

PROGETTO TIW 2023

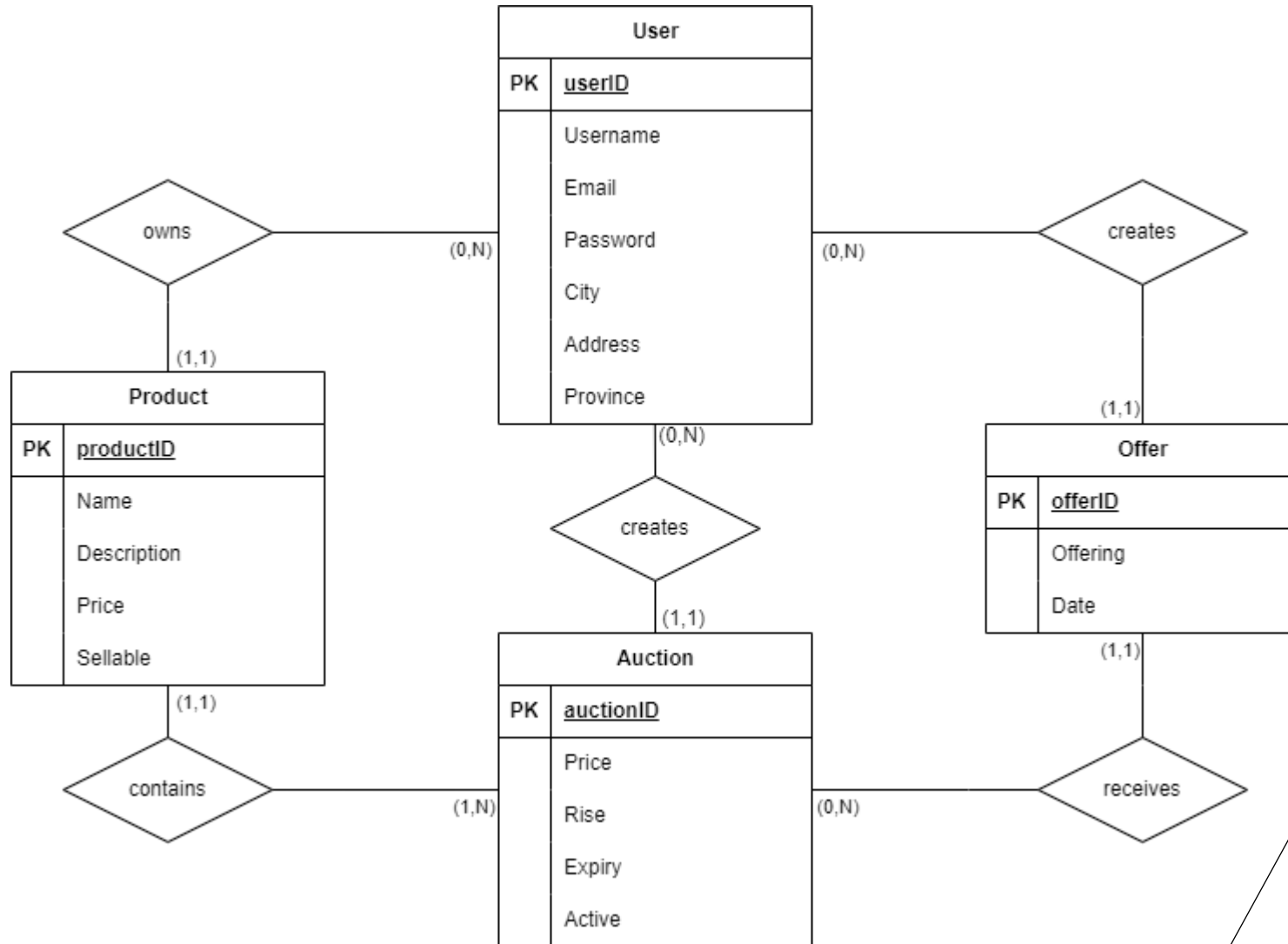
Progetto di Christian Rossi e Kirolos Sharoubim

Prof. Fraternali Piero

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 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.

DATABASE



DATABASE

```
CREATE TABLE IF NOT EXISTS Offer (  
  offerID      SERIAL      PRIMARY KEY,  
  offering     INT         NOT NULL,  
  date         DATETIME    NOT NULL,  
  userID       BIGINT      UNSIGNED NOT NULL REFERENCES USER (userID),  
  auctionID    BIGINT      UNSIGNED NOT NULL REFERENCES AUCTION (auctionID)  
);
```

```
CREATE TABLE IF NOT EXISTS Article (  
  productID    SERIAL      PRIMARY KEY,  
  name         VARCHAR(32)  NOT NULL,  
  description   VARCHAR(256) NOT NULL,  
  price        INT         NOT NULL,  
  sellable     BOOLEAN     NOT NULL,  
  userID       BIGINT      UNSIGNED NOT NULL REFERENCES USER (userID),  
  auctionID    BIGINT      UNSIGNED NOT NULL REFERENCES AUCTION (auctionID)  
);
```

DATABASE

CREATE TABLE IF NOT EXISTS **User** (

userID	SERIAL
username	VARCHAR(32)
email	VARCHAR(256)
password	CARCHAR(64)
city	VARCHAR(32)
address	VARCHAR(64)
province	VARCHAR(32)

PRIMARY KEY,
NOT NULL UNIQUE,
NOT NULL UNIQUE,
NOT NULL,
NOT NULL,
NOT NULL,
NOT NULL

);

CREATE TABLE IF NOT EXISTS **Auction** (

auctionID	SERIAL
price	INT
rise	INT
expiry	DATETIME
active	BOOLEAN
userID	BIGINT

PRIMARY KEY,
NOT NULL,
NOT NULL,
NOT NULL,
NOT NULL,
UNSIGNED NOT NULL REFERENCES USER (**userID**)

);

ANALISI DEI 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.

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.

View & view component

ANALISI DEI 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.

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.

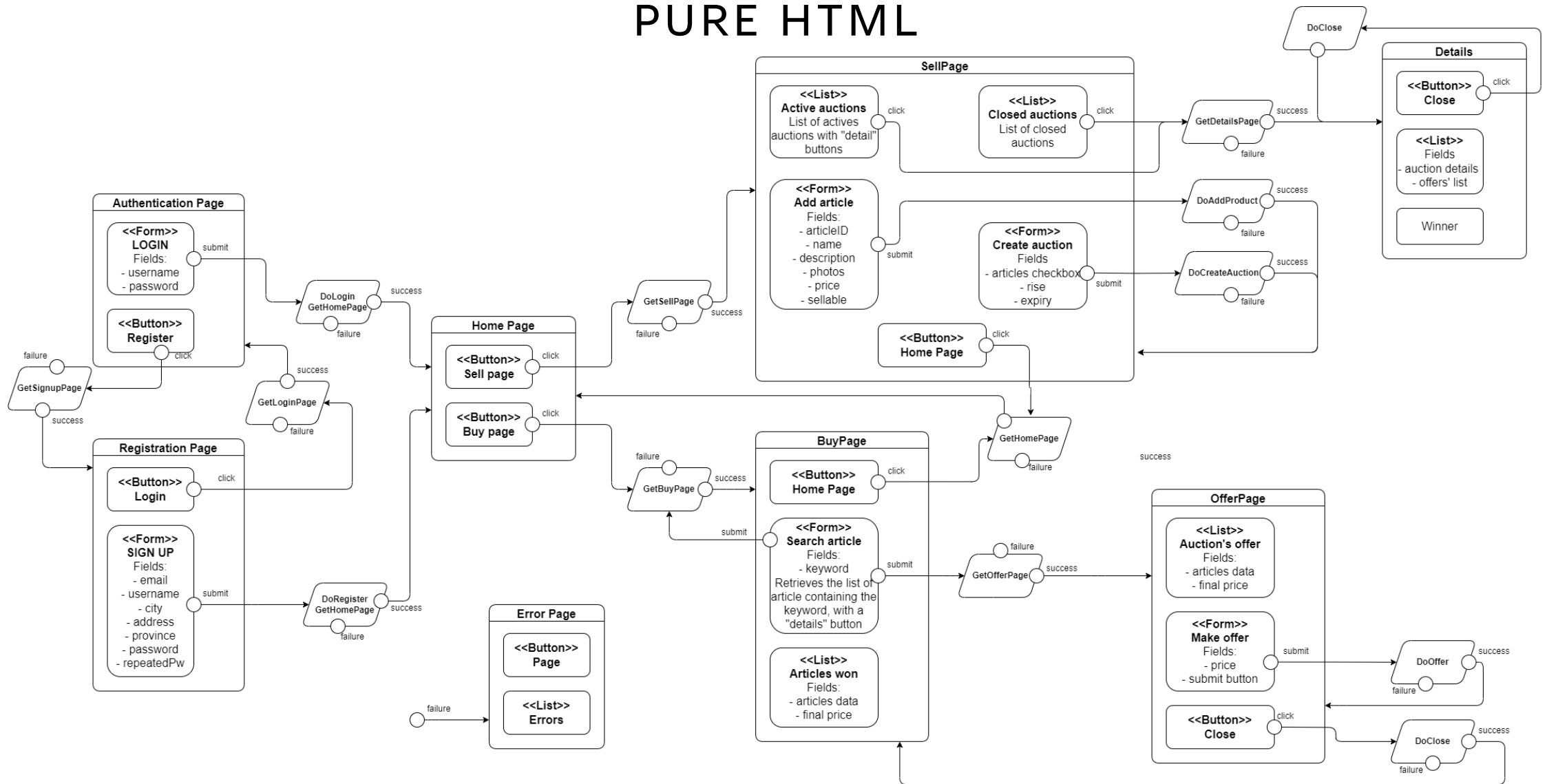
Event & Action



IPOTESI

- Username: stringa alfanumerica tra quattro e sei caratteri
- Password: stringa alfanumerica con almeno un simbolo speciale, una lettera e un numero da otto a sedici caratteri
- Email: stringa spezzata da una chiocciola con lunghezza massima duecentocinquantasei
- Città e provincia: stringa alfabetica da uno a trentadue caratteri
- Indirizzo: stringa con al più sessantaquattro caratteri
- Nome del prodotto: stringa con lettere e spazi con massimo trentadue caratteri
- Descrizione: stringa con al più duecentocinquantasei caratteri
- Data di scadenza: data successiva alla corrente
- Offerta: offerta in valore numerico intero

PURE HTML



PURE HTML

Data Access objects:

- AuctionDAO: createAuction (price, rise, expiry, userID), getUserAuctions (userID, active, loginTime), getAuctionsById (userID, loginTime), getWinner (auctionID), getWonAuctions (userID), getAuctionByKeyword (article, userID, loginTime), close (auctionID, userID), isAuctionOwner (userID, auctionID), isAuctionActive (auctionID)
- OfferDAO: getOffers (auctionID), addOffer (offer, userID, auctionID, date), getMaxOffer (auctionID)
- ProductDAO: addProduct (name, description, price, userID), getUserProducts (userID), update (productid, auctionID), getPrice (productID), checkProduct (productid, userID), getProductFromAuction (auctionID)
- UserDao: authenticate (username, password), addUser (username, email, city, address, province, password), getUser (userID), getUserByUsername (username), getUserByEmail (email)

Model objects:

- AuctionBean
- OfferBean
- ProductBean
- UserBean

Controllers:

- DoAddProduct
- DoClose
- DoCreateAuction
- DoGetImage
- DoLogin
- DoLogout
- DoOffer
- DoSignup

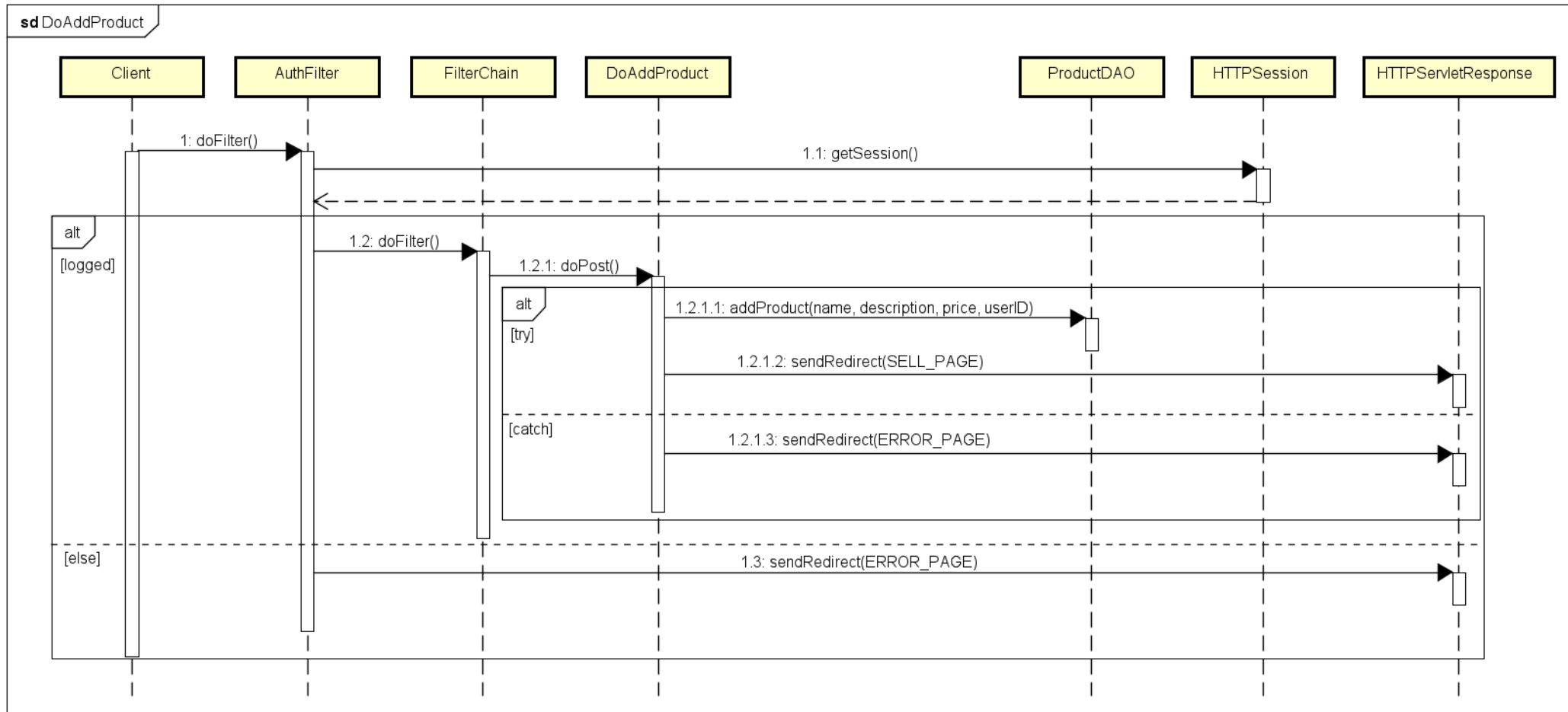
Filters:

- AuthFilter
- OfferFilter
- UnAuthFilter

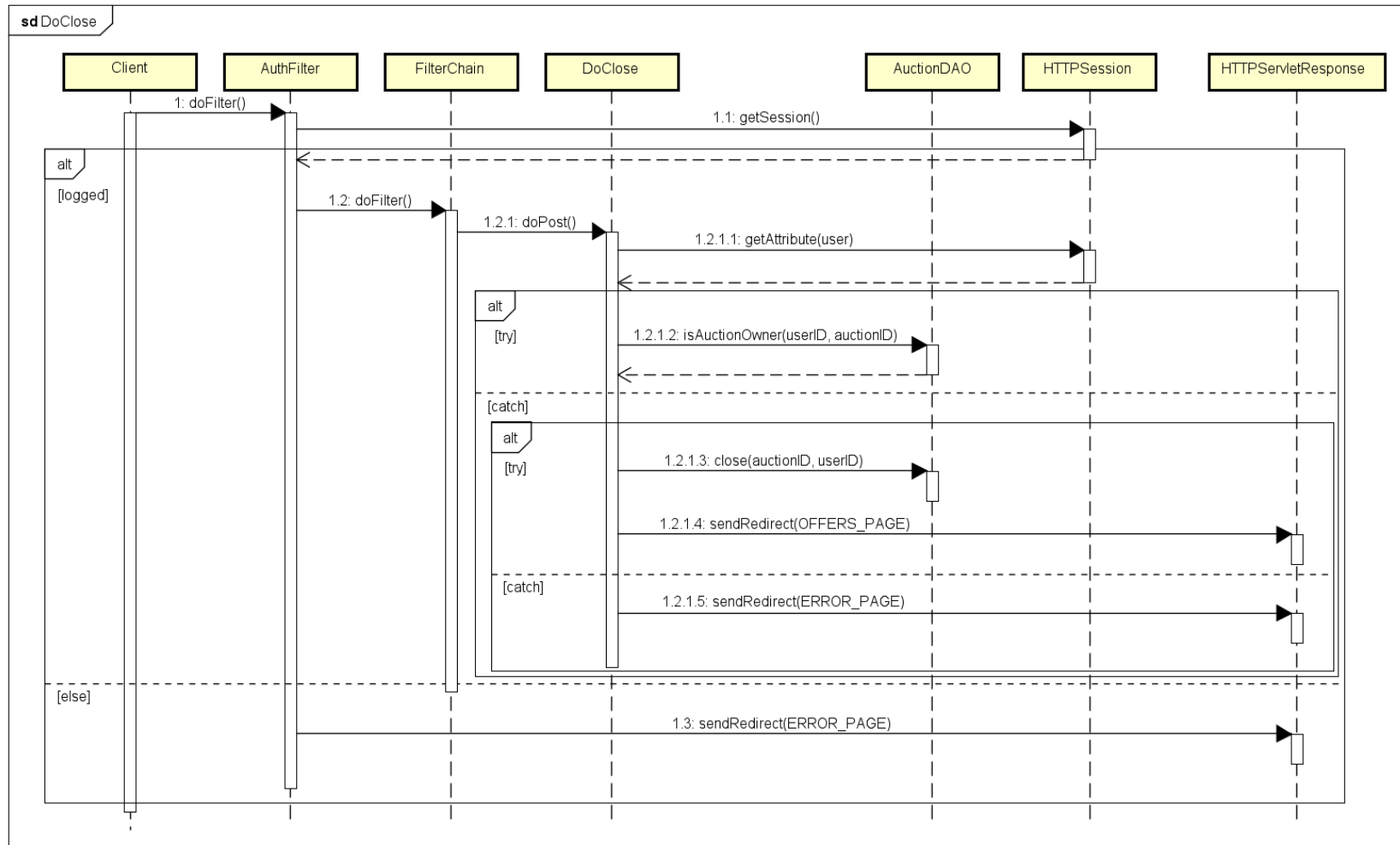
Views:

- GetBuyPage
- GetDetailsPage
- GetErrorPage
- GetHomePage
- GetLoginPage
- GetOffersPage
- GetSellPage
- GetSignupPage

