

Web Technologies

Esercizio 2 - Versione PureHTML

Gruppo 2

Negrini Samuele - Matr: 866797 - CP: 10539341

Plenzich Aldo - Matr: 868058 - CP: 10538310

Sala Mattia - Matr: 869766 - CP: 10540582

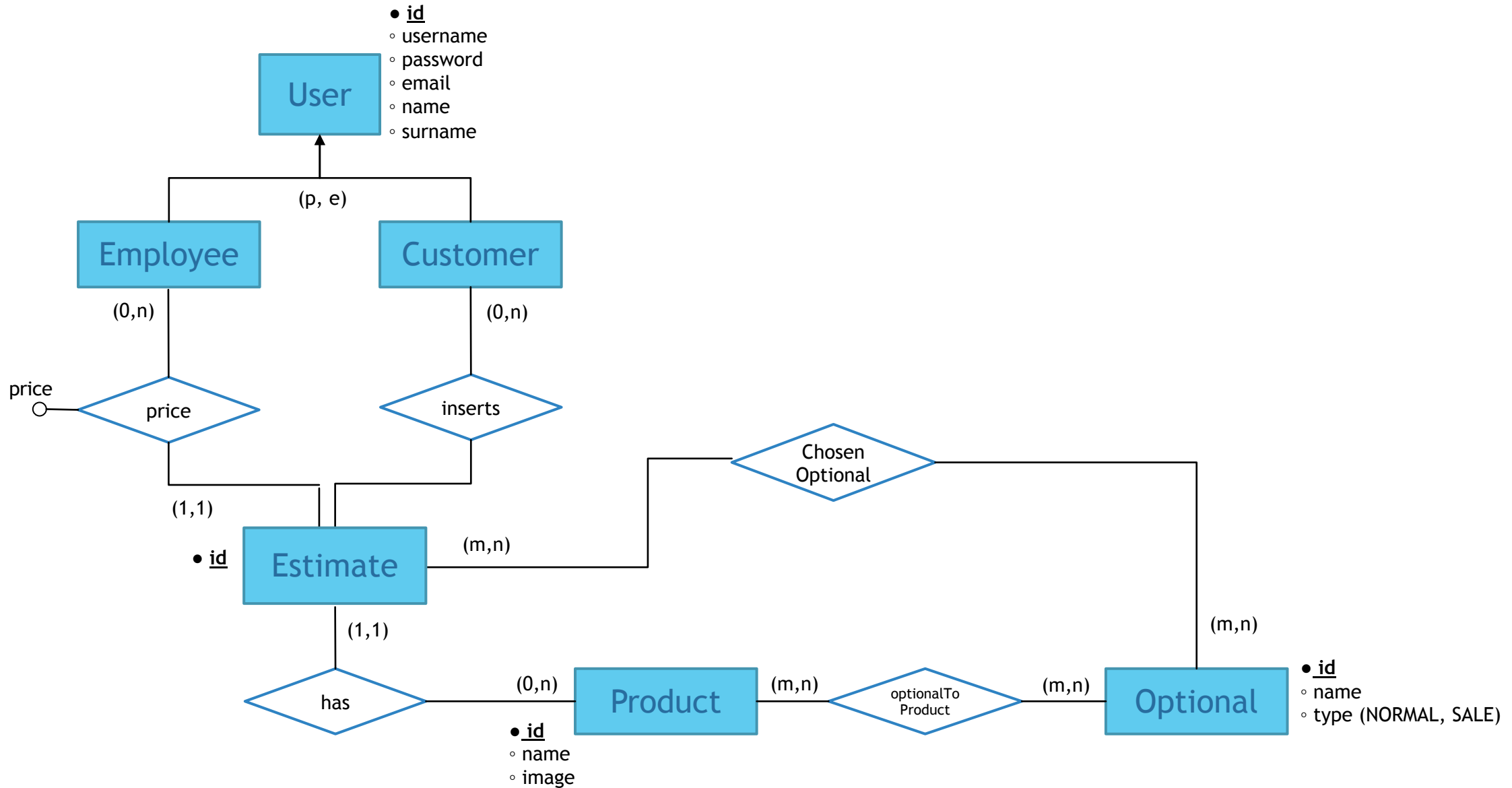
Gestione Preventivi

Un'applicazione web consente la gestione di richieste di preventivi per prodotti personalizzati. Un preventivo è associato a un prodotto, al cliente che l'ha richiesto e all'impiegato che l'ha gestito. Il preventivo comprende una o più opzioni per il prodotto a cui è associato, che devono essere tra quelle disponibili per il prodotto. Un prodotto ha un codice, un'immagine e un nome. Un'opzione ha un codice, un tipo ("normale", "in offerta") e un nome. Un preventivo ha un prezzo, definito dall'impiegato. Quando l'utente (cliente o impiegato) accede all'applicazione, appare una LOGIN PAGE, mediante la quale l'utente si autentica con username e password. Quando un cliente fa login, accede a una pagina HOME PAGE CLIENTE che contiene una form per creare un preventivo e l'elenco dei preventivi creati dal cliente. Mediante la form l'utente per prima cosa sceglie il prodotto; scelto il prodotto, la form mostra le opzioni di quel prodotto.

Gestione Preventivi

L'utente sceglie le opzioni (almeno una) e conferma l'invio del preventivo mediante il bottone INVIA PREVENTIVO. Quando un impiegato fa login, accede a una pagina HOME PAGE IMPIEGATO che contiene l'elenco dei preventivi gestiti da lui in precedenza e quello dei preventivi non ancora associati a nessun impiegato. Quando l'impiegato seleziona un elemento dall'elenco dei preventivi non ancora associati a nessuno, compare una pagina PREZZA PREVENTIVO che mostra i dati del cliente (username) e del preventivo e una form per inserire il prezzo del preventivo. Quando l'impiegato inserisce il prezzo e invia i dati con il bottone INVIA PREZZO, compare di nuovo la pagina HOME PAGE IMPIEGATO con gli elenchi dei preventivi aggiornati.

