

# Documentazione di progetto

TRACCIA 2 – GESTIONE PREVENTIVI – PURE HTML

CARLO CHIODAROLI

# Specifica

## Traccia 2 – Gestione preventivi

### Versione pure HTML

Un'applicazione web consente la gestione di richieste di preventivi per prodotti personalizzati. L'applicazione supporta registrazione e login di clienti e impiegati mediante una pagina pubblica con opportune form. La registrazione controlla l'unicità dello username. 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. Selezionando uno dei preventivi il cliente ne visualizza i dettagli. Mediante la form di creazione di un preventivo 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 effettua il 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. Il prezzo definito dall'impiegato risulta visibile al cliente quando questi accede all'elenco dei propri preventivi e visualizza i dettagli del preventivo. La pagina PREZZA PREVENTIVO contiene anche un collegamento per tornare alla HOME PAGE IMPIEGATO. L'applicazione consente il logout dell'utente.

### Versione con JavaScript

Si realizzi un'applicazione client server web che modifica le specifiche precedenti come segue:

- La registrazione controlla la validità sintattica dell'indirizzo di e-mail e l'uguaglianza tra i campi "password" e "ripeti password", anche a lato client.
- Dopo il login, l'intera applicazione è realizzata con un'unica pagina per ciascuno dei ruoli: una pagina singola per il ruolo di cliente e una pagina singola per il ruolo di impiegato.
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica del contenuto da aggiornare a seguito dell'evento.
- Nella pagina del cliente, la scelta del prodotto comporta la successiva visualizzazione delle opzioni senza produrre un'ulteriore chiamata al server.
- L'invio del preventivo da parte del cliente deve produrre la verifica dei dati anche a lato client (almeno un'opzione scelta).
- Nella pagina dell'impiegato, il controllo del prezzo (non nullo e maggiore di zero) deve essere fatto anche a lato client.
- Eventuali errori a lato server devono essere segnalati mediante un messaggio di allerta all'interno della pagina del cliente o dell'impiegato.

# Base di dati

## Appunti su specifica

Legenda: **Entità** **Attributi** **Relazioni**

### Versione pure HTML

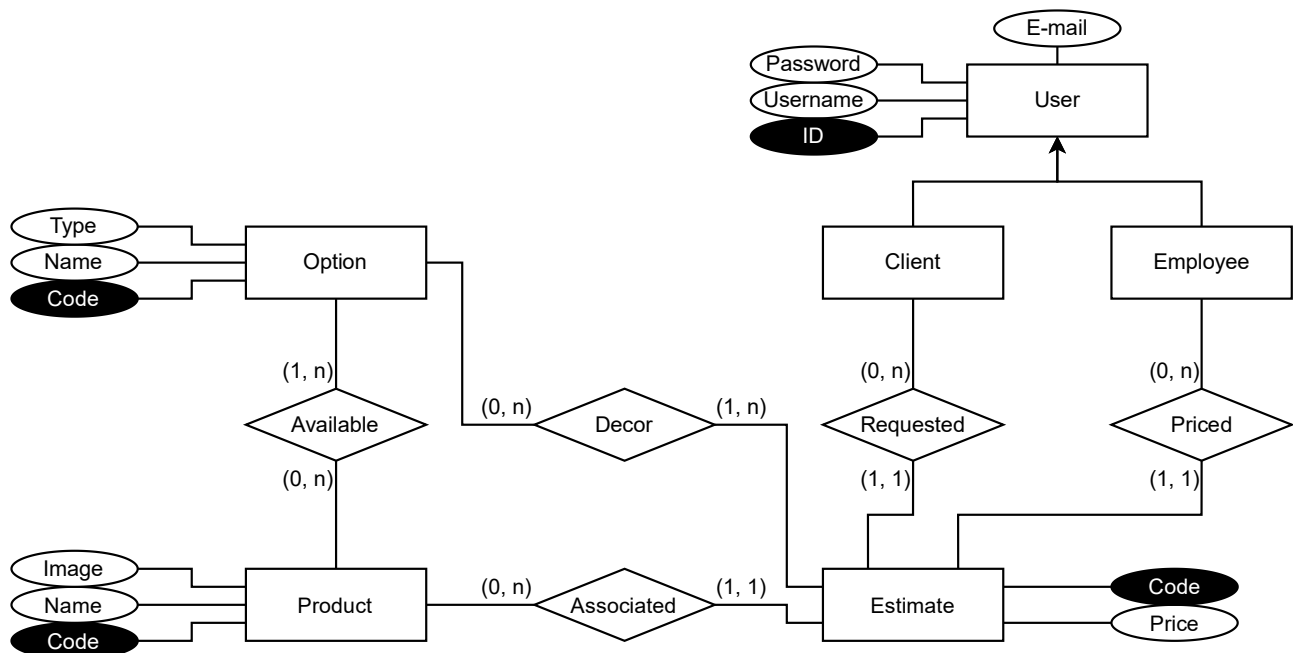
Un'applicazione web consente la gestione di richieste di **preventivi** per **prodotti** personalizzati. L'applicazione supporta registrazione e login di **clienti** e **impiegati** mediante una pagina pubblica con opportune form. La registrazione controlla l'**unicità** dello **username**. Un **preventivo** 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. Selezionando uno dei preventivi il cliente ne visualizza i dettagli. Mediante la form di creazione di un preventivo 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 effettua il 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. Il prezzo definito dall'impiegato risulta visibile al cliente quando questi accede all'elenco dei propri preventivi e visualizza i dettagli del preventivo. La pagina PREZZA PREVENTIVO contiene anche un collegamento per tornare alla HOME PAGE IMPIEGATO. L'applicazione consente il logout dell'utente.

### Versione con JavaScript

Si realizzi un'applicazione client server web che modifica le specifiche precedenti come segue:

- La registrazione controlla la validità sintattica dell'**indirizzo di e-mail** e l'uguaglianza tra i campi "password" e "ripeti password", anche a lato client.
- Dopo il login, l'intera applicazione è realizzata con un'unica pagina per ciascuno dei ruoli: una pagina singola per il ruolo di cliente e una pagina singola per il ruolo di impiegato.
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica del contenuto da aggiornare a seguito dell'evento.
- Nella pagina del cliente, la scelta del prodotto comporta la successiva visualizzazione delle opzioni senza produrre un'ulteriore chiamata al server.
- L'invio del preventivo da parte del cliente deve produrre la verifica dei dati anche a lato client (almeno un'opzione scelta).
- Nella pagina dell'impiegato, il controllo del prezzo (non nullo e maggiore di zero) deve essere fatto anche a lato client.
- Eventuali errori a lato server devono essere segnalati mediante un messaggio di allerta all'interno della pagina del cliente o dell'impiegato.