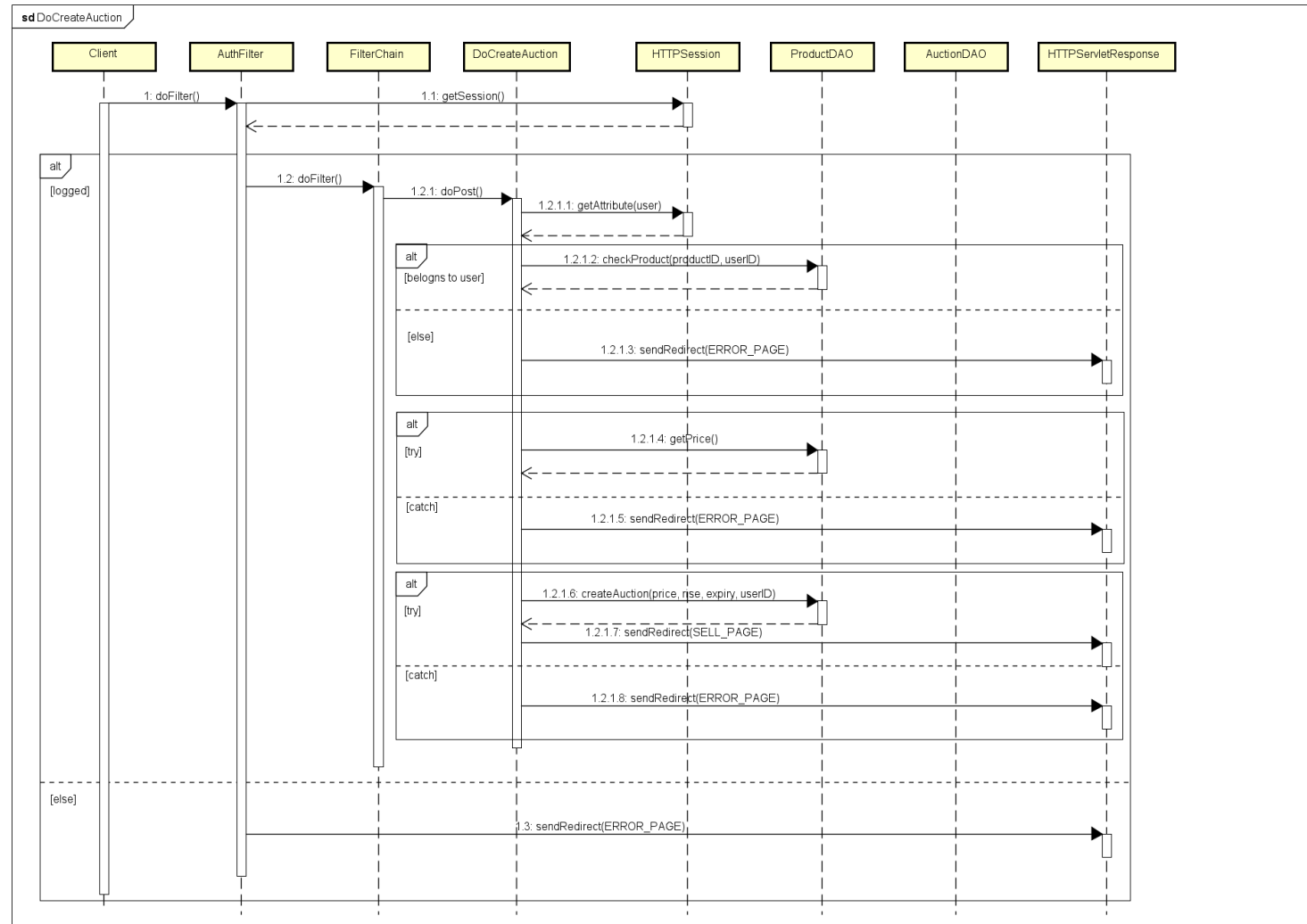
PURE HTML



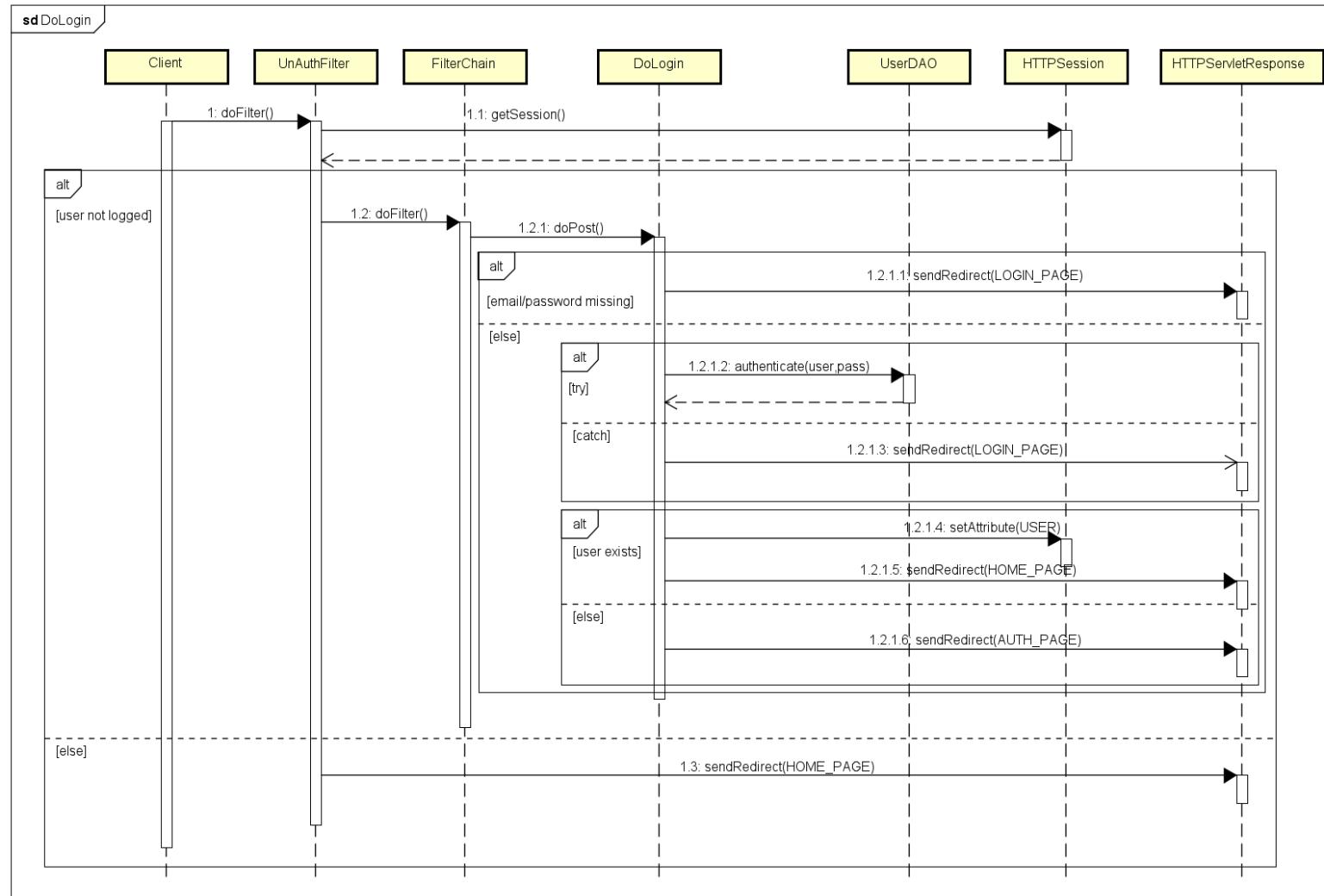
PURE HTML



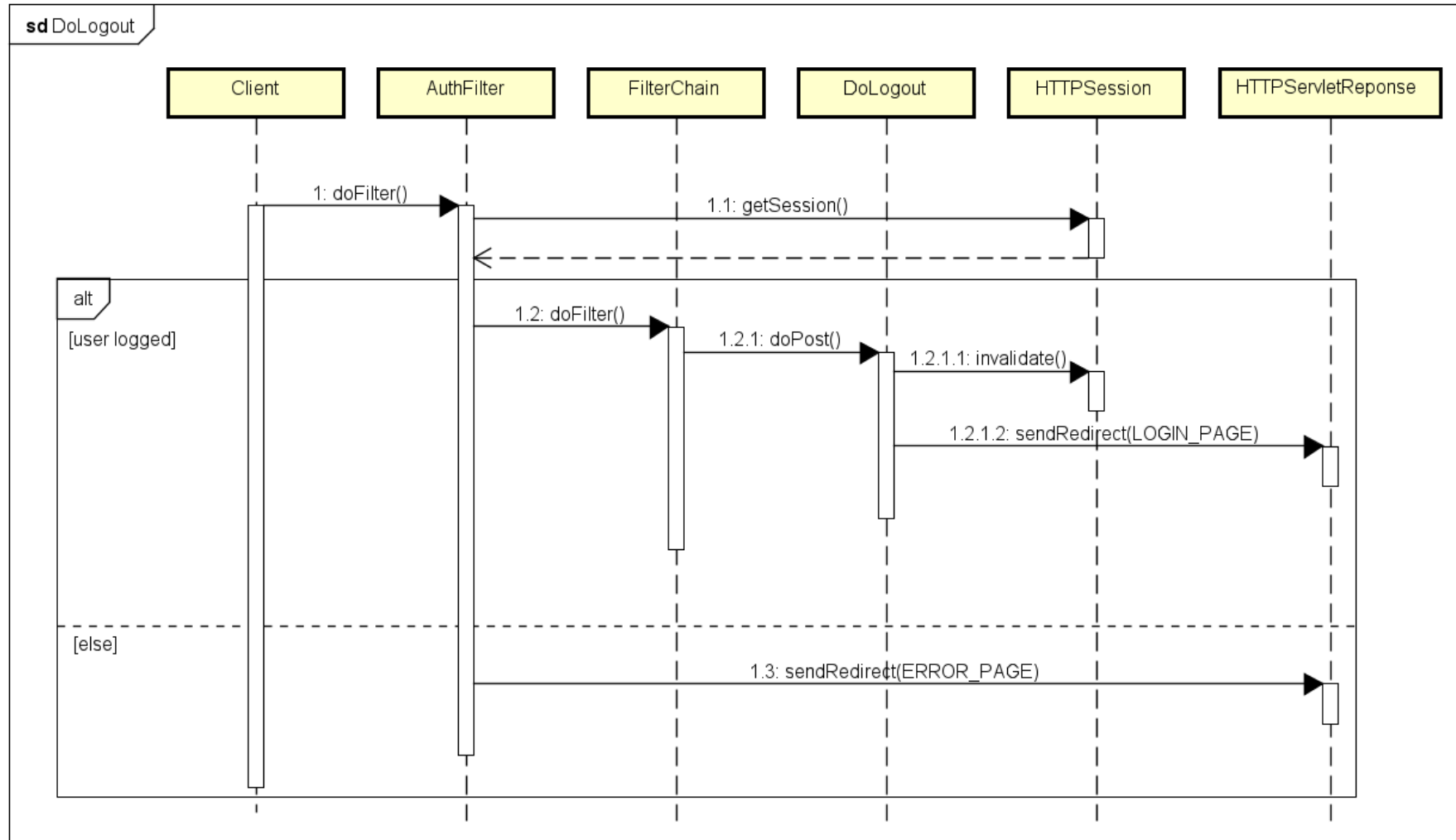
PURE HTML



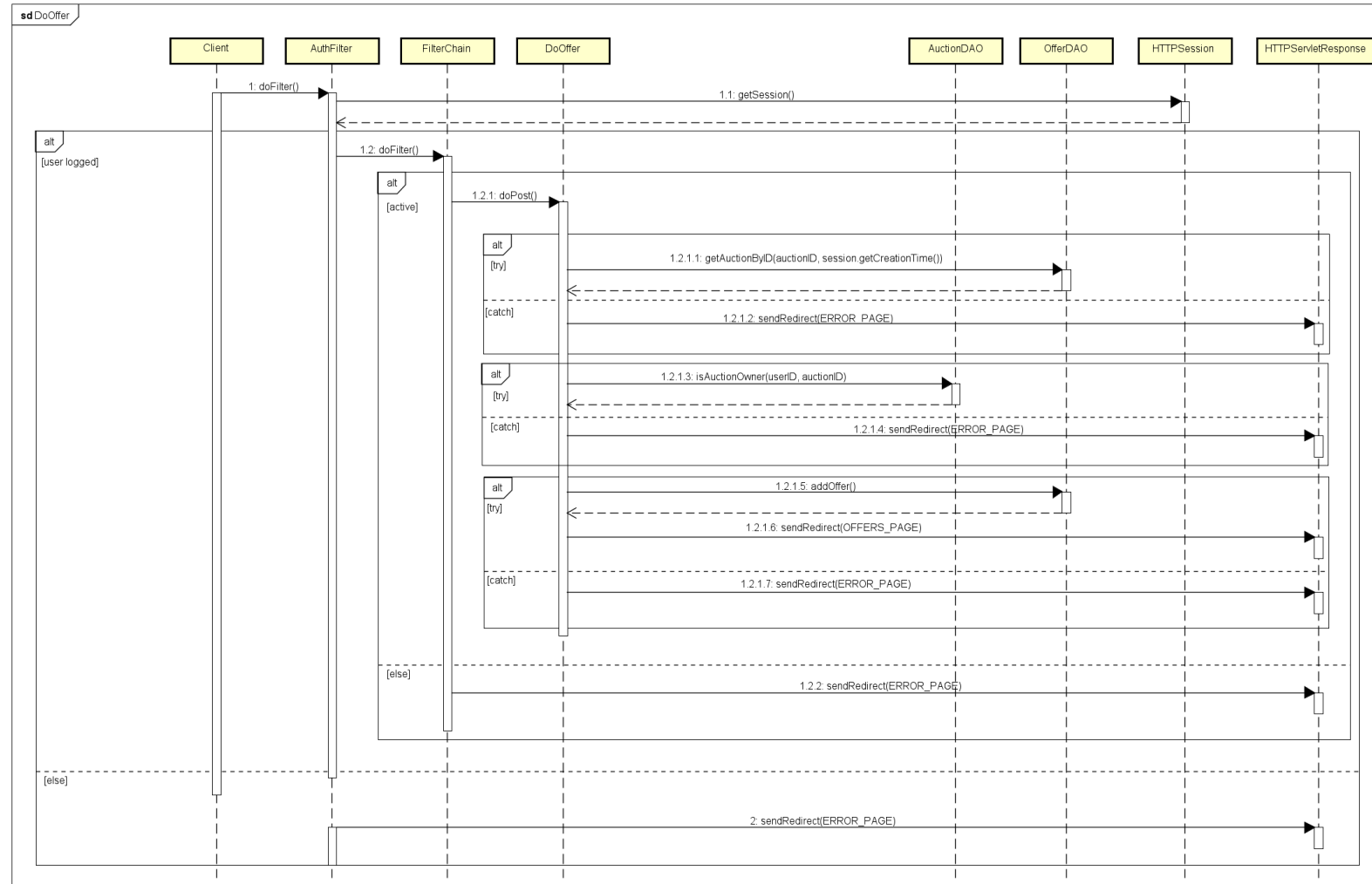
PURE HTML



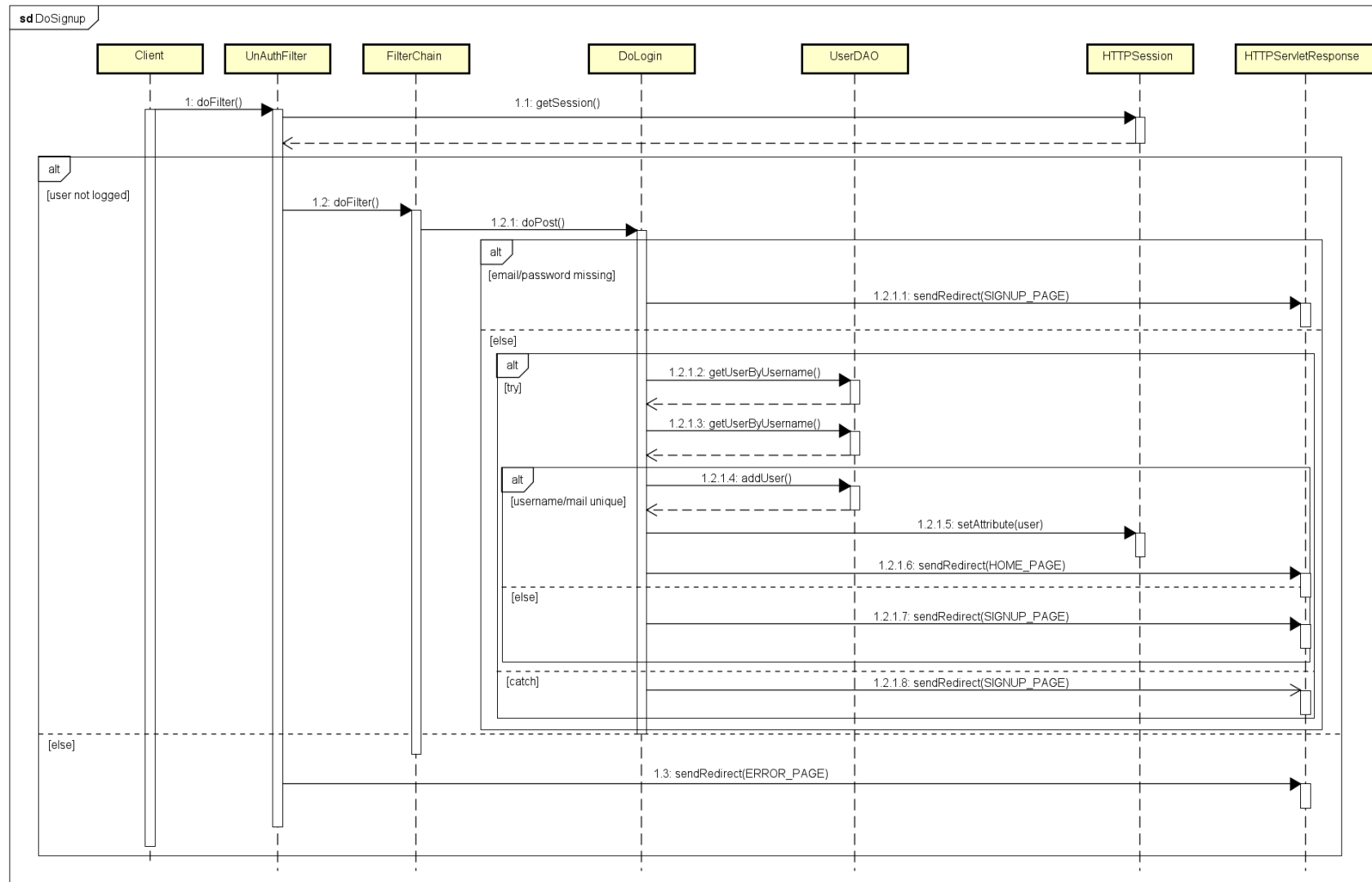
PURE HTML



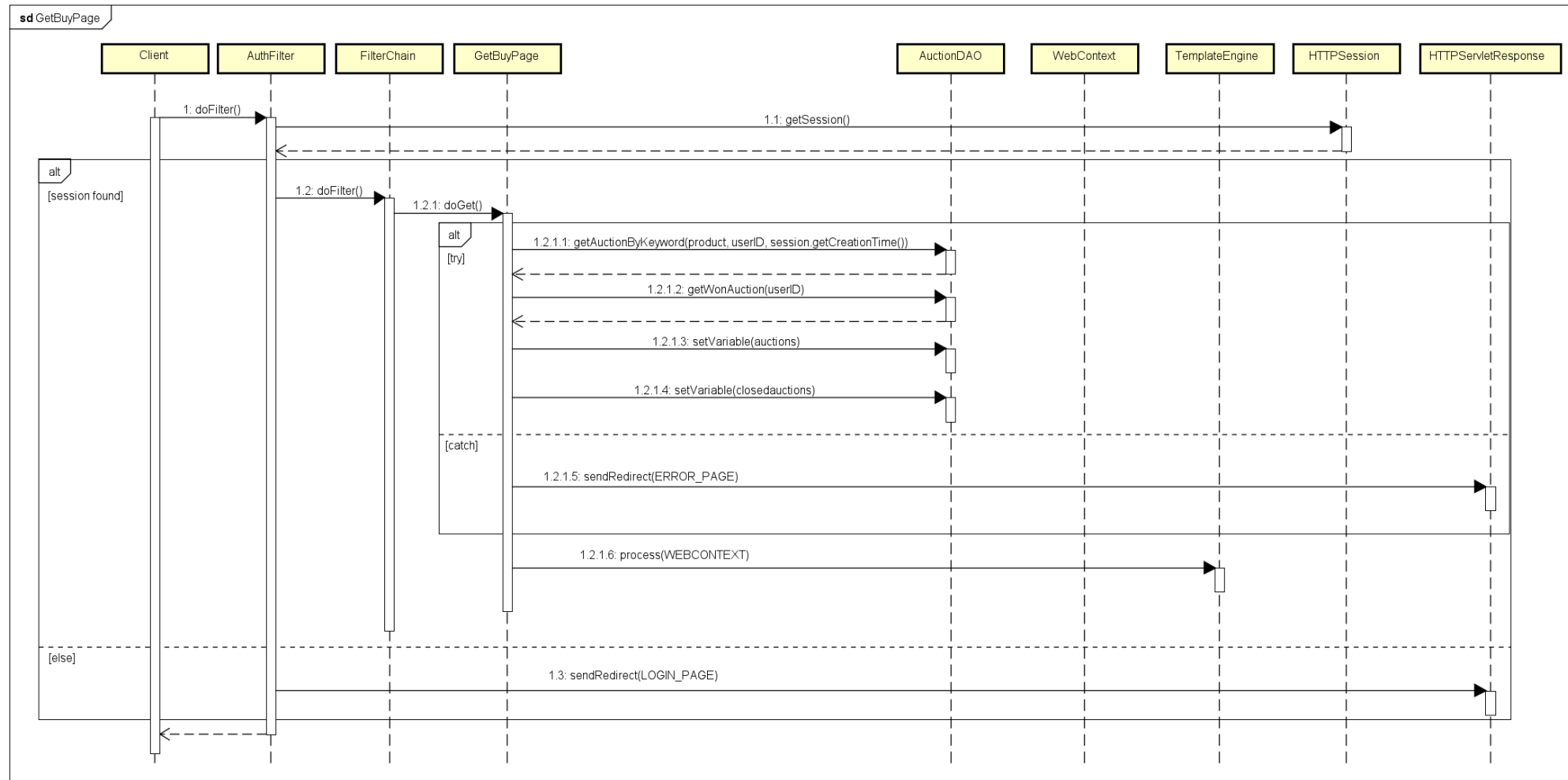
PURE HTML



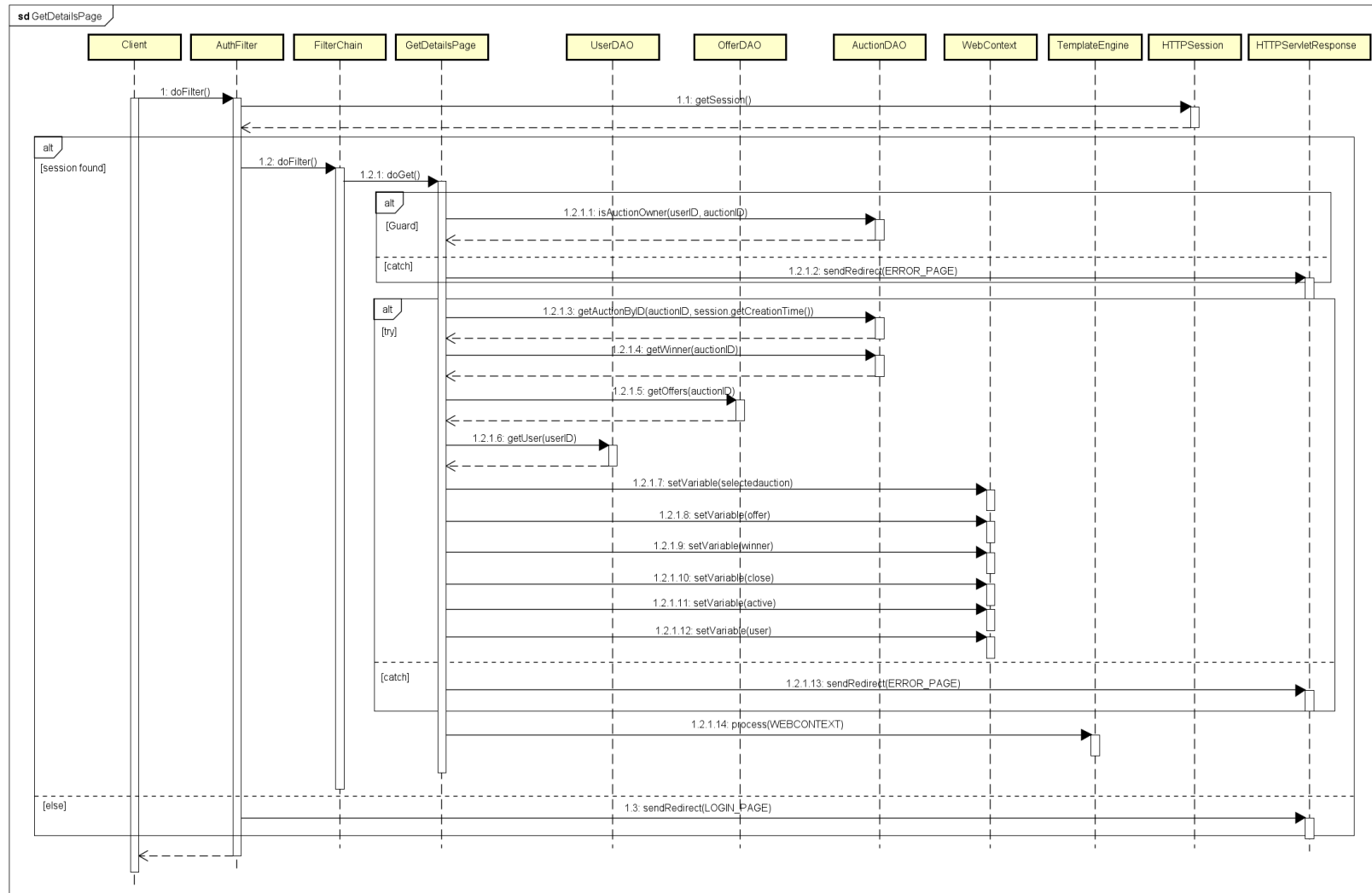
PURE HTML



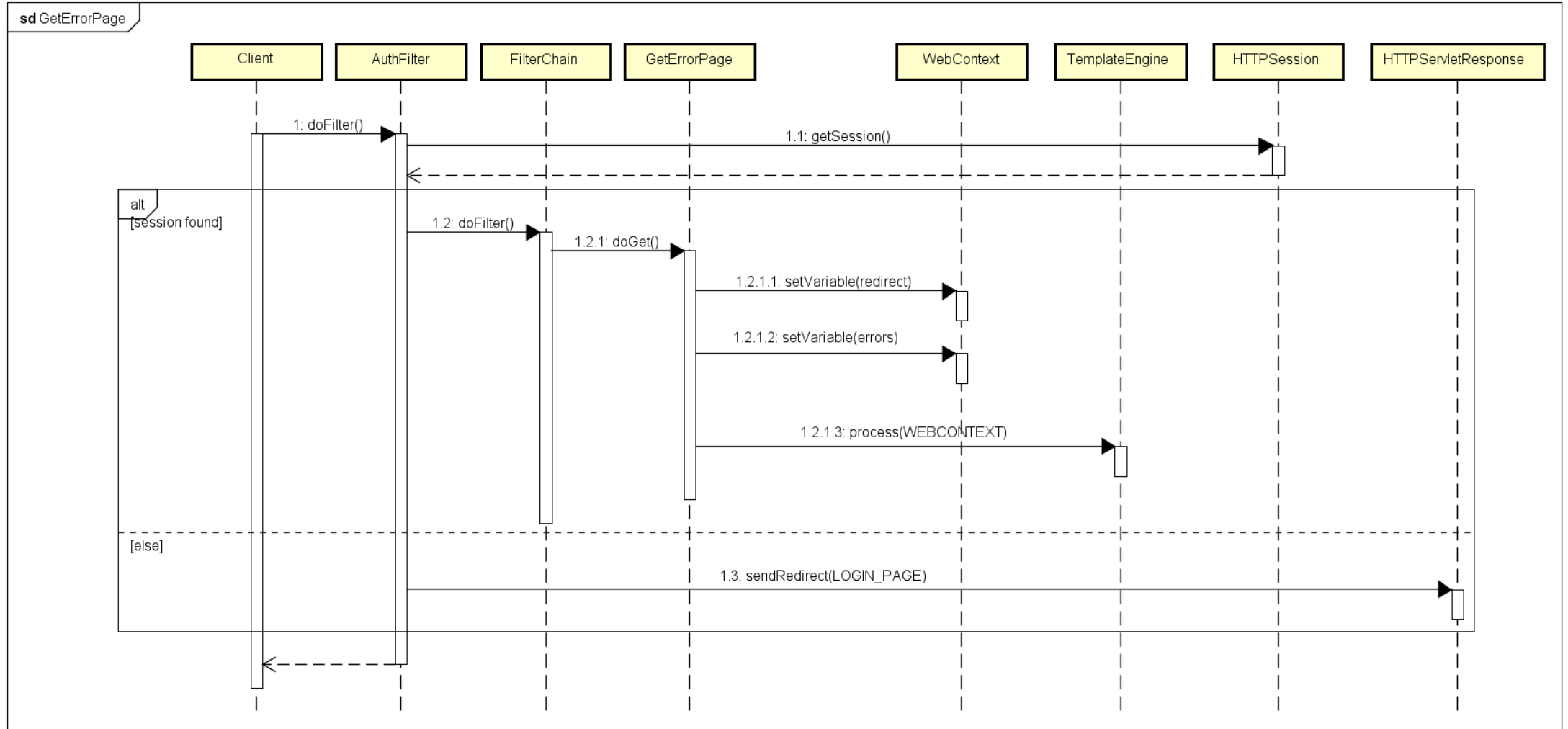
PURE HTML



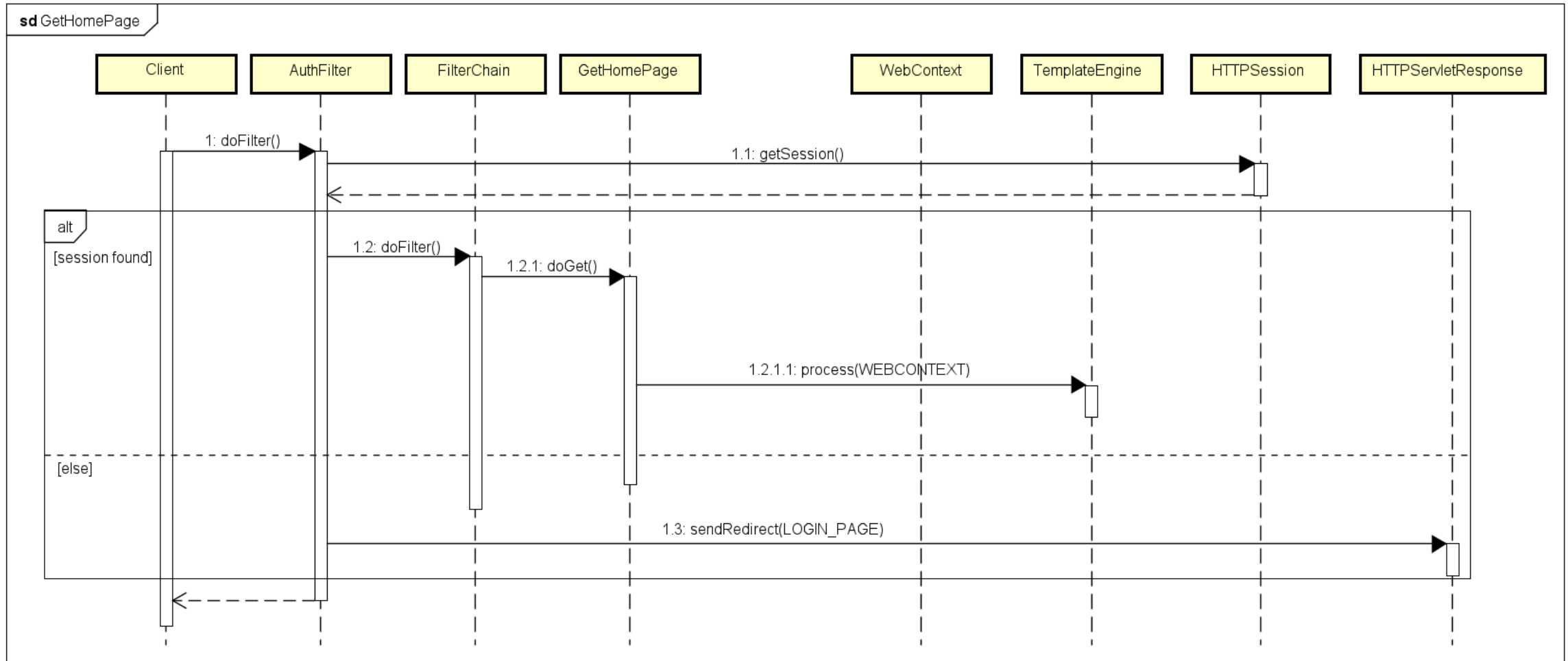
PURE HTML



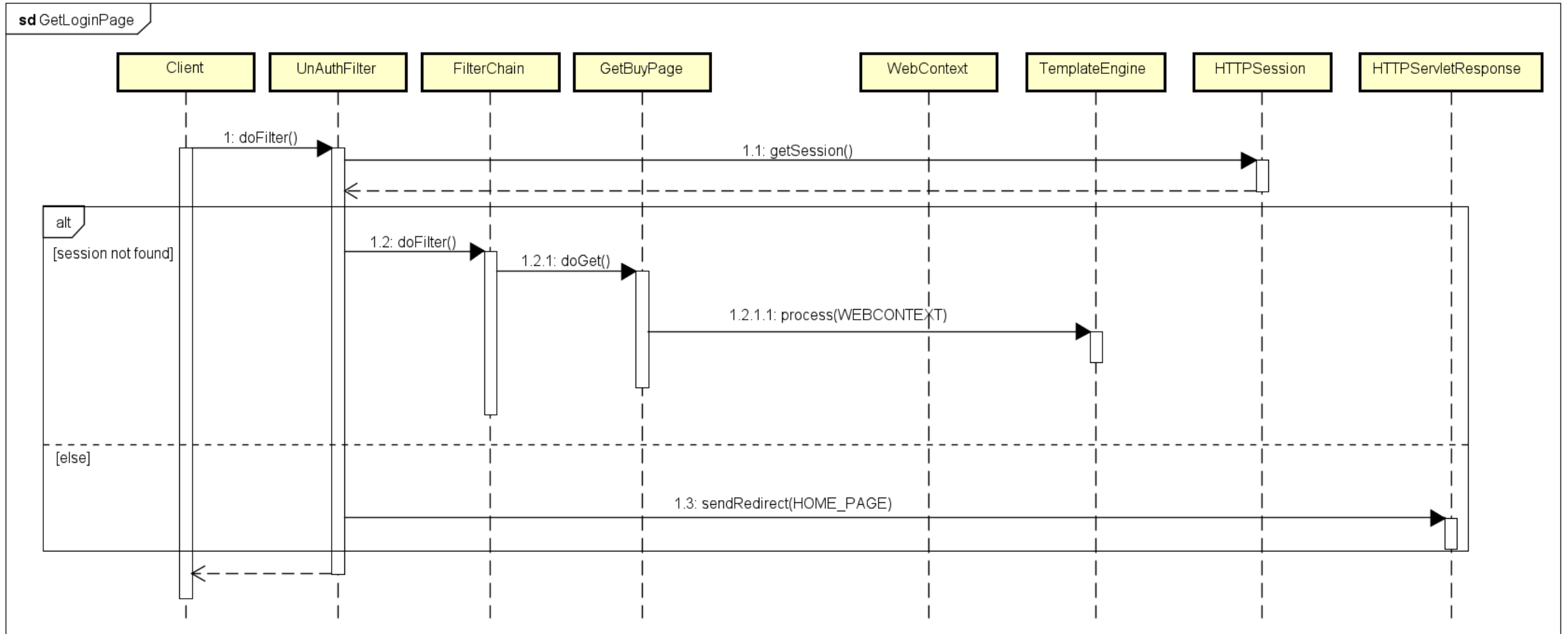
PURE HTML



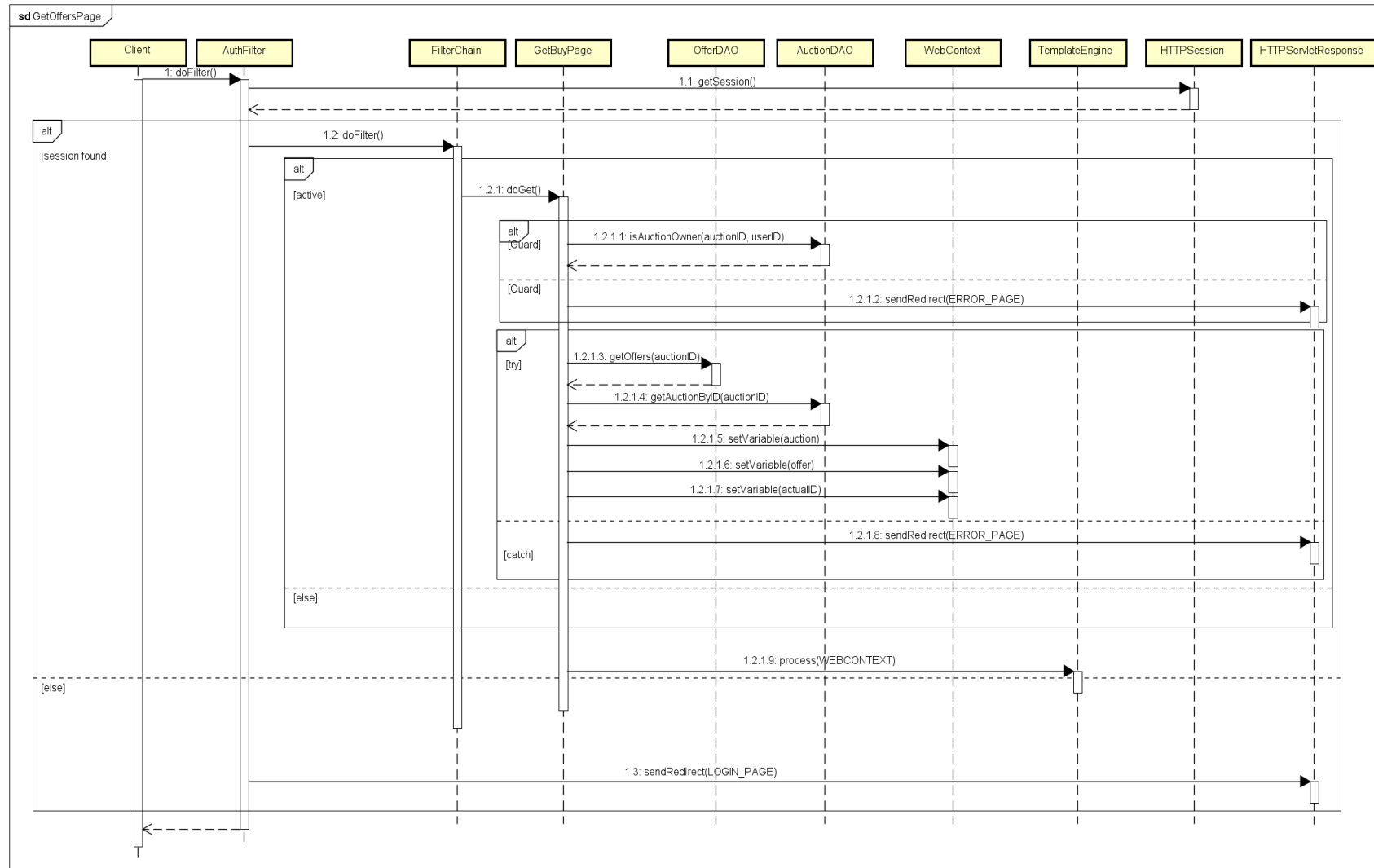
PURE HTML



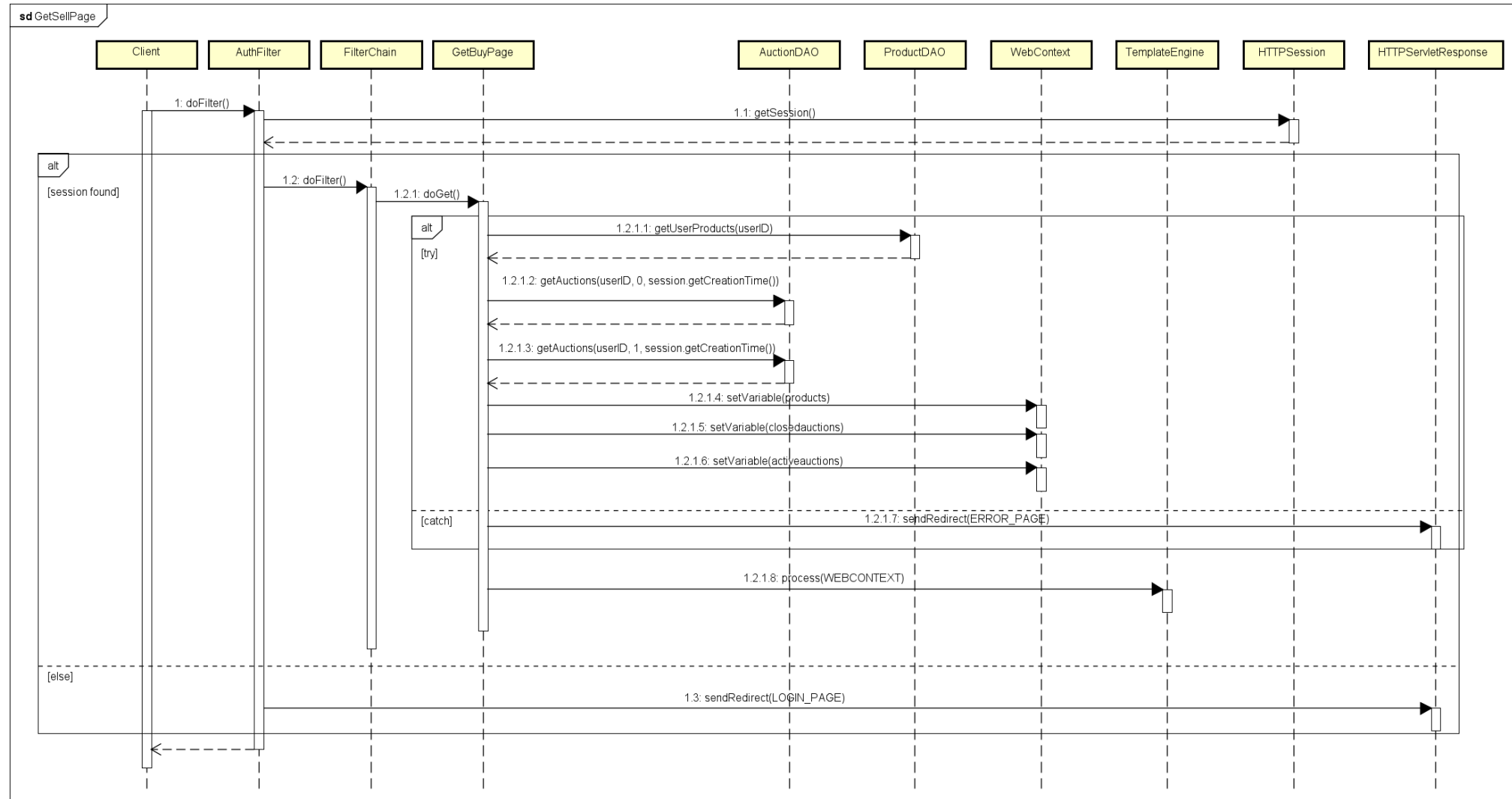
PURE HTML



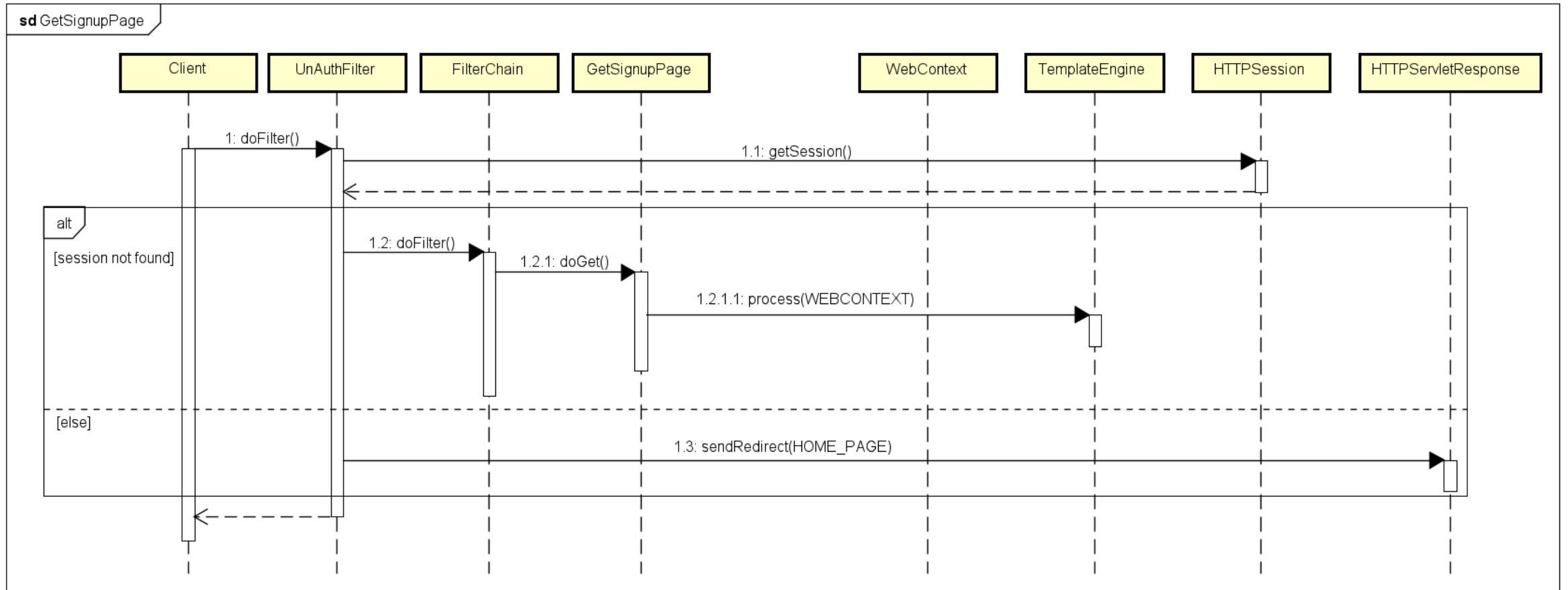
PURE HTML



PURE HTML



PURE HTML





JAVASCRIPT

Model objects:

- AuctionBean
- OfferBean
- ProductBean
- UserBean

Data Access objects:

- AuctionDAO: createAuction (price, rise, expiry, userID), getUserAuctions (userID, active, loginTime), getWinner (auctionID), isAuctionOwner (userID, auctionID), isAuctionActive (auctionID), getWonAuctions (userID), getAuctionByKeyword (article, userID, loginTime), close (auctionID, userID)
- OfferDAO: getOffers (auctionID), addOffer (offer, userID, auctionID, date), getMaxOffer (auctionID)
- ProductDAO: addProduct (name, description, price, userID), getUserProducts (userID), update (productid, auctionID), getPrice (productID), checkProduct (productid, userID), getProductFromAuction (auctionID)
- UserDAO: authenticate (username, password), addUser (username, email, city, address, province, password), getUser (userID), getUserByUsername (username), getUserByEmail (email)

Model objects:

- AuctionBean
- OfferBean
- ProductBean
- UserBean

Filters:

- AuthFilter
- UnAuthFilter

Controllers:

- DoAddProduct
- DoClose
- DoCreateAuction
- DoGetImage
- DoLogin
- DoLogout
- SoSignup
- GetAuctionDetails
- GetMatchingAuctions
- GetSearchedAuctions
- GetSell
- GetWonAuctions
- LoadDefaultPage

JAVASCRIPT

Data Access objects:

- AuctionDAO: createAuction (price, rise, expiry, userID), getUserAuctions (userID, active, loginTime), getWinner (auctionID), isAuctionOwner (userID, auctionID), isAuctionActive (auctionID), getWonAuctions (userID), getAuctionByKeyword (article, userID, loginTime), close (auctionID, userID)
- OfferDAO: getOffers (auctionID), addOffer (offer, userID, auctionID, date), getMaxOffer (auctionID)
- ProductDAO: addProduct (name, description, price, userID), getUserProducts (userID), update (productid, auctionID), getPrice (productID), checkProduct (productid, userID), getProductFromAuction (auctionID)
- UserDAO: authenticate (username, password), addUser (username, email, city, address, province, password), getUser (userID), getUserByUsername (username), getUserByEmail (email)

Model objects:

- AuctionBean
- OfferBean
- ProductBean
- UserBean

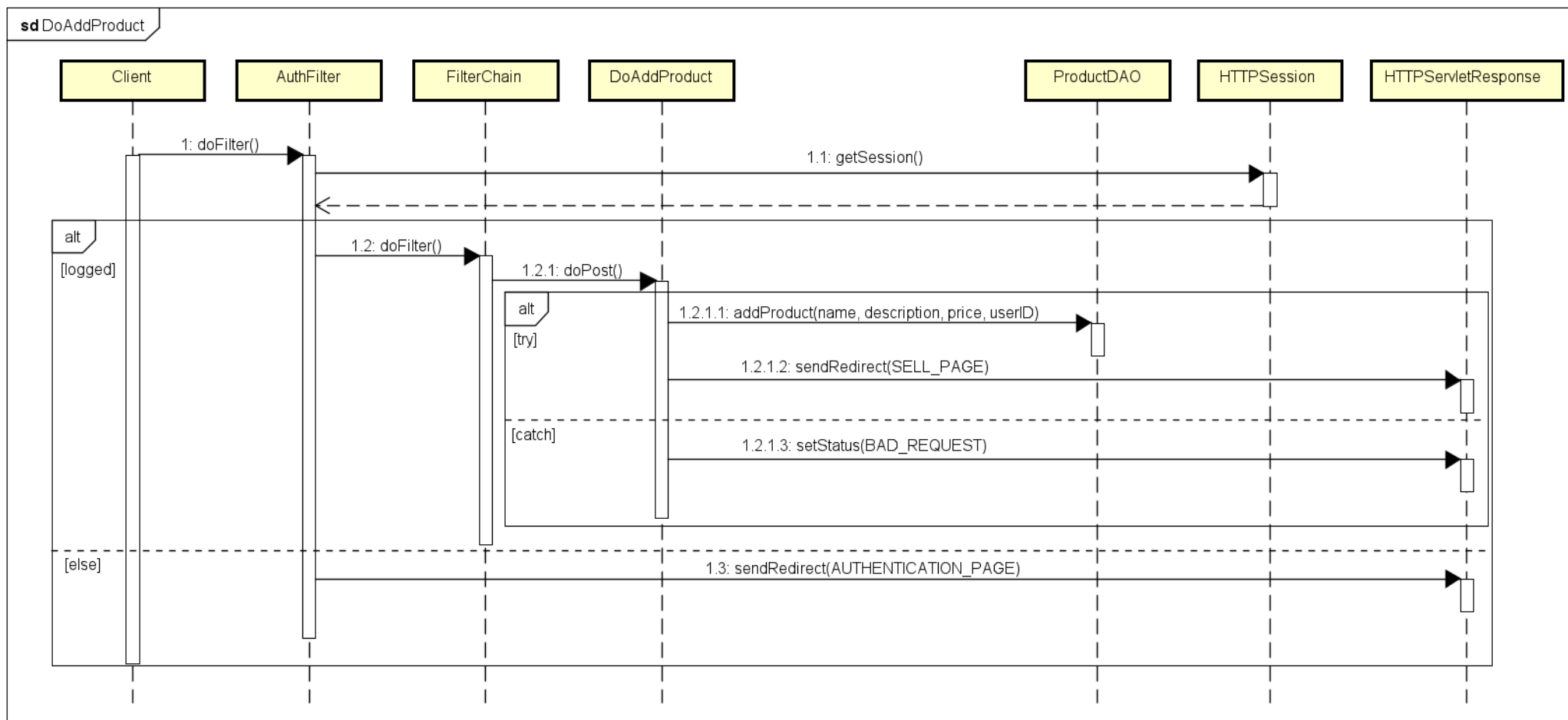
Filters:

- AuthFilter
- UnAuthFilter

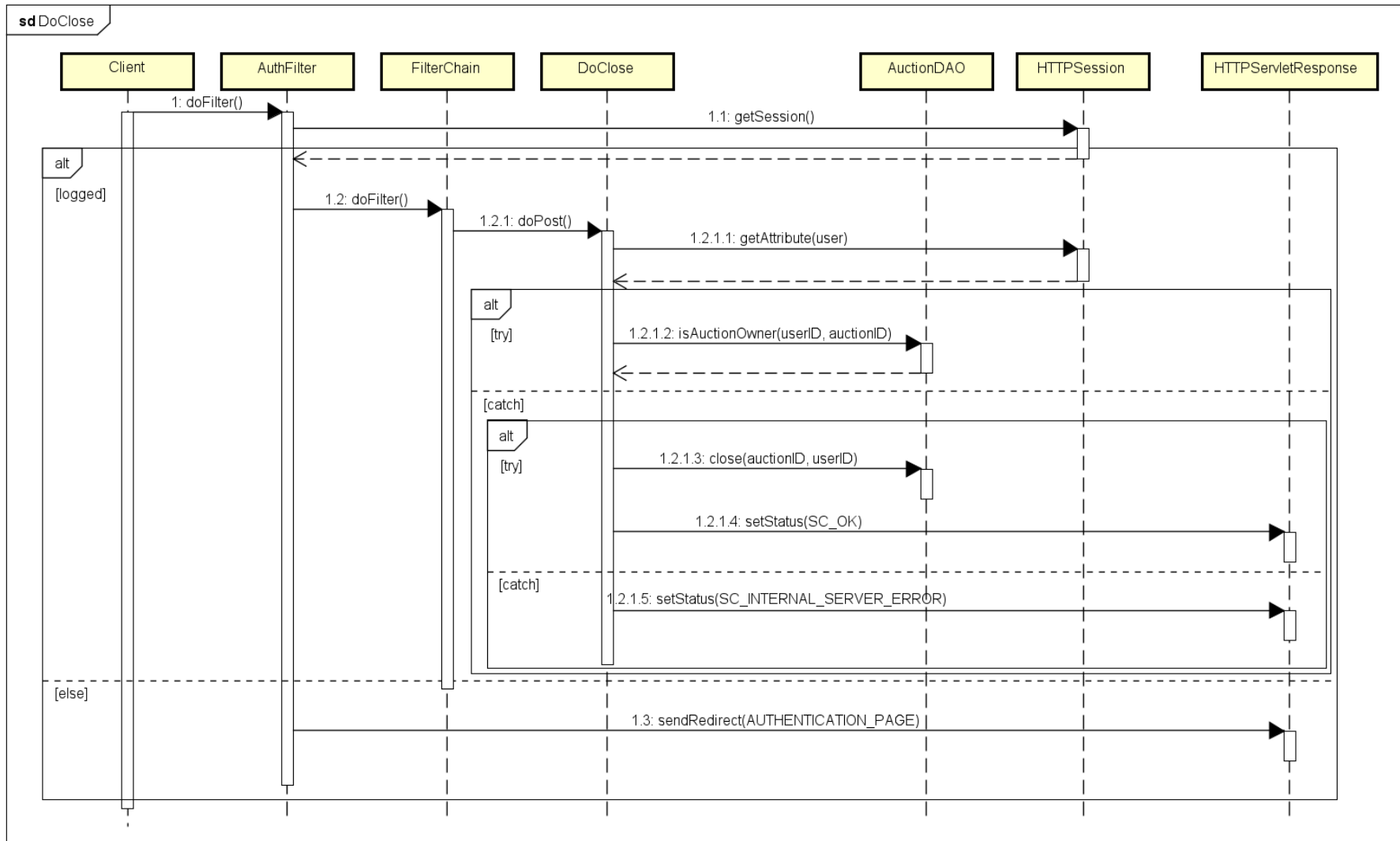
Controllers:

- DoAddProduct
- DoClose
- DoCreateAuction
- DoGetImage
- DoLogin
- DoLogout
- SoSignup
- GetAuctionDetails
- GetMatchingAuctions
- GetSearchedAuctions
- GetSell
- GetWonAuctions
- LoadDefaultPage