Database design



Local database schema

```
CREATE TABLE IF NOT EXISTS `user` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(50) NOT NULL,  
  `password` varchar(50) NOT NULL,  
  `email` varchar(50) NOT NULL,  
  `name` varchar(50) NOT NULL,  
  `surname` varchar(50) NOT NULL,  
  `role` enum('customer','employee') NOT NULL DEFAULT 'customer',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `username` (`username`),  
  UNIQUE KEY `email` (`email`)  
)
```

```
CREATE TABLE IF NOT EXISTS `product` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(50) NOT NULL,  
  `image` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
)
```

```
CREATE TABLE IF NOT EXISTS `optionaltopproduct` (  
  `prdid` int(11) NOT NULL,  
  `optid` int(11) NOT NULL,  
  PRIMARY KEY (`prdid`, `optid`),  
  KEY `FK_optionaltopproduct_optional` (`optid`),  
  
  CONSTRAINT `FK_optionaltopproduct_optional`  
  FOREIGN KEY (`optid`) REFERENCES `optional`  
  (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  
  CONSTRAINT `FK_optionaltopproduct_product`  
  FOREIGN KEY (`prdid`) REFERENCES `product`  
  (`id`) ON DELETE CASCADE ON UPDATE CASCADE  
)
```

```
CREATE TABLE IF NOT EXISTS `optional` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(50) NOT NULL,  
  `type` enum('NORMAL','SALE') NOT NULL DEFAULT 'NORMAL',  
  PRIMARY KEY (`id`)  
)
```

Local database schema

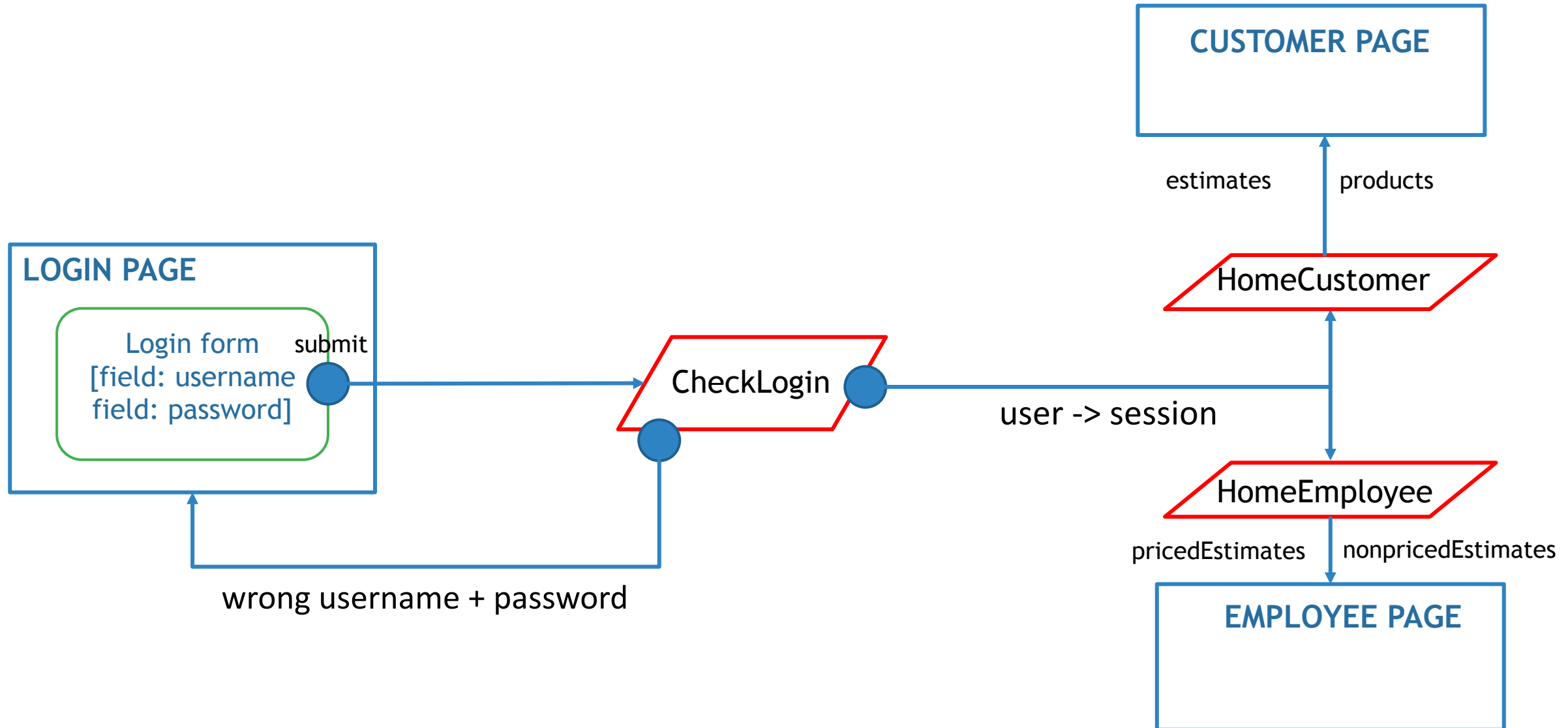
```
CREATE TABLE IF NOT EXISTS `estimate` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `usrid` int(11) NOT NULL DEFAULT '0',  
  `prdid` int(11) NOT NULL,  
  `empid` int(11) DEFAULT NULL,  
  `price` double DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `FK__user` (`usrid`),  
  KEY `FK_estimate_product` (`prdid`),  
  KEY `FK_order_user` (`empid`),  
  
  CONSTRAINT `FK_estimate_product`  
  FOREIGN KEY (`prdid`) REFERENCES `product` (`id`)  
  ON DELETE CASCADE ON UPDATE CASCADE,  
  
  CONSTRAINT `FK_order_user`  
  FOREIGN KEY (`empid`) REFERENCES `user` (`id`)  
  ON DELETE CASCADE ON UPDATE CASCADE  
)
```

```
CREATE TABLE IF NOT EXISTS `chosenoptional` (  
  `estid` int(11) NOT NULL,  
  `optid` int(11) NOT NULL,  
  PRIMARY KEY (`estid`, `optid`),  
  KEY `FK_chosenoptional_optionaltopproduct` (`optid`)  
  
  CONSTRAINT `FK_chosenoptional_estimate`  
  FOREIGN KEY (`estid`) REFERENCES `estimate` (`id`)  
  ON DELETE CASCADE ON UPDATE CASCADE,  
  
  CONSTRAINT `FK_chosenoptional_optional`  
  FOREIGN KEY (`optid`) REFERENCES `optional` (`id`)  
  ON DELETE CASCADE ON UPDATE CASCADE  
)
```

