

Pure HTML

ESERCIZIO 1

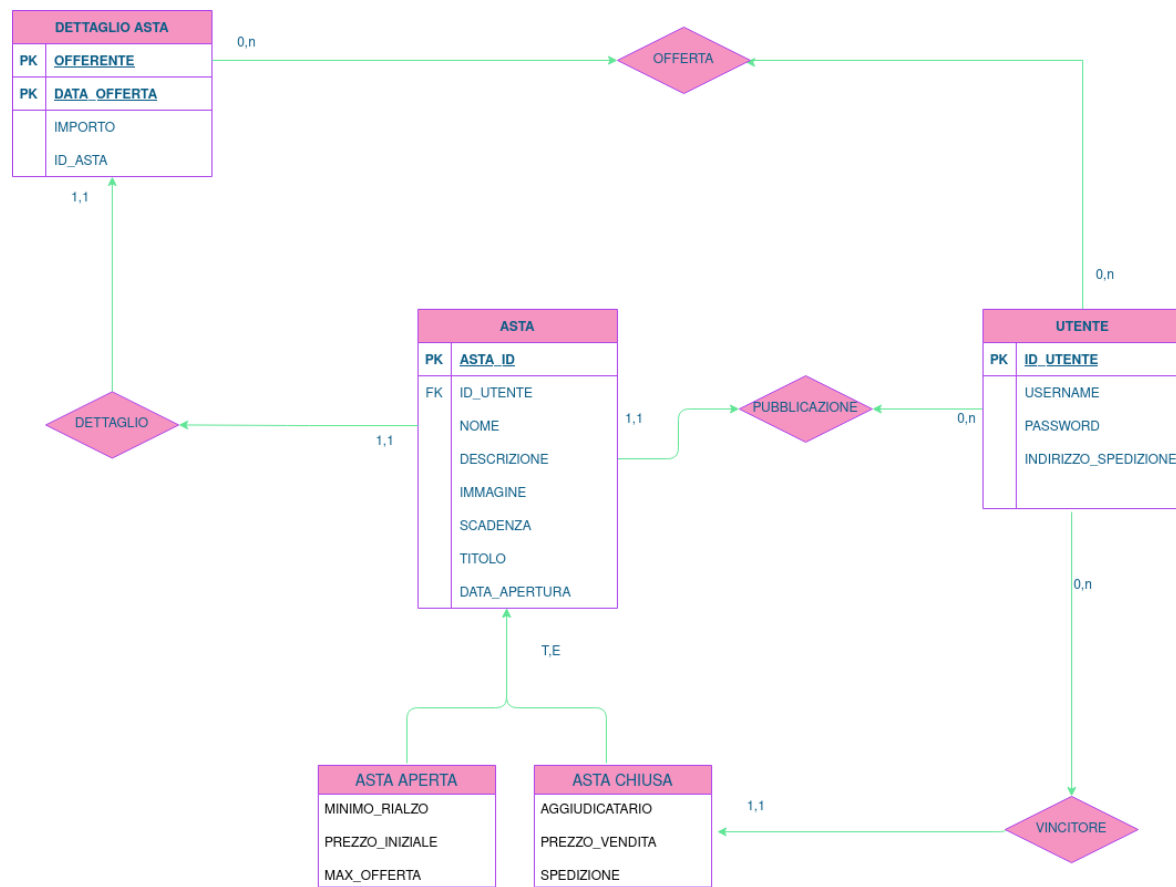
Armillotta Domenico - Borghetti Matteo

Data requirements analysis:

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** e una **scadenza** (data e ora, es 19-04-2021 alle 23:00). La lista delle aste è ordinata per data+ora crescente e riporta: **codice e nome dell'articolo, offerta massima, tempo mancante** (numero di giorni e ore) tra il momento 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). Se **l'asta è chiusa**, la pagina riporta i **dati dell'asta, il nome dell'aggiudicatario, il prezzo finale e l'indirizzo 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 si ricarica e mostra un elenco di aste aperte (la cui scadenza è posteriore all'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, ore e minuti) 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. 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.

