Sito di aste Online

Versione FAT CLIENT

(Javascript)

STUDENTE : DEAN NICOLA

Mat:911817

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.

Specifiche(2/2)

Specifiche RIA

- 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.

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

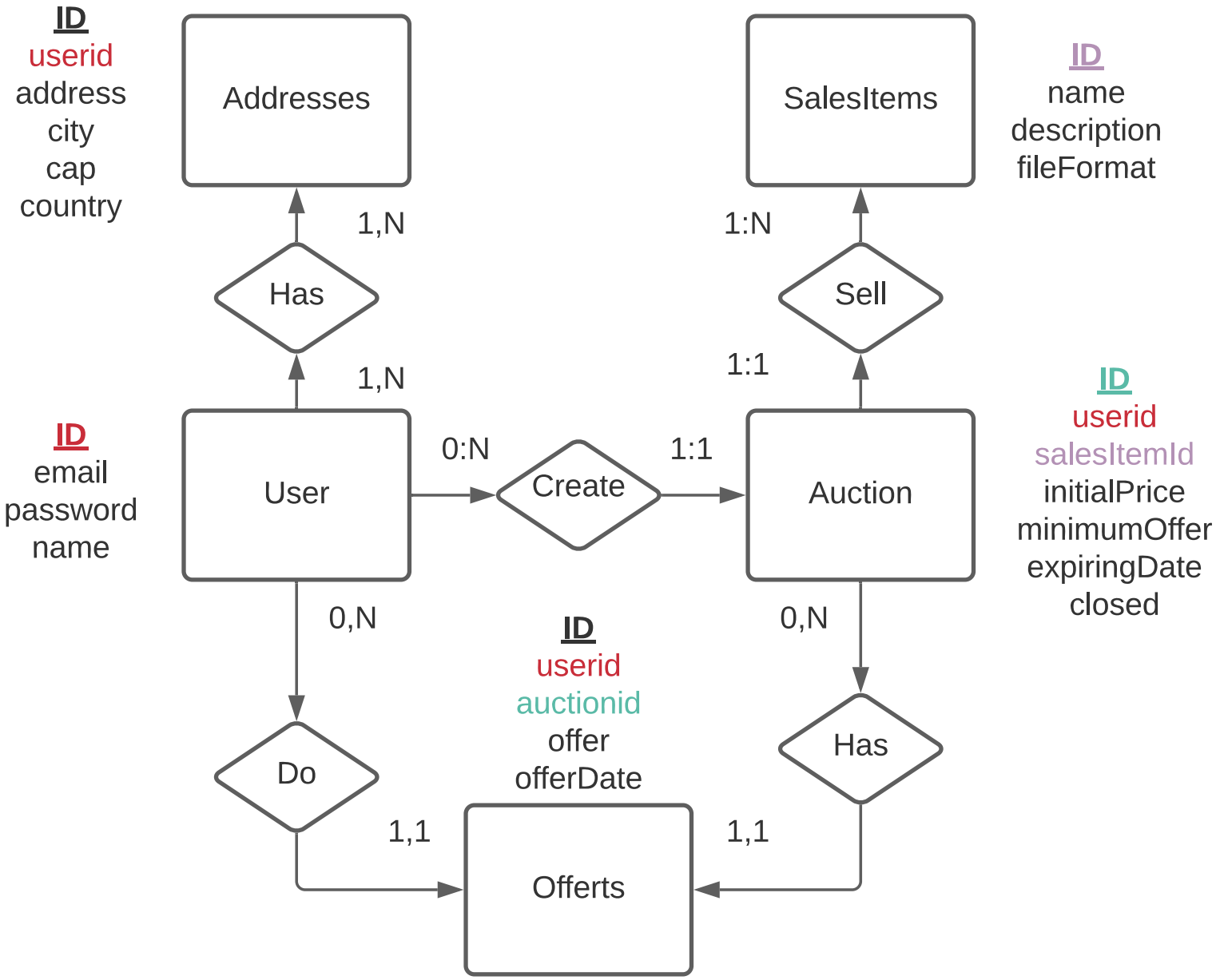


Diagramma View Database

Addresses | Users | Auctions | SalesItems | Offerts

AuctionsData

auctionid
itemid
userid
name
description
fileFormat
initialPrice
minimumOffer
expiringDate
closed
winnerId
username
maxOffer
maxOfferDate
(All Address Parameter)

OffertsData

auctionid
userid
name
offer
offerdate
(All Address Parameter)