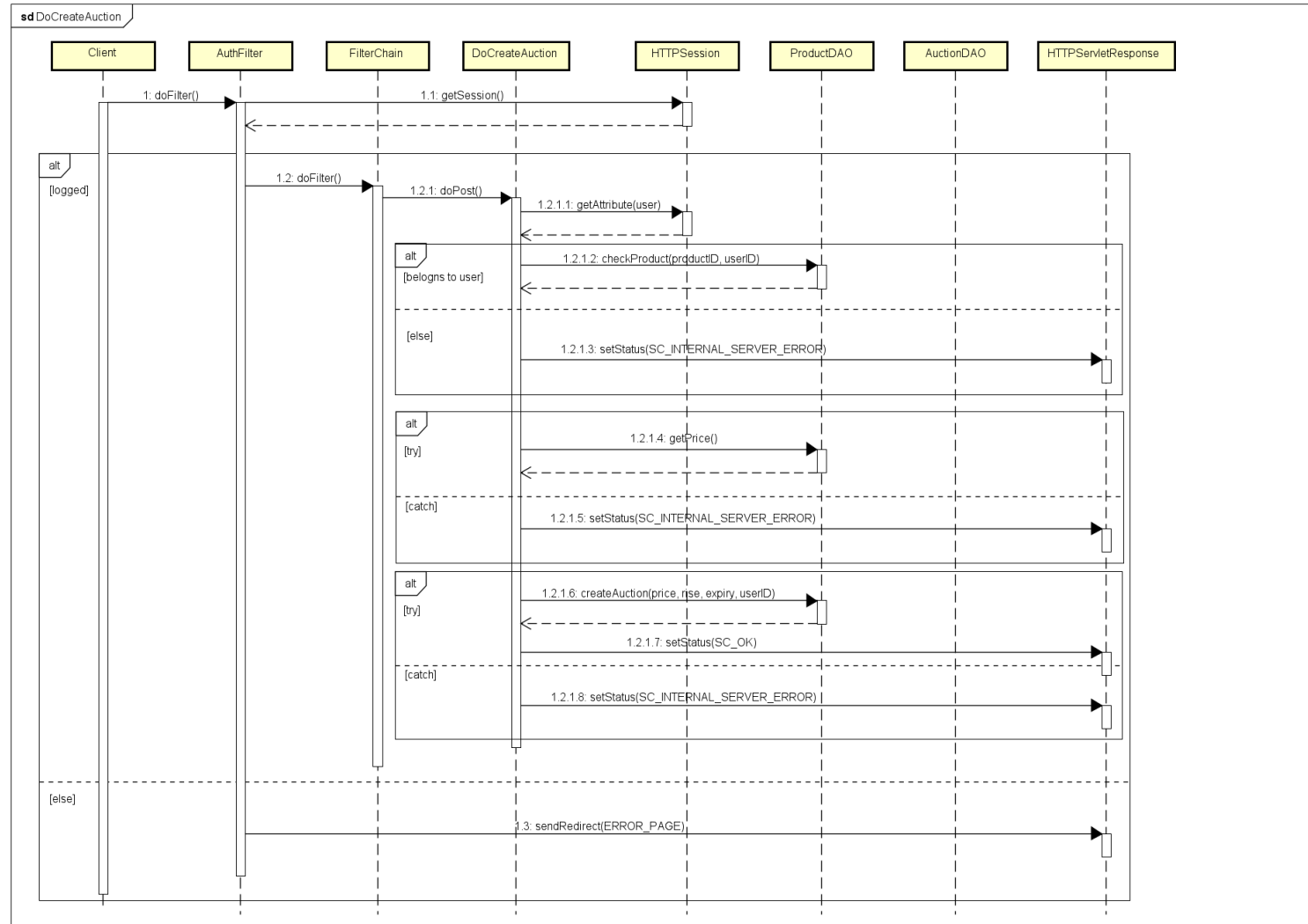
JAVASCRIPT



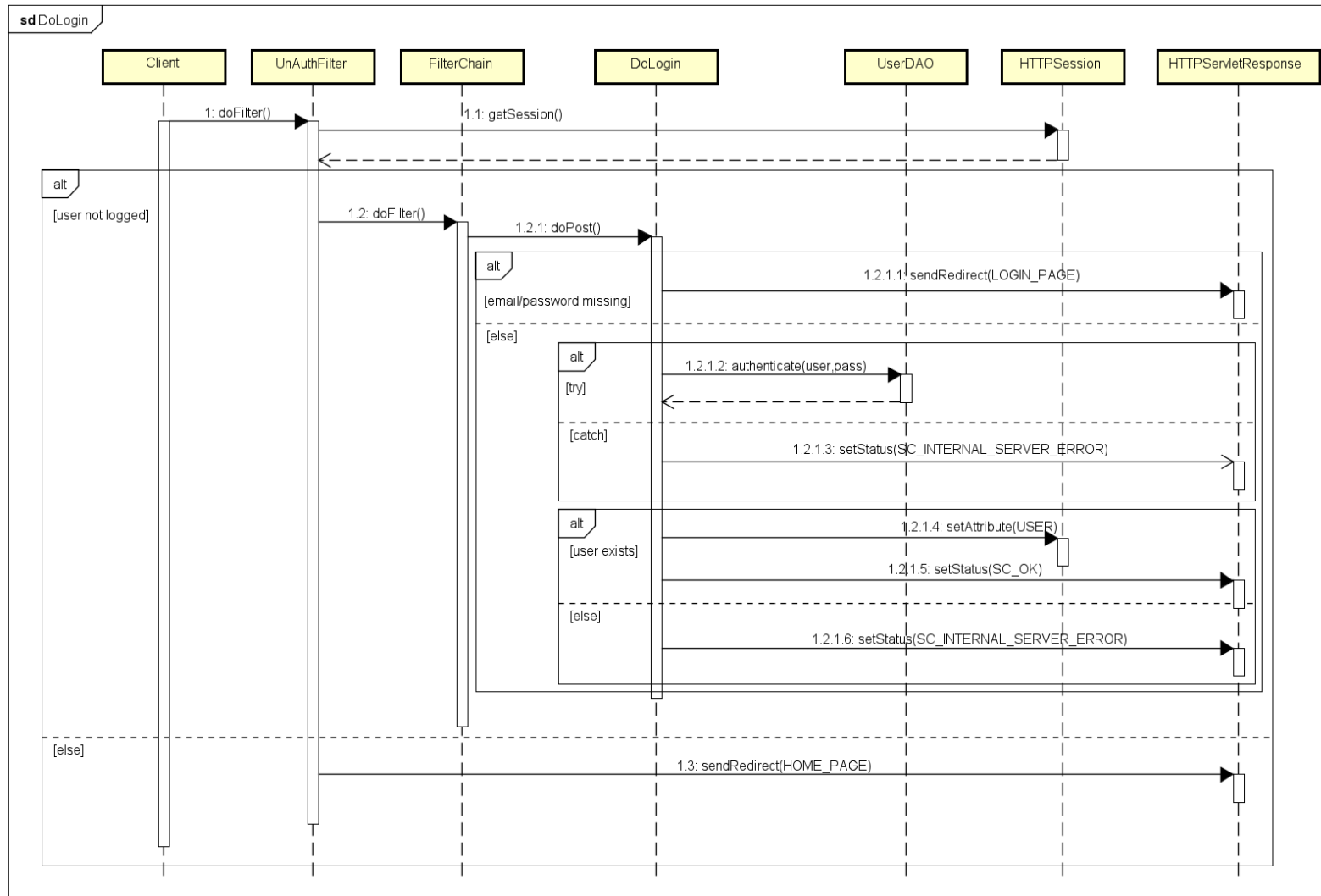
JAVASCRIPT



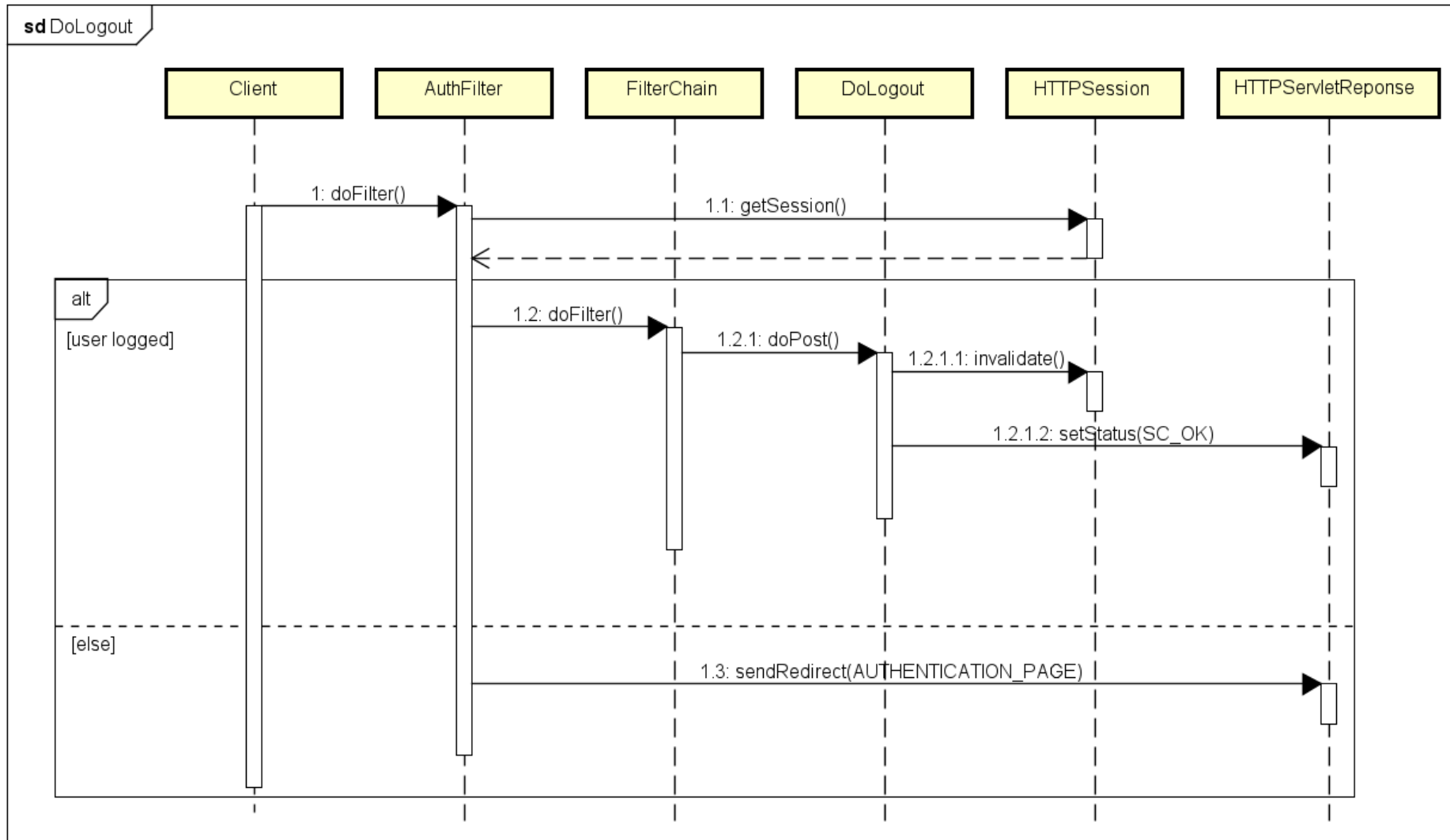
JAVASCRIPT



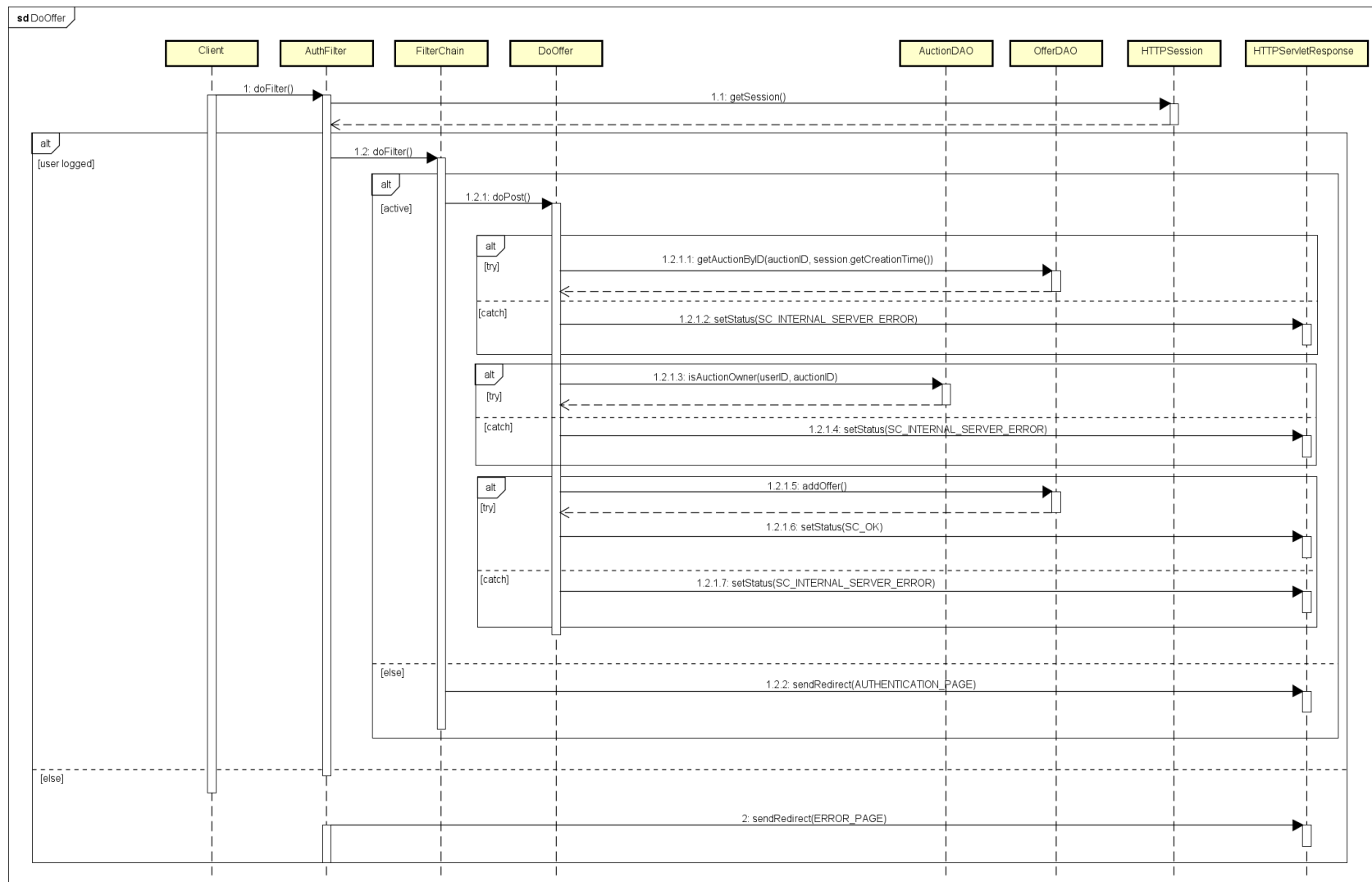
JAVASCRIPT



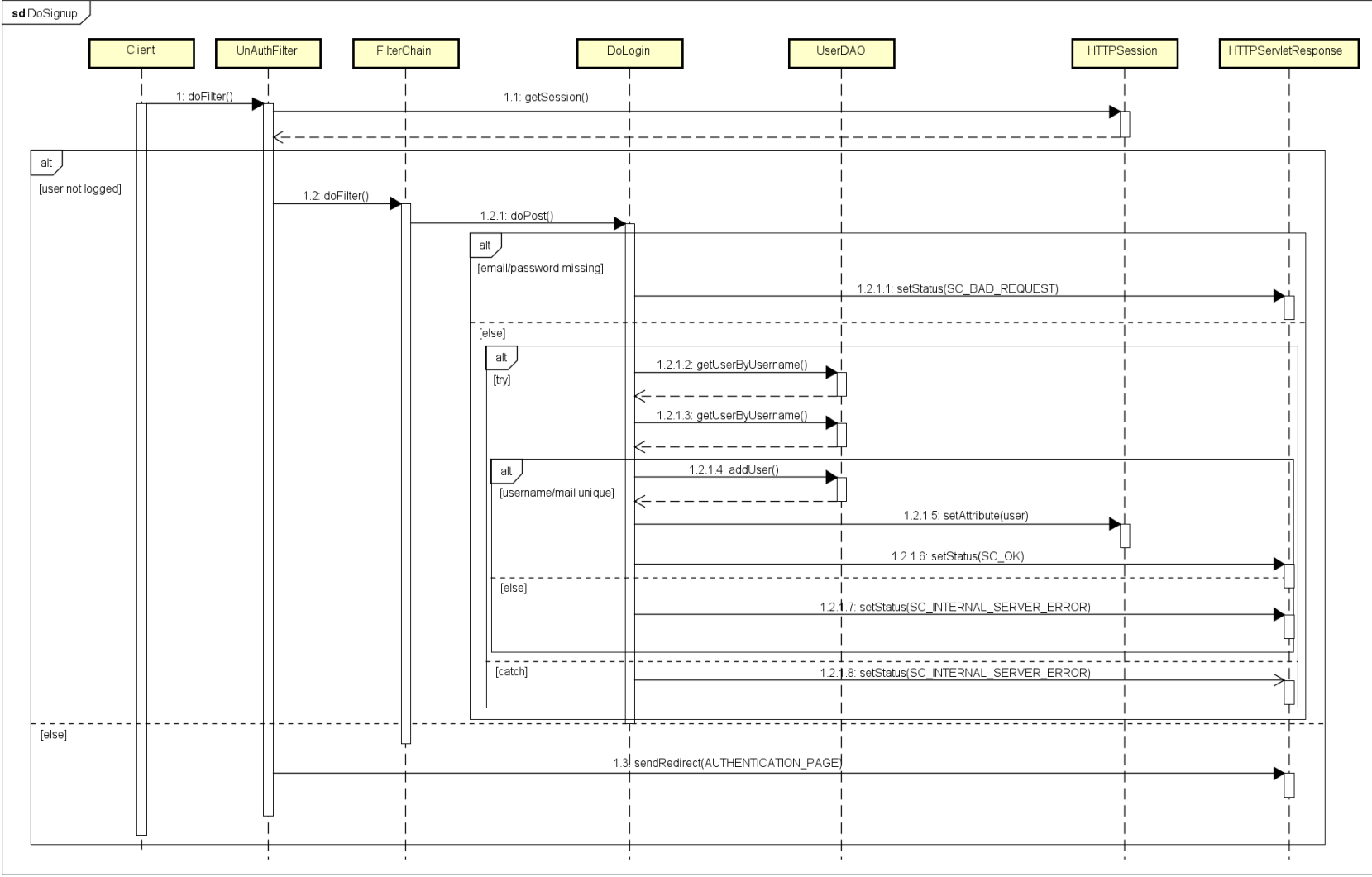
JAVASCRIPT