Entities, attributes, relationships

Database Design

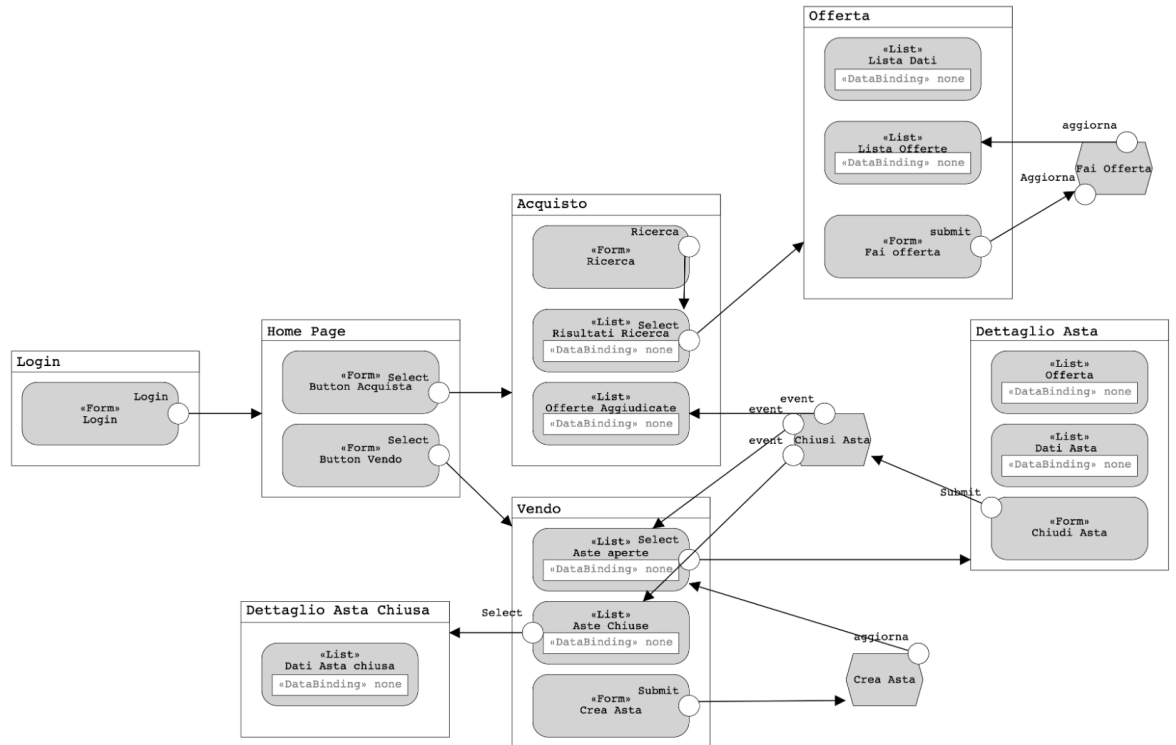


Application requirements analysis:

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 e una scadenza (data e ora, es 19-04-2021 alle 23:00). La lista delle aste è ordinata per data+ora crescente e riporta: codice e nome dell'articolo, offerta massima, tempo mancante (numero di giorni e ore) tra il momento del login e la data e ora di chiusura dell'asta. **Cliccando su un'asta compare una pagina DETAGLIO 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). Se l'asta è chiusa, la pagina riporta i dati dell'asta, il nome dell'aggiudicatario, il prezzo finale e l'indirizzo 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 **si ricarica e mostra un elenco di aste aperte** (la cui scadenza è posteriore all'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, ore e minuti) 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. 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.

pages(views),view components,events,actions

Application Design (IFML)

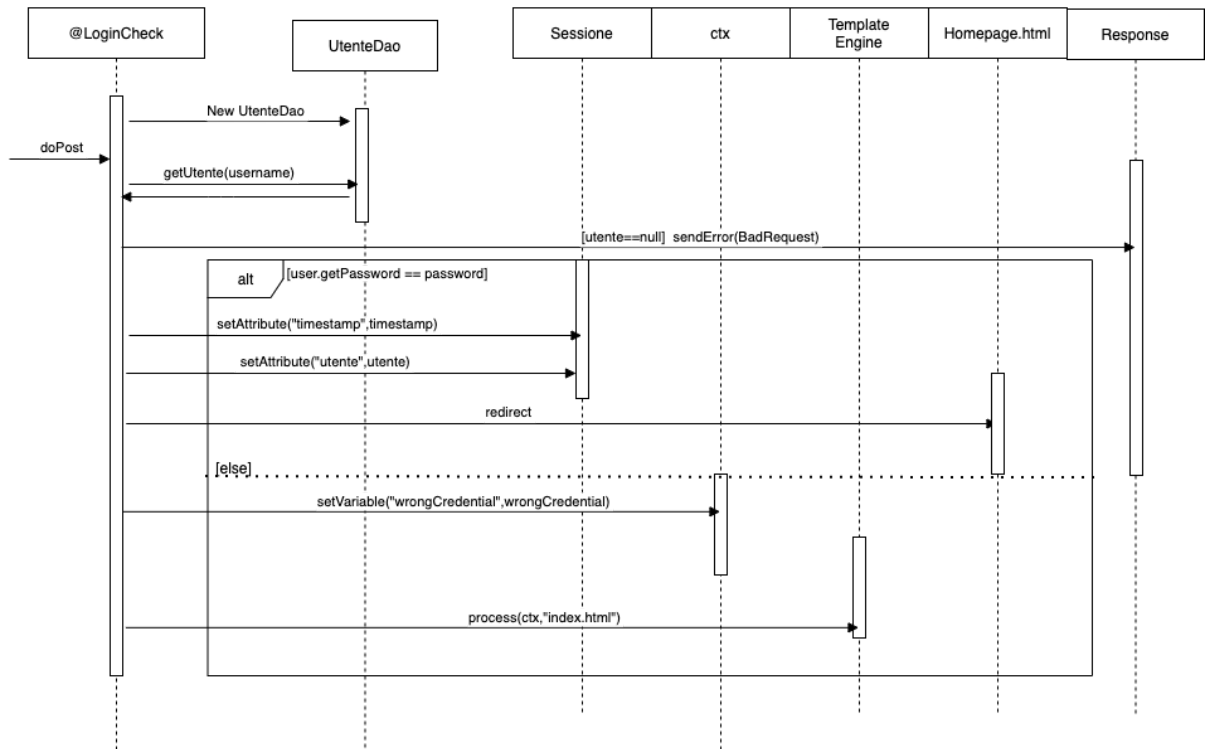


Components:

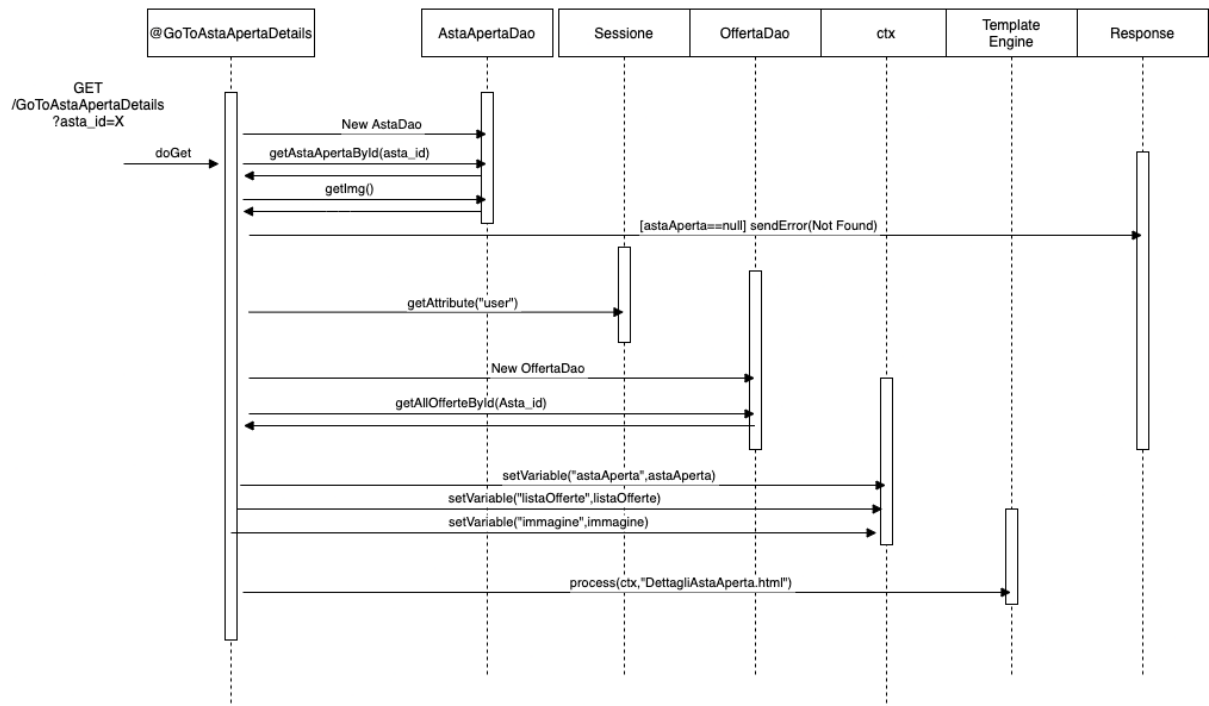
- **Model Object (Beans)**
 - ❖ Asta
 - ❖ AstaAperta
 - ❖ AstaChiusa
 - ❖ Offerta
 - ❖ Utente
- **Data Access Object (Classes)**
 - ❖ AstaApertaDao
 - ❖ AstaChiusaDao
 - ❖ loginDao
 - ❖ OffertaDao
 - ❖ UtenteDao
- **Controllers (servlets)**
 - ❖ ChiudiAstaAperta
 - ❖ CreaAstaAperta
 - ❖ GetHomepage
 - ❖ GoAcquisto
 - ❖ GoToAstaApertaDetails
 - ❖ GoToAstaApertaForOfferta
 - ❖ GoToAstaChiusaDetails
 - ❖ GoToVendita
 - ❖ LoginCheck
 - ❖ Logout
- **Views(Template)**
 - ❖ Acquisto
 - ❖ DettagliAstaAperta
 - ❖ DettagliAstaChiusa
 - ❖ Home
 - ❖ index
 - ❖ OffertaPerAstaAperta
 - ❖ Vendita
- **Filters**
 - ❖ Checker

Sequence diagrams :