## Schema Entità Relazione



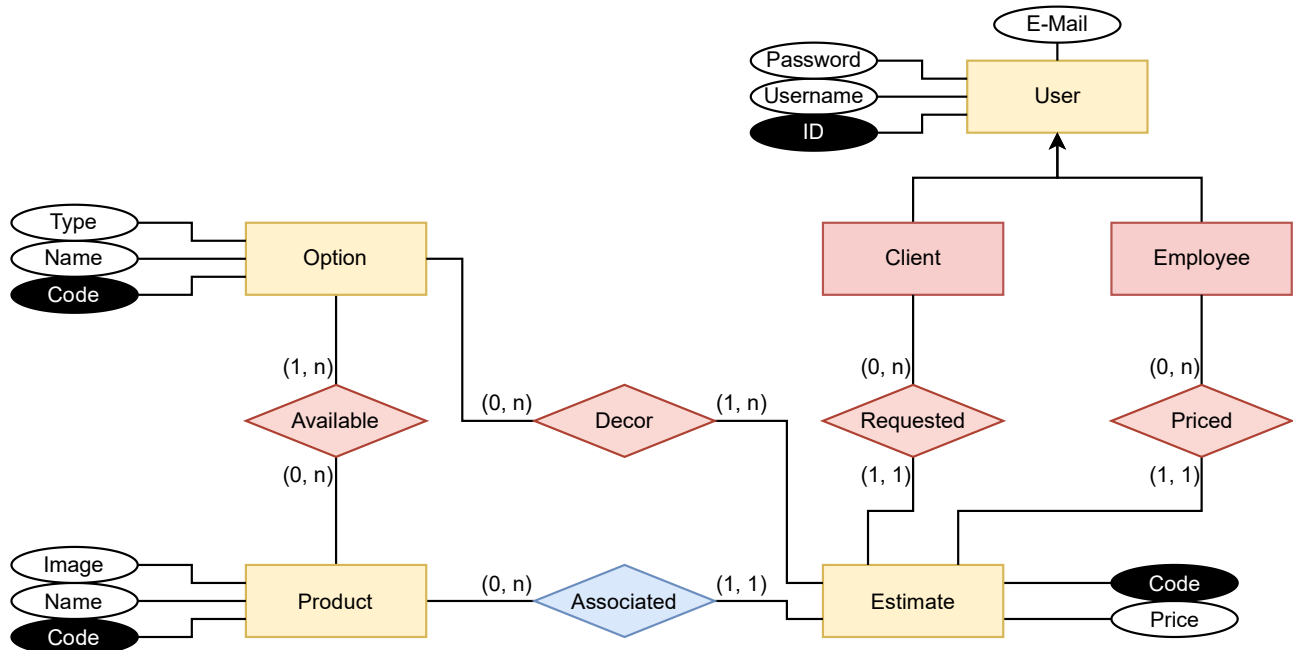
### ER con commenti

Entità gialle: Entità diventate tabelle

Entità Rosse: Entità incorporate nella entità padre

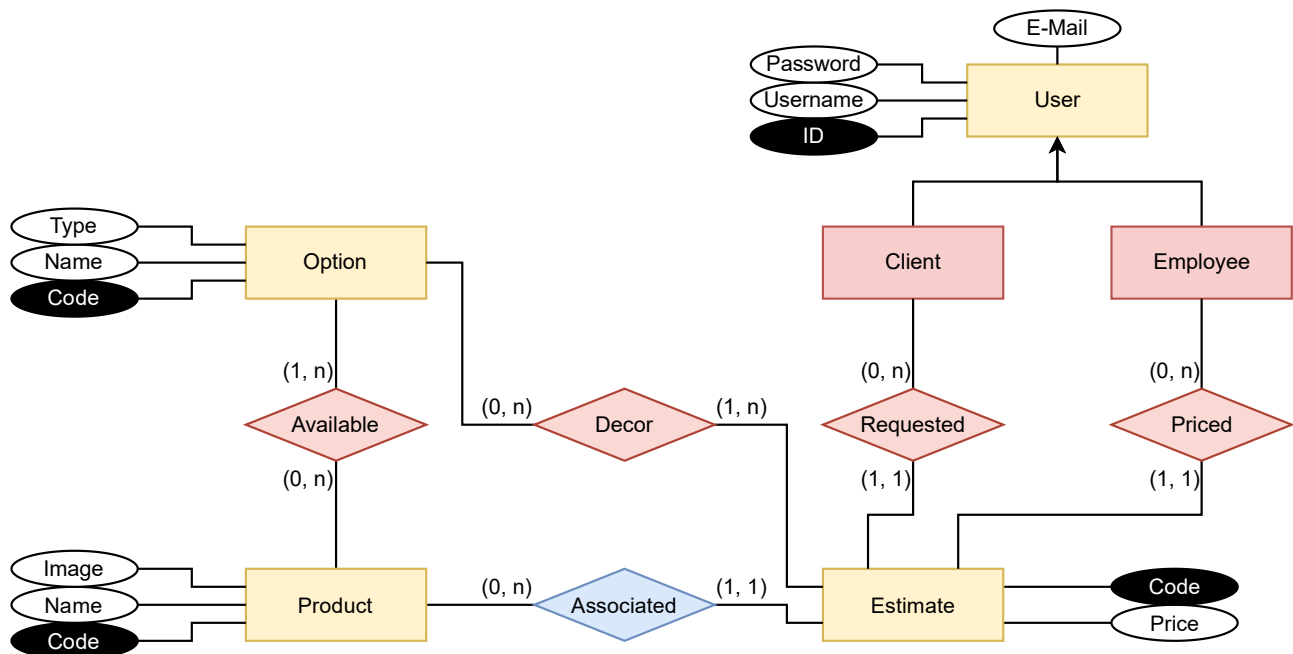
Relazioni azzurre: Relazioni implementate mediante l'uso di attributi

Relazioni verdi: Relazioni implementate mediante l'uso di tabella di relazione

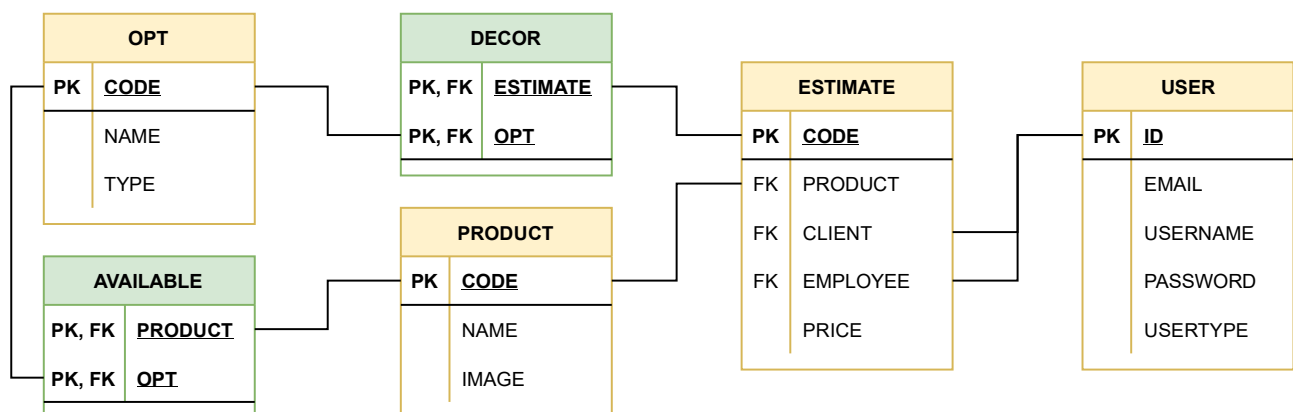


## Schema ER e Schema Logico a confronto

Entità Relazione:

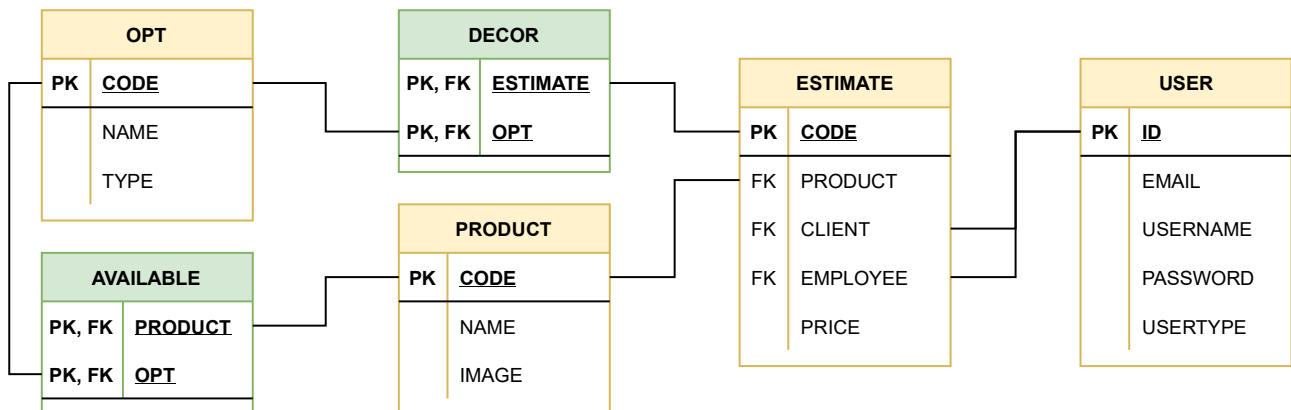


Logico:



## Istruzioni SQL Create

Schema Logico di riferimento



Istruzioni:

```
CREATE TABLE PRODUCT(
    CODE INT NOT NULL AUTO_INCREMENT,
    NAME VARCHAR(45) NOT NULL,
    IMAGE VARCHAR(45),
    PRIMARY KEY (CODE));
```

```
CREATE TABLE OPT(
    CODE INT NOT NULL AUTO_INCREMENT,
    TYPE VARCHAR(45) NOT NULL,
    NAME VARCHAR(45) NOT NULL,
    PRIMARY KEY (CODE));
```

```
CREATE TABLE AVAILABILITY(
    PRODUCT INT NOT NULL,
    OPT INT NOT NULL,
    PRIMARY KEY (PRODUCT, OPT),
    CONSTRAINT AVFOR
        FOREIGN KEY (PRODUCT)
            REFERENCES PRODUCT (CODE)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT AVBY
        FOREIGN KEY (OPT)
            REFERENCES OPT (CODE)
            ON UPDATE CASCADE
            ON DELETE CASCADE);
```

```
CREATE TABLE USER(
    ID INT NOT NULL AUTO_INCREMENT,
    EMAIL VARCHAR(45) NOT NULL,
    USERNAME VARCHAR(45) NOT NULL UNIQUE,
    PASSWORD VARCHAR(45) NOT NULL,
    USERTYPE VARCHAR(45) NOT NULL,
    PRIMARY KEY (ID));
```

```
CREATE TABLE ESTIMATE(
    CODE INT NOT NULL AUTO_INCREMENT,
    PRODUCT INT NOT NULL,
    CLIENT INT NOT NULL,
    EMPLOYEE INT DEFAULT NULL,
    PRICE DOUBLE DEFAULT NULL,
    PRIMARY KEY (CODE),
    CONSTRAINT PROD
        FOREIGN KEY (PRODUCT)
            REFERENCES PRODUCT (CODE)
            ON UPDATE CASCADE
            ON DELETE CASCADE);
```

```
CREATE TABLE DECOR(
    ESTIMATE INT NOT NULL,
    OPT INT NOT NULL,
    PRIMARY KEY (ESTIMATE, OPT),
    CONSTRAINT DECORATING
        FOREIGN KEY (ESTIMATE)
            REFERENCES ESTIMATE (CODE)
            ON UPDATE CASCADE
            ON DELETE CASCADE,
    CONSTRAINT DECORATOR
        FOREIGN KEY (OPT)
            REFERENCES OPT (CODE)
            ON UPDATE CASCADE
            ON DELETE CASCADE);
```

# Applicazione pure HTML

## Appunti su specifica

Legenda: **Pagine**, **Componenti vista**, **Eventi**, **Azioni**

### Versione pure HTML

Un'applicazione web consente la gestione di richieste di preventivi per prodotti personalizzati. L'applicazione supporta **registrazione** e **login** di clienti e impiegati mediante una pagina pubblica con opportune form. La registrazione controlla l'unicità dello username. 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**. **Selezionando uno dei preventivi** il cliente ne **visualizza i dettagli**. Mediante la **form di creazione** di un preventivo 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 effettua il 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. Il **prezzo definito dall'impiegato** risulta visibile al cliente quando questi accede all'elenco dei propri preventivi e **visualizza i dettagli del preventivo**. La pagina **PREZZA PREVENTIVO** contiene anche un collegamento per **tornare** alla **HOME PAGE IMPIEGATO**. L'applicazione consente il **logout dell'utente**.

### Versione con JavaScript

Si realizzi un'applicazione client server web che modifica le specifiche precedenti come segue:

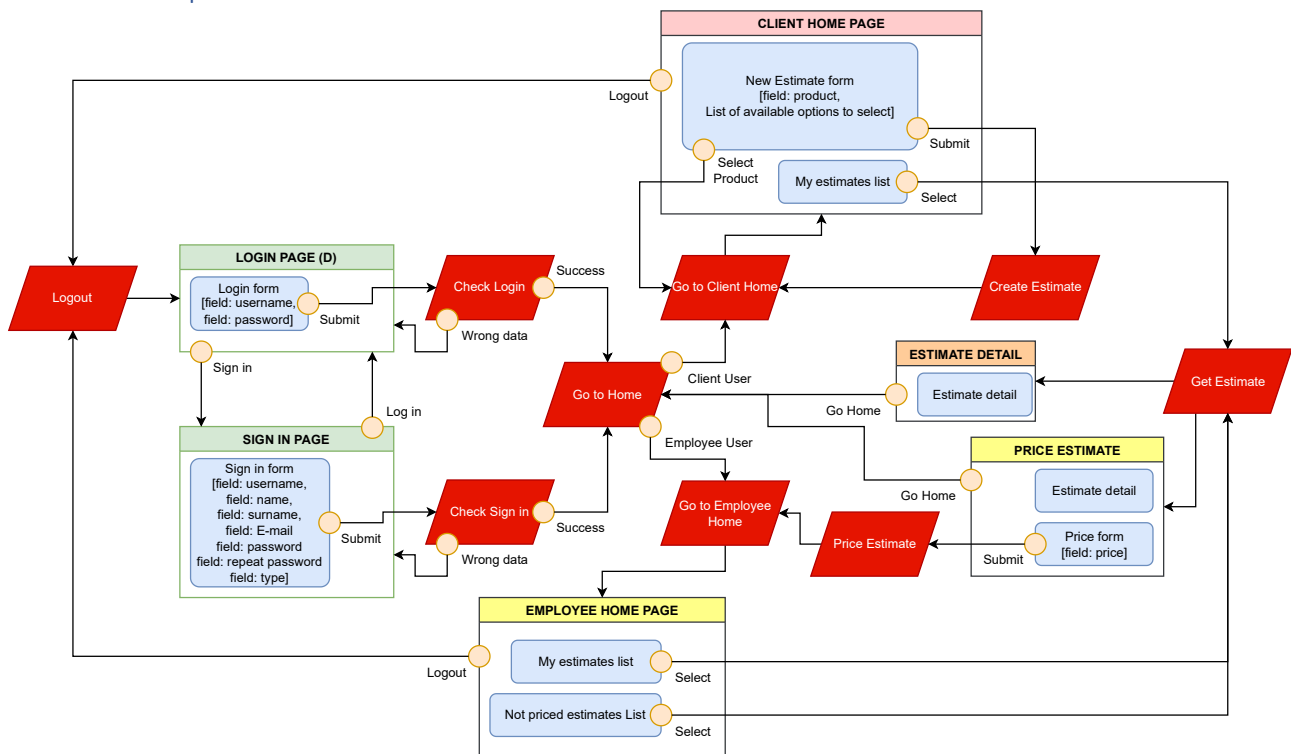
- La registrazione controlla la validità sintattica **dell'indirizzo di e-mail** e l'uguaglianza tra i **campi "password" e "ripeti password"**, anche a lato client.
- Dopo il login, l'intera applicazione è realizzata con un'unica pagina per ciascuno dei ruoli: una pagina singola per il ruolo di cliente e una pagina singola per il ruolo di impiegato.
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica del contenuto da aggiornare a seguito dell'evento.
- Nella pagina del cliente, la scelta del prodotto comporta la successiva visualizzazione delle opzioni senza produrre un'ulteriore chiamata al server.
- L'invio del preventivo da parte del cliente deve produrre la verifica dei dati anche a lato client (almeno un'opzione scelta).
- Nella pagina dell'impiegato, il controllo del prezzo (non nullo e maggiore di zero) deve essere fatto anche a lato client.
- Eventuali errori a lato server devono essere segnalati mediante un messaggio di allerta all'interno della pagina del cliente o dell'impiegato.

## Completamento delle Specifiche

- Pagina di login e Pagina di registrazione
- Pagina di registrazione contiene una form di registrazione con campi: Username, e-mail, password e ripeti password; contiene pulsanti "Submit".
- Pagina DETTAGLI PREVENTIVO per vedere i dettagli del preventivo
- Anche un'impiegato, cliccando su un preventivo gestito in precedenza, può vederne i dettagli
- Tasto di Logout e di Home in ogni pagina delle zone private
- Controllo che il prezzo immesso dall'impiegato abbia un valore positivo non nullo
- Pagina dettagli preventivo mostra l'immagine del prodotto
- Username sono unici tra diversi tipi di utenti (Non si può avere un impiegato con uno stesso username di un cliente).