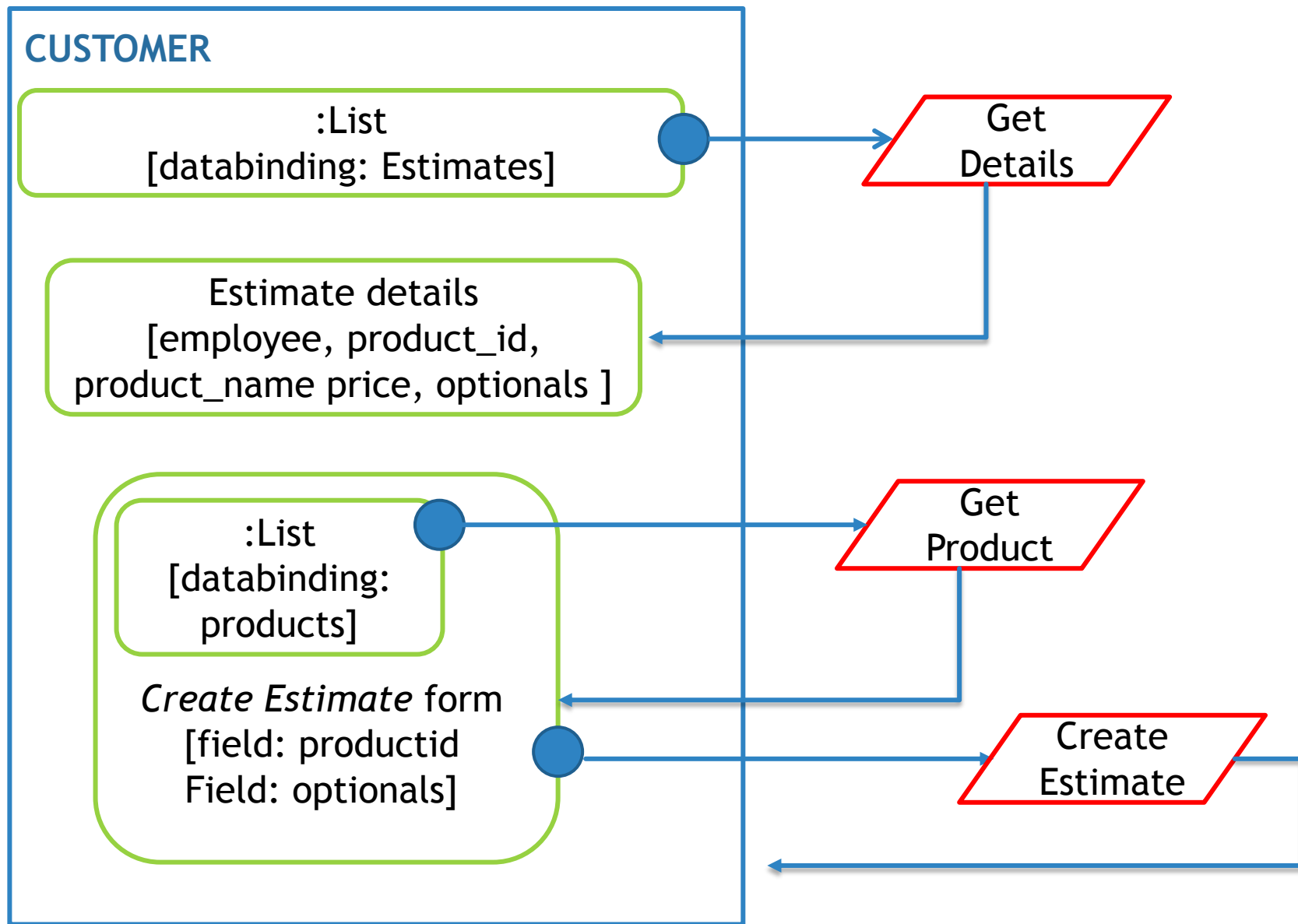
Gestione Preventivi

Un'applicazione web consente la gestione di richieste di preventivi per prodotti personalizzati. Un preventivo è associato a un prodotto, al cliente che l'ha richiesto e all'impiegato che l'ha gestito. Il preventivo comprende una o più opzioni per il prodotto a cui è associato, che devono essere tra quelle disponibili per il prodotto. Un prodotto ha un codice, un'immagine e un nome. Un'opzione ha un codice, un tipo ("normale", "in offerta") e un nome. Un preventivo ha un prezzo, definito dall'impiegato. Quando l'utente (cliente o impiegato) accede all'applicazione, appare una **LOGIN PAGE**, mediante la quale l'utente si autentica con username e password. Quando un cliente fa login, accede a una **pagina HOME PAGE CLIENTE** che contiene una **form per creare un preventivo** e **l'elenco dei preventivi creati dal cliente**. Mediante la form l'utente per prima cosa sceglie il prodotto; scelto il prodotto, la **form mostra le opzioni di quel prodotto**. L'utente sceglie le opzioni (almeno una) e conferma l'invio del preventivo mediante il **bottone INVIA PREVENTIVO**. Quando un impiegato fa login, accede a una **pagina HOME PAGE IMPIEGATO** che **contiene l'elenco dei preventivi** gestiti da lui in precedenza e quello dei **preventivi non ancora associati a nessun impiegato**. Quando l'impiegato seleziona un elemento dall'elenco dei preventivi non ancora associati a nessuno, **compare** una **pagina PREZZA PREVENTIVO** che **mostra i dati del cliente** (username) e del preventivo e una **form per inserire il prezzo del preventivo**. Quando l'impiegato **inserisce il prezzo** e **invia i dati** con il **bottone INVIA PREZZO**, **compare** di nuovo la **pagina HOME PAGE IMPIEGATO** **con gli elenchi dei preventivi aggiornati**.

