Online Auctions
Pure HTML

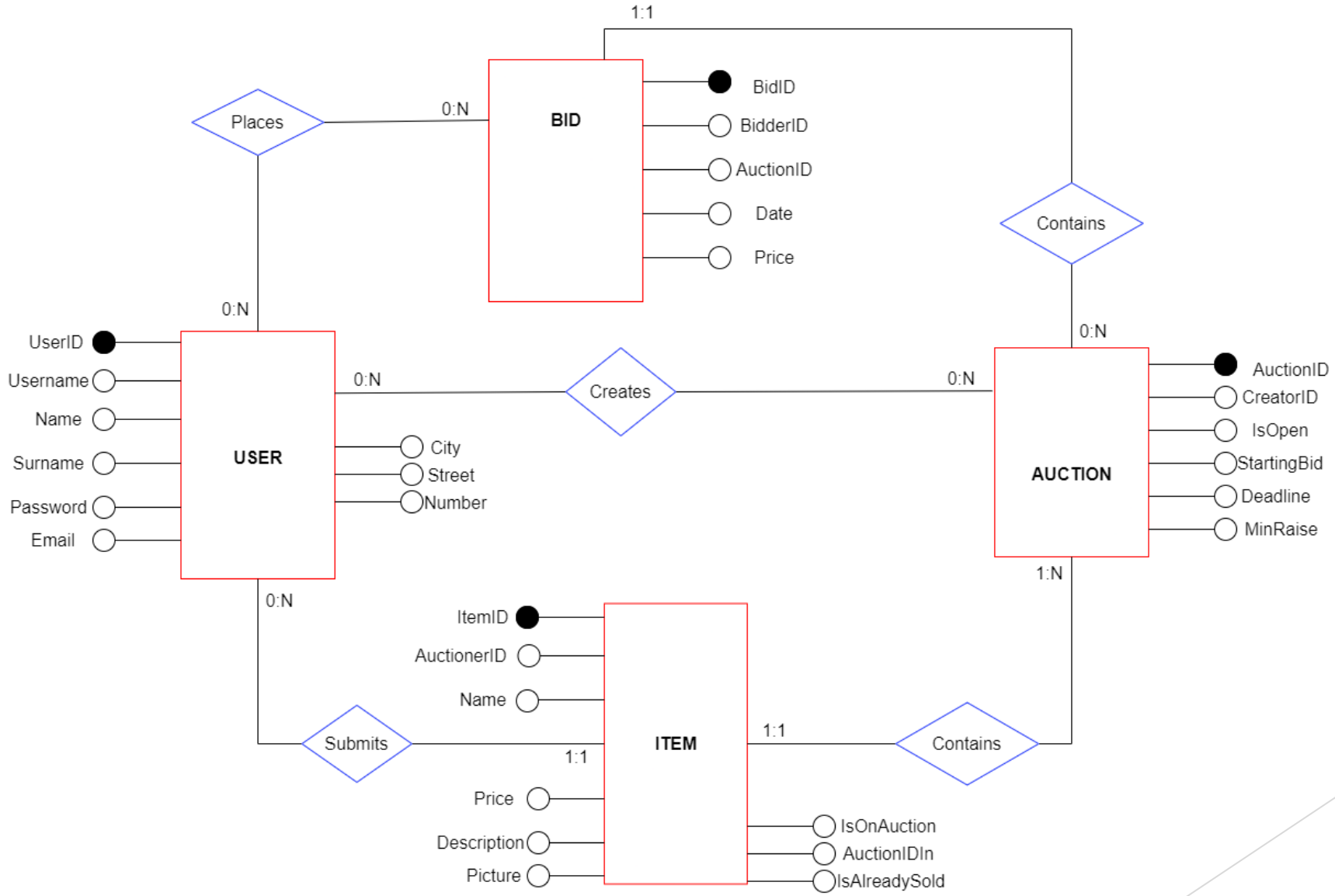
Benelle Francesco
10727489
959528

Specifications

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.

Entities, attributes, relationships

Database Design



Database Schema

```
CREATE TABLE `user` (
  `userid` int unsigned NOT NULL AUTO_INCREMENT,
  `username` varchar(45) NOT NULL,
  `name` varchar(45) NOT NULL,
  `surname` varchar(45) NOT NULL,
  `email` varchar(100) NOT NULL,
  `password` varchar(45) NOT NULL,
  `city` varchar(45) NOT NULL,
  `street` varchar(45) NOT NULL,
  `number` int unsigned NOT NULL,
  PRIMARY KEY (`userid`),
  UNIQUE KEY `userid_UNIQUE` (`userid`)
)
CREATE TABLE `bid` (
  `bidid` int unsigned NOT NULL AUTO_INCREMENT,
  `bidderid` int unsigned NOT NULL,
  `auctionid` int unsigned NOT NULL,
  `date` timestamp NOT NULL,
  `price` decimal(15,2) unsigned NOT NULL,
  PRIMARY KEY (`bidid`),
  UNIQUE KEY `bidid_UNIQUE` (`bidid`),
  KEY `auctionid_idx` (`auctionid`),
  KEY `bidderid_idx` (`bidderid`),
  CONSTRAINT `auctionid` FOREIGN KEY (`auctionid`) REFERENCES `auction`
    (`auctionid`) ON UPDATE CASCADE,
  CONSTRAINT `bidderid` FOREIGN KEY (`bidderid`) REFERENCES `user`
    (`userid`) ON UPDATE CASCADE
)
```

Database Schema

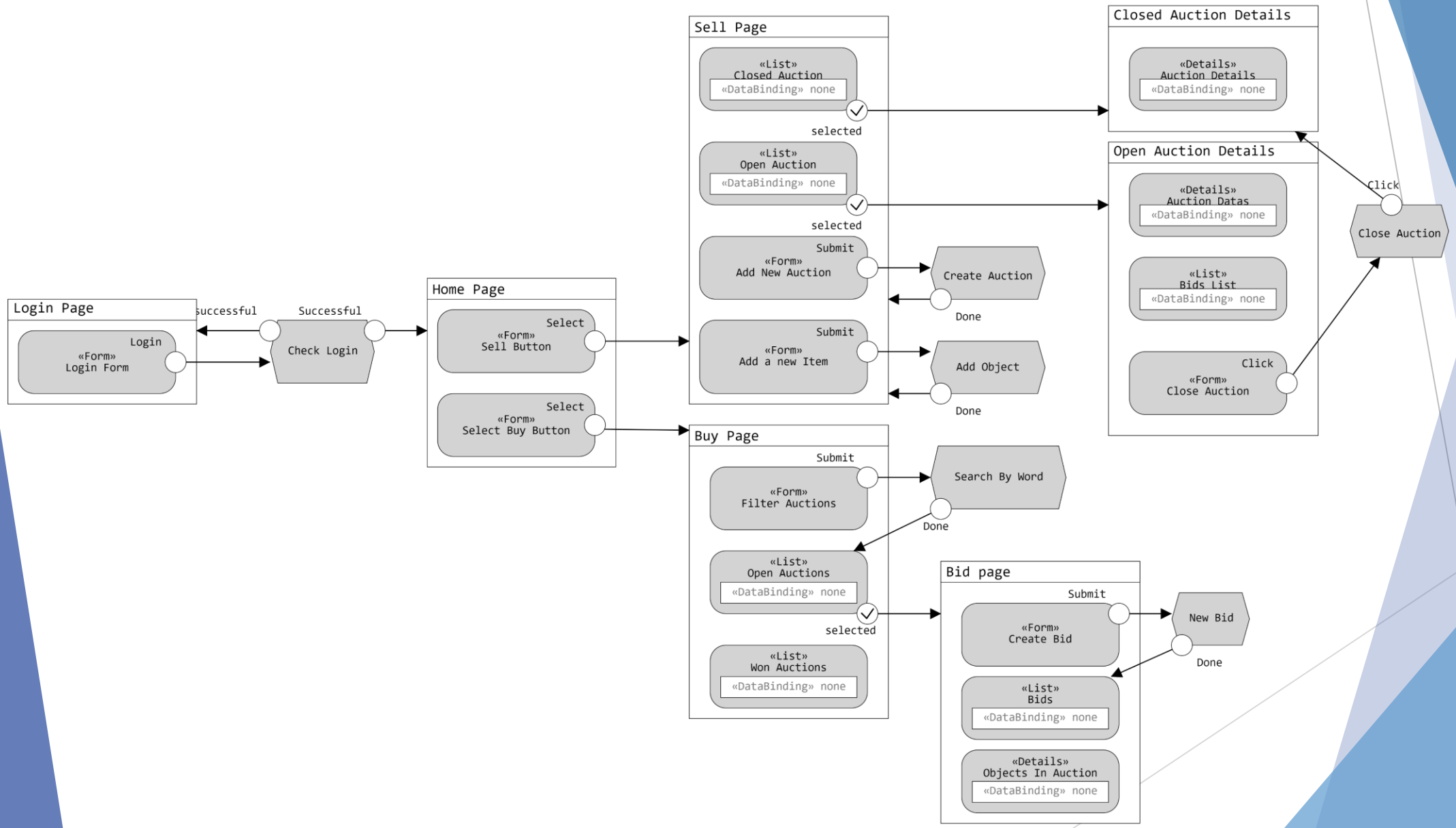
```
CREATE TABLE `item` (
  `itemid` int unsigned NOT NULL AUTO_INCREMENT,
  `auctionerid` int unsigned NOT NULL,
  `name` varchar(45) NOT NULL,
  `price` decimal(10,2) unsigned NOT NULL,
  `description` varchar(500) NOT NULL,
  `picture` varchar(200) NOT NULL,
  `isonauction` tinyint NOT NULL DEFAULT '0',
  `auctionidin` int DEFAULT NULL,
  `isalreadysold` tinyint NOT NULL DEFAULT '0',
  PRIMARY KEY (`itemid`),
  UNIQUE KEY `itemid_UNIQUE` (`itemid`),
  KEY `auctionerid_idx` (`auctionerid`),
  CONSTRAINT `auctionerid` FOREIGN KEY (`auctionerid`) REFERENCES `user`
    (`userid`)
)
CREATE TABLE `auction` (
  `auctionid` int unsigned NOT NULL AUTO_INCREMENT,
  `creatorid` int unsigned NOT NULL,
  `isopen` tinyint NOT NULL,
  `startingbid` decimal(15,2) unsigned NOT NULL,
  `deadline` timestamp NOT NULL,
  `minraise` decimal(15,2) unsigned NOT NULL,
  PRIMARY KEY (`auctionid`),
  UNIQUE KEY `auctionid_UNIQUE` (`auctionid`),
  KEY `creatorid_idx` (`creatorid`),
  CONSTRAINT `creatorid` FOREIGN KEY (`creatorid`) REFERENCES `user`
    (`userid`)
)
```