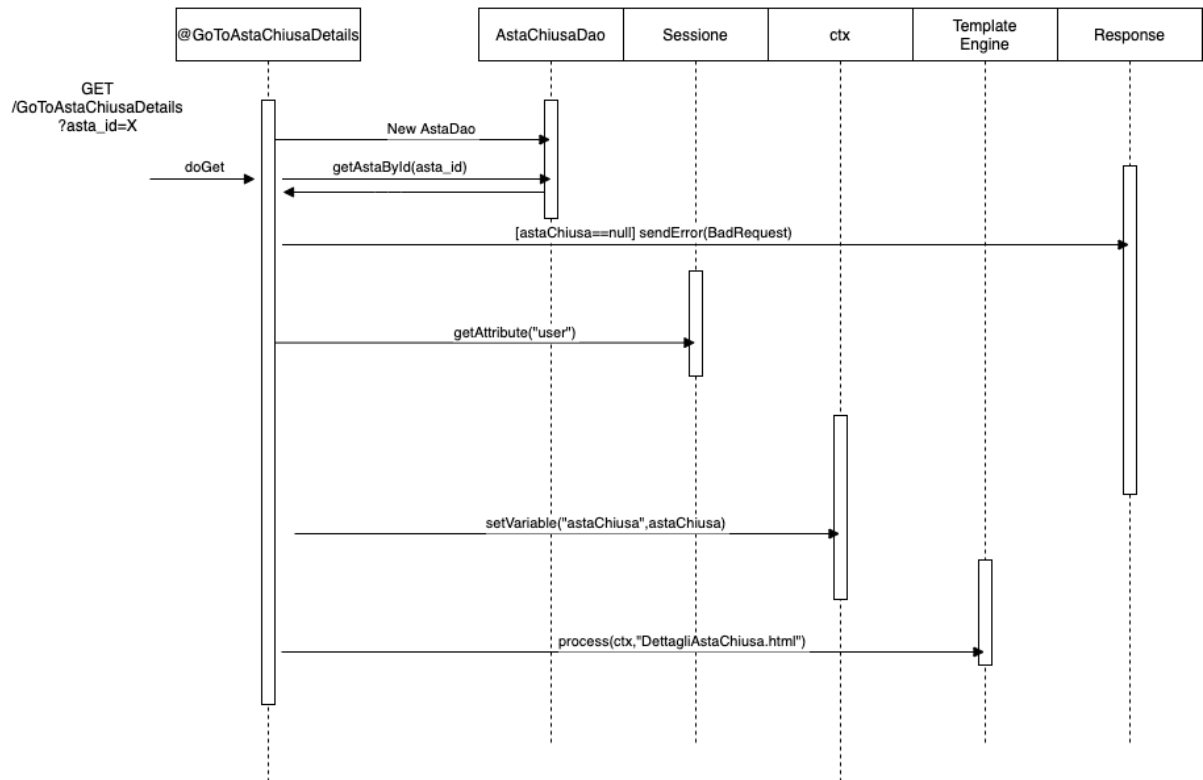
@LoginCheck



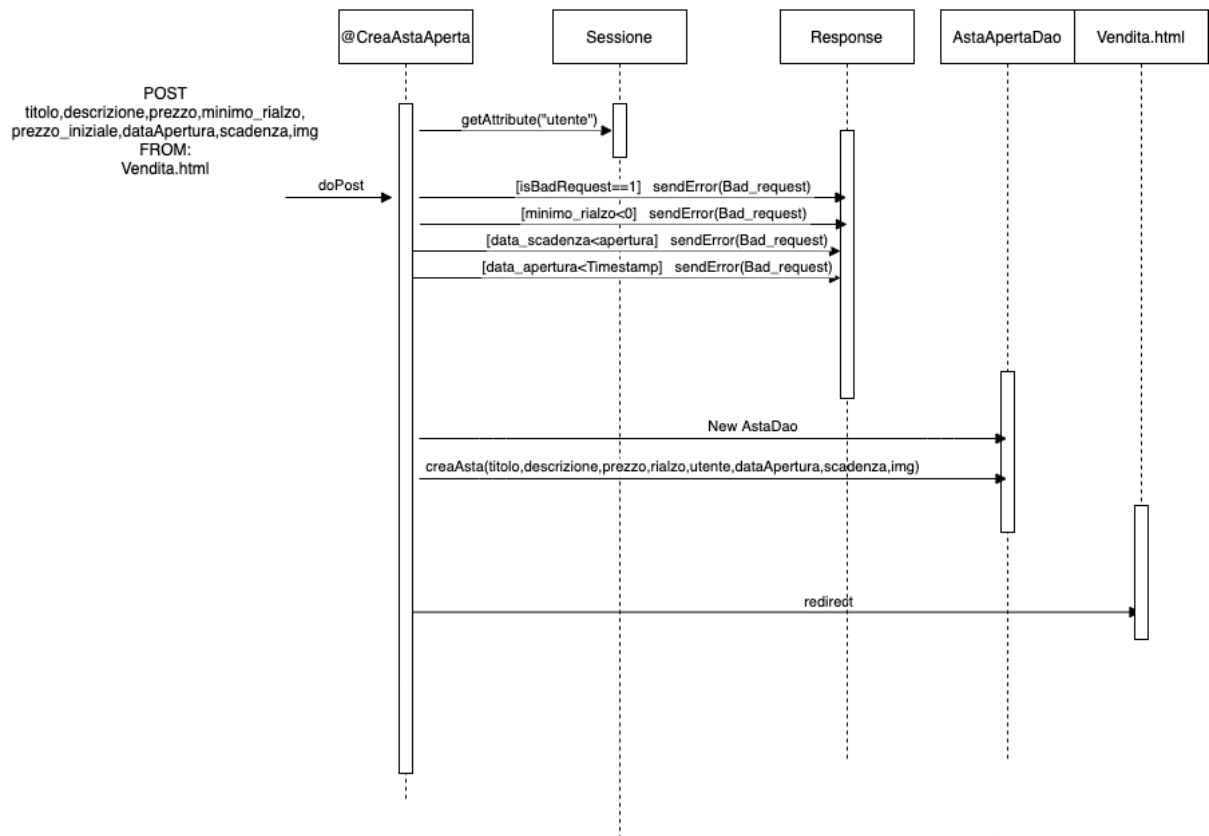
@GoToAstaApertaDetails



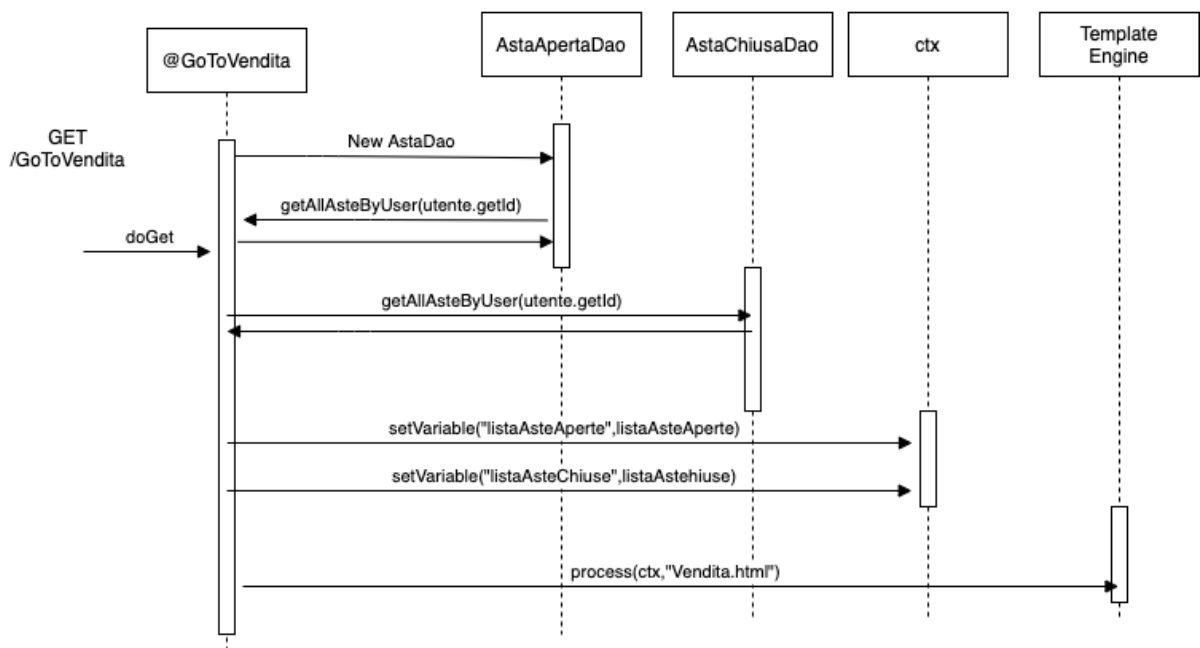