Application design - login



Application design - Customer



Application design - Employee

EMPLOYEE

:List
[databinding: Estimates]

select

Get
Details

Estimate details
[customer, product_id,
product_name price, optionals]

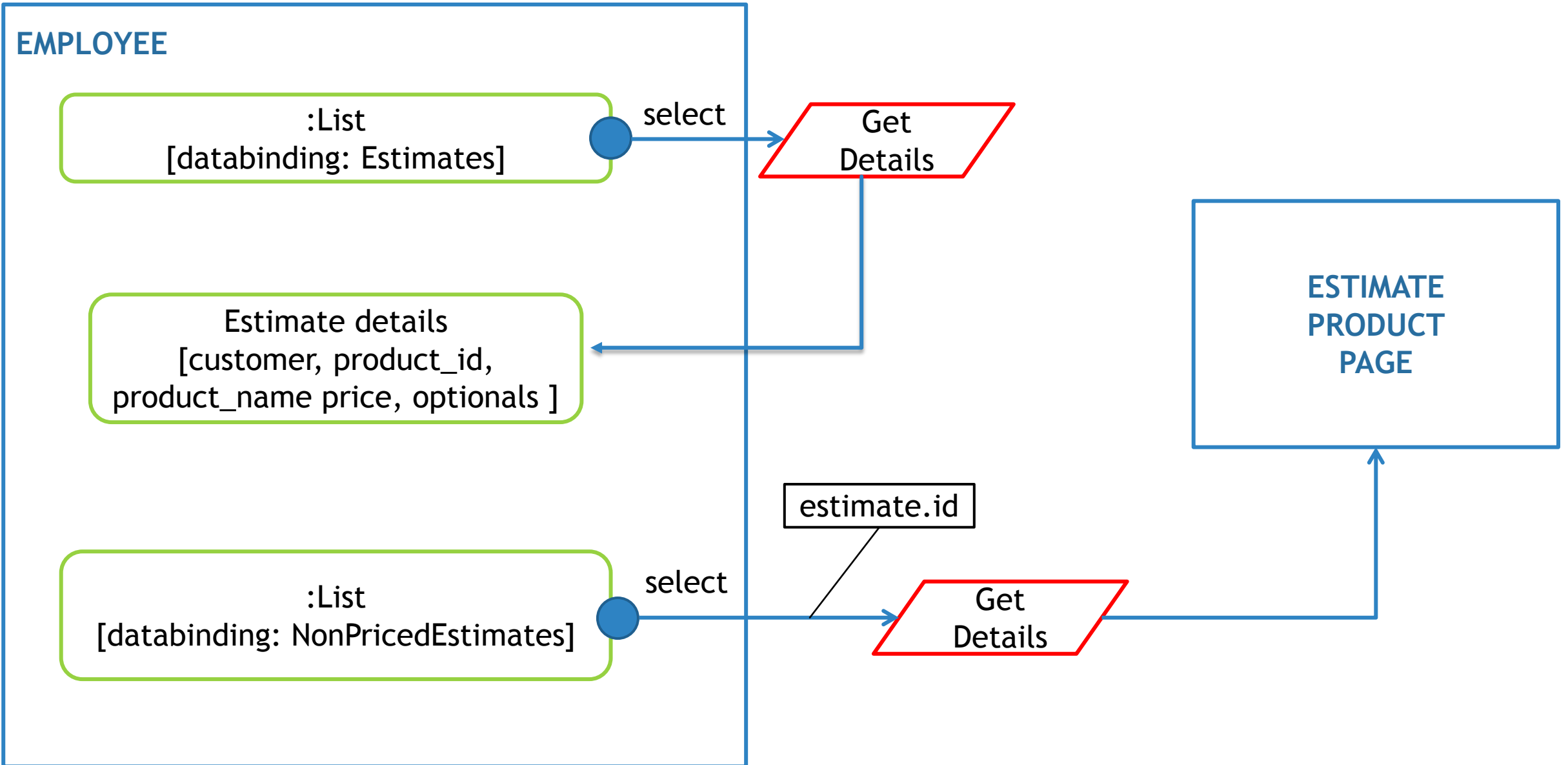
:List
[databinding: NonPricedEstimates]