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 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 emostra 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.

Pages (views), view components, events, actions

Application Design (IFML)



Components

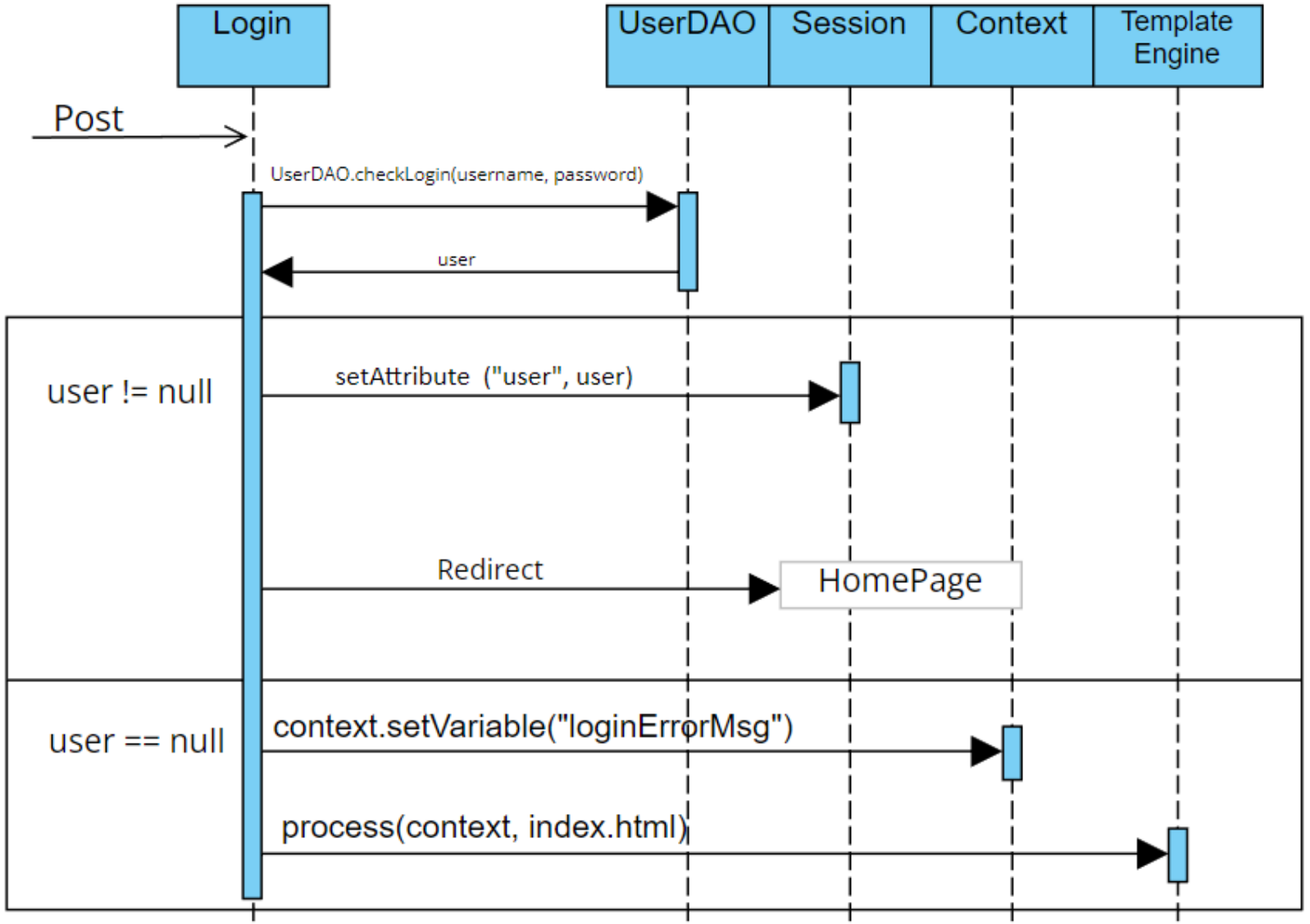
Model Objects (Beans)
- User
- Auction
- Item
- Bid

Data Access Objects (DAO)
- UserDAO
- AuctionDAO
- ItemDAO
- BidDAO

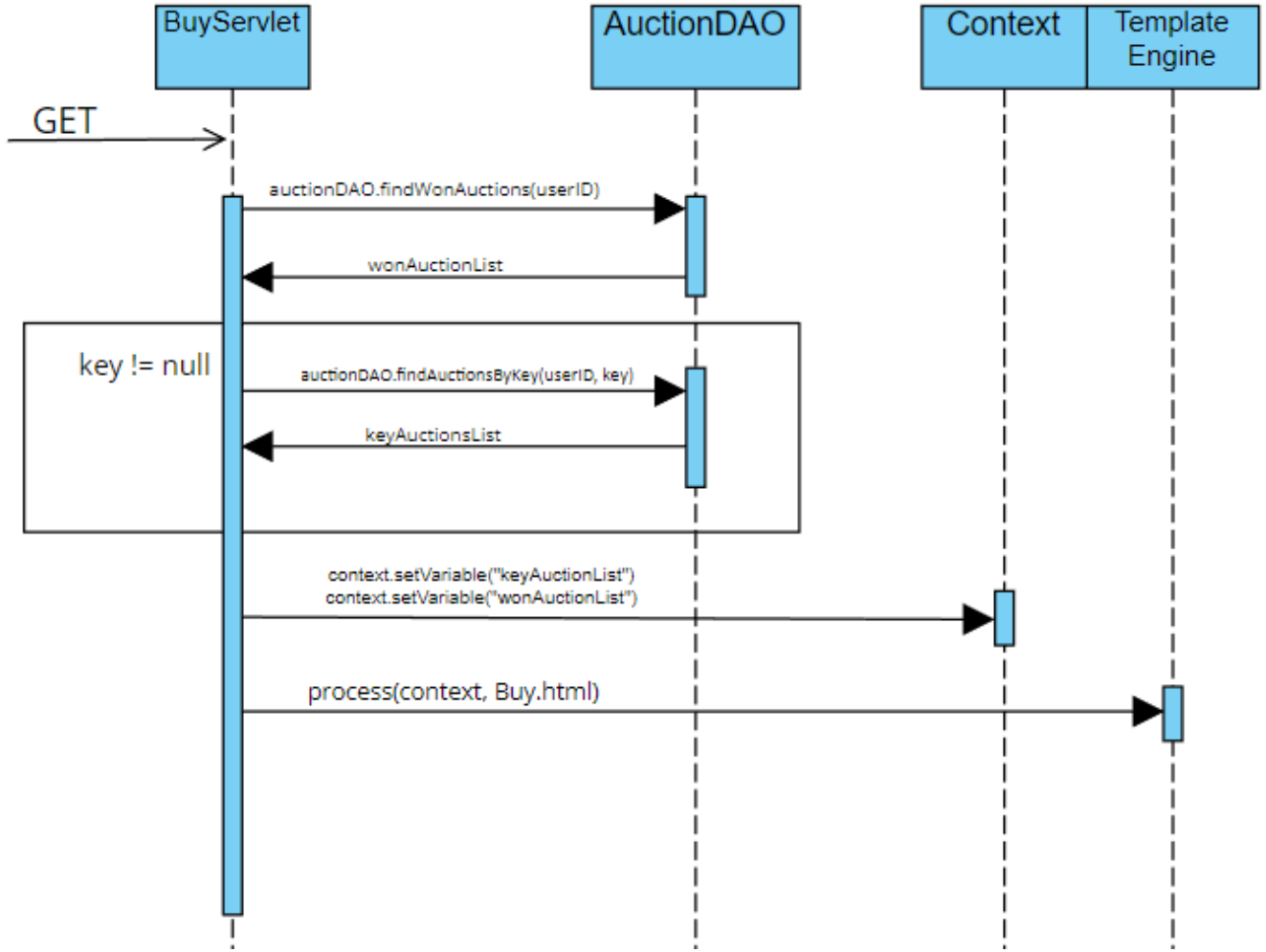
Controllers (Servlet)
- Login
- Logout
- GoToHomePage
- BuyServlet
- SellServlet
- ClosedAuctionDetails
- OpenAuctionDetails
- CloseAuction
- NewBid
- NewItem
- NewAuction
- GetPicture

