Sito di aste Online

Versione THICK CLIENT

(javascript)

STUDENTE : DEAN NICOLA

Mat: 911817

nicola.dean@mail.polimi.it

Diagramma Tabelle Database
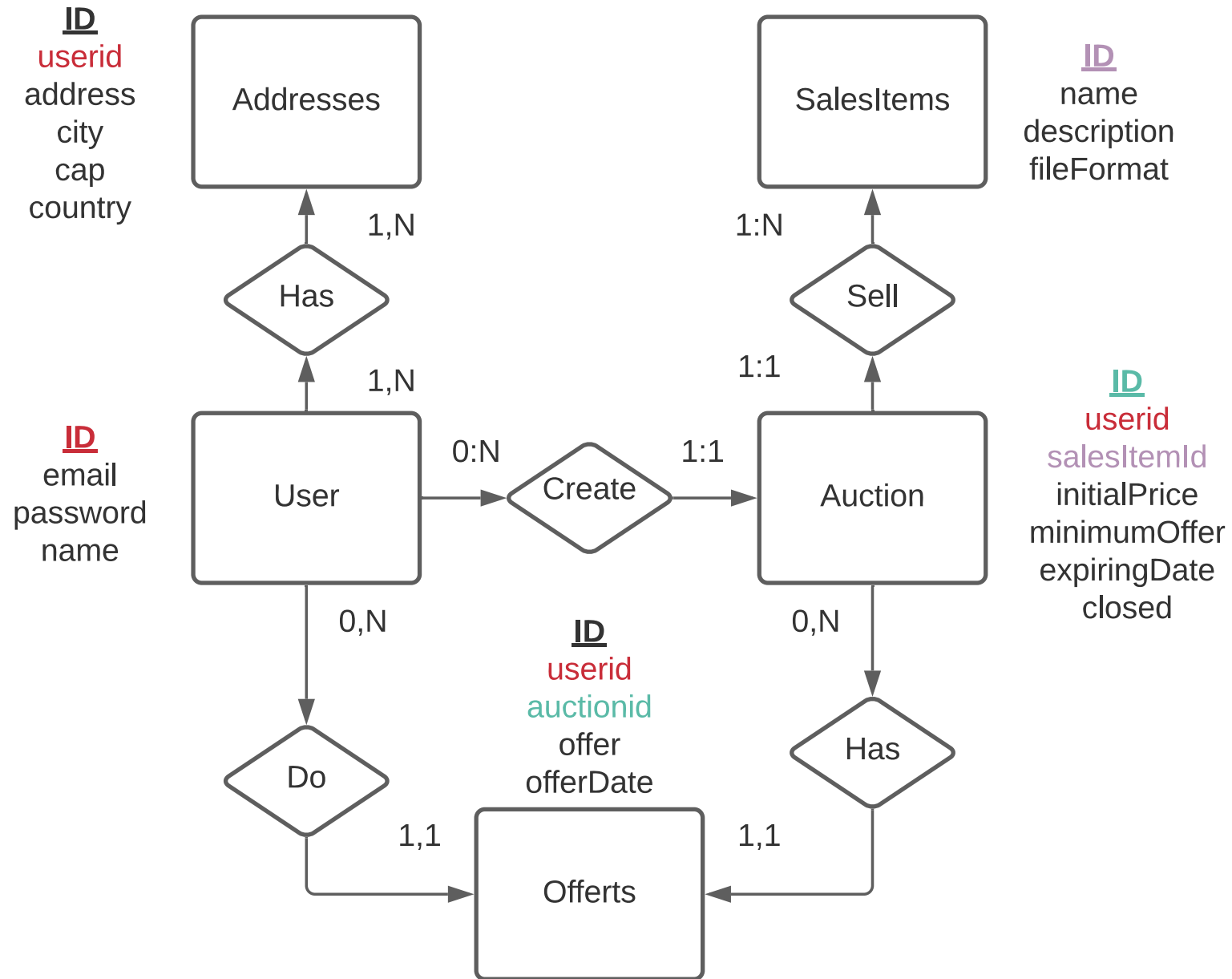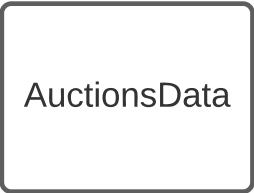
ID
userid
address
city
cap
country