MaxOfferData

auctionid
userid
name
offer
offerdate
(All Address Parameter)

```
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
);
```

```sql
create view offertsData as
(
    select auctionsid,offerts.userid,name,offer,offerDate,addresses.address,addresses.city,addresses.cap,addresses.country
    from (offerts join users
                on offerts.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 O where O.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(1/2)

- 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

Analisi Requisiti(2/2)

- E’ necessario salvare nei cookie lultima operazione svolta dall’utente
- E’ necessario salvare nei cookie un array di id corrispondenti alle ultime aste visitate e ancora aperte

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 assenza di aste suggerite viene visualizzato un allert
- 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 Addizionali/Aggiunte

- Per migliorare la visualizzazione/esperienza utente le aste vinte vengono visualizzate nella «UserPage» anziche nella pagina «Compro»
- La tabella Addresses è stata aggiunta per potenziali aggiornamenti futuri che permettano allutente 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)

Server Side Components

Models

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

HTML files

- Home.html
- Index.html

Filters

- LoggedFilter
- NoLoggedFilter

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>
  - getWinnedAuction(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
- AuctionSuggestions
- GetUserProfileData
- ImageGetter
- LogoutUser
- CloseAuction

Client Side Components

Login(index.html)

- LoginForm -> (manage submit and errorMsg)

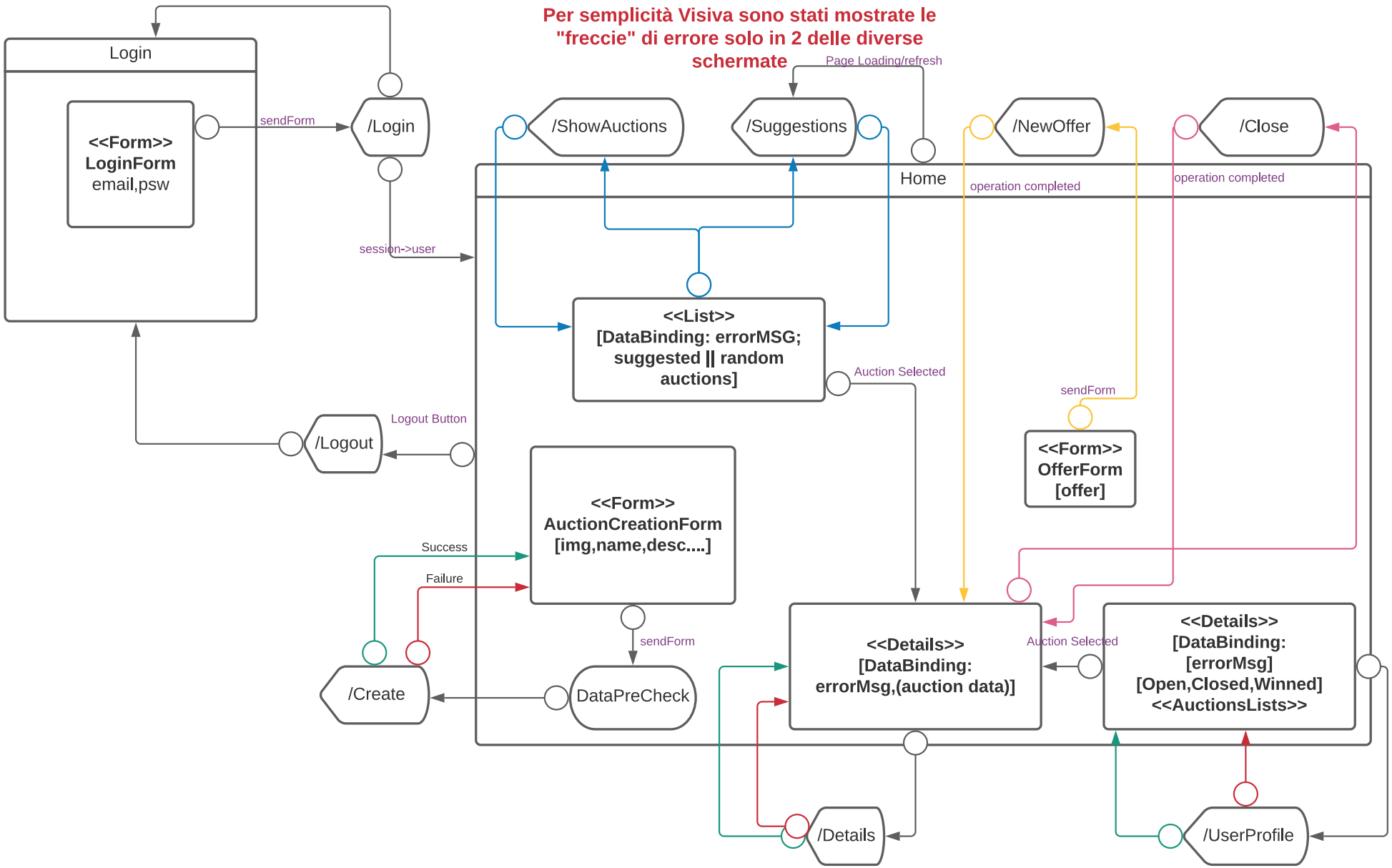
Home(home.html)

- HomeManager
  - LoadPage() (see LIFE CYCLE PAGE)
  - onWindowLoad(..)
  - HideAllElements()
  - (minors Button AddEventHandler)
- UserPage
  - printAuction(auctions,div) print auctions list inside div container