View (Template)
- Homepage.html
- Buy.html
- Sell.html
- ClosedAuction.html
- OpenAuction.html
- Bid.html
- Index.html

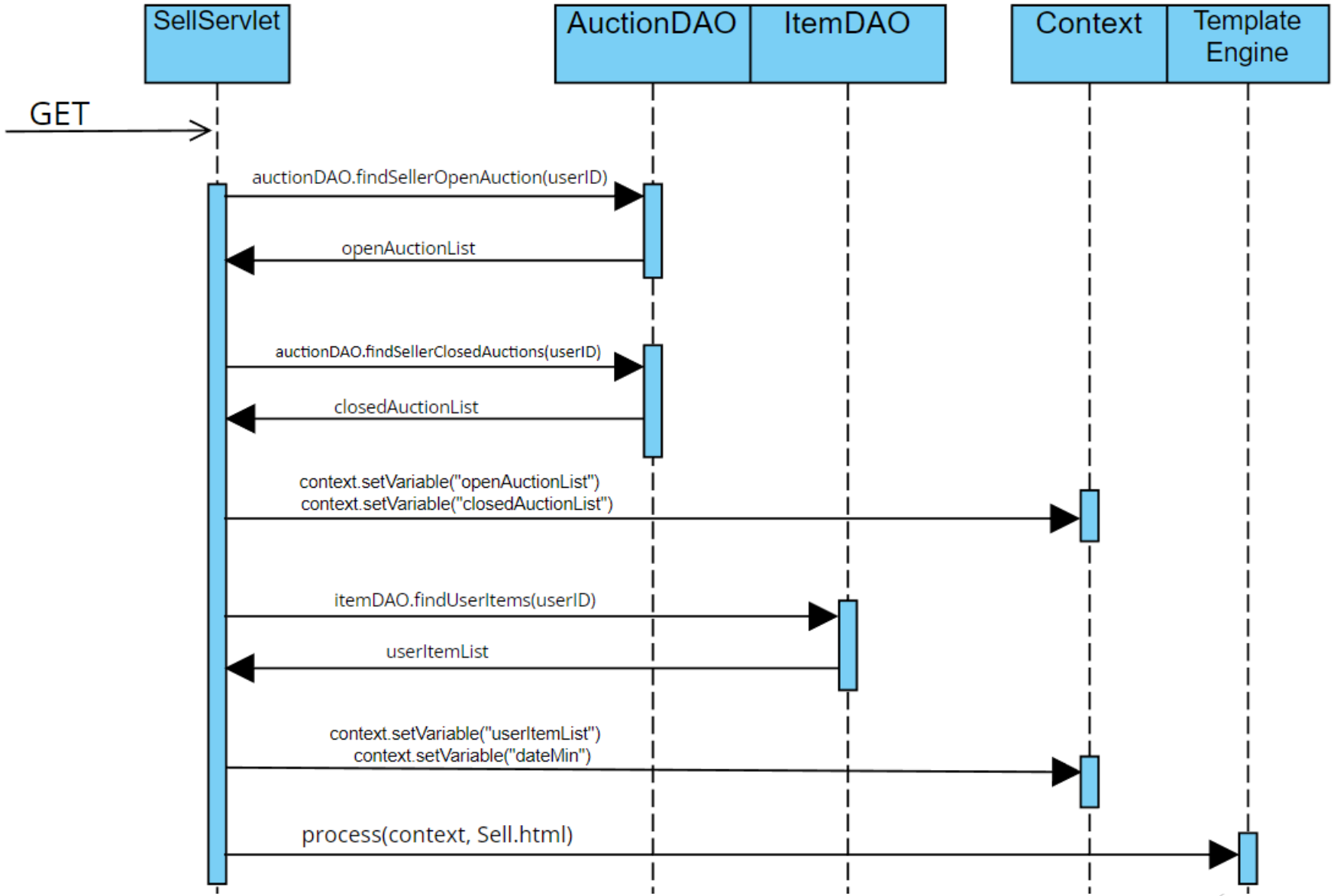
Sequence Diagrams: Login



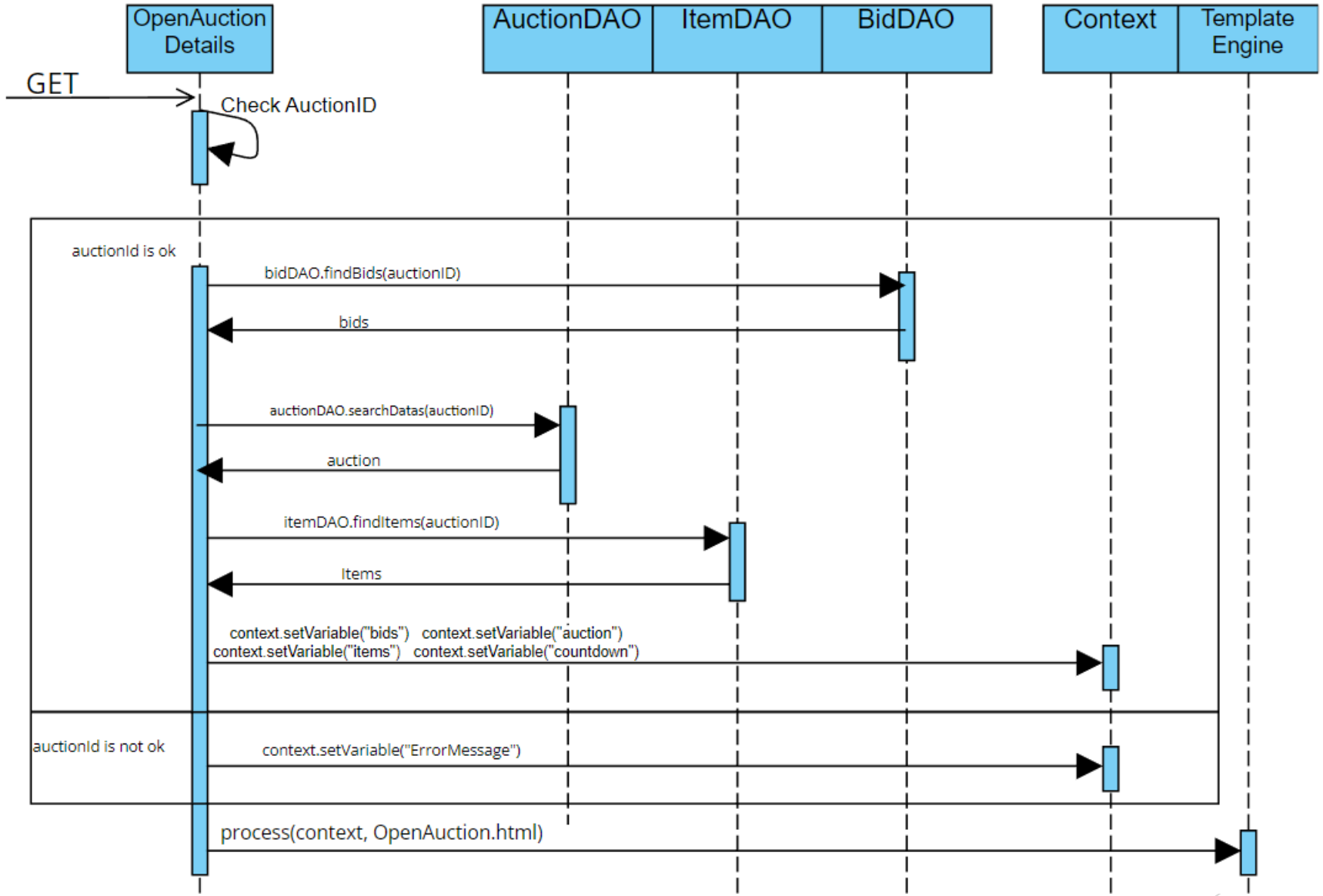