@GoToChiusaDetails



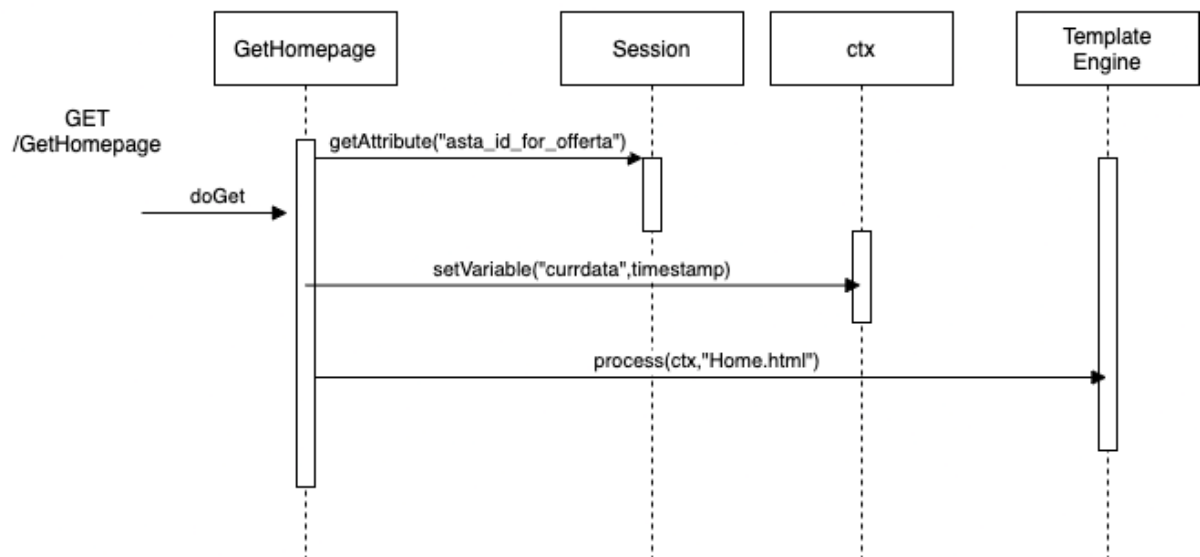
@CreaAstaAperta



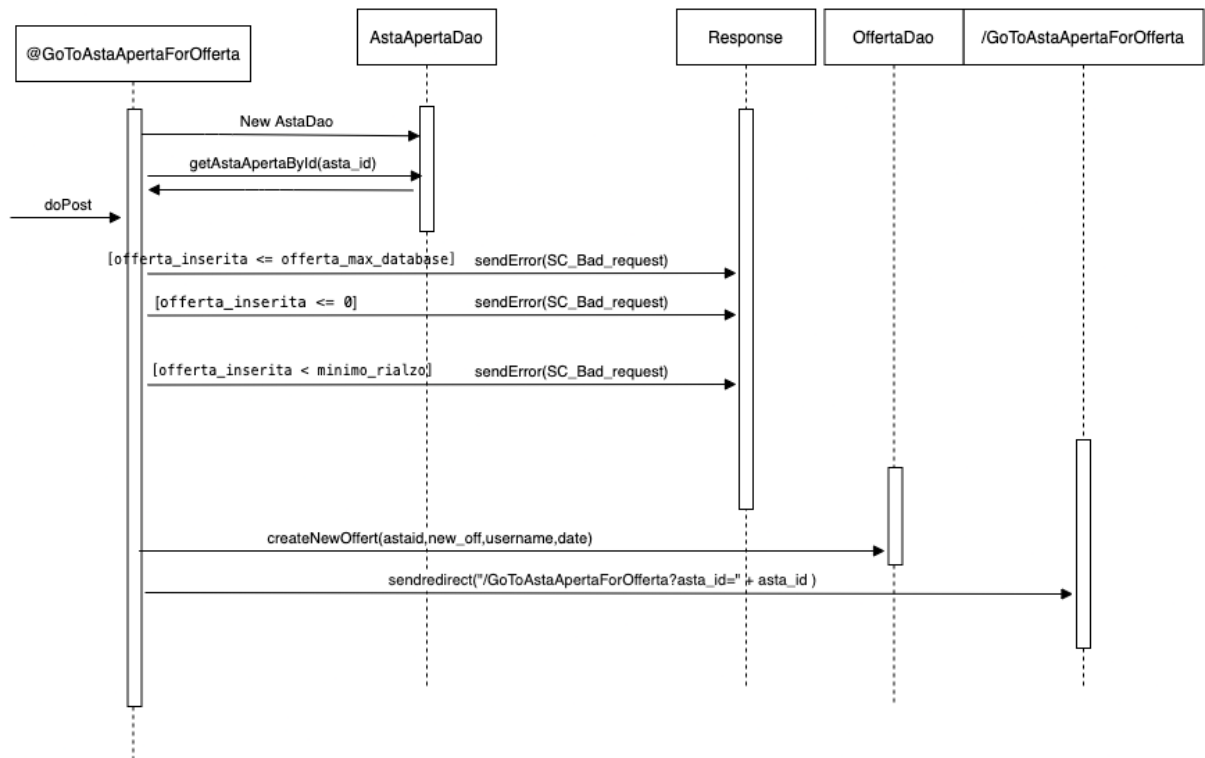
@GoToVendita