Addresses

ID
name
description
fileFormat

SalesItems

1,N

Has

1:N

Sell

1,N

1:1

ID
email
password
name

User

0:N

Create

1:1

Auction

ID
userid
salesItemId
initialPrice
minimumOffer
expiringDate
closed

0,N

ID
userid
auctionid
offer
offerDate

Do

0,N

Has

1,1

Offerts

1,1

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)

```sql
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
);
```

```sql
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
);
```

Evento Di Login
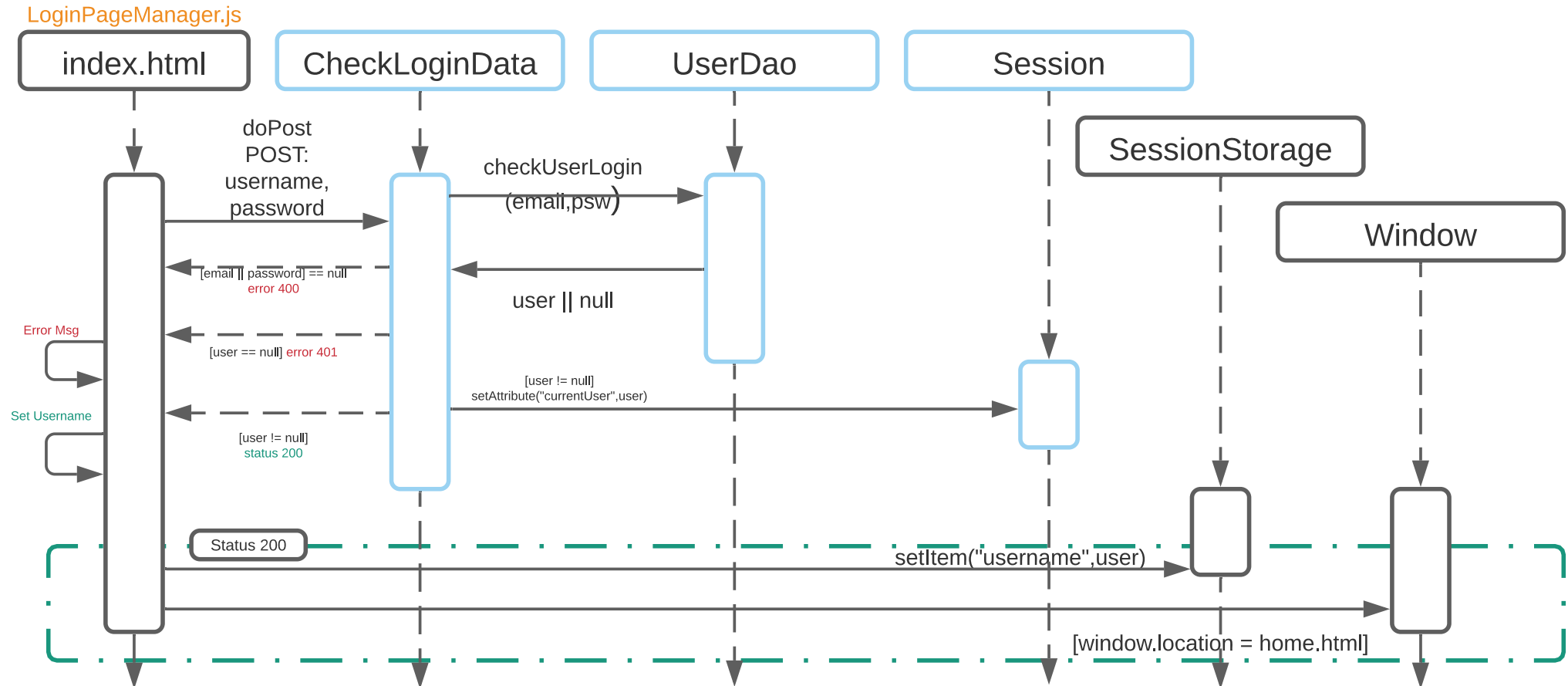
Evento Di Caricamento HomeManager
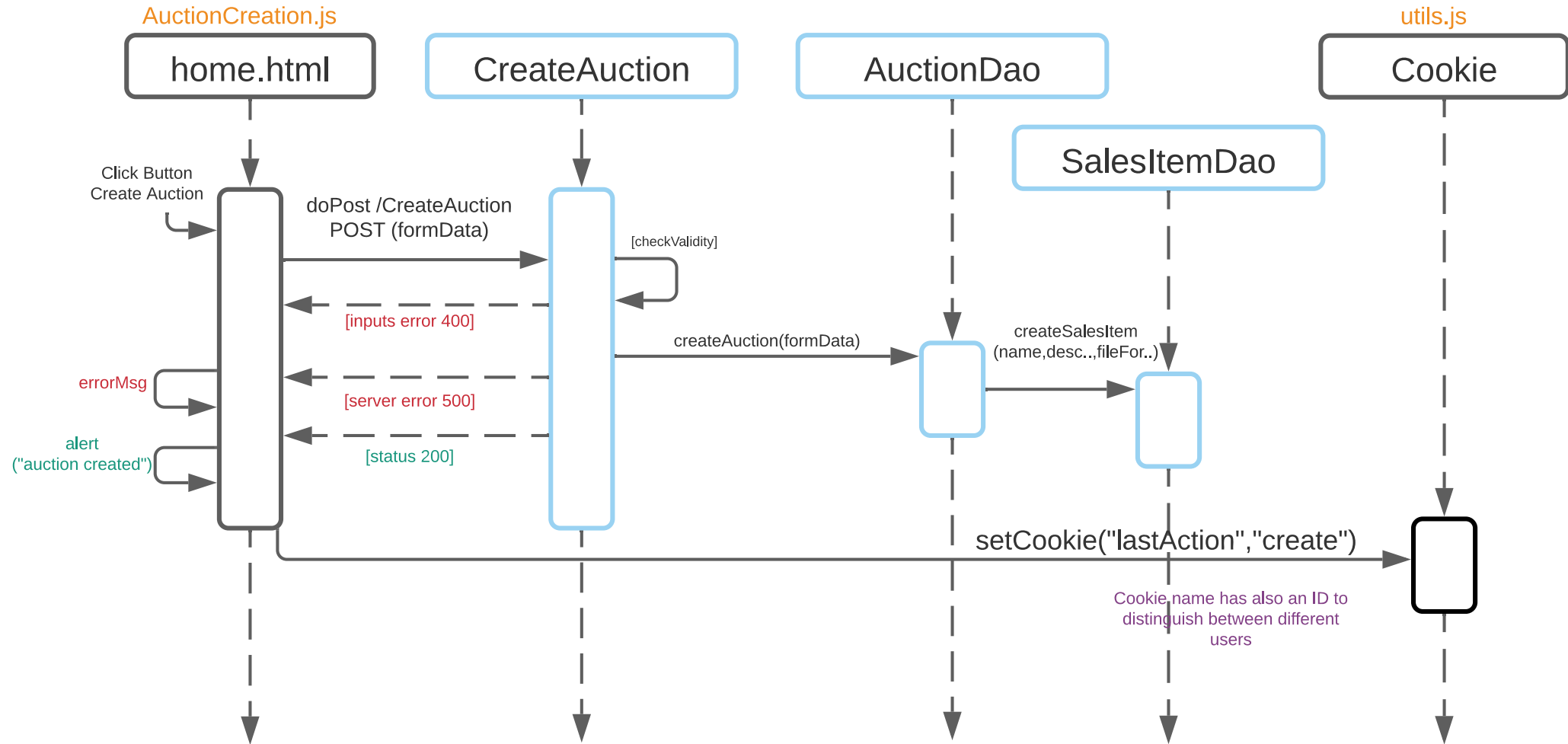
Home manager è lo script che si occupa di