Sequence Diagrams: Go to BUY page



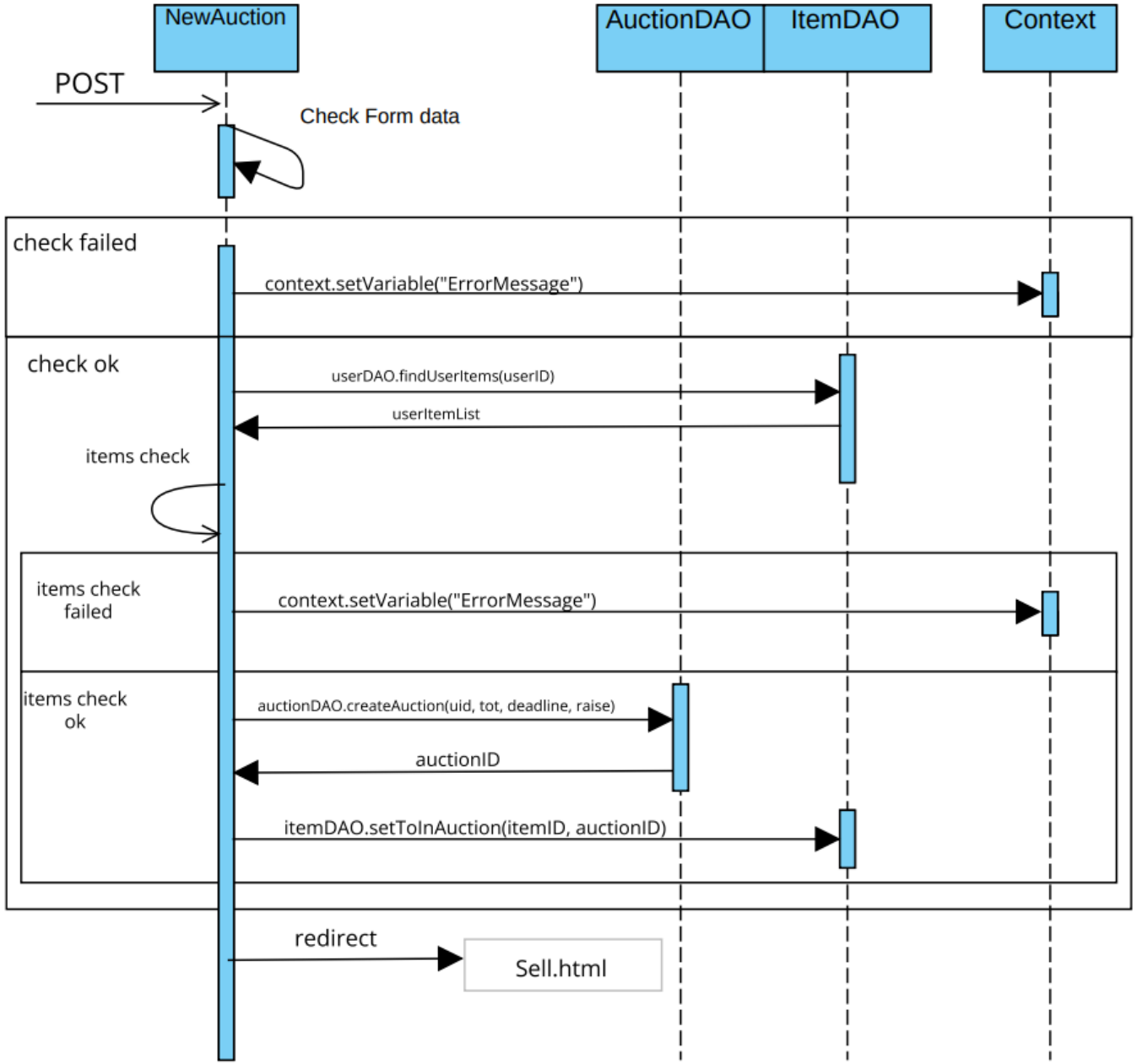
Sequence Diagrams: Go to SELL page



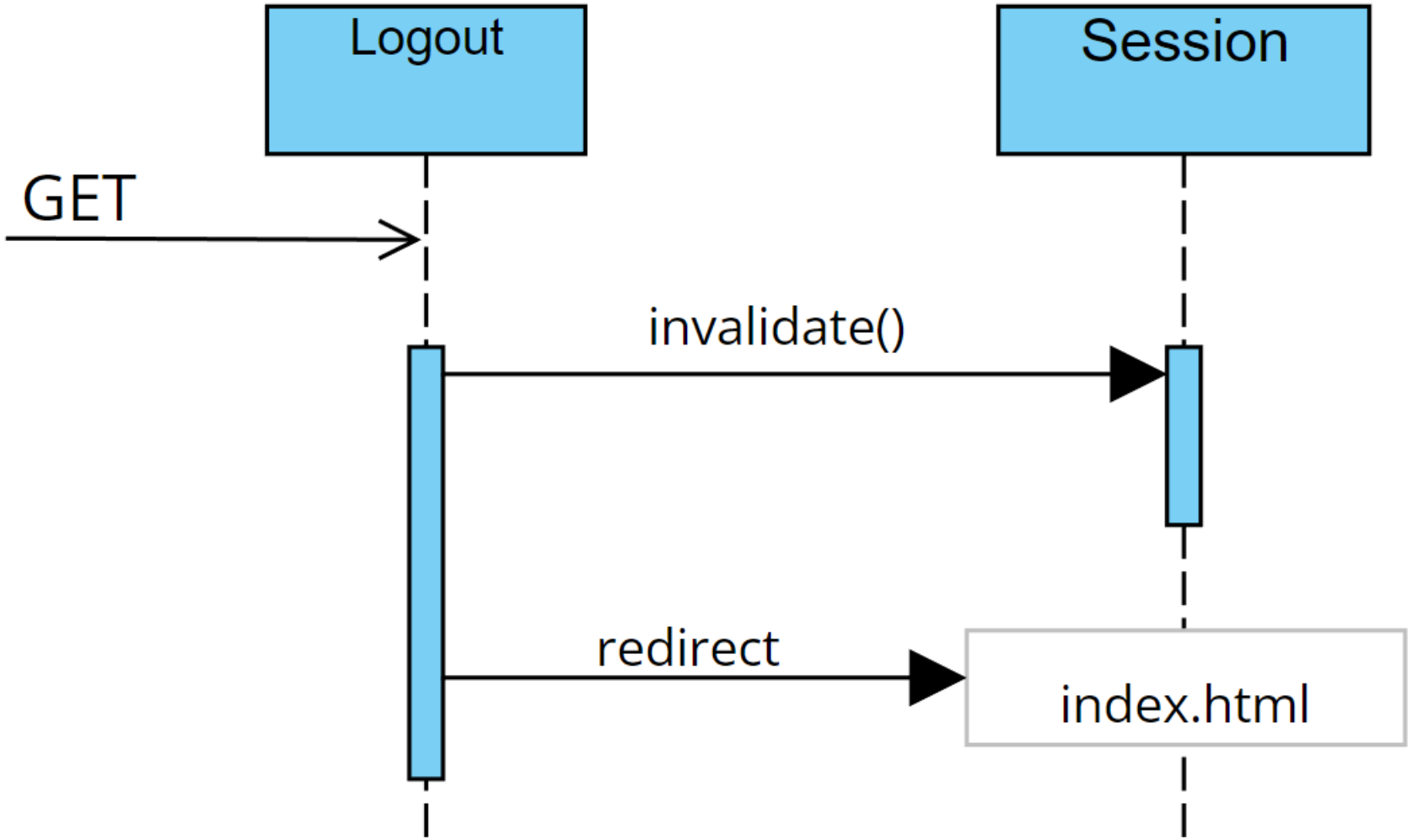
Sequence Diagrams: Get Open Auction Details