select

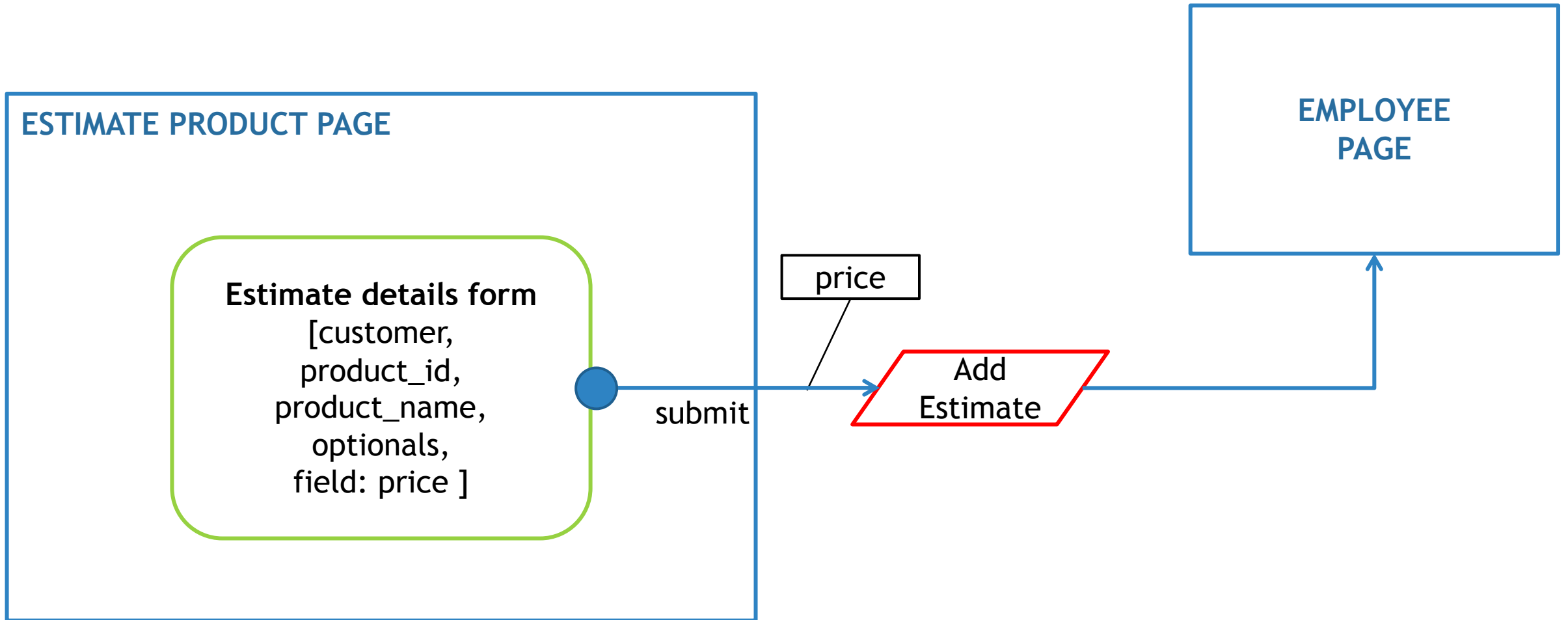
Get
Details

estimate.id

ESTIMATE
PRODUCT
PAGE



Application design - Employee



Components

Model Objects (Beans):

- Estimate
- Optional
- Optional Type
- Product
- User

Data Access Objects (Classes)

- EstimateDAO
- OptionalDAO
- ProductDAO
- UserDAO

Filters

- AuthChecker
- CustomerChecker
- EmployeeChecker

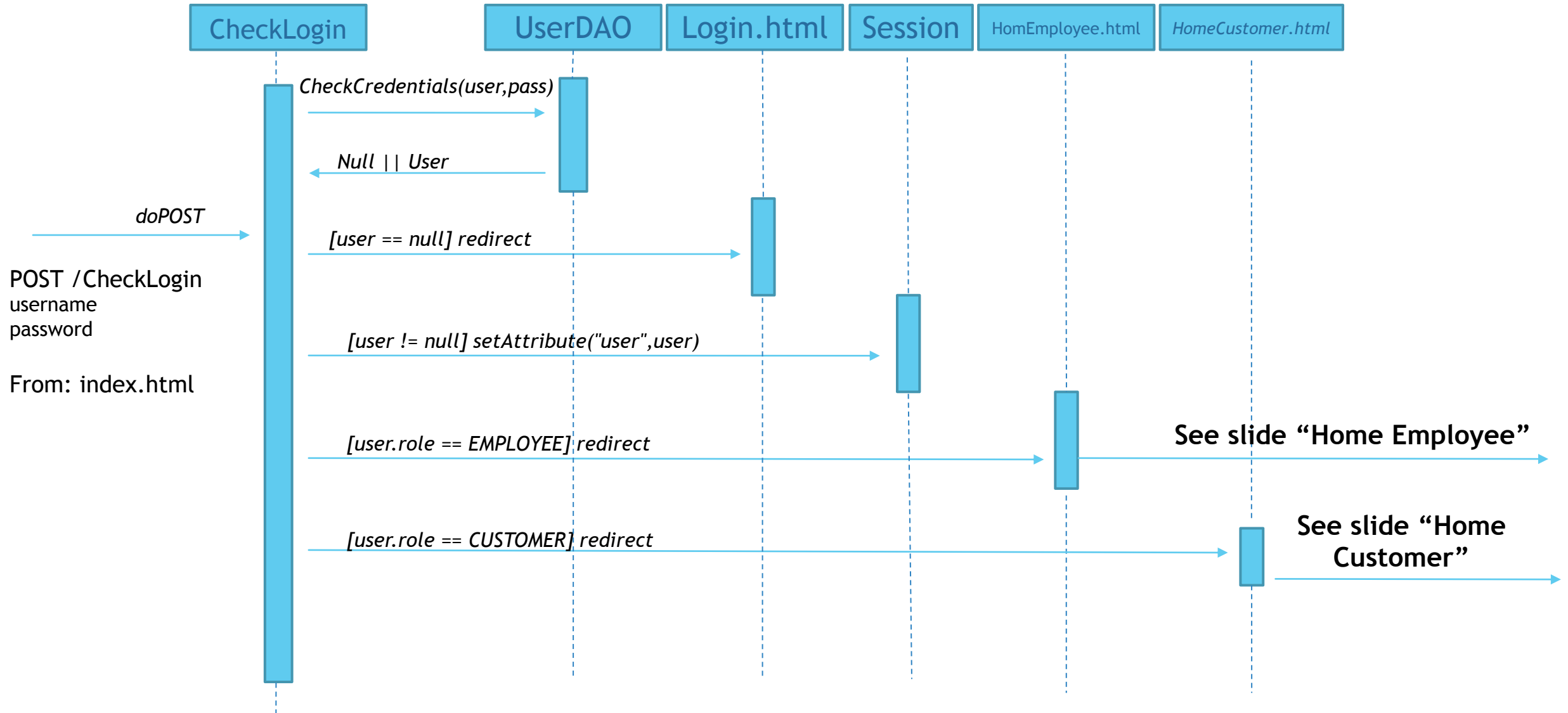
Controllers (Servlets):

- CheckLogin
- AddEstimatePrice
- CreateEstimate
- CreateEstimatePrice
- HomeCustomer
- HomeEmployee
- Logout