## Schema di applicazione

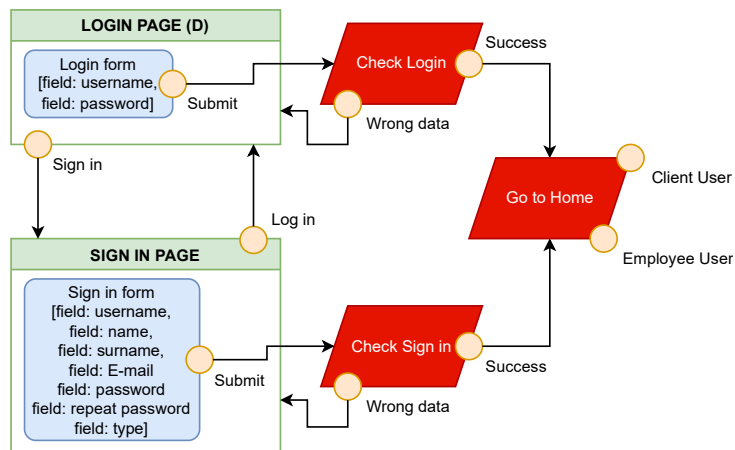
### Schema Complessivo



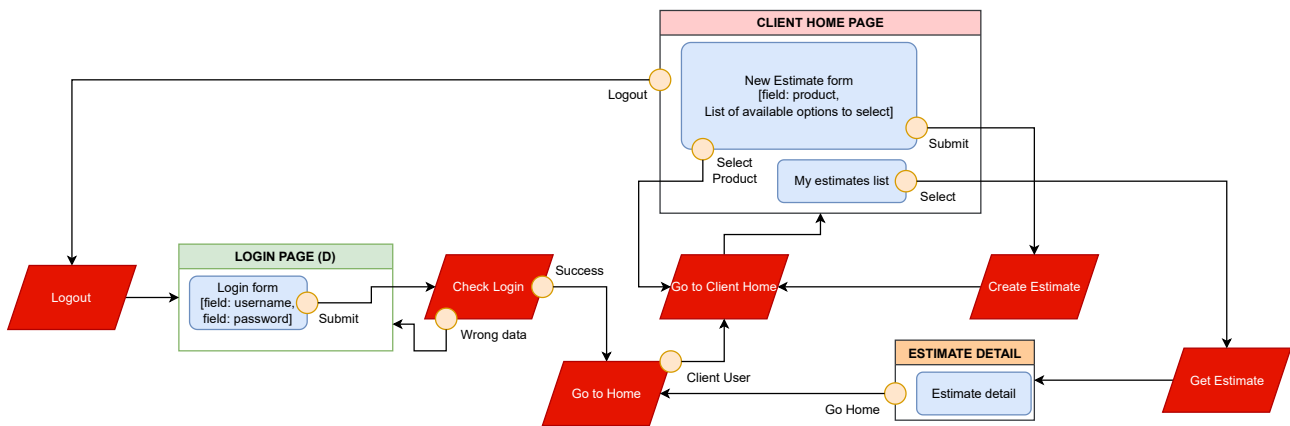


## Zone autorizzate per tipo di utente

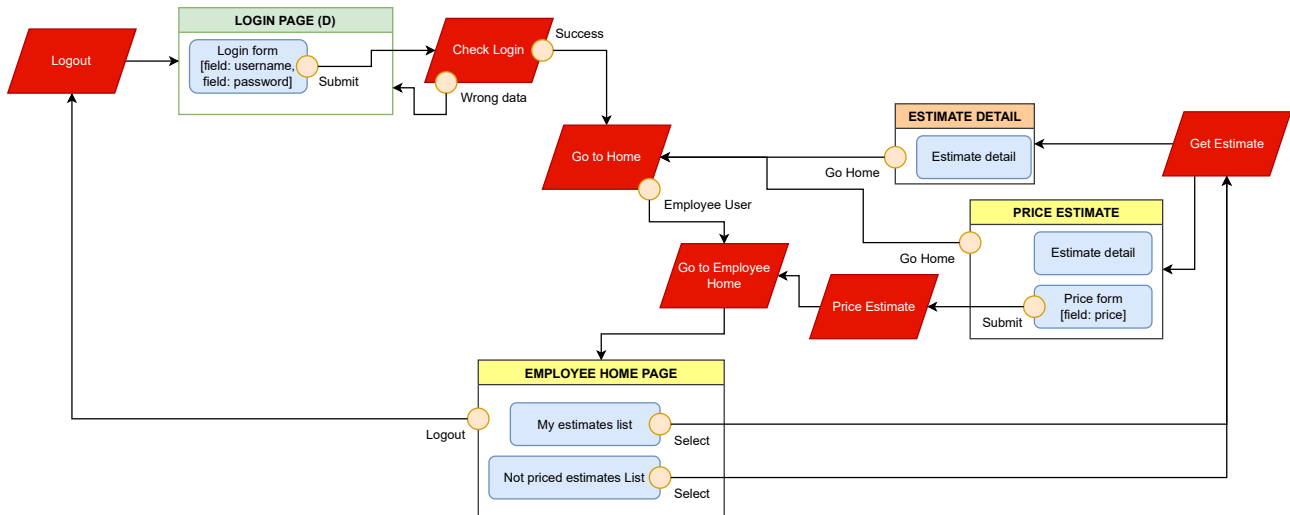
### Accesso Pubblico



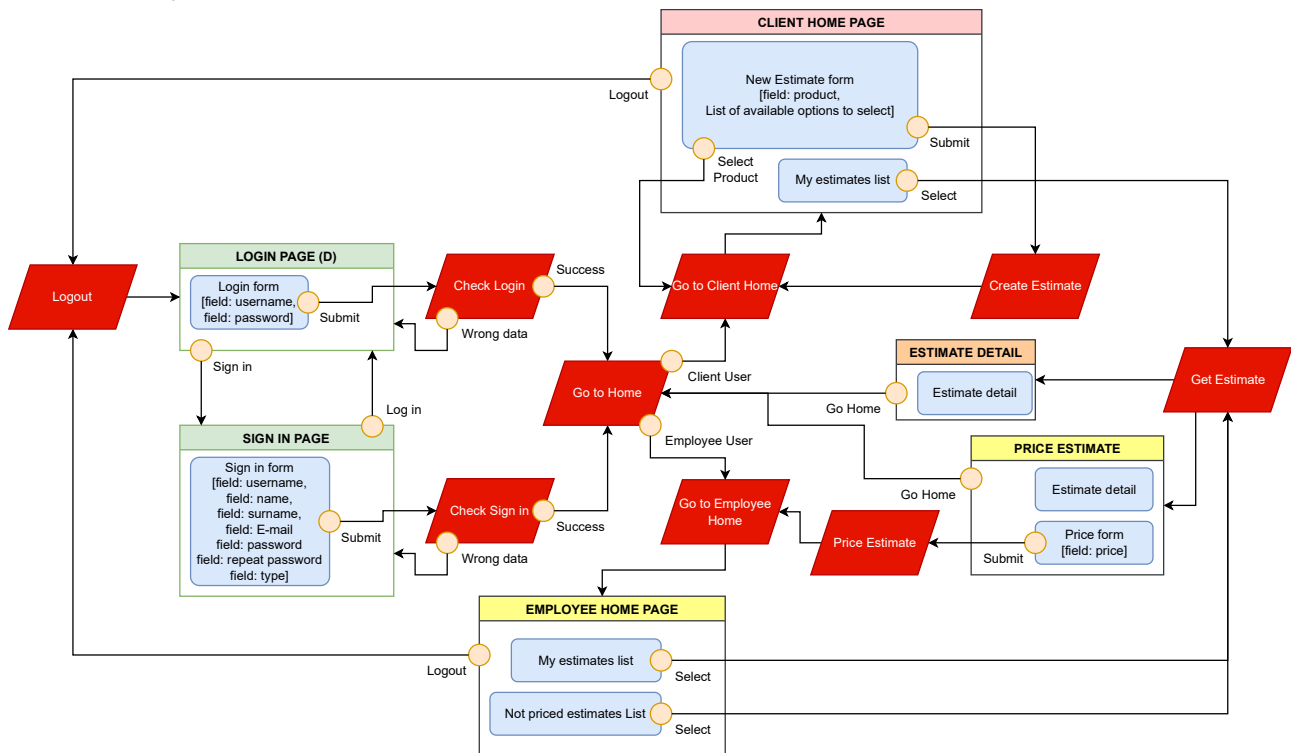
### Accesso Cliente



### Accesso Impiegato



## Lista componenti



## DAOs (Data Access Objects)

### UserDAO

```

checkCredentials(username, password)
registerCredentials(username, e-mail, password, userType)
isUsernamePresent(username)
getById(userId)
    
```

### OptionDAO

```

optionsFromProduct(productCode)
optionsFromEstimate(estimateCode)
getFromCode(optionCode)
    
```

### ProductDAO

```

getAll()
getByCode(productId)
    
```

### EstimateDAO

```

getByUser(user)
getByCode(estimateCode)
getNotPriced()
createEstimate(clientId, productCode, optionCodeList)
priceEstimate(estimate, employeeId, price)
    
```