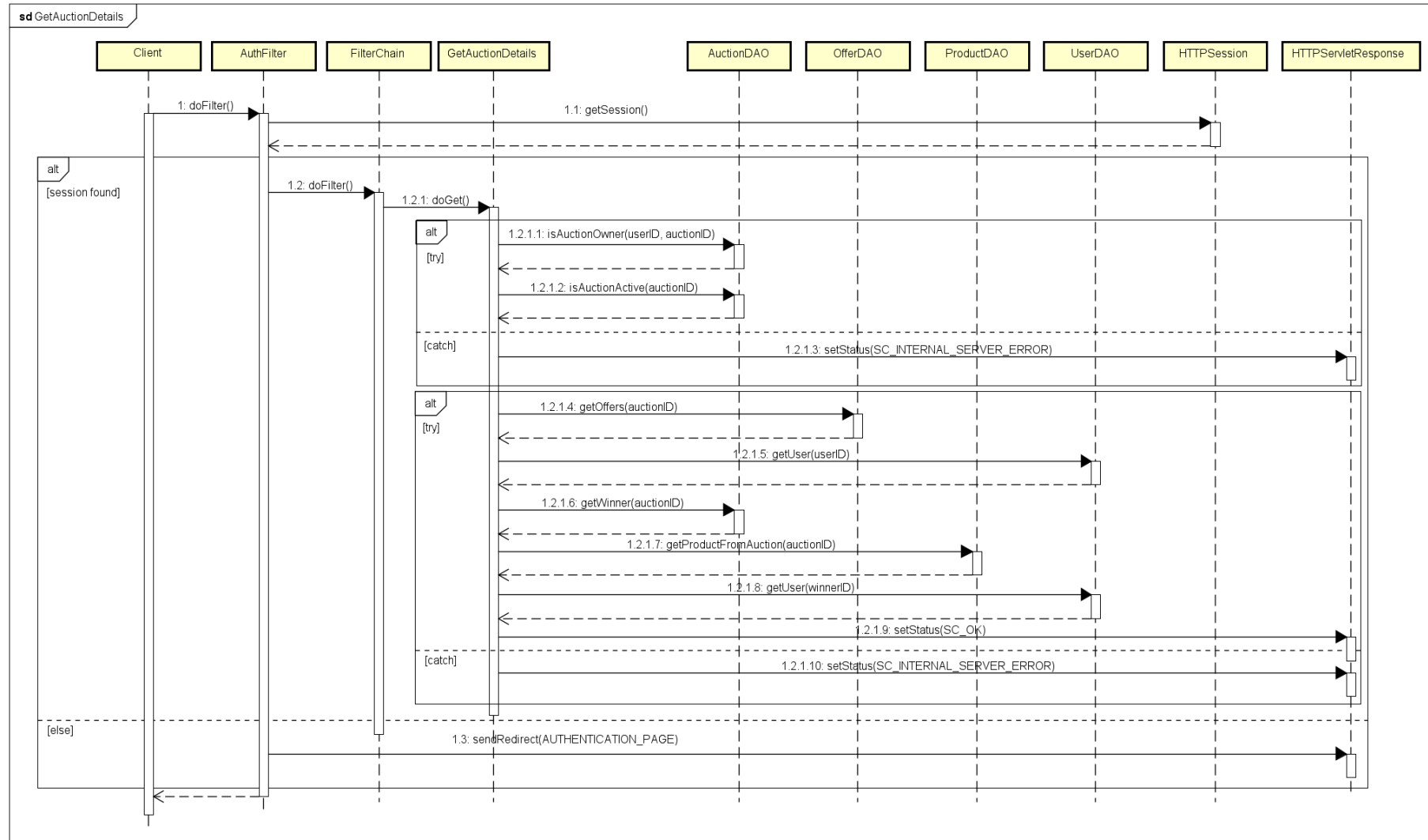
JAVASCRIPT



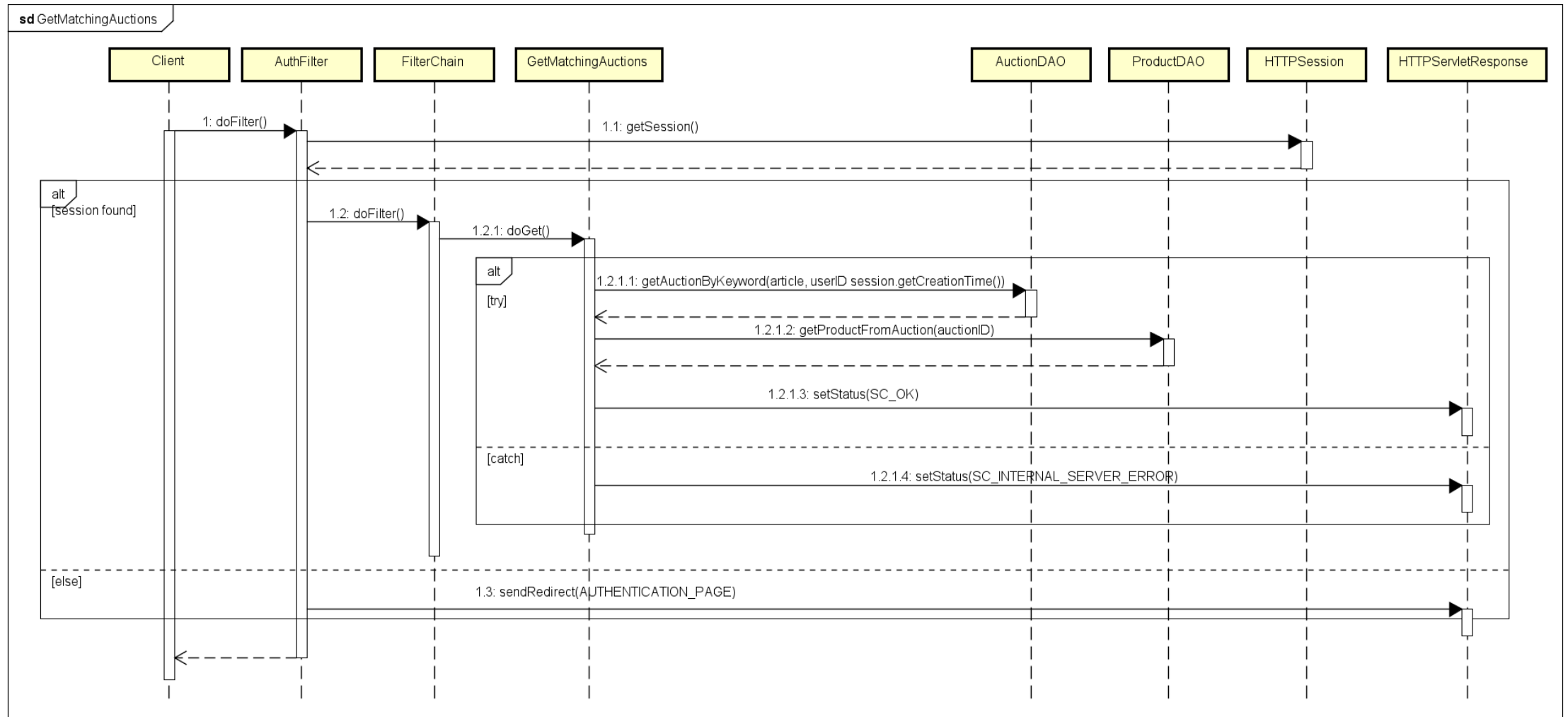
JAVASCRIPT



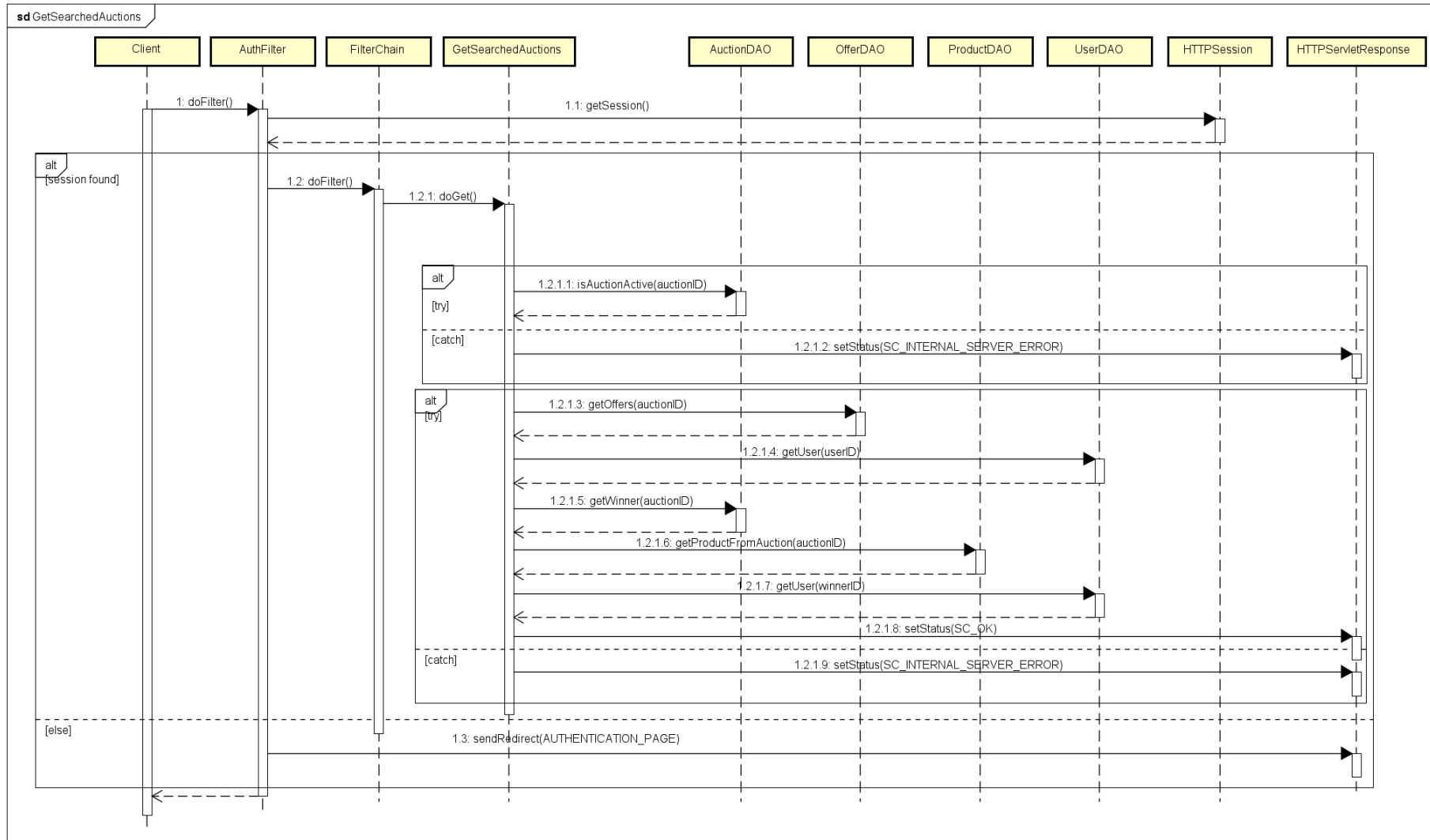
JAVASCRIPT



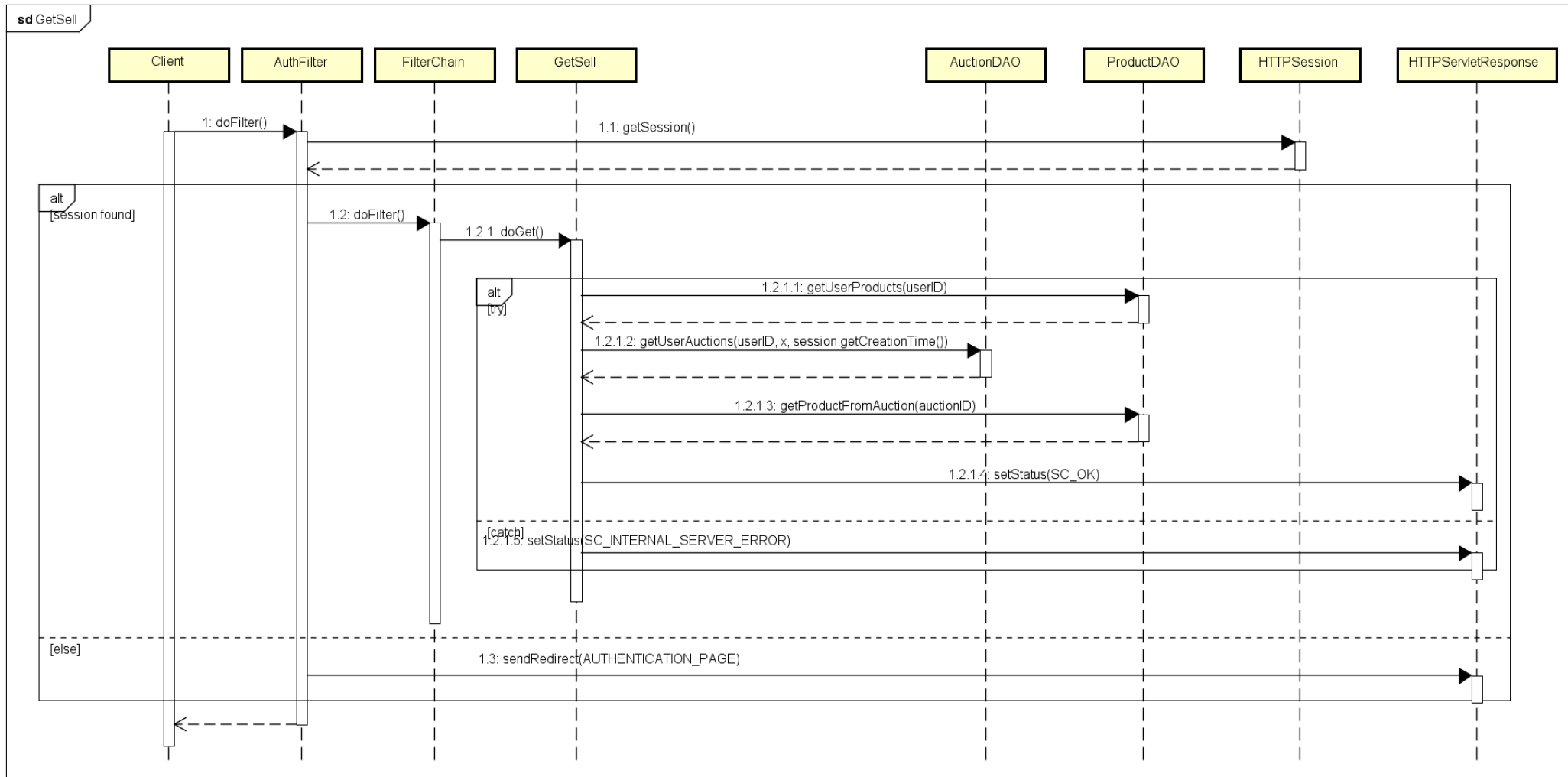
JAVASCRIPT



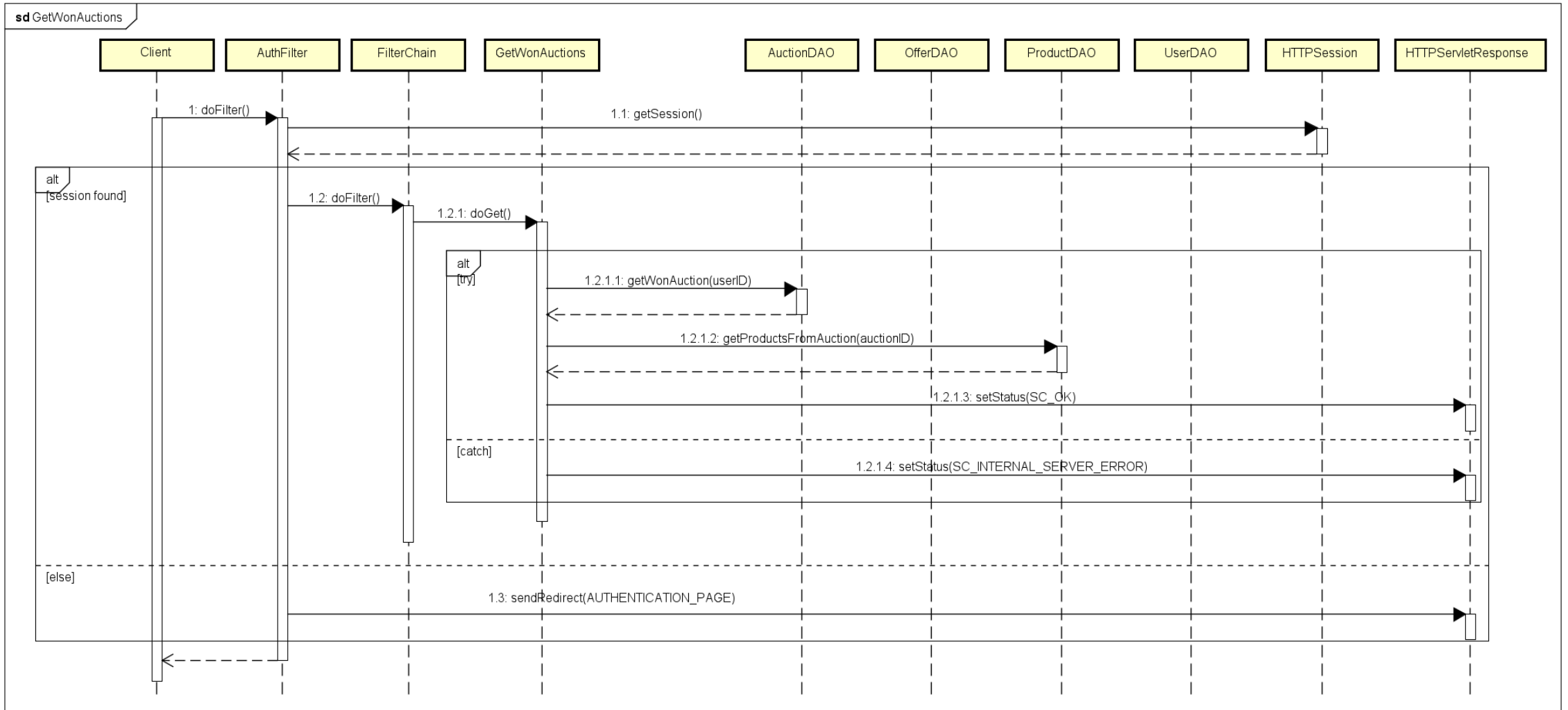
JAVASCRIPT



JAVASCRIPT



JAVASCRIPT



JAVASCRIPT