  - printAuctions(): print the 3 list inside userPage(open,closed,winned)
  - parseUserData()
  - Load() makecall
- AuctionCreation
  - sendAuctionCreation(form) : do makecall
  - SetVisible,Invisible
- AuctionDetails
  - minorPrintingAutomations methods
  - retriveData(id) get auction details and print them
  - closeAuction(id)
  - offertCreation(form) do makecall
  - Reset() delete old details
  - SetVisible,Invisible
- Offerts
  - printOffert(offer,HTMLid)
  - parseJsonOfferts(json)
- ShowAuctions
  - fillAuctions() :show auctions
  - getSuggestions(suggestions) retrive suggestions from db
  - getAuctions() retrive auctions from db
  - applyFilter(filter) hide auctions not similar to filter
  - Reset() delete old auctions
  - setVisible.Invisible
- Utils
  - makeCall(method,url,form,request..)
  - getCookie(name)
  - setCookie(name,content,expiring)
  - Some minors printing/DOM methods

Architettura

- Tutti i controller dell'App (lato server) estendono la classe «BasicServerlet»
- Tale classe ha lo scopo di evitare codice ripetuto in quanto presenta metodo usati in maniera ricorrente in tutti i controller
- Tali metodi sono sendJSON(..) e respondError(..)
- E' presente pure una classe «CustomException» che utilizza in automatico il metodo respondError in caso di azioni non valide da parte dell'utente
- Tutte le classi del Modello implementano «JsonSerializable» cosi da poter essere serializzate facilmente con la generica funzione sopracitata

Diagramma Applicativo IFML



Eventi & Azioni

La granparte dei metodi (tranne il Login) è richiamabile sia in Post che in Get

Client side		Server side	
Evento	**Azione**	**Evento**	**Azione**
index → login form → submit	Controllo dati	POST username password	Controllo credenziali
Home page→ load || Home page→ search	LoadJsControllers -> mostra aste suggerite	POST (JsonIdSuggeriti)	Legge aste suggerite dal DB
	LoadJsControllers ->suggestions vuote -> mostra aste casuali	GET (AuctionId)	Legge tutte le aste aperte dal DB (in ordine di scadenza piu recente)
