

# Catalogazione di immagini

Tecnologie informatiche per il web

Esercizio 3 – Versione HTML

Leonardo Guerra – 10524955

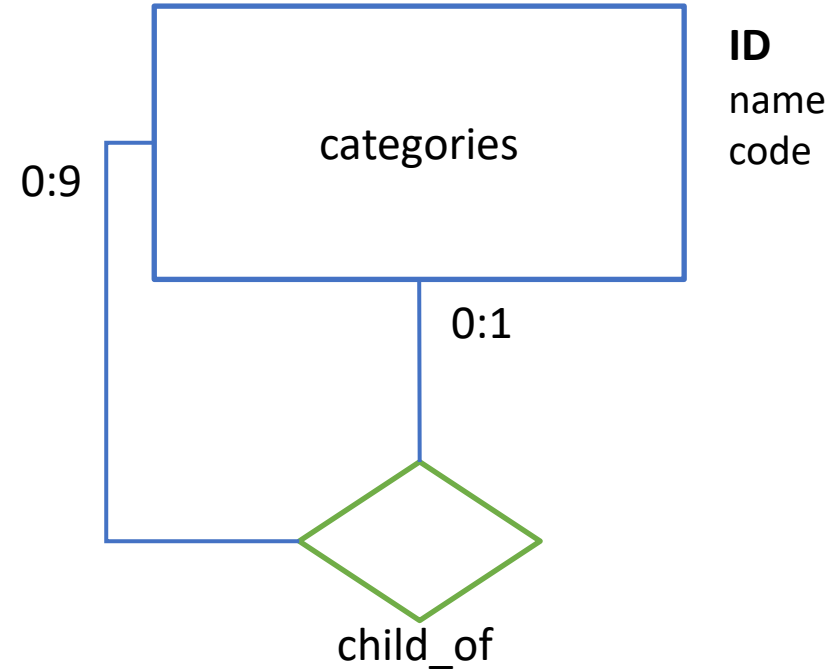