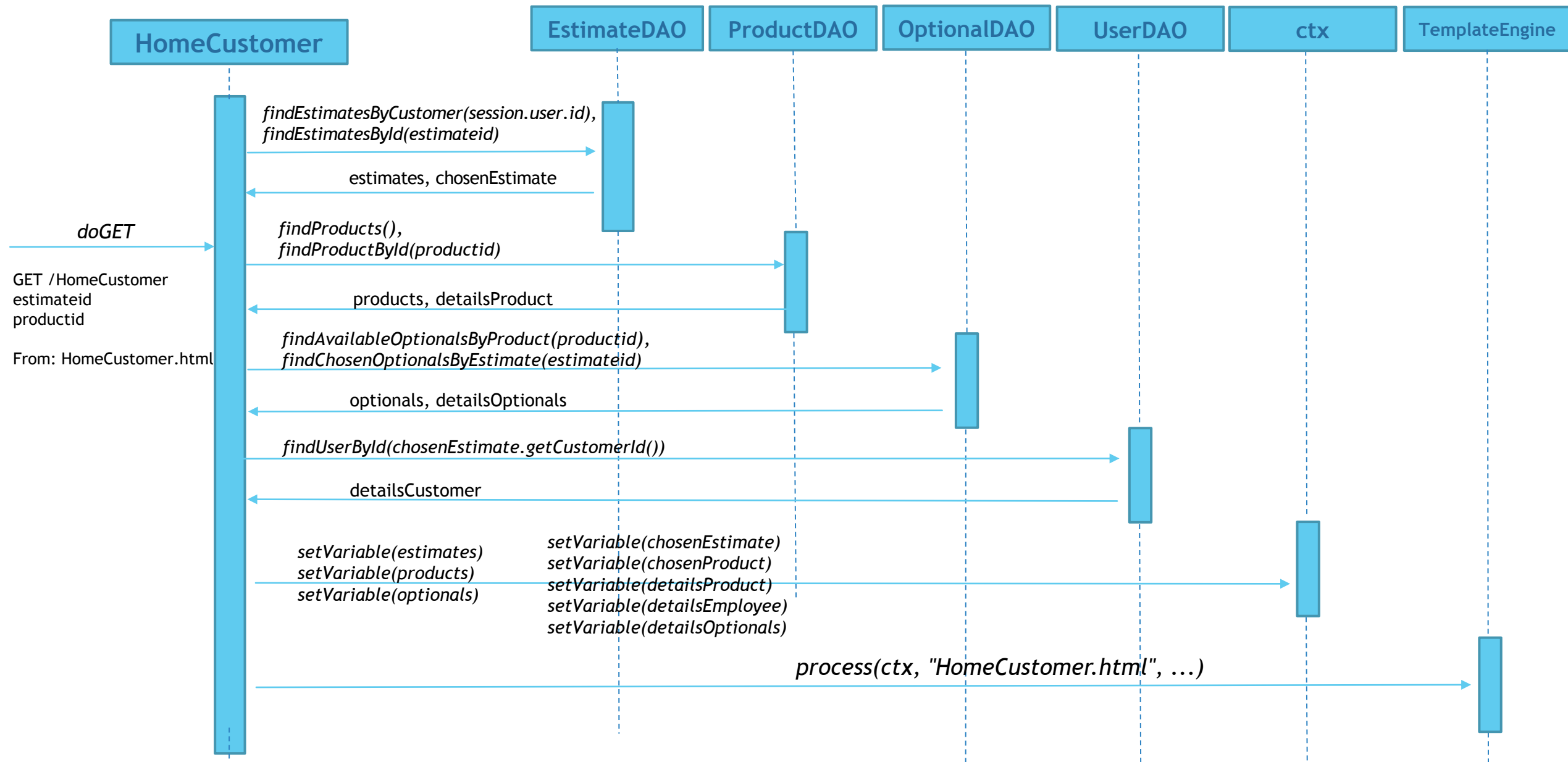
Views (Templates)