creare tutti gli oggetti necessari affinche il sito funziona, Questo script viene lanciato in automatico
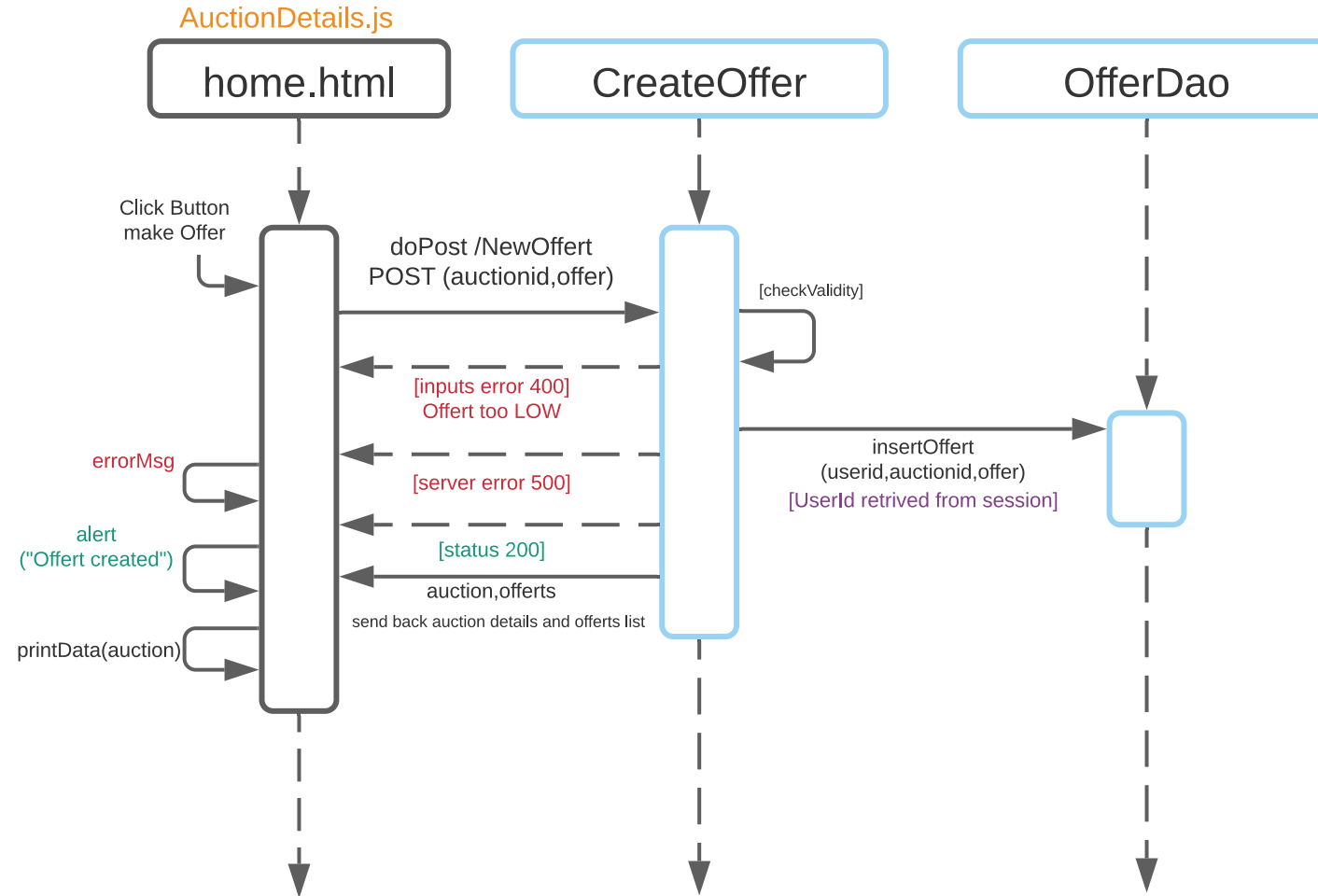UNA VOLTA SOLA
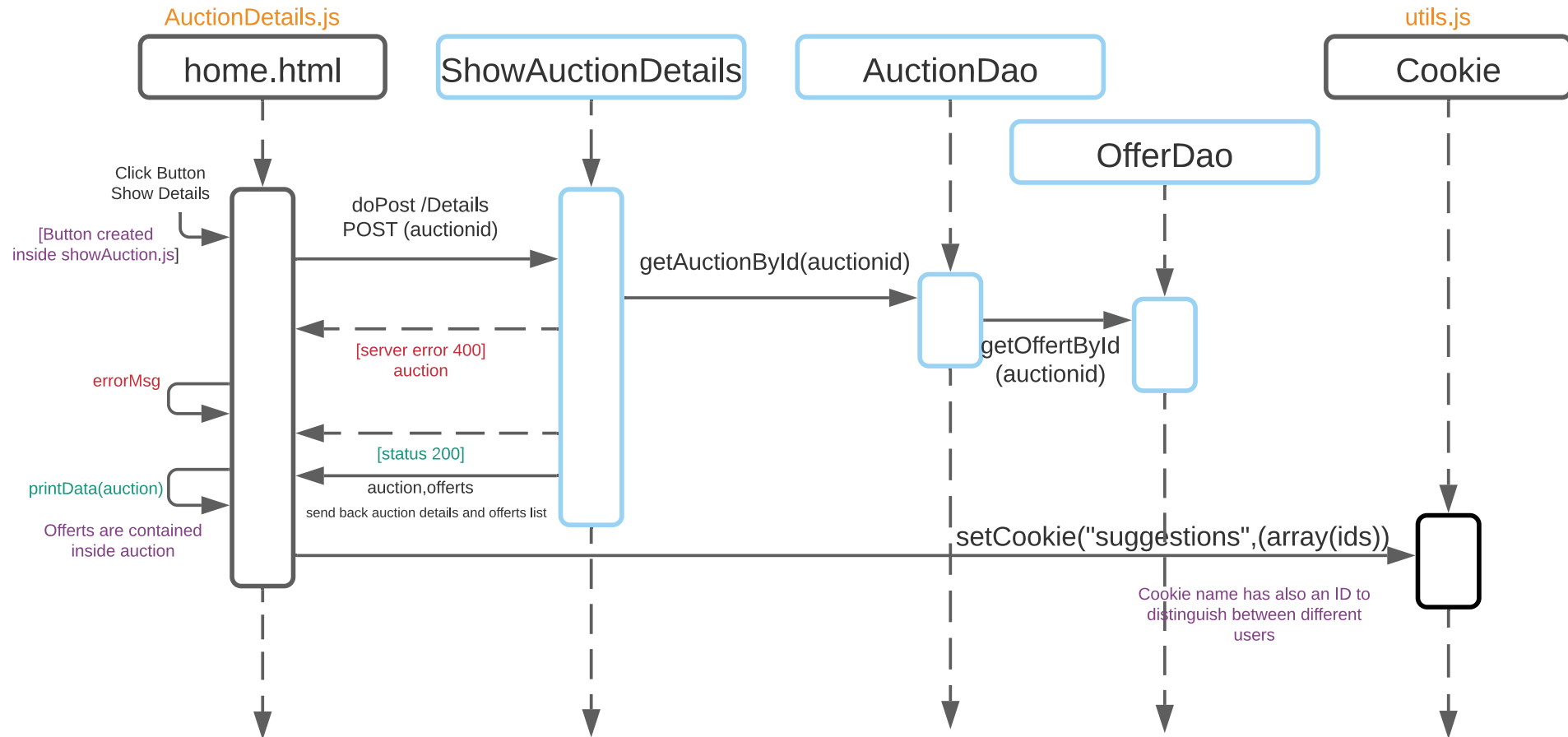Subito dopo il Login