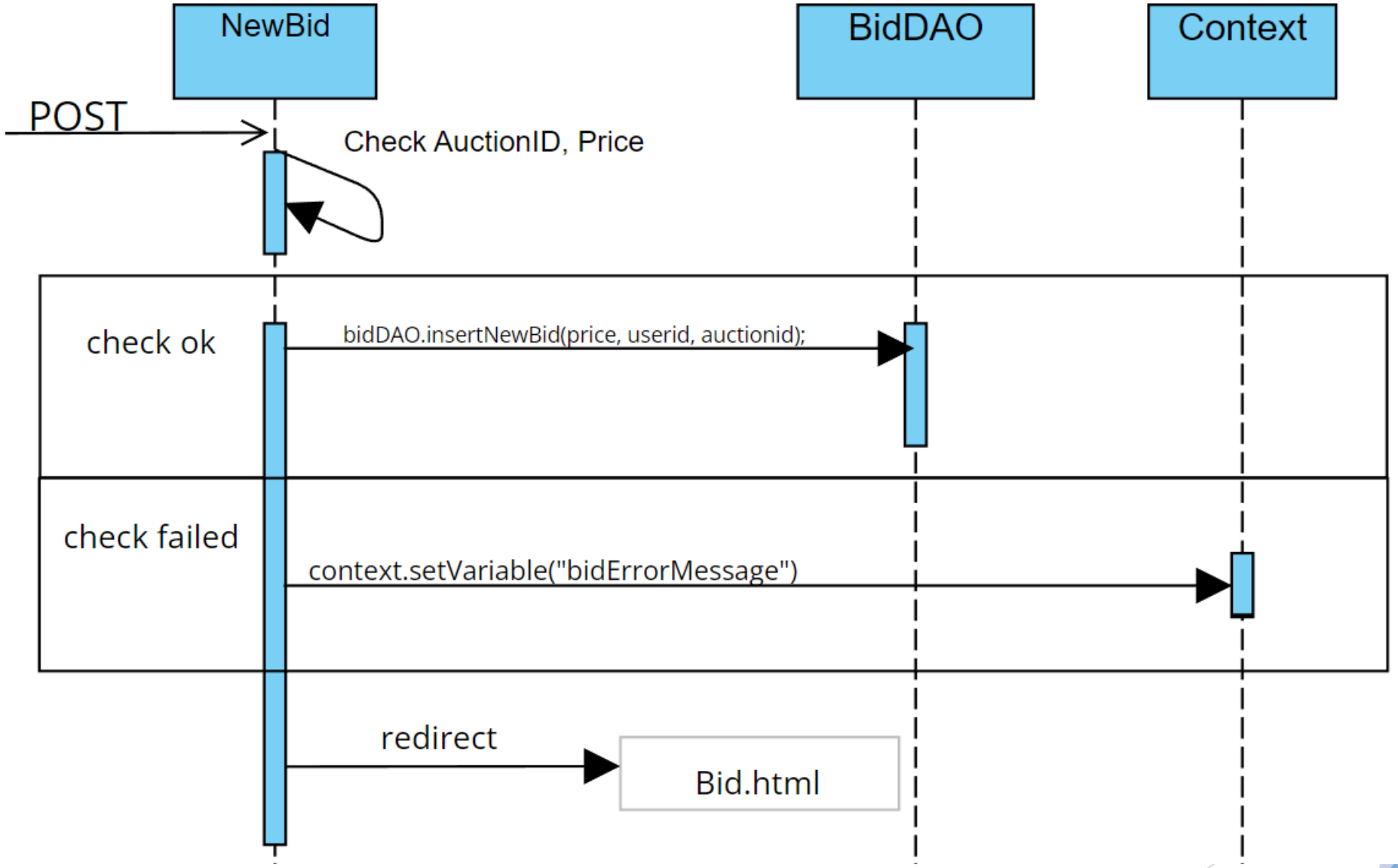
Sequence Diagrams: Create a New Auction



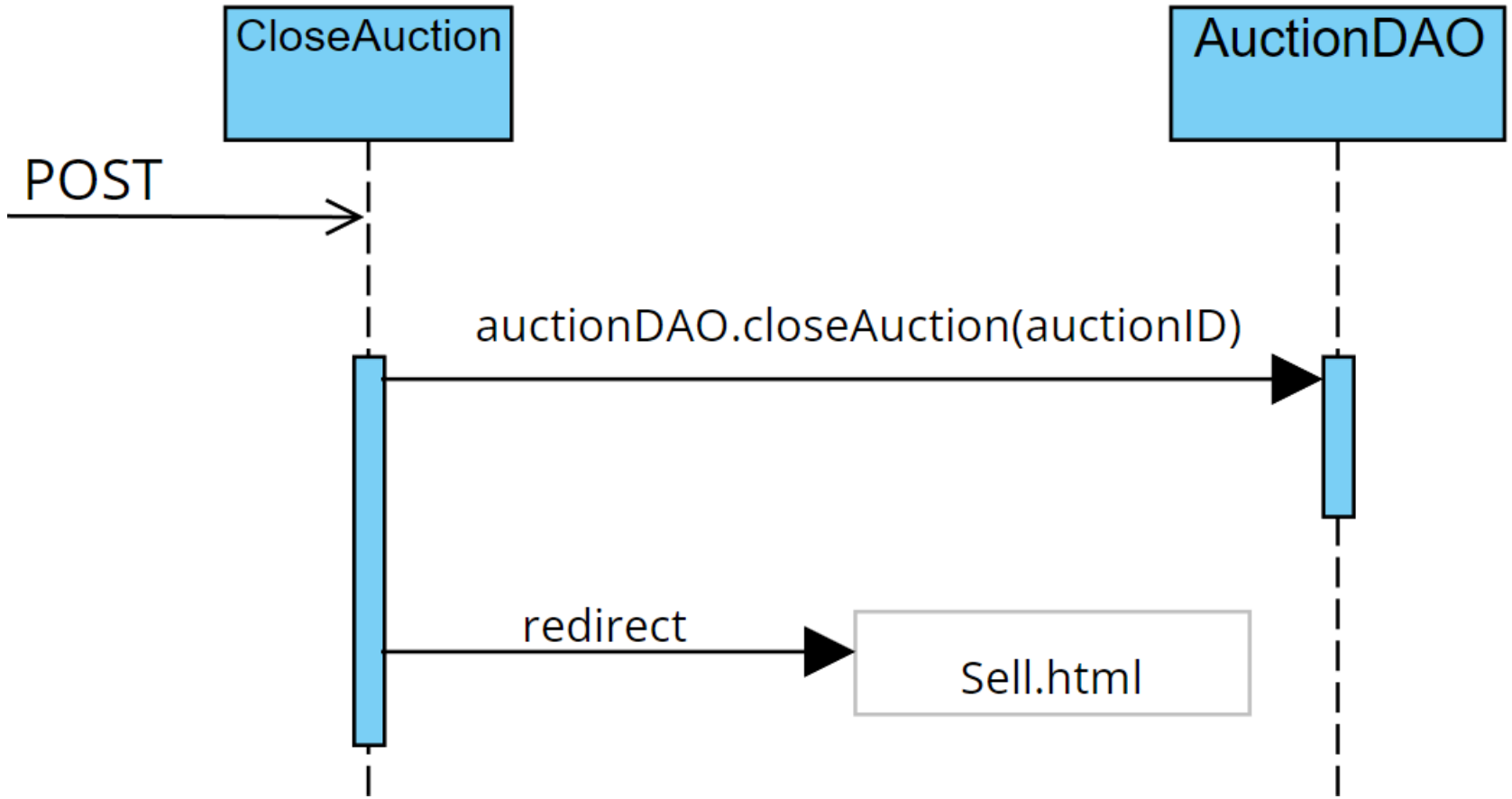
Sequence Diagrams: Logout



Sequence Diagrams: Add a new bid



Sequence Diagrams: Close Auction



Online Auctions
RIA

Benelle Francesco
10727489
959528

JavaScript-only Specifications

- 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.

Components

Model Objects (Beans)
- User
- Auction
- Item
- Bid

Data Access Objects (DAO)
- UserDAO
- AuctionDAO
- ItemDAO
- BidDAO

Views
- Home.html
- Index.html

Controllers
- Login
- Logout
- BidServlet
- BuyServlet
- SellServlet
- GetItems
- GetUserItems
- GetWinner
- NewBid
- NewItem
- NewAuction
- CloseAuction
- GetCookieAuctions
- KeySearchServlet