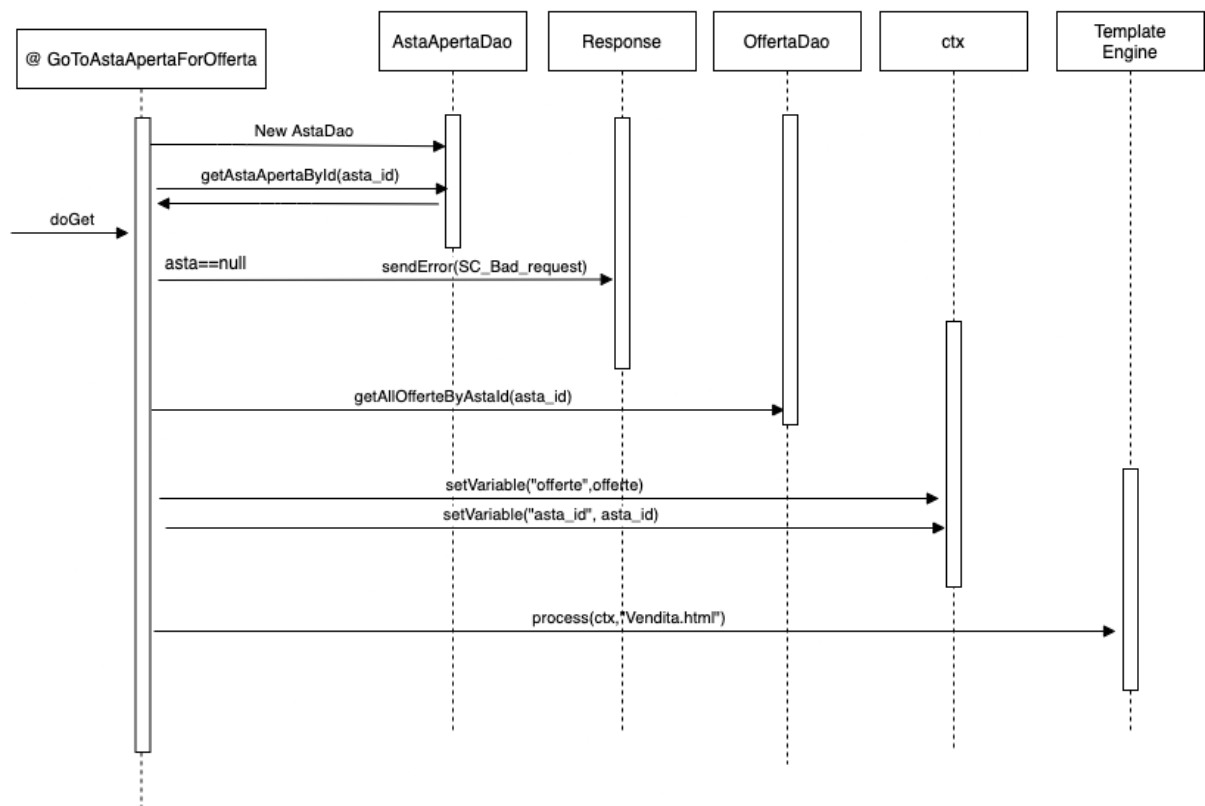
@GetHomepage



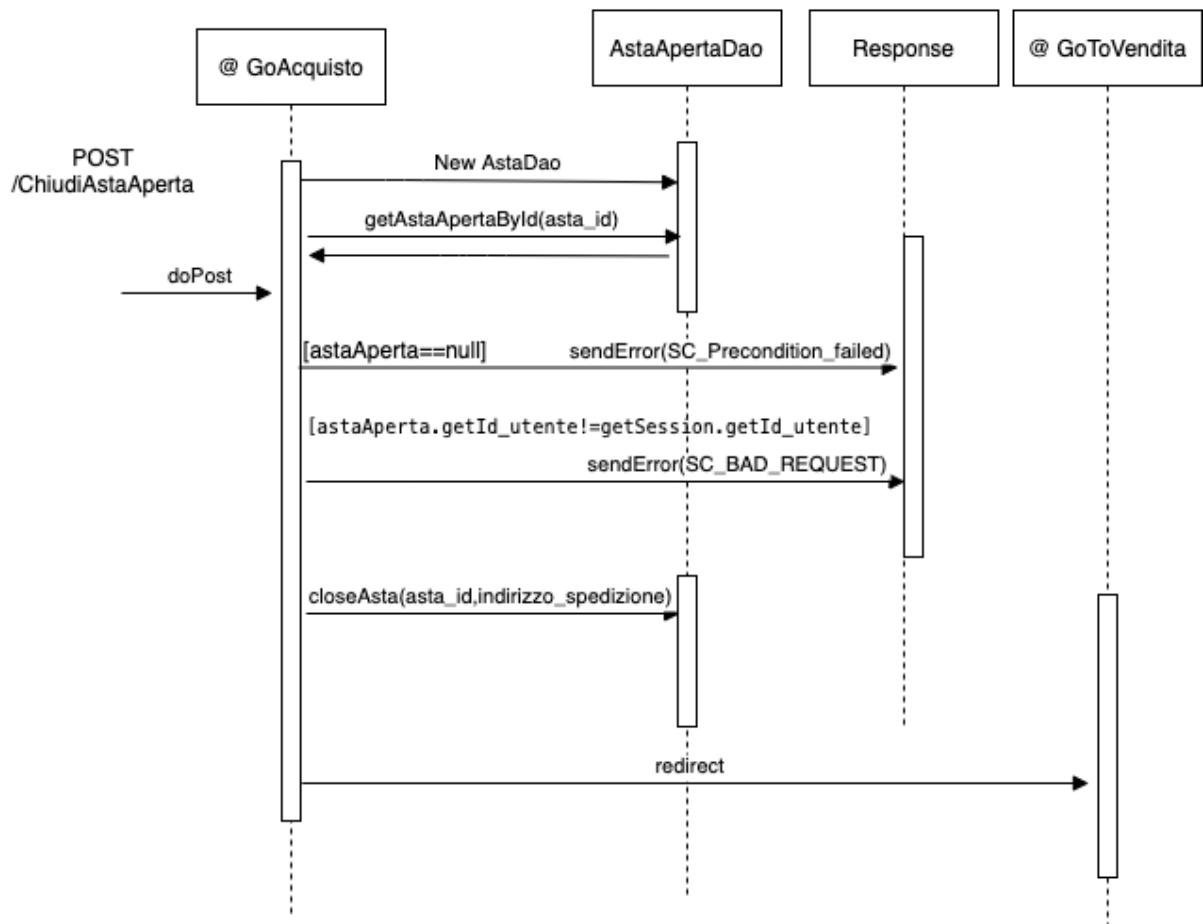
@GoToAstaApertaForOfferta



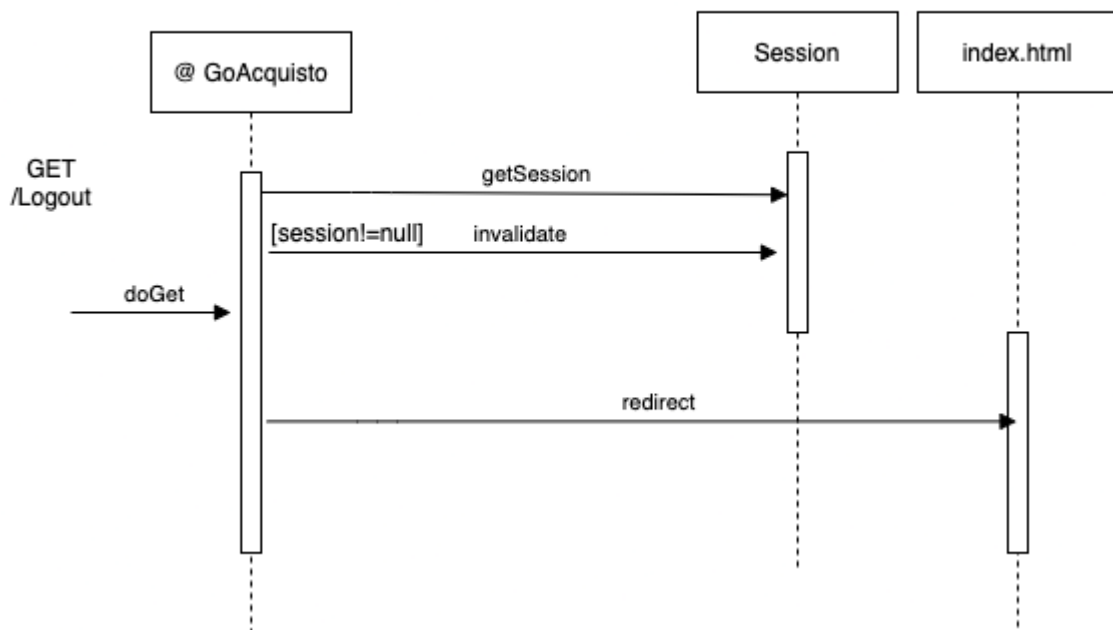