- Login
- HomeCustomer
- HomeEmployee

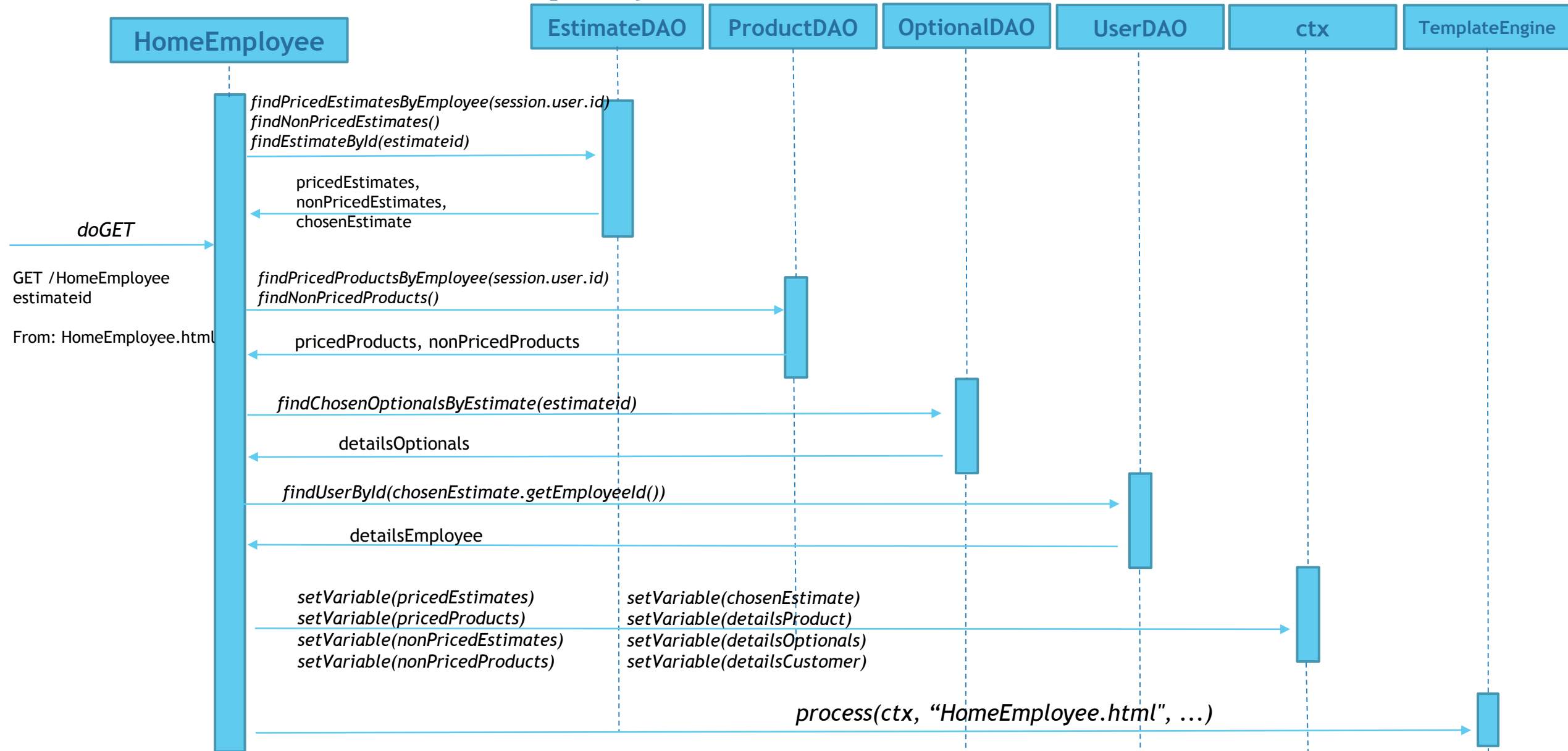
Event: login



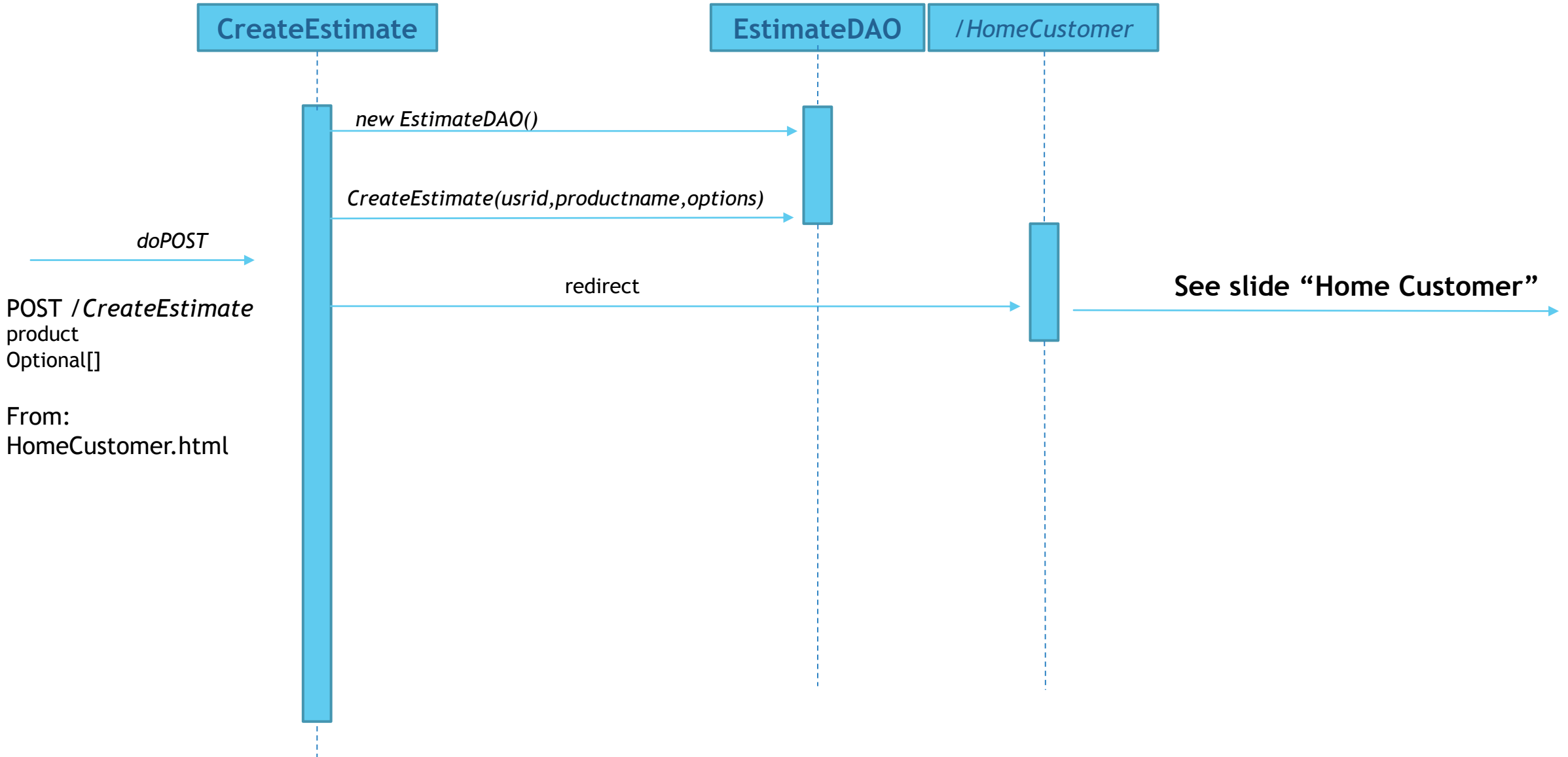
Event: Home Customer



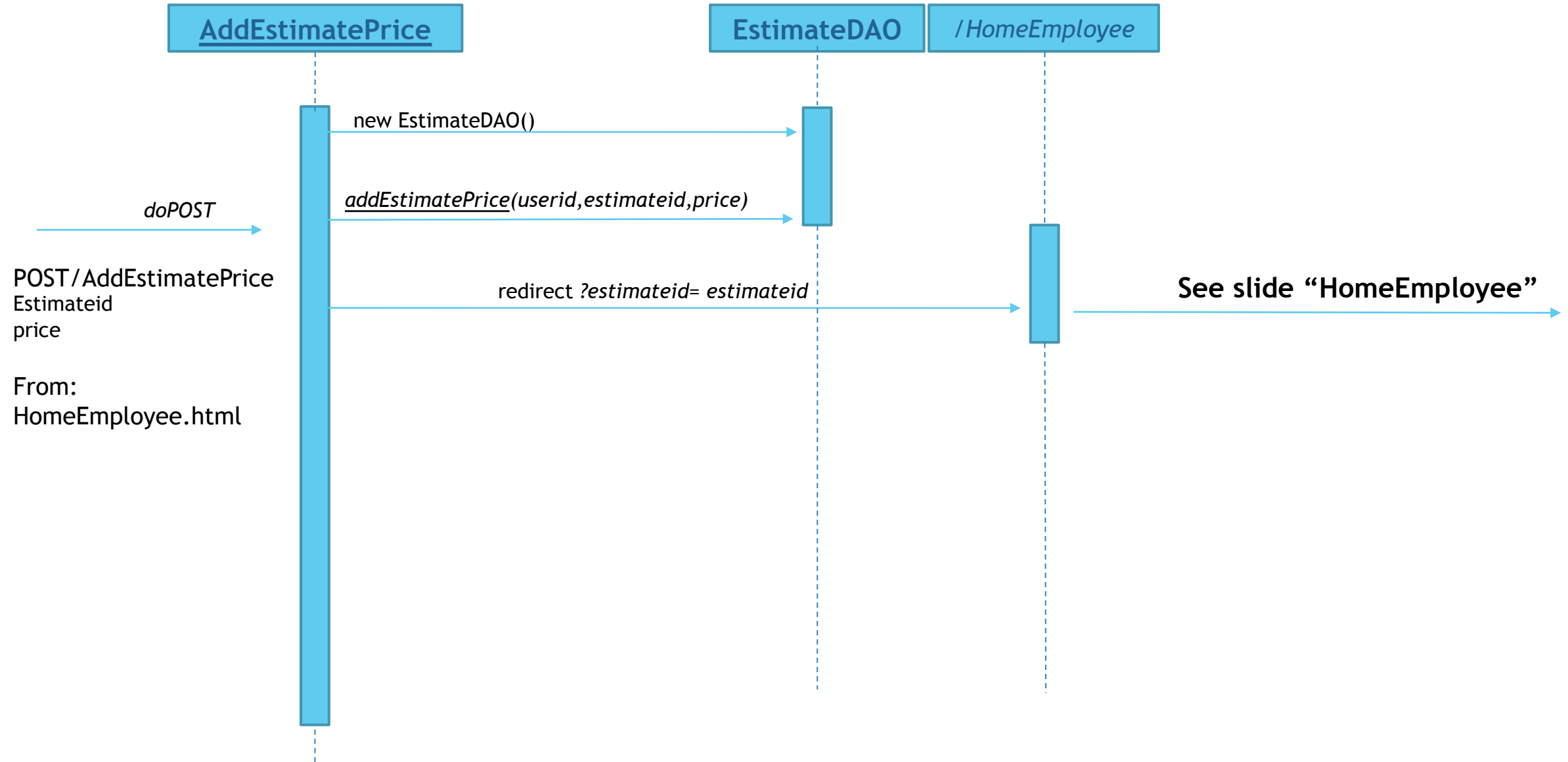