Home page → click «userProfile»	HideAllElements-> MostraUserProfile	POST(UserId)	Legge lista di aste aperte,chiuse,vinte dall'utente
Home page→ elenco aste→ click «showDetails»|| UserPage → click «auctionName»	HideAllElements-> Mostra i dettagli asta	GET(AuctionID)	Legge i dettagli dell'asta dal DB
AuctionDetails → Click «close auction>>	Ristampa i dettagli asta	GET(AuctionId)	Aggiorna il DB settando lasta come chiusa (se possibile)
AuctionDetails → offerForm→submit	Controllo dati	POST(AuctionId,Offer)	Creazione Offerta
UserPage->click»CreateAuction» →form→submit	Controllo dati	POST(Img,name,desc….)	Crea un Asta
Logout		GET	Terminazione della sessione

makeCall indica una richiesta HTTP asincrona

Controller & EventHandler

Client side		Server side	
Evento	**Controllore**	**Evento**	**Controllore**
index → login form → submit	Function makeCall	POST username password	CheckLogin(serverlet)
Home page→ load || Home page→ search	Function LoadPage || fillAuctions()->getSuggestions()	POST (JsonIdSuggeriti)	AuctionSuggestions(serverlet)
	[se suggestions è vuota] fillAuctions()->getAuctions()	GET (AuctionId)	ShowAuctions(serverlet)
Home page → click «userProfile»	Function makeCall	POST(UserId)	GetUserProfileData(serverlet)
Home page→ elenco aste→ click «showDetails»|| UserPage → click «auctionName»	Function retriveData()	GET(AuctionID)	ShowAuctionDetails(serverlet)
AuctionDetails → Click «close auction>>	Function closeAuction(id)	GET(AuctionId)	CloseAuctions(serverlet)
AuctionDetails → offerForm→submit	Function offertCreation(form)	POST(AuctionId,Offer)	CreateOffert(serverlet)
UserPage->click»CreateAuction» →form→submit	Function sendAuctionCreation	POST(Img,name,desc….)	CreateAuction(serverlet)
Logout		GET	Logout(serverlet)

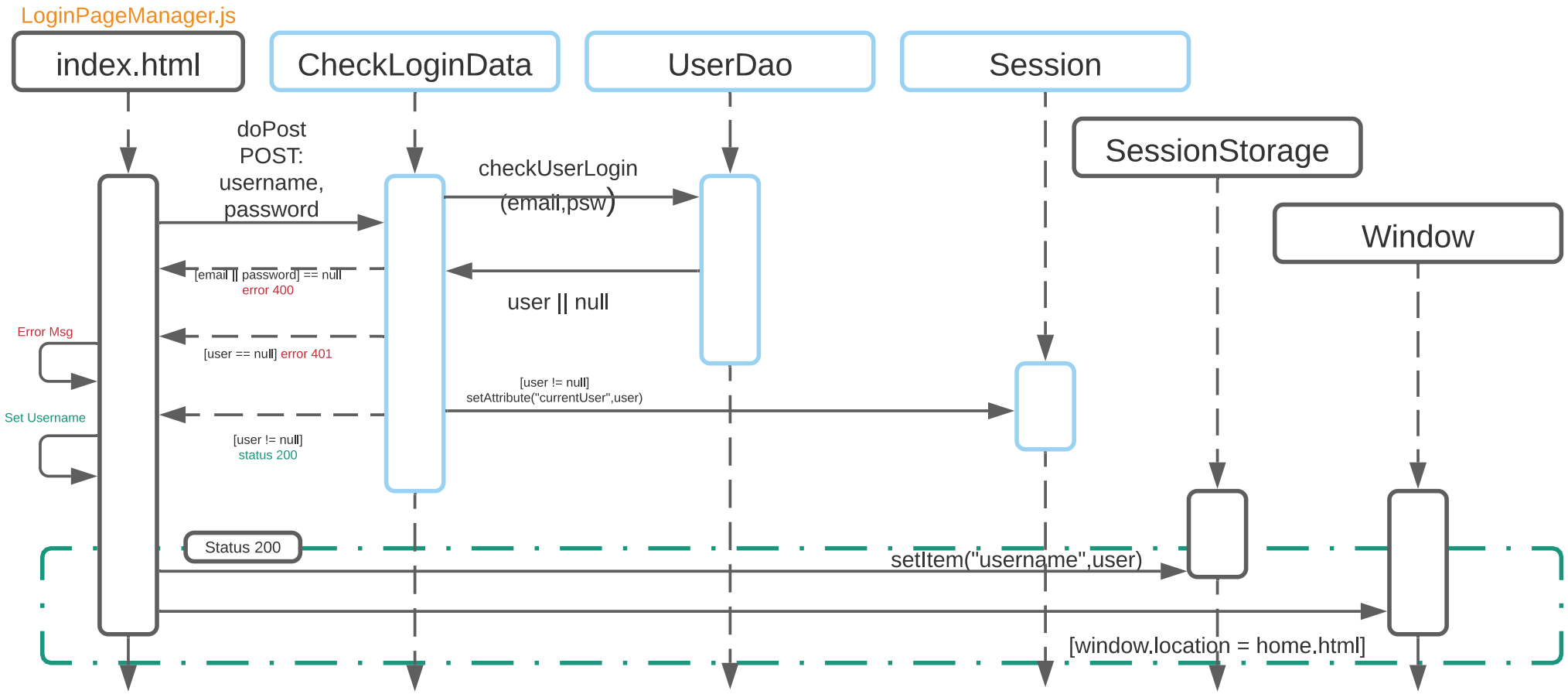
Sequence Diagram

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

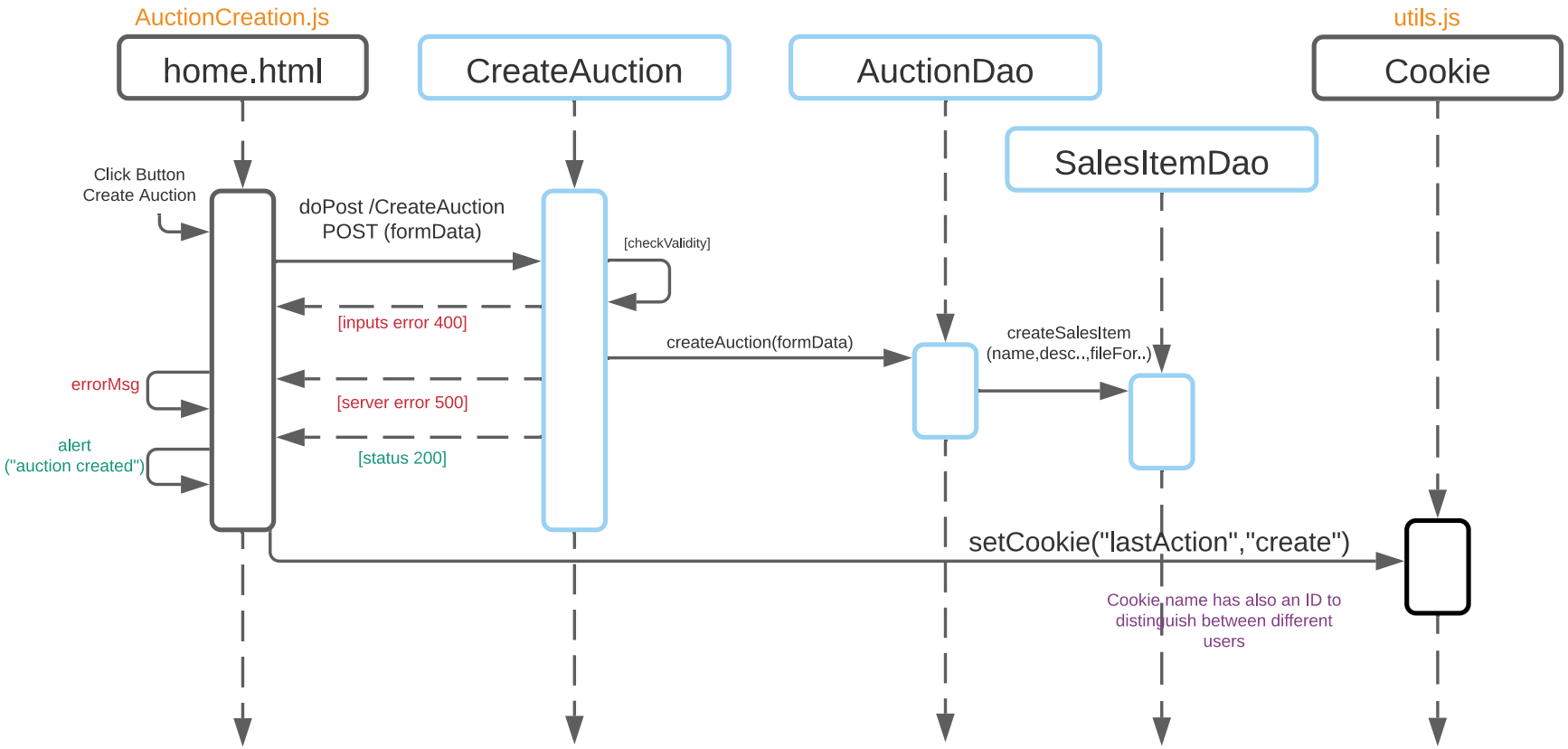
Evento Di Login



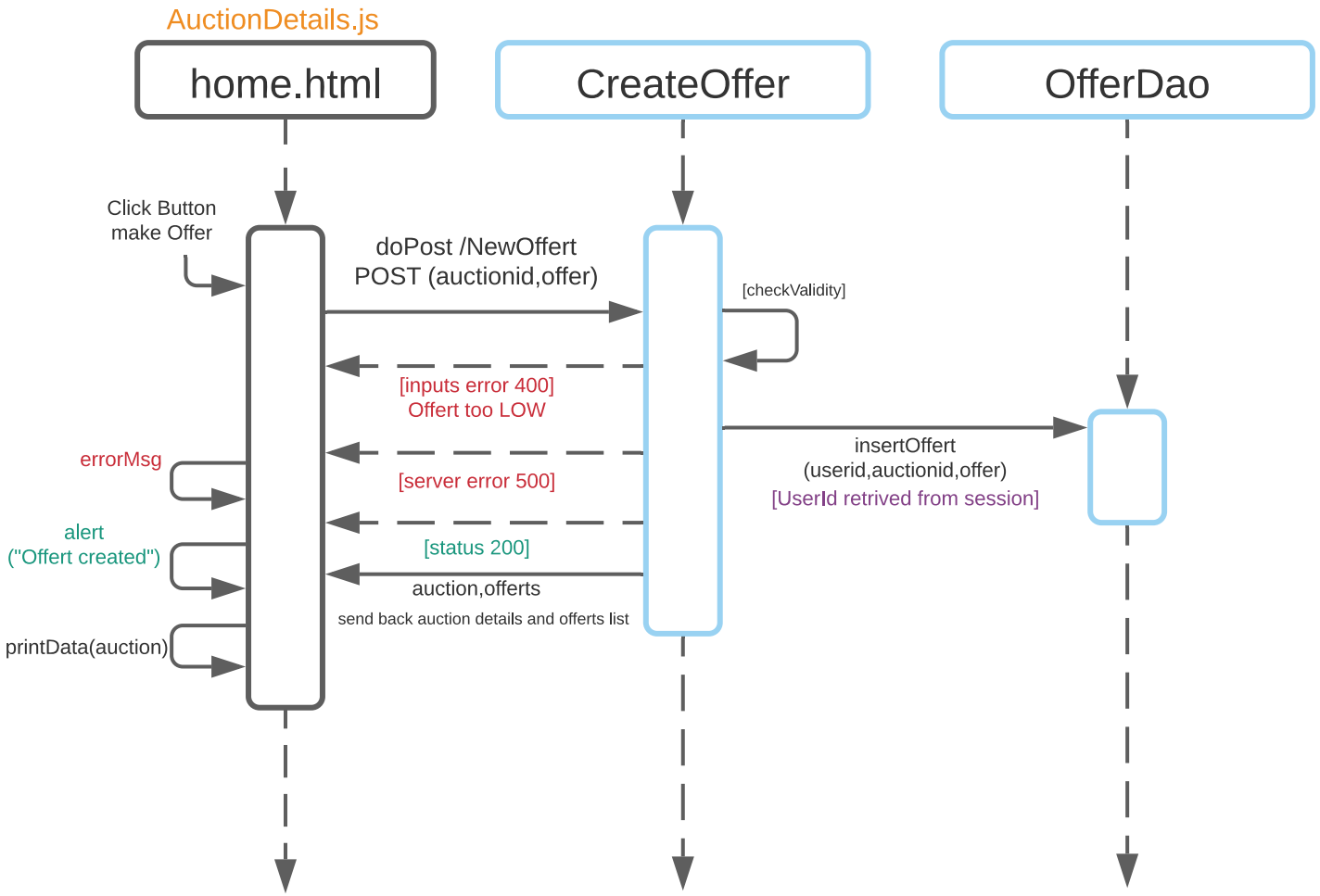
