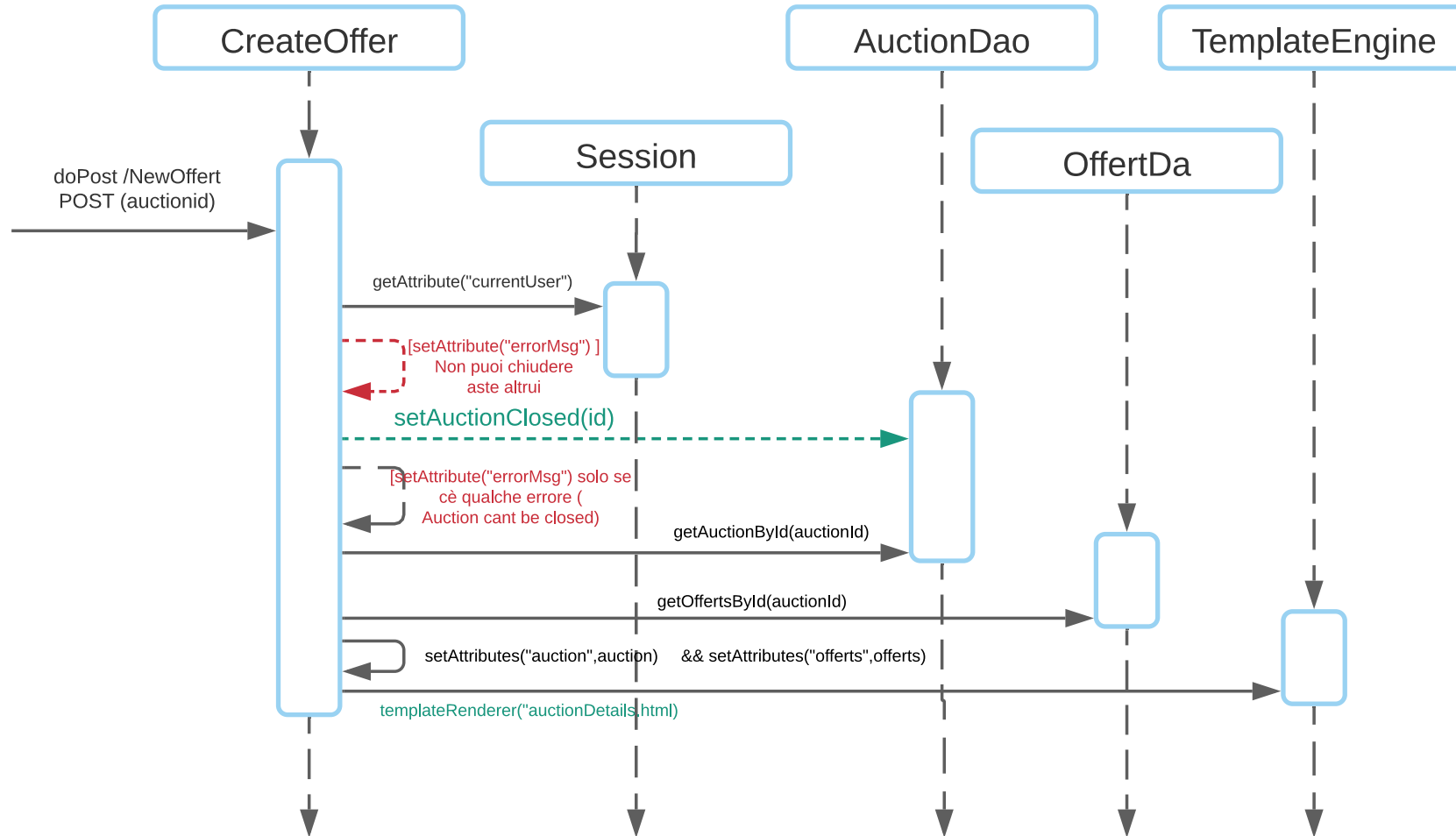
# Evento Di Caricamento UserPage



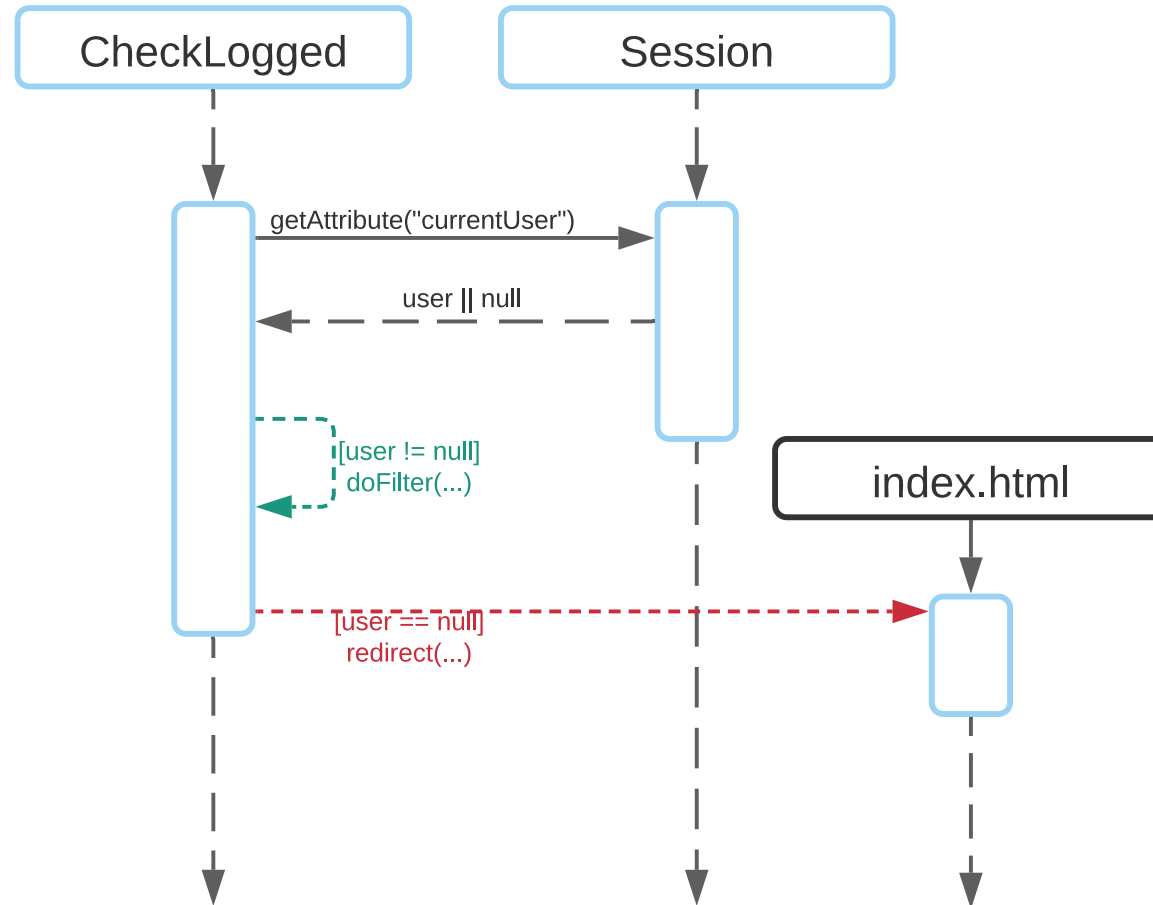
# Evento Di Stampa Lista Aste



# Evento Di Chiusura Asta



# Evento Di Filtro CheckLogin



Il Filtro **CheckNotLogged** è esattamente speculare (logina inversa)

# Evento Di Logout