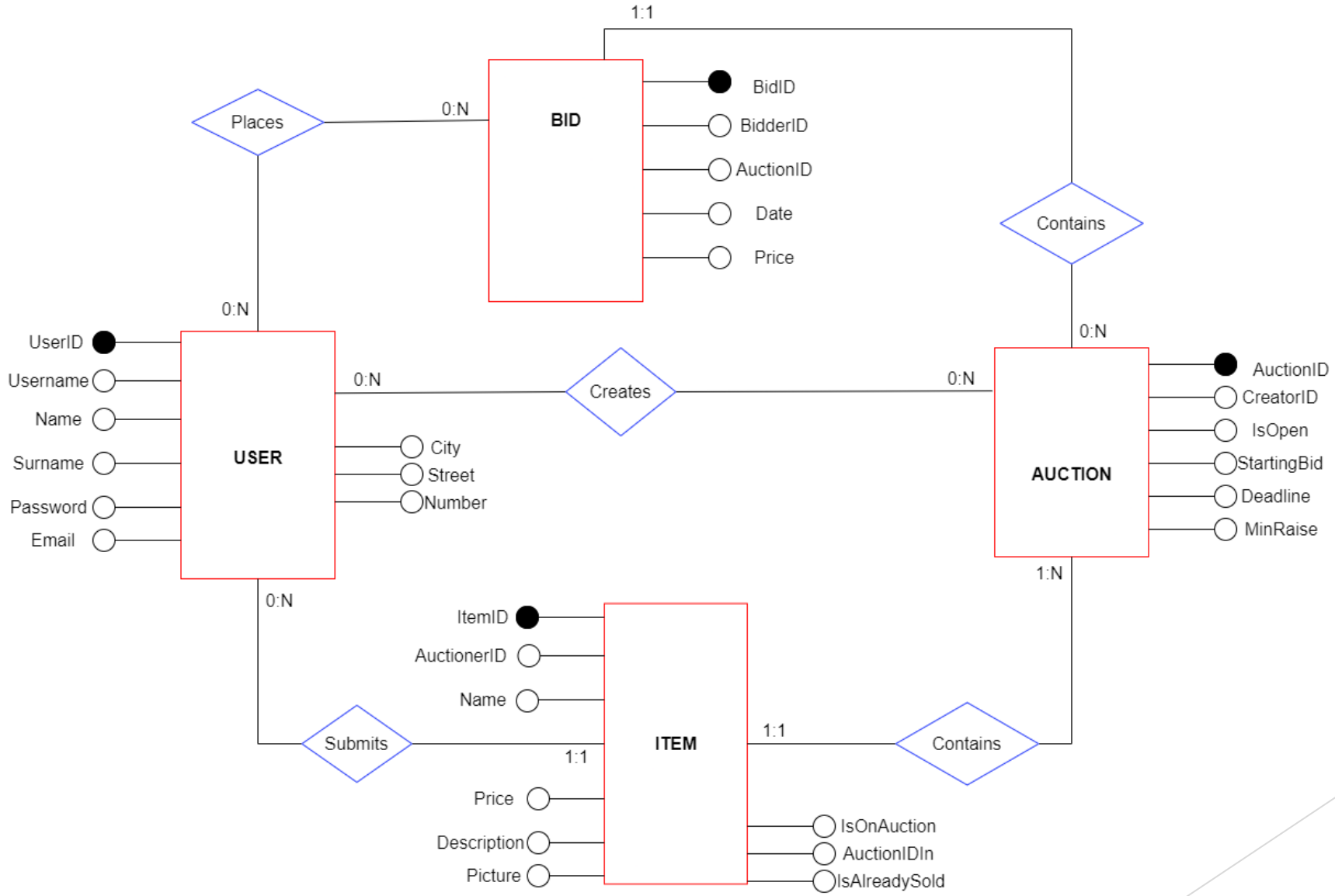
JavaScript Components

- **PageOrchestrator**: initializes and loads the Home page components
- **Manager**: shows and hides the various components of the page

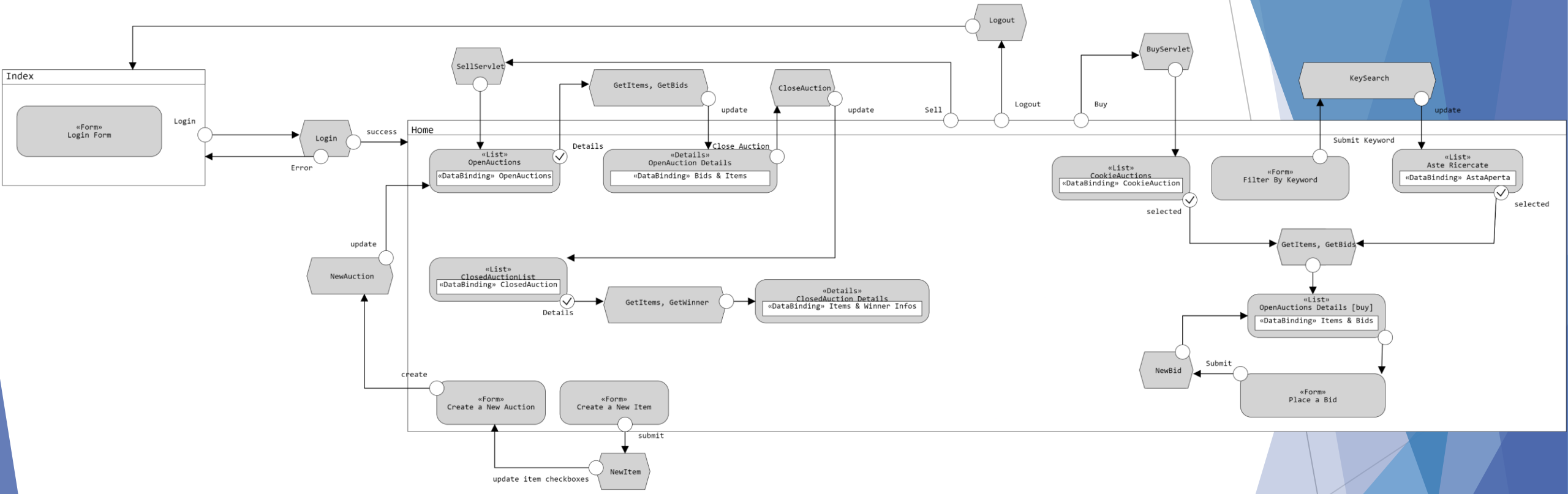
- **OpenAuctions**: shows a list of the current user's open auctions, does makeCall in order to retrieve the auction's details
- **ClosedAuctions** : shows a list of the current user's closed auctions, does makeCall in order to retrieve the auction's details
- **OpenAuctionBids**: shows a list of all the bids that were placed on a certain auction
- **OpenAuctionsItems**: shows a list of all the all the items in an open auction
- **ClosedAuctionsDetails**: shows a list of all the all the items in a closed auction
- **ClosedAuctionsWinner**: shows the info about the winner of a certain auction
- **NewItem**: retrieves the data from a form in order to record in the DB a submitted item
- **NewAuction**: retrieves the data from a form in order to record in the DB a new auction

- **KeywordForm**: retrieves the keyword inserted in order to filter the open auctions
- **KeyAuctions**: shows a list of the filtered auctions, does makeCall in order to retrieve the details and to permit the user to place a bid
- **CookieAuctions**: shows a list of the most recently viewed auctions
- **BuyAuctionDetails**: shows the details (item and bids) of a certain auction
- **PlaceBid**: retrieves the submitted import from a form in order to place a bid on the auction
- **WonAuctions**: shows a list of all the user's won auctions

Database Design



Application Design (IFML)



Events & Actions

CLIENT SIDE		SERVER SIDE	
EVENT	ACTION	EVENT	ACTION
Index.html ⟶ login form ⟶ Submit	Check submitted datas	POST (Username, Password)	Username & Password check
Home ⟶ load	Show BUY or SELL page according to the cookies	GET	
Home ⟶ Show Sell	Shows the components of the SELL page	GET	Retrieves the user's open and closed auctions
Home ⟶ Sell ⟶ OpenAuctions ⟶ Show Details	Shows an open auction's details	GET (AuctionID)	Retrieves the items and the bids of a certain auction
Home ⟶ Sell ⟶ ClosedAuctions ⟶ Show Details	Shows a closed auction's details	GET (AuctionID)	Retrieves the items of a certain auction and the info about the winner
Home ⟶ Sell ⟶ NewItem ⟶ Submit	Creates a new item	POST (form datas)	Adds the new Item to the DB if it passes the checks
Home ⟶ Sell ⟶ NewAuction ⟶ Submit	Creates a new auction	POST (form datas)	Adds the new Auction to the DB if it passes the checks, updates the items' state
Home ⟶ Sell ⟶ CloseAuction	Closes the Auction	POST (AuctionID)	Sets the Auction to closed and its items to sold

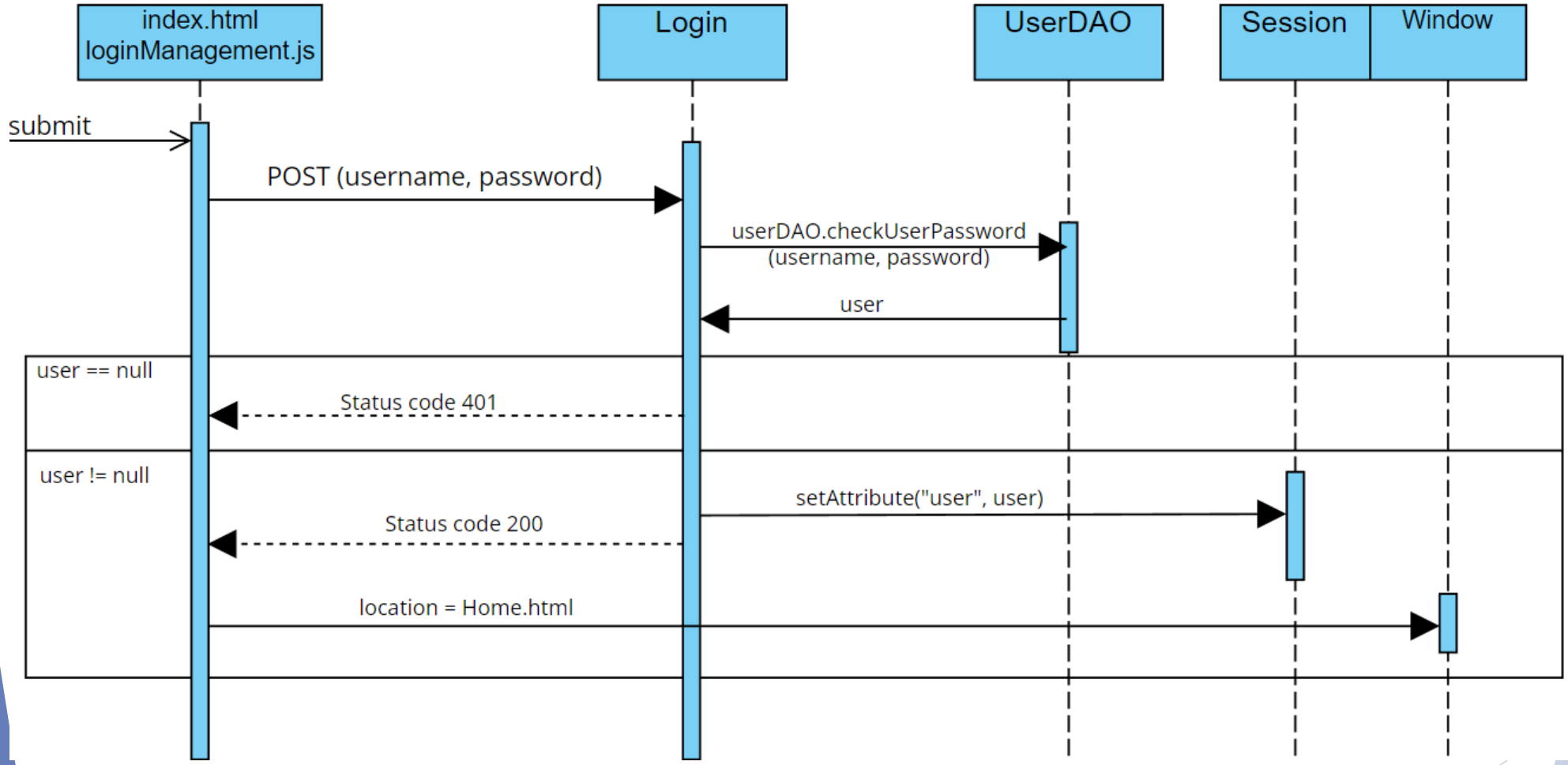
Events & Actions [2]