### Bean

```

User
Estimate
Option
Product
    
```

### Controllori

```

CheckLogin
CheckSignIn
GoToHome
GetEstimate
GetAvailableOptions
CreateEstimate
Logout
    
```

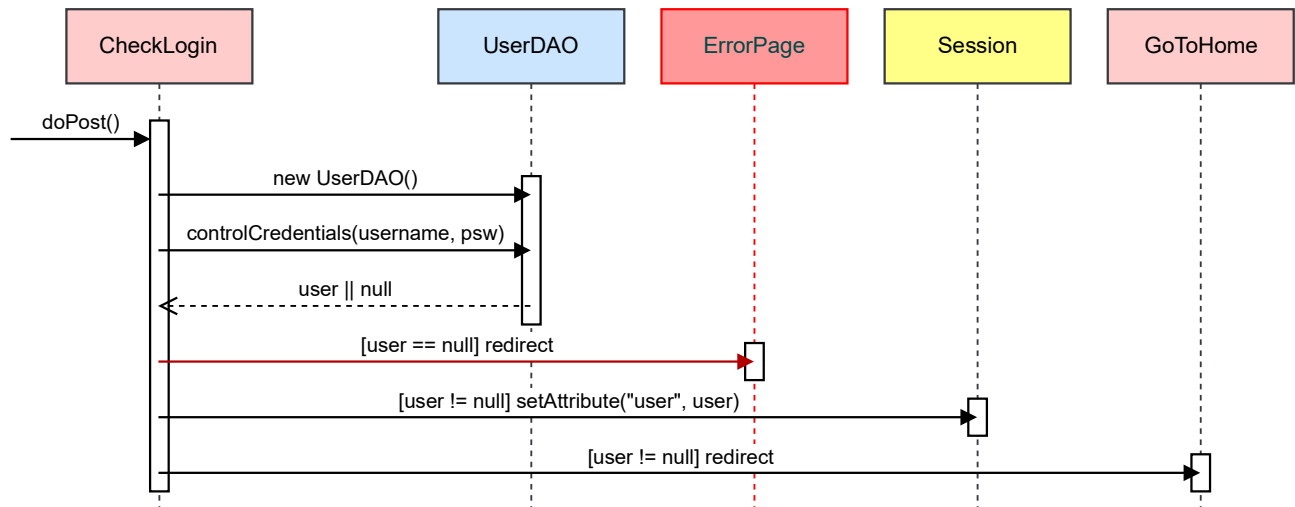
### Viste

```

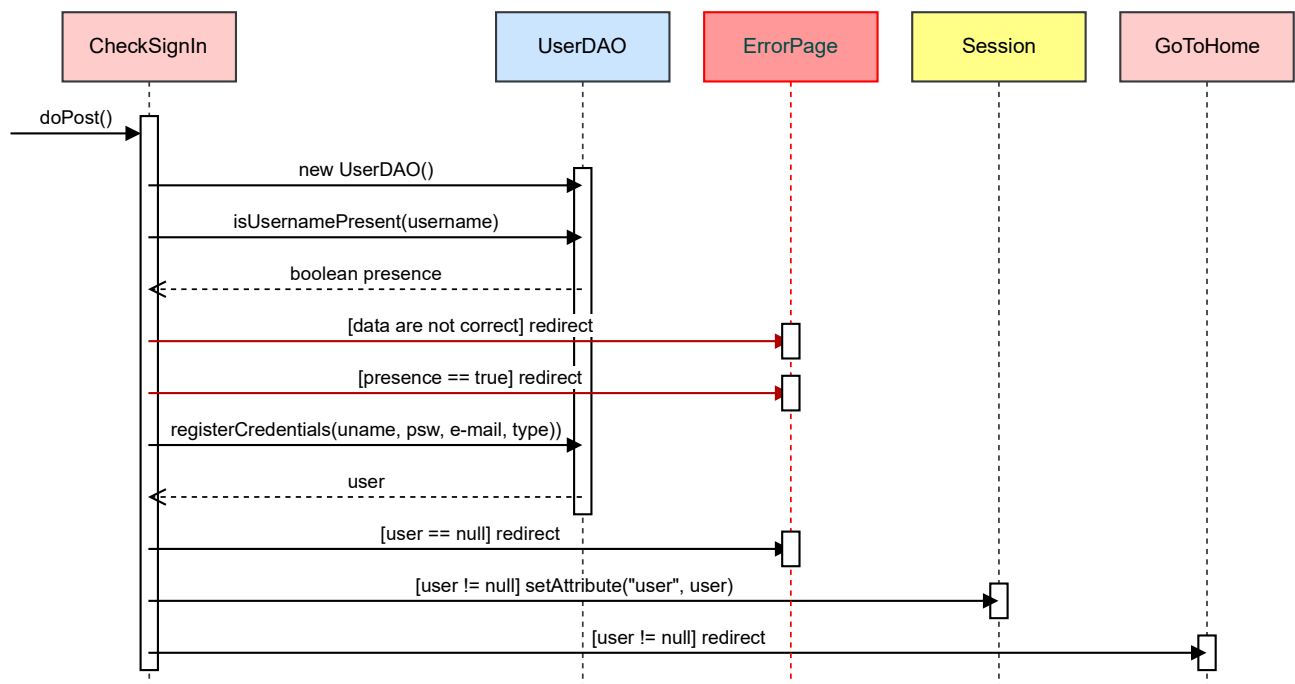
Login
SignIn
ClientHomePage
EmployeeHomePage
EstimateDetail
PriceEstimate
    
```