Evento Di Creazione Asta

Evento Di Creazione Offerta

AuctionDetails.js

home.html | CreateOffer | OfferDao

Click Button
make Offer

doPost /NewOffert
POST (auctionid,offer)

[checkValidity]

[inputs error 400]
Offert too LOW

insertOffert
(userid,auctionid,offer)
[UserId retrived from session]

errorMsg

[server error 500]

alert
("Offert created")

[status 200]

auction,offerts
send back auction details and offerts list

printData(auction)

Evento Di Caricamento Dettagli Asta

Evento Di Caricamento UserPage

Evento Di Stampa Lista Aste (Random)

ShowAuctions.js

home.html

ShowAuction

AuctionDao

utils.js

Cookie

Click Button Search inside Header or Button "Random Auction"

doPost /Auctions POST

getAuctions()

[server error 400] no auctions available

errorMsg

[status 200]

auctions list

fillAuctions(auctions)

setCookie("lastAction","buy")

Cookie name has also an ID to distinguish between different users
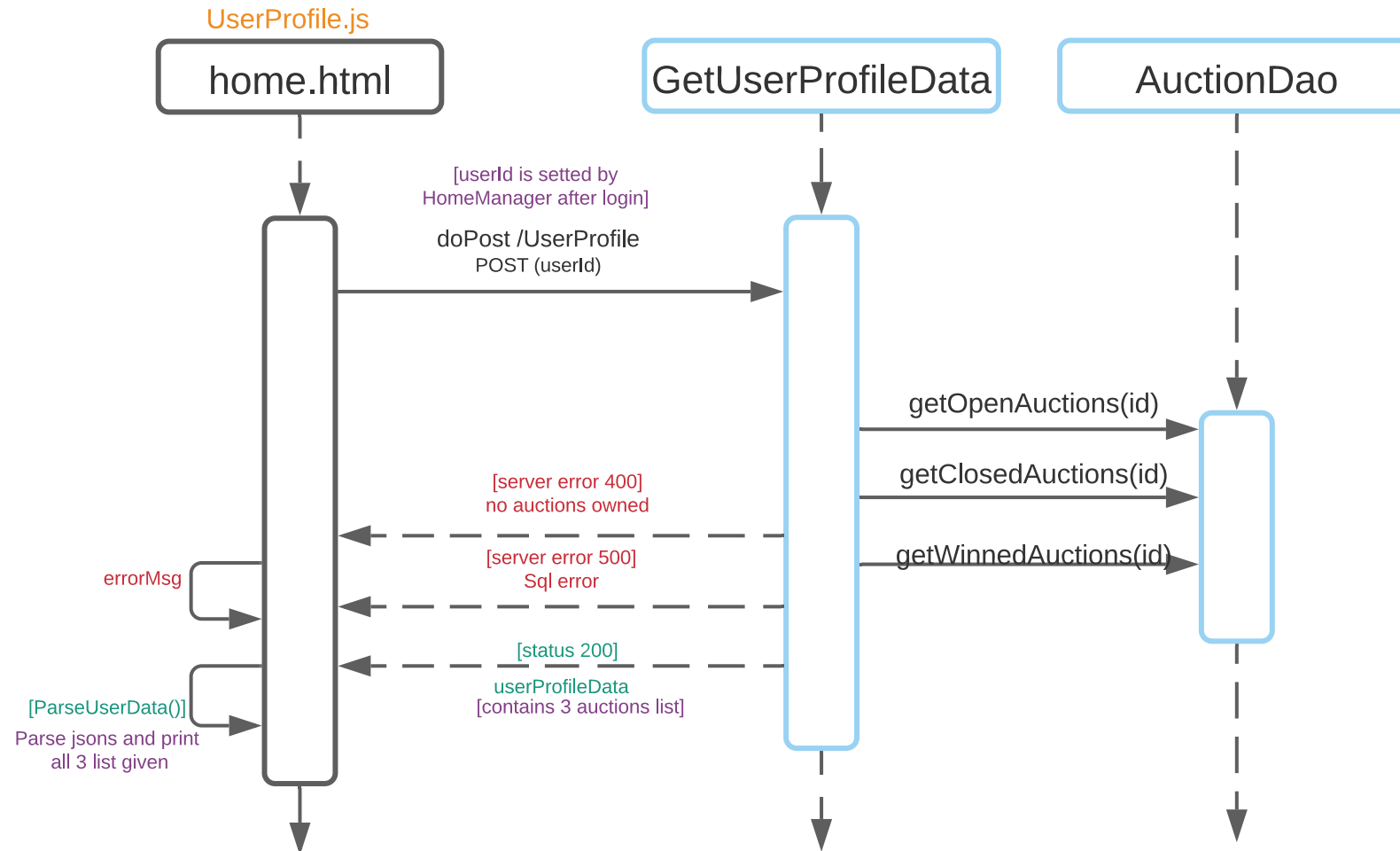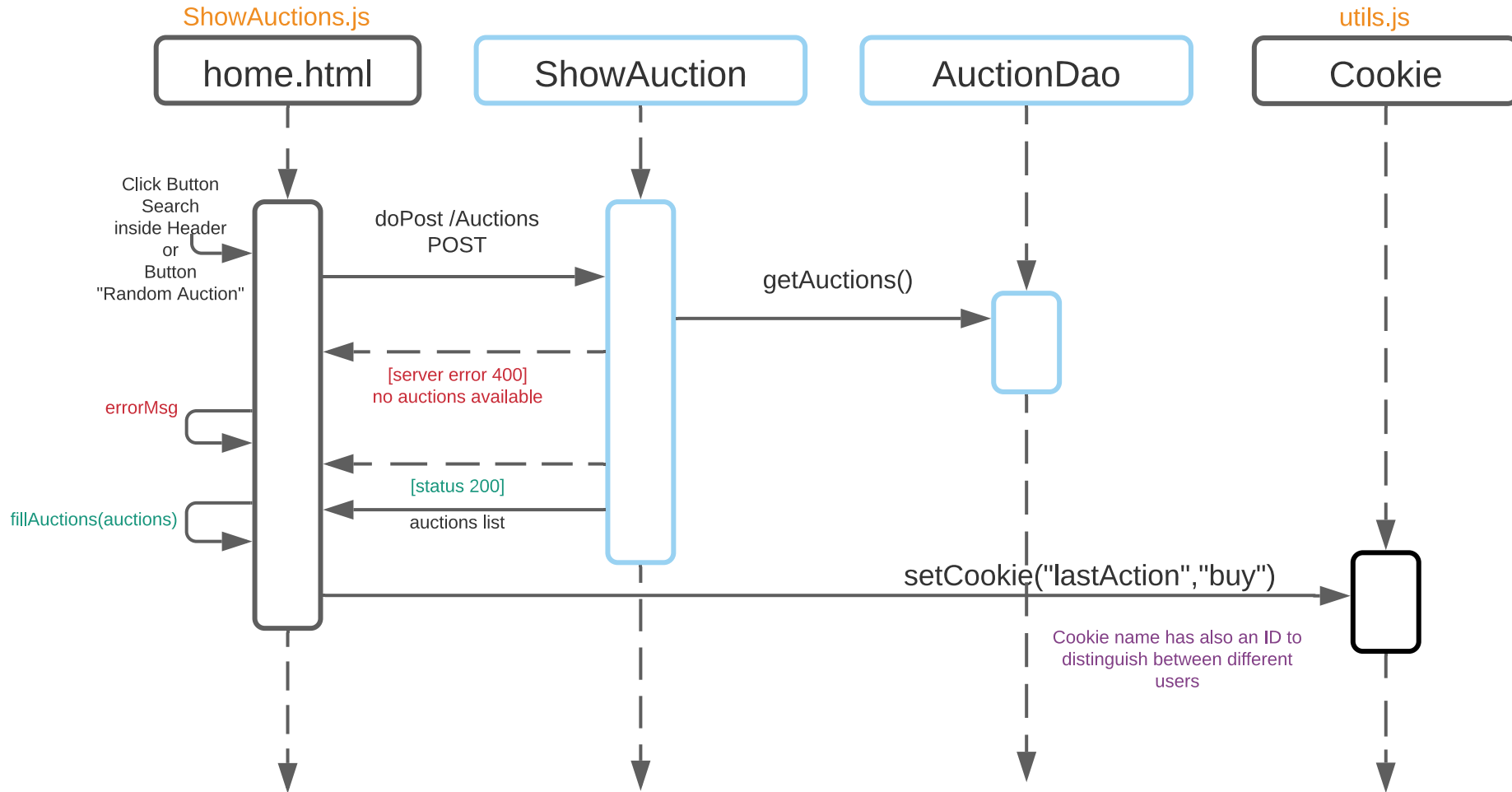
Evento Di Stampa Lista Aste (Recent Clicked)

ShowAuctions.js — home.html

utils.js — Cookie

ShowAuction

AuctionDao

getCookie("suggestions")

array of ids || null

alert("No suggestions")
[case null]

[case not null]

doPost /Suggestions
POST (json with auctionsid)

getAuctions(id)

[foreach: id]

fillAuctions(auctions)

return list of auctions

setCookie("lastAction","buy")

Cookie name has also an ID to
distinguish between different
users

PER SEMPLICITA VISIVA
E' STATO OMESSO IL
MESSAGGIO DI
ERRORE INEL CASO DI
ECCEZZIONI LATO
SERVER

Evento Di Chiusura Asta

Evento Di Logout