@GoToAstaApertaForOfferta



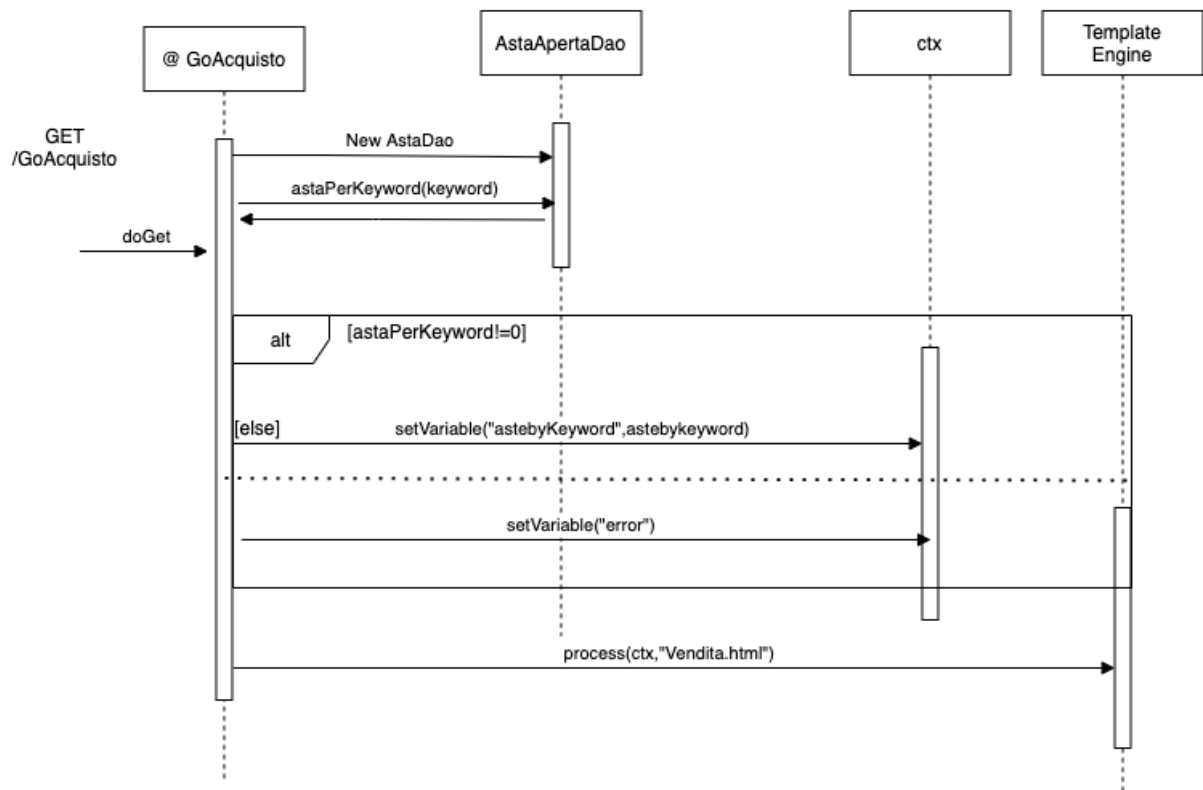
@ChiudiAstaAperta



@Logout



@GoAcquisto



Strategic choices:

- all data entered in the forms are also checked on the server side, to avoid errors
- the "closeAsta" query, which contains two database operations, has been made atomic