## Eventi

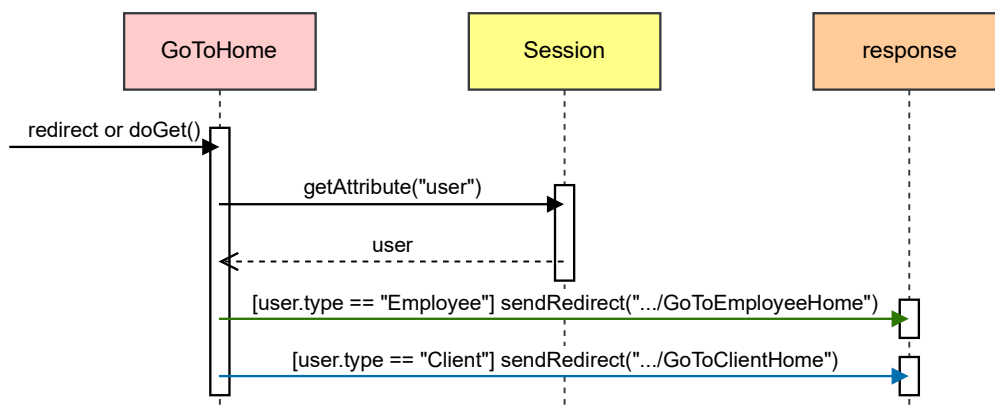
### Log in



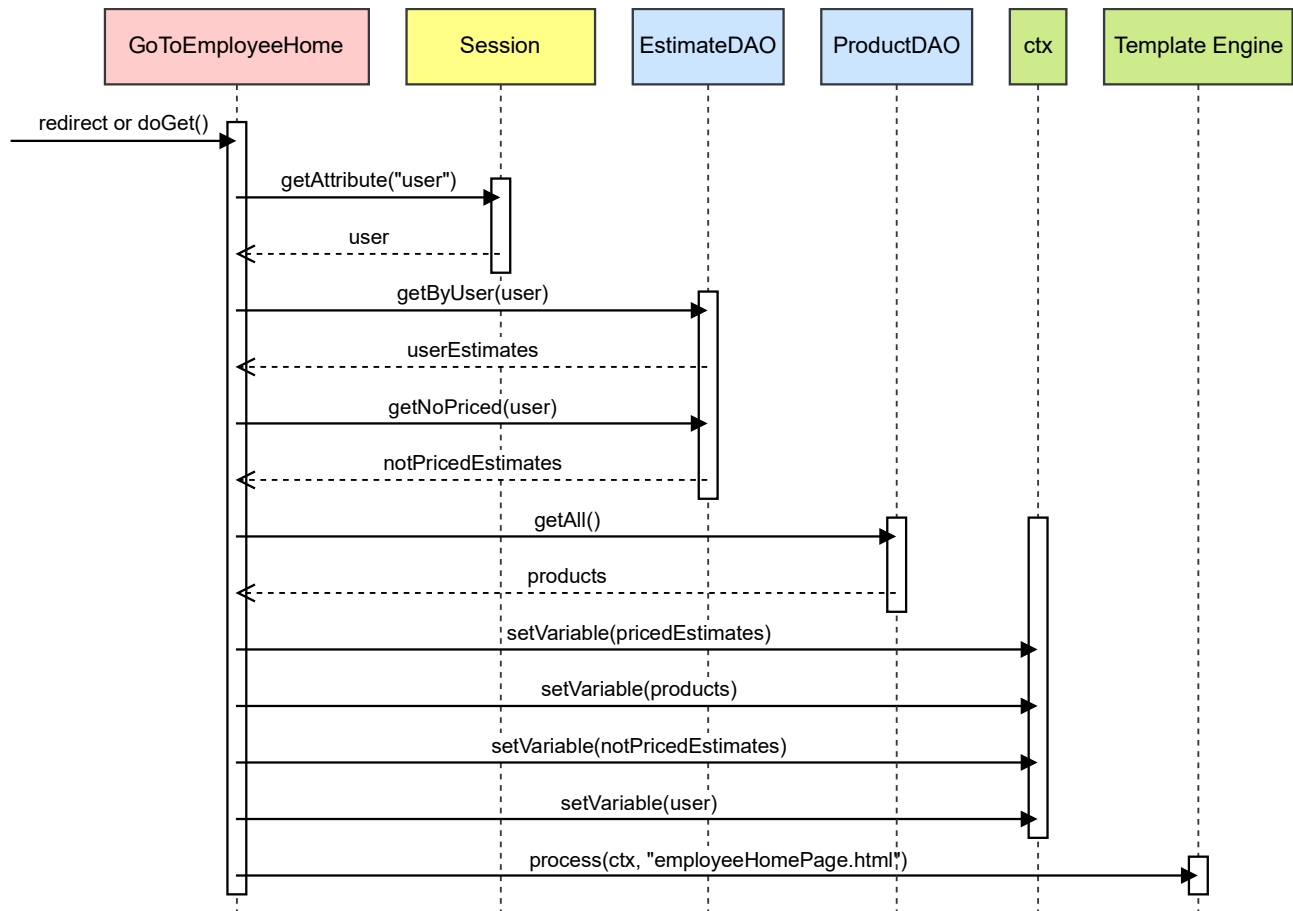
### Sign up



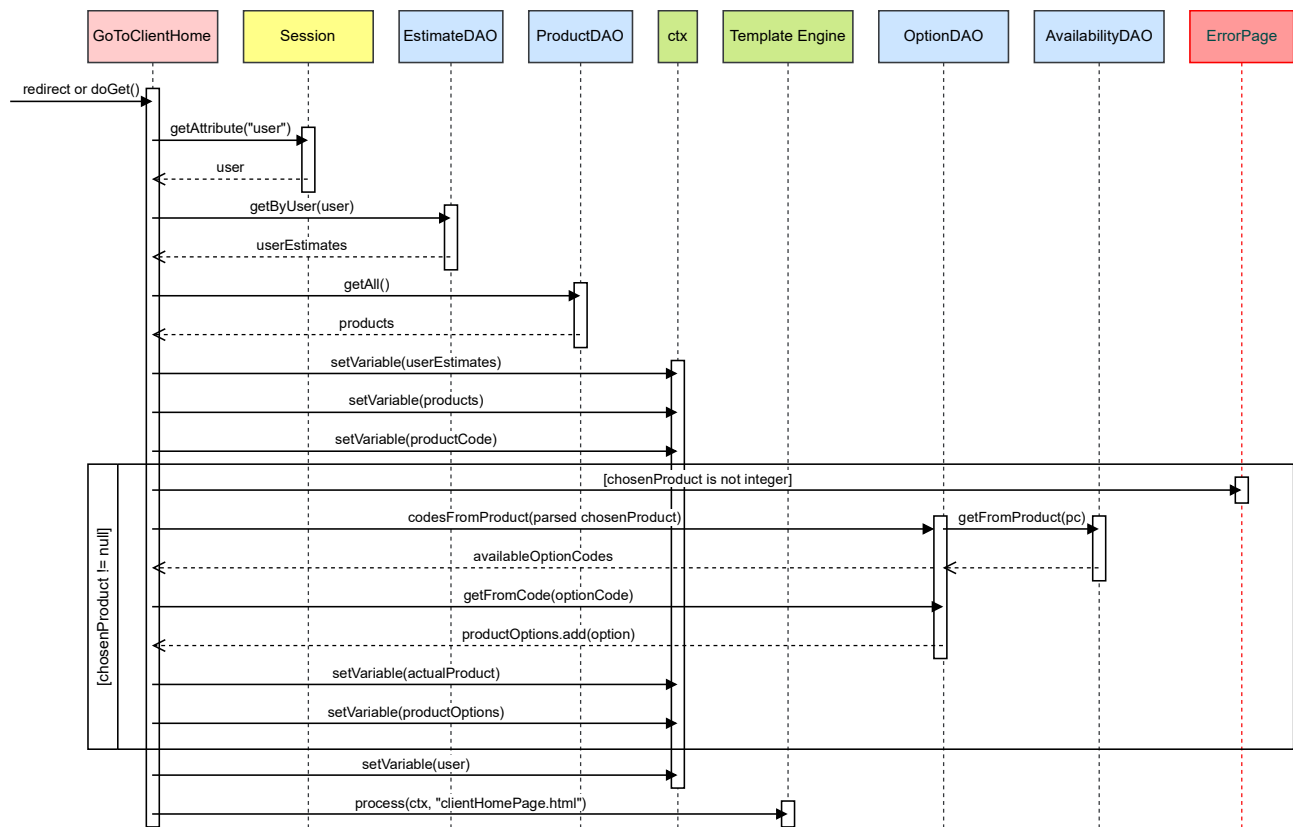
### Go To Home



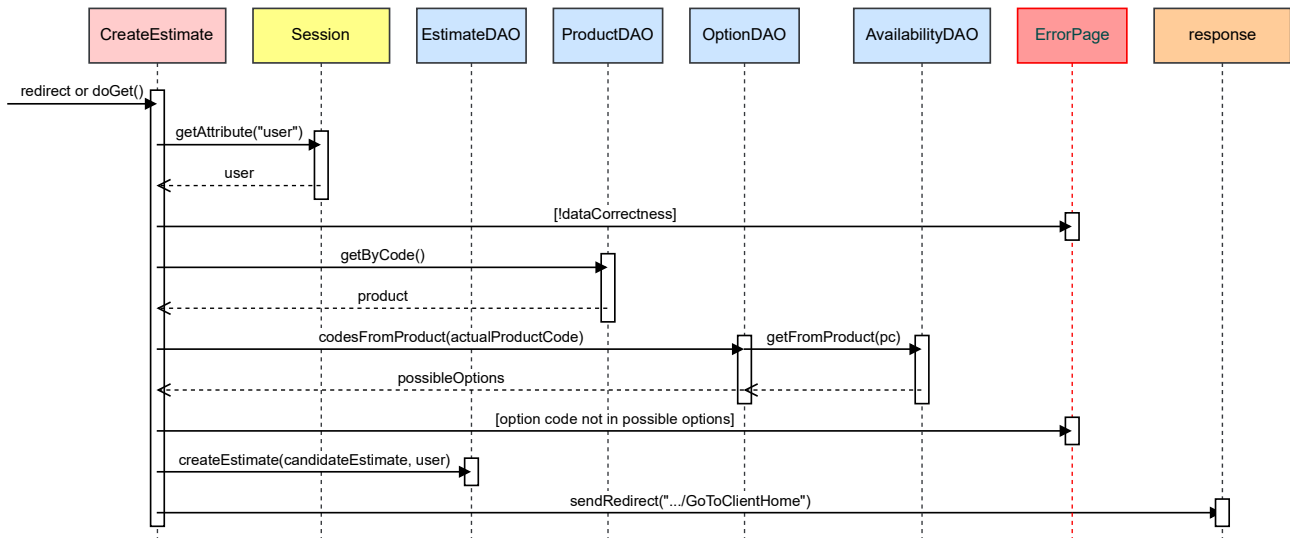