Event: Home Employee



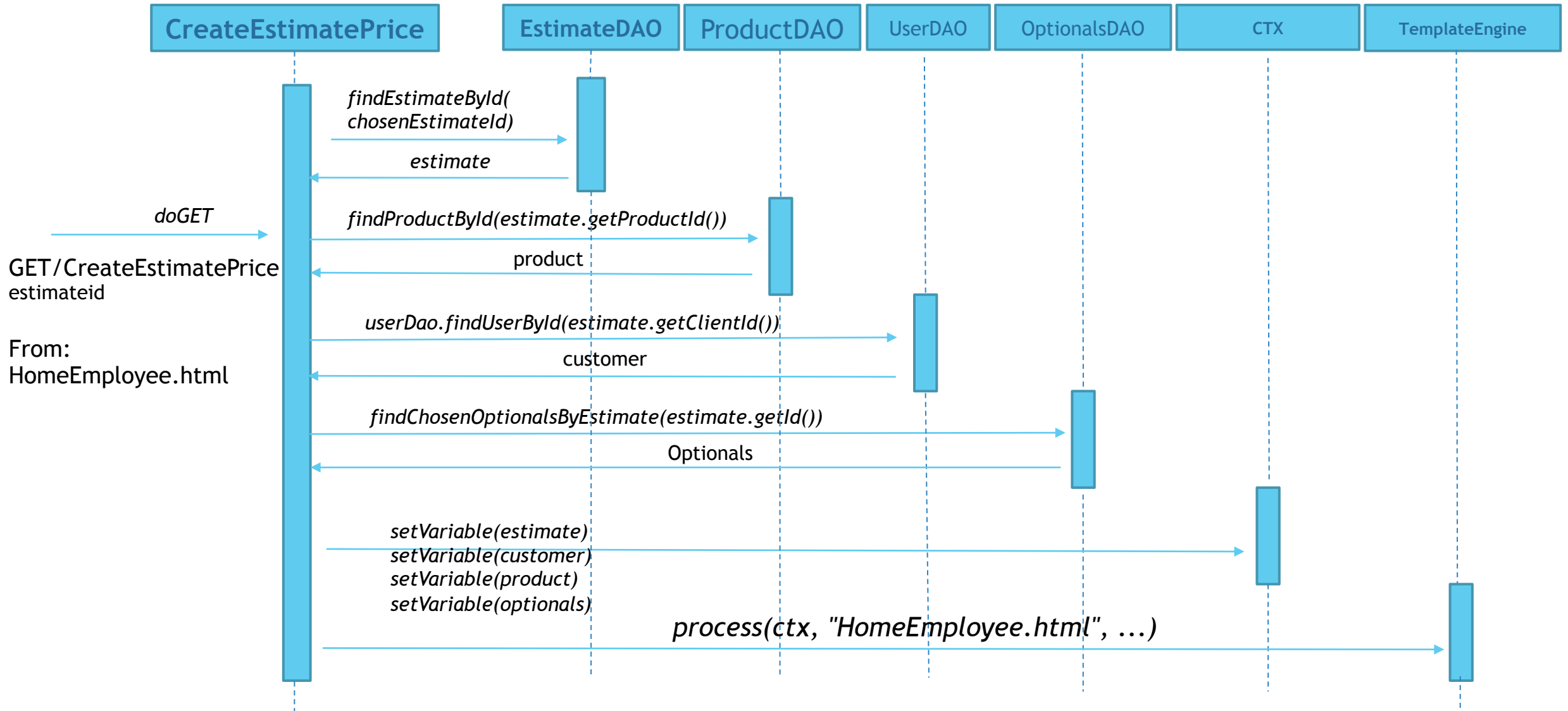
Event: CreateEstimate



Event: AddEstimatePrice



Event: CreateEstimatePrice



Event: Logout