CLIENT SIDE		SERVER SIDE	
EVENT	ACTION	EVENT	ACTION
Home ⟶ Show Buy	Shows the components of the BUY page	GET (Cookie AuctionIDs)	Retrieves the user's latest visited auctions and the won auctions
Home ⟶ Buy ⟶ SearchByKeyword	Filters the auctions according to the keyword	GET (keyword)	Retrieves the list of auctions that contain the keyword in some way
Home ⟶ Buy ⟶ AuctionList ⟶ ShowDetails	Shows an auction's details	GET (AuctionID)	Retrieves the items and the bids of a certain auction
Home ⟶ Buy ⟶ AuctionDetails ⟶ NewBid	Places a Bid on the auction	POST (AuctionID, Import)	Adds the Bid to the DataBase, after the checks
Home ⟶ Logout		GET	Invalidate session

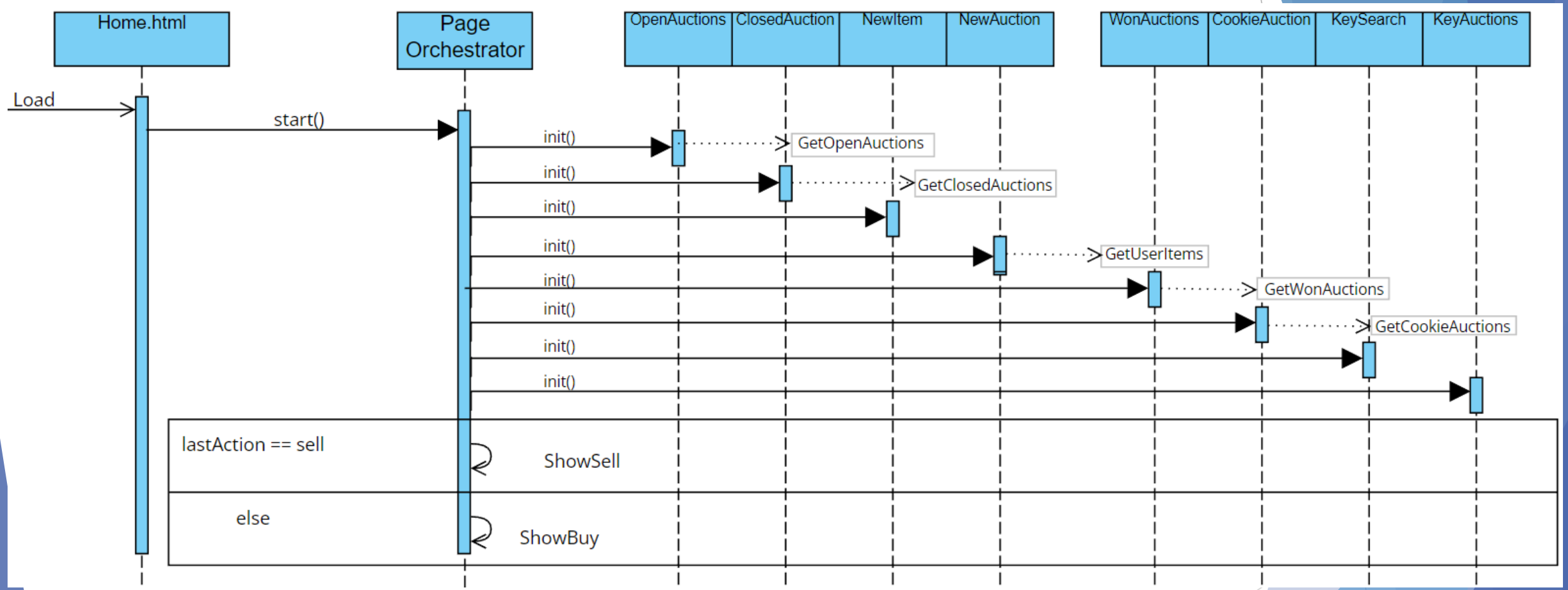
Events & Controllers

CLIENT SIDE		SERVER SIDE	
EVENT	CONTROLLER	EVENT	CONTROLLER
Index.html → login form → Submit	makeCall	POST (Username, Password)	Login
Home → load	PageOrchestrator	GET	
Home → Show Sell	OpenAuctions, ClosedAuctions	GET	SellServlet
Home → Sell → OpenAuctions → Show Details	OpenAuctionsBids, OpenAuctionsItems	GET (AuctionID)	GetItems, GetBids
Home → Sell → ClosedAuctions → Show Details	ClosedAuctionsDetails, ClosedAuctionsWinner	GET (AuctionID)	GetItems, GetWinner
Home → Sell → NewItem → Submit	makeCall	POST (form datas)	NewItem
Home → Sell → NewAuction → Submit	makeCall	POST (form datas)	NewAuction
Home → Sell → CloseAuction	makeCall	POST (AuctionID)	CloseAuction
Home → Show Buy	CookieAuctions, WonAuctions	GET (Cookie AuctionIDs)	GetCookieAuctions
Home → Buy → SearchByKeyword	KeywordForm, KeyAuctions	GET (keyword)	KeySearchServlet
Home → Buy → AuctionList → ShowDetails	BuyAuctionDetails	GET (AuctionID)	BuyServlet
Home → Buy → AuctionDetail → NewBid	makeCall	POST (AuctionID, Import)	NewBid
Home → Logout		GET	Logout