Life Cycle

- Tutte le pagine seguenti vengono generate da HomeManager al caricamento di home.html
- Home manager è uno script che permette di caricare tutti gli elementi DOM e gli oggetti JS che compongono l'APP, questo script viene RUNNATO solo una volta al caricamento del file home.html
- All'interno di questo script vengono anche assegnati gli eventi di click/input ai vari bottoni (Logout,CreateAuction….)
- Viene anche generato l'evento sulla barra di ricerca che permette un filtro realtime delle aste per parola chiave

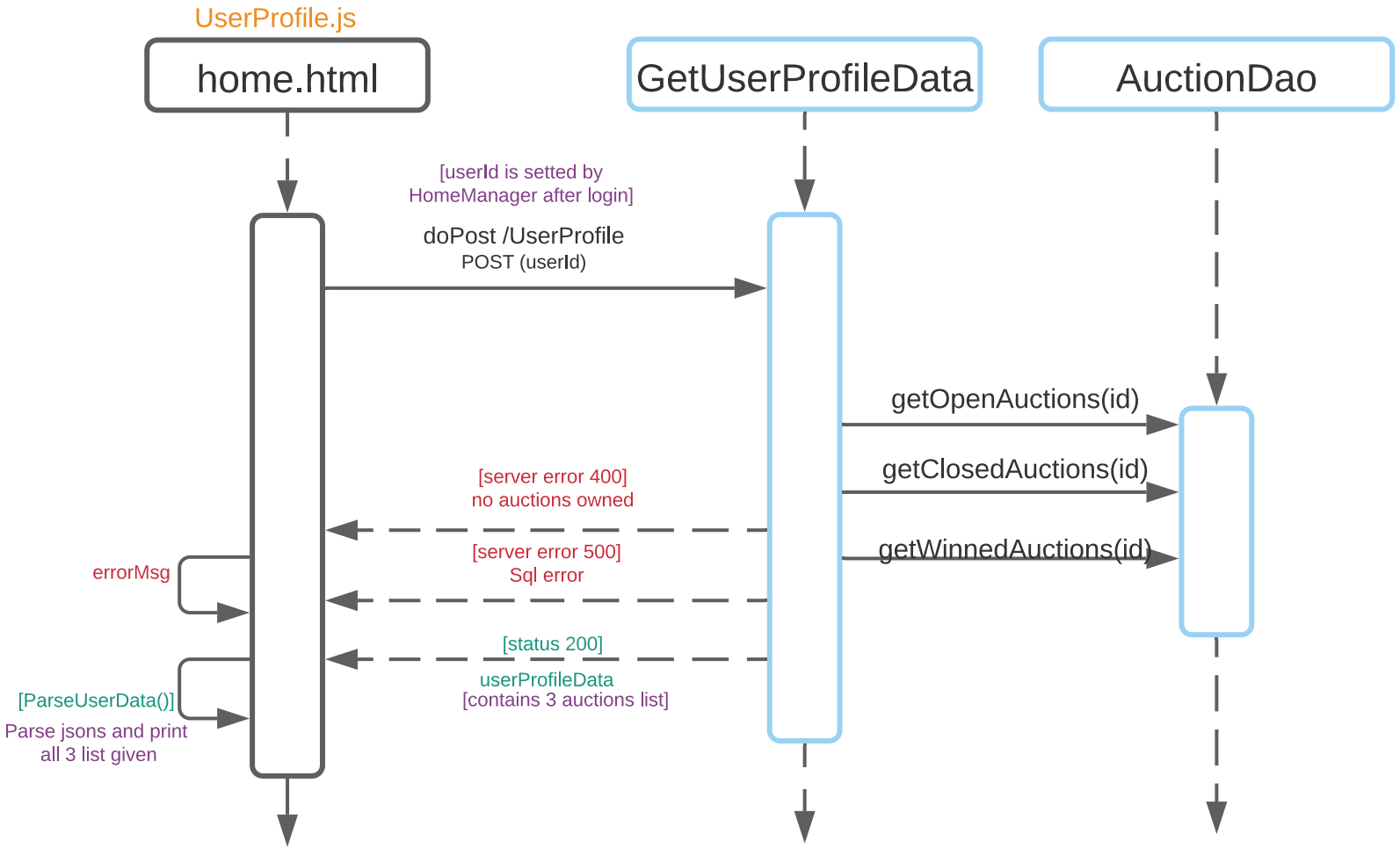
Evento Di Creazione Asta



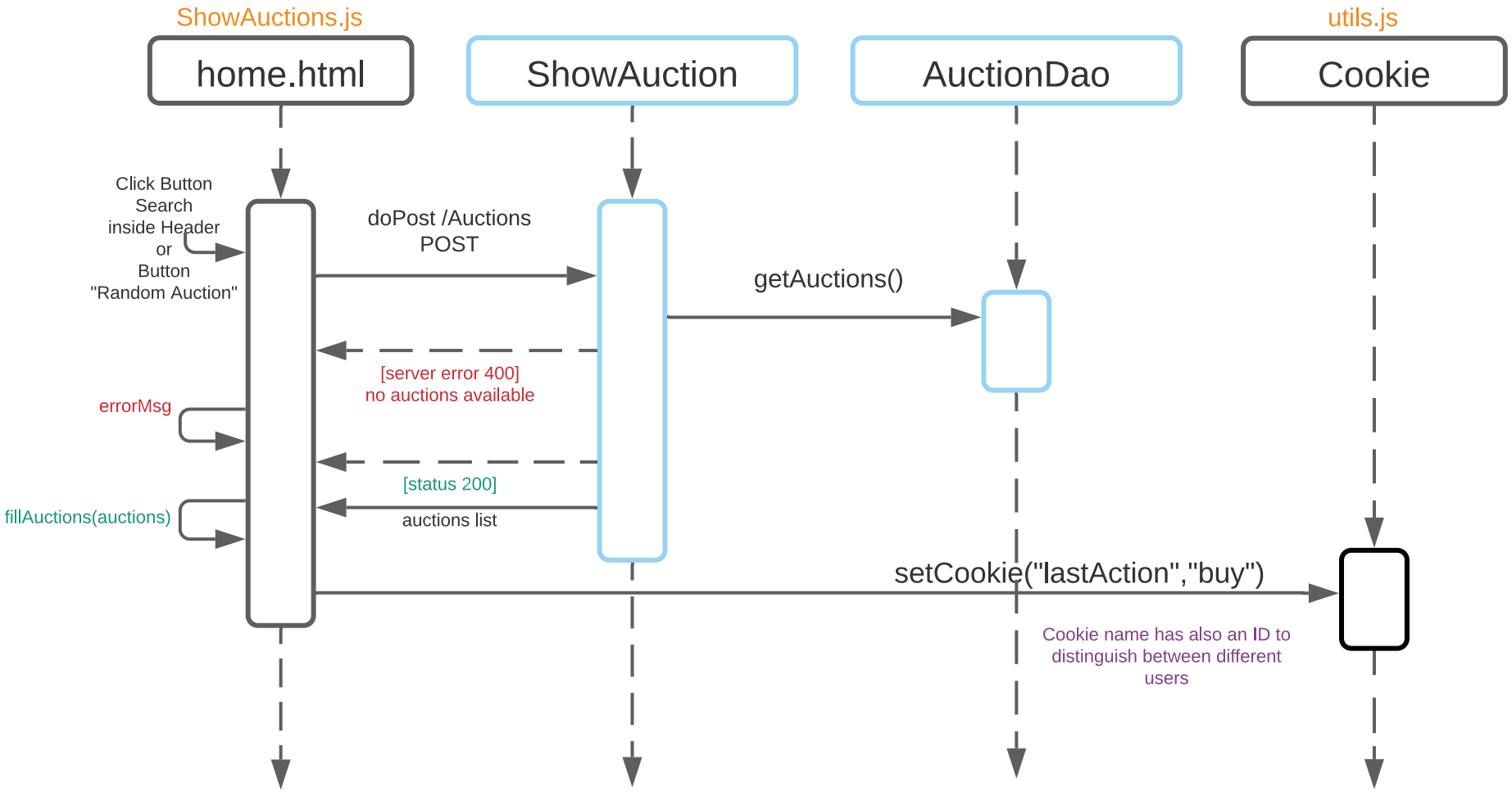
Evento Di Creazione Offerta



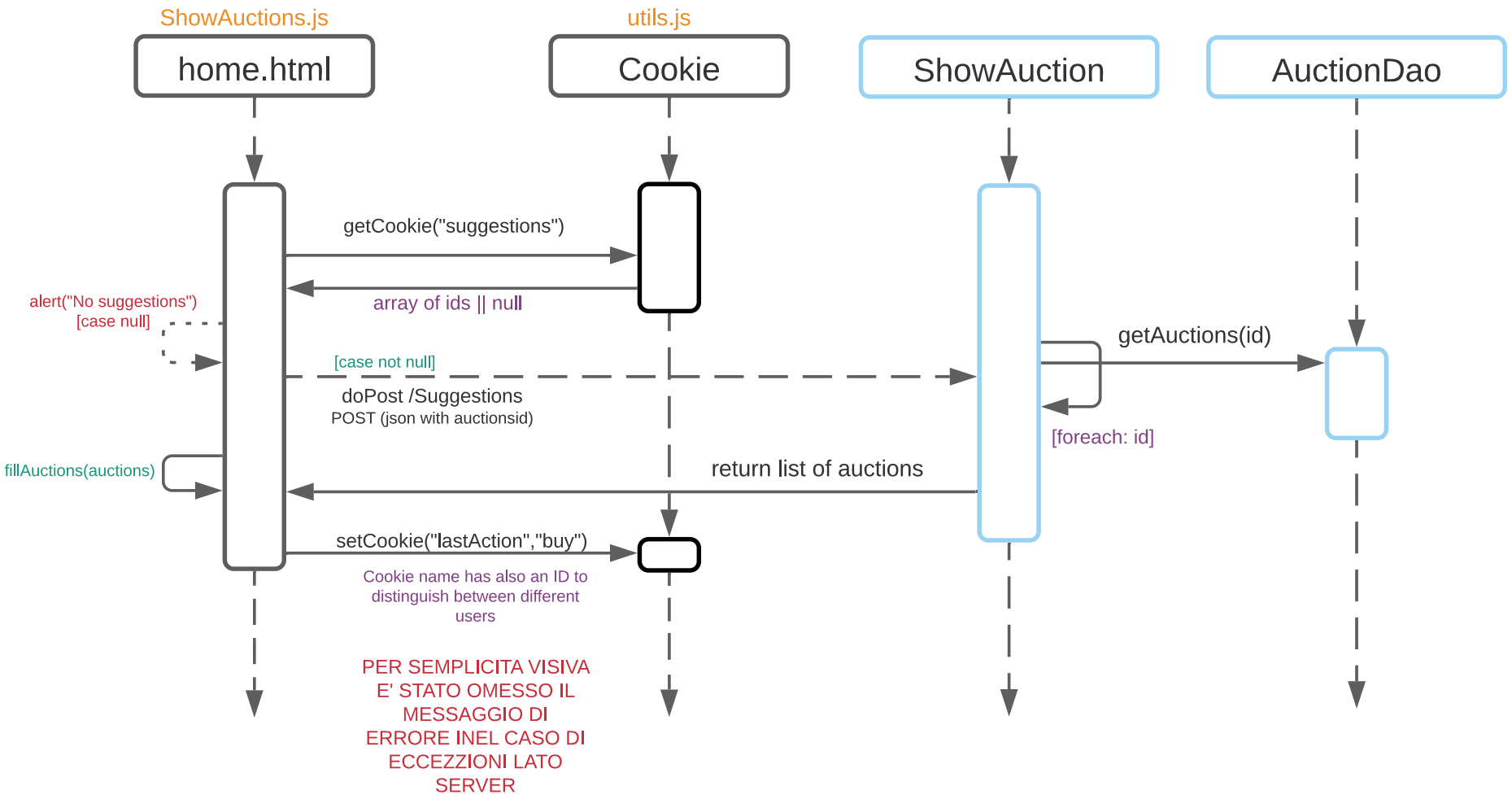
Evento Di Caricamento UserPage



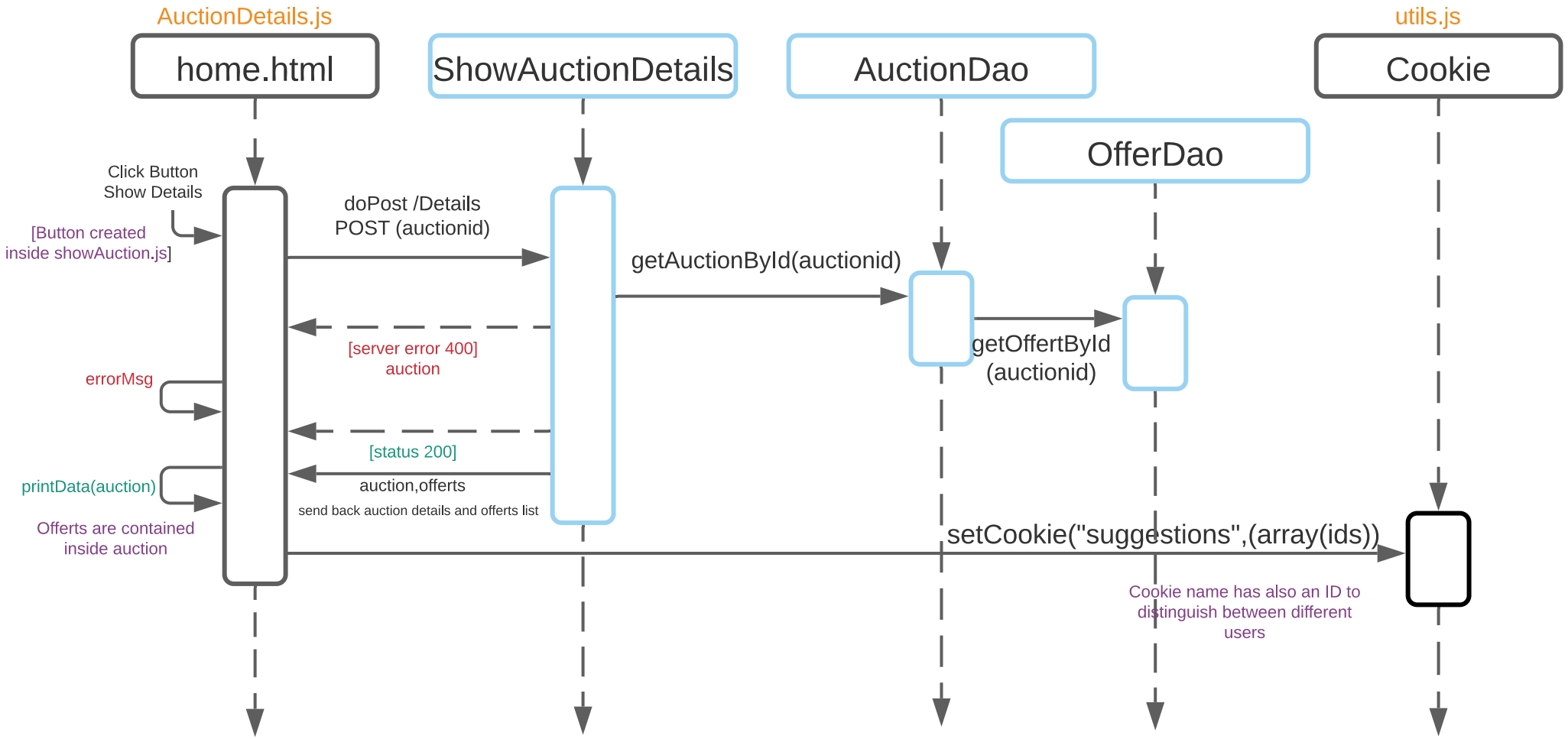
Evento Di Stampa Lista Aste(Random)



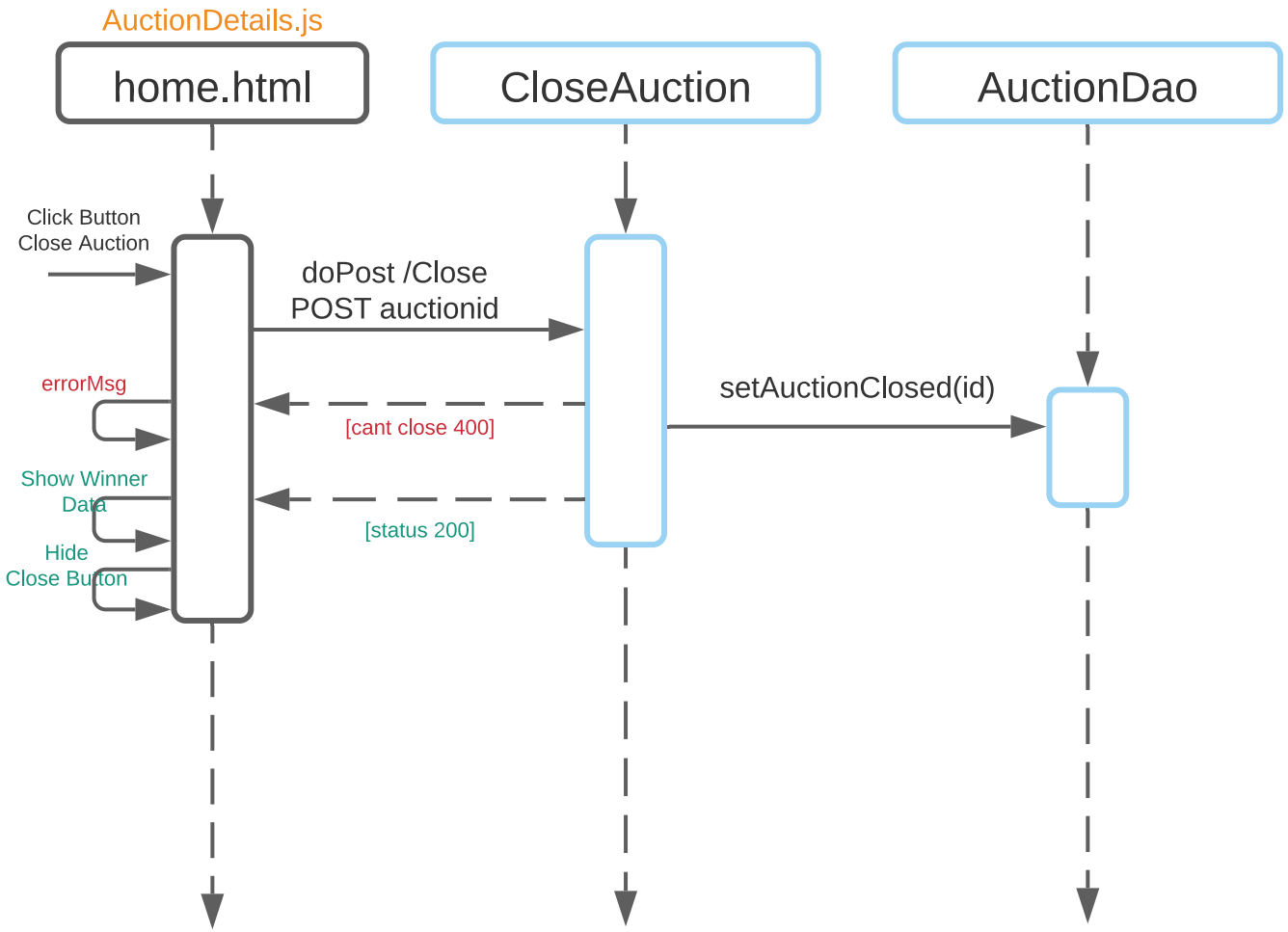
Evento Di Stampa Lista Aste(Recent Clicked)



Evento Di Stampa Dettagli Asta



Evento Di Chiusura Asta



Evento Di Logout