## Go To Employee Home



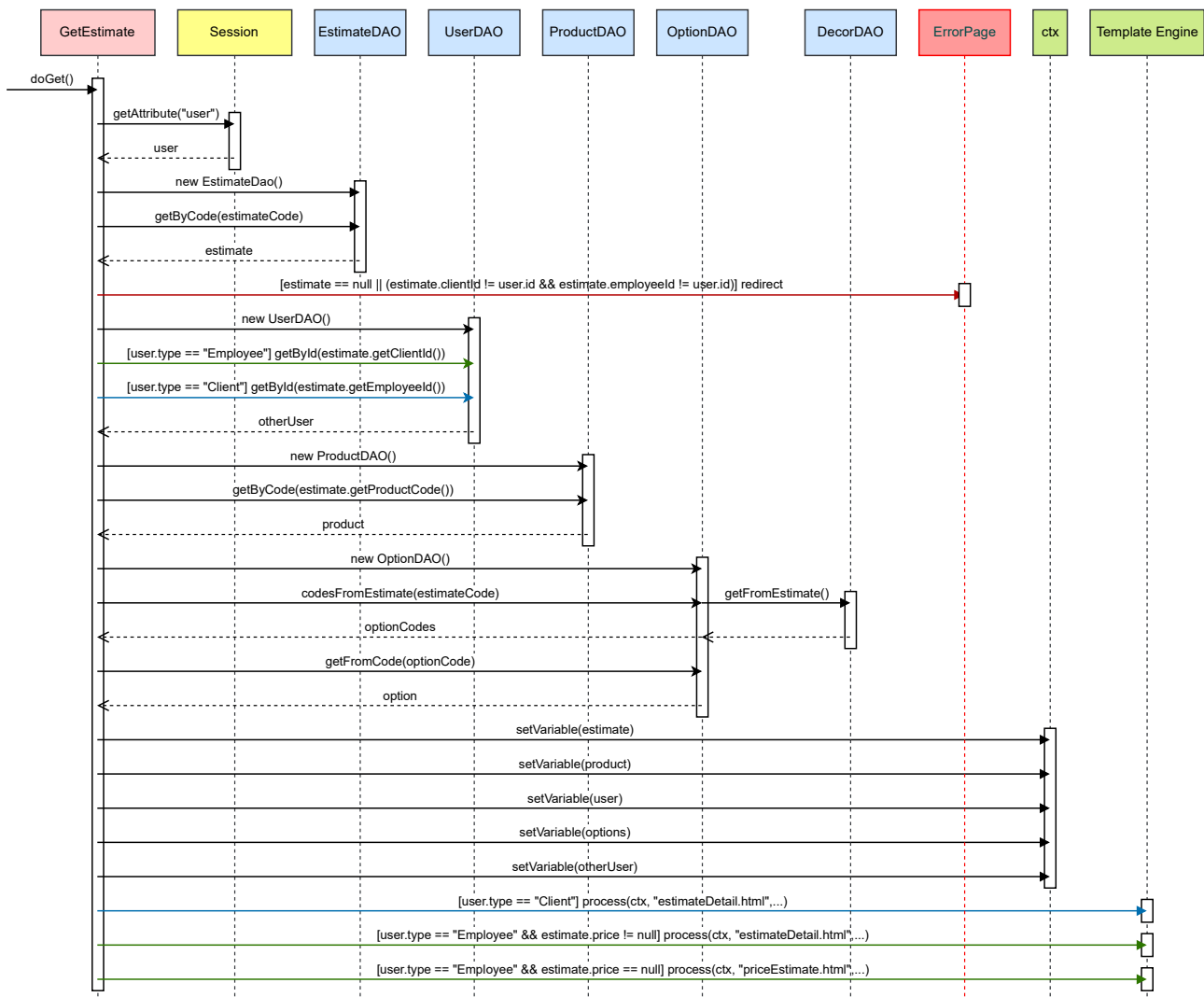
## Go To Client Home



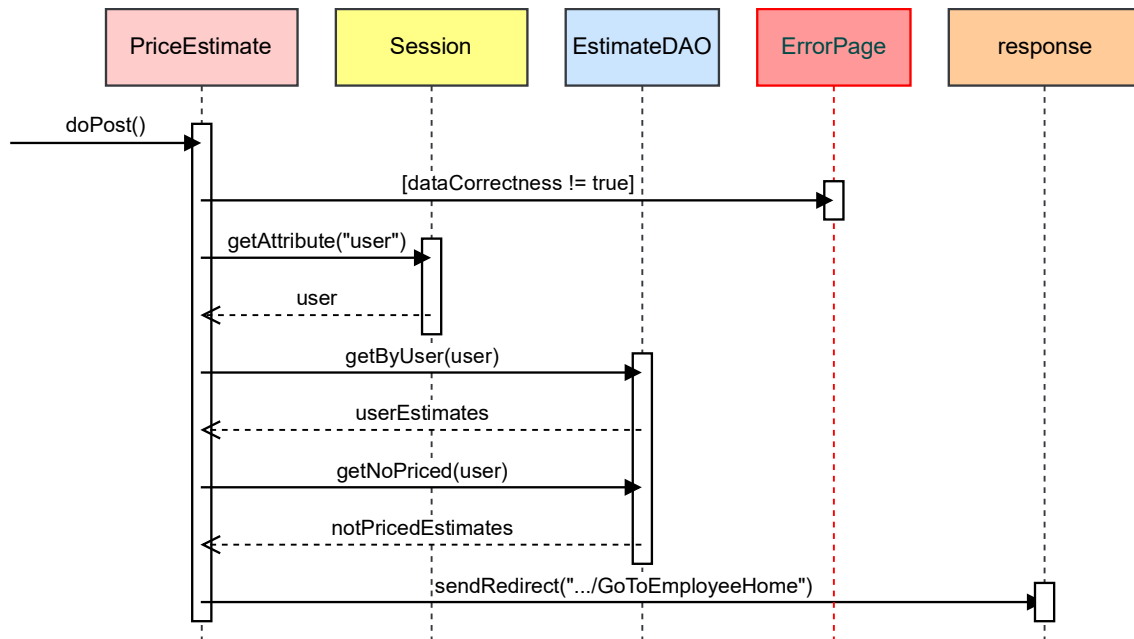
## CreateEstimate



## EstimateDetail



## PriceEstimate



## Log out