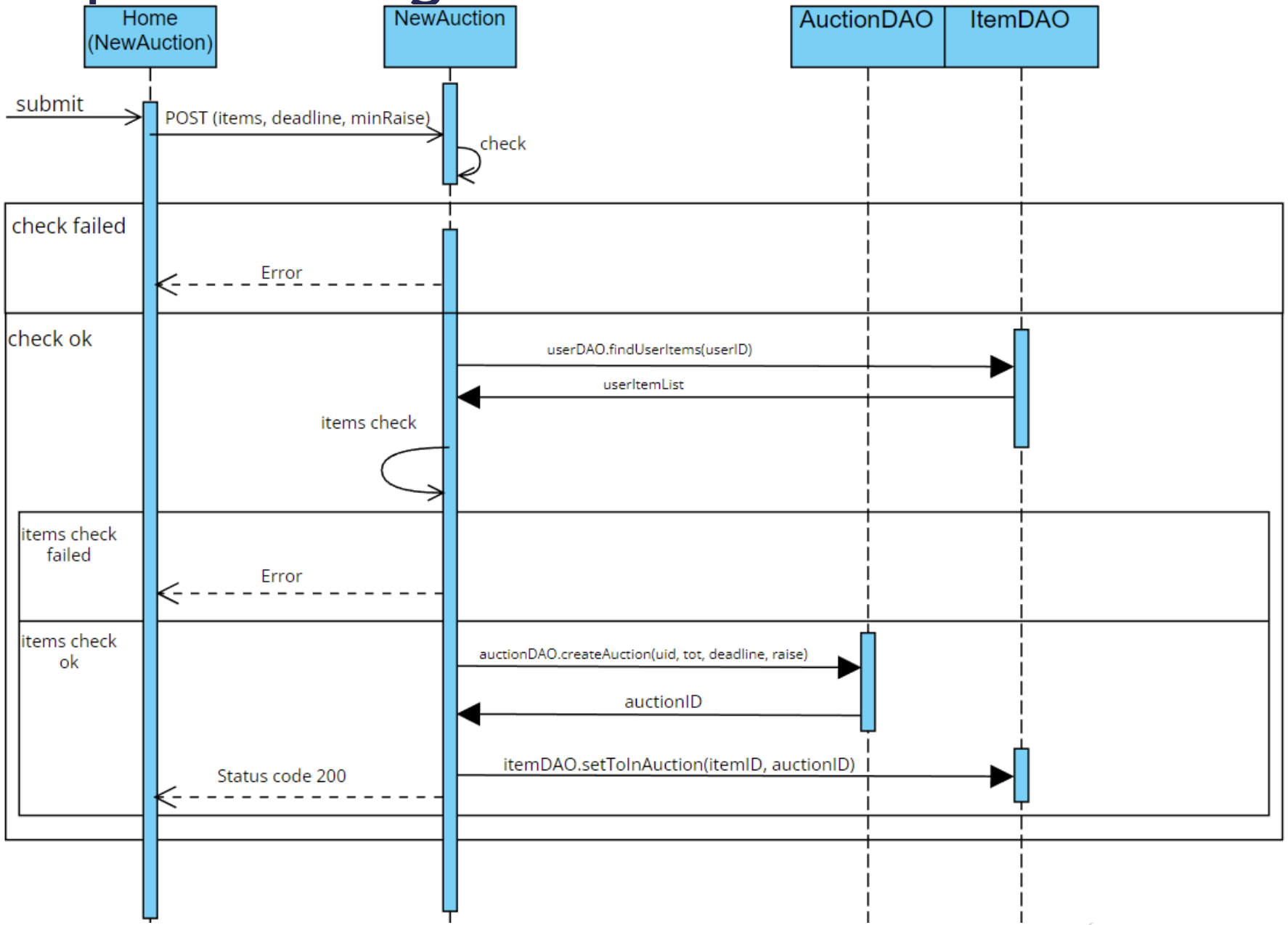
Sequence Diagrams: Login



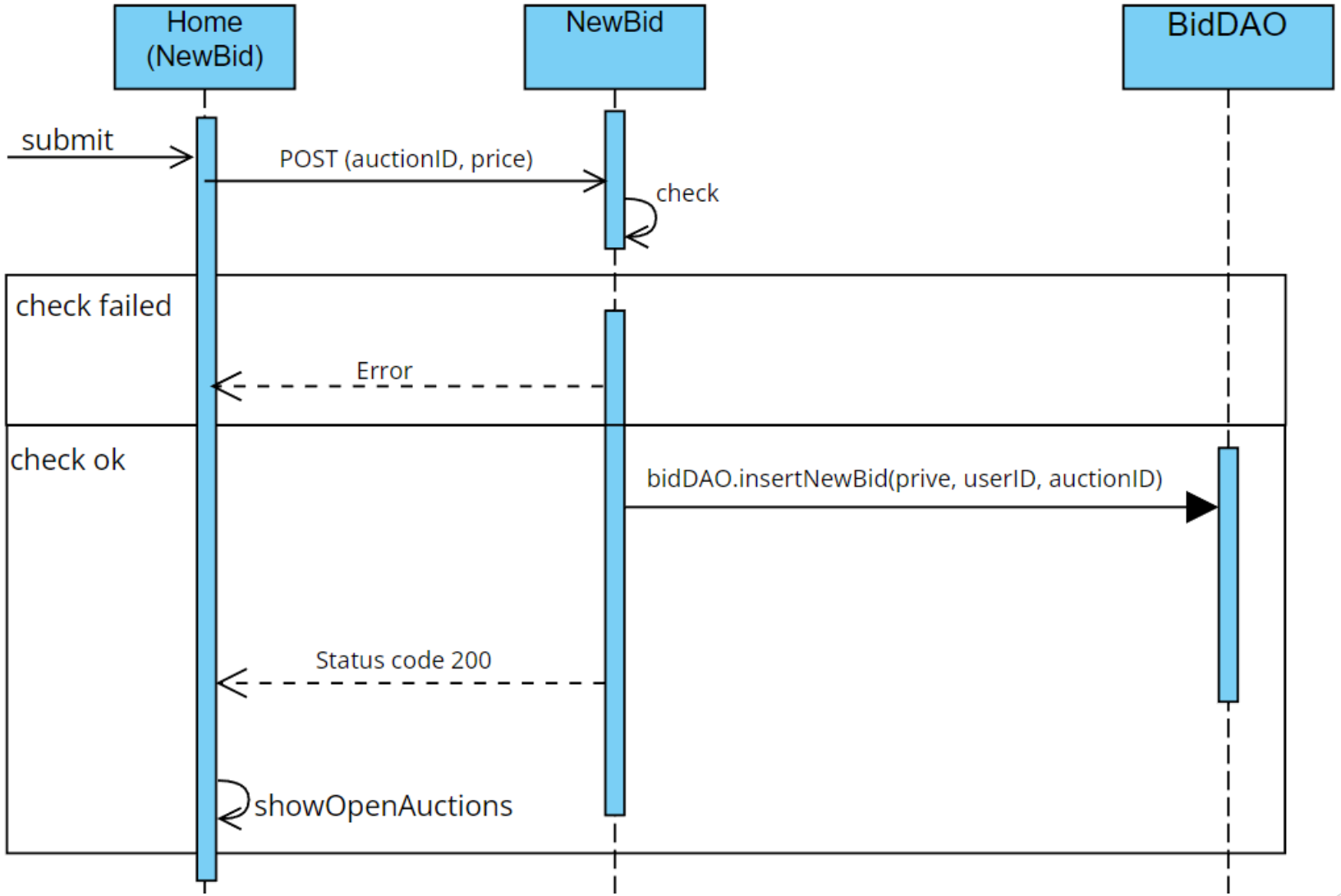
Sequence Diagrams



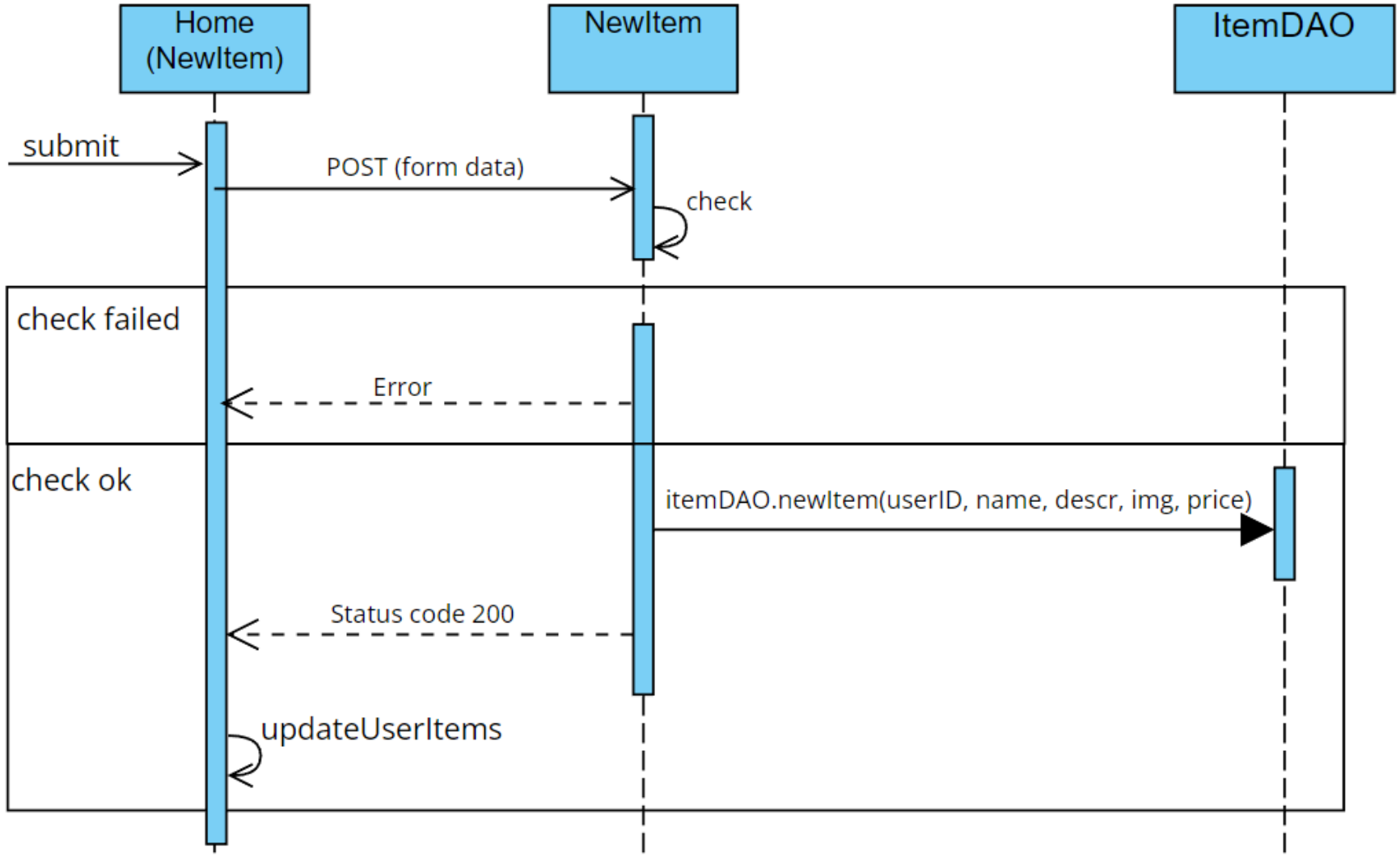
Sequence Diagrams: Add a New Auction



Sequence Diagrams: Add a New Bid



Sequence Diagrams: Add a New Item



Sequence Diagrams: Close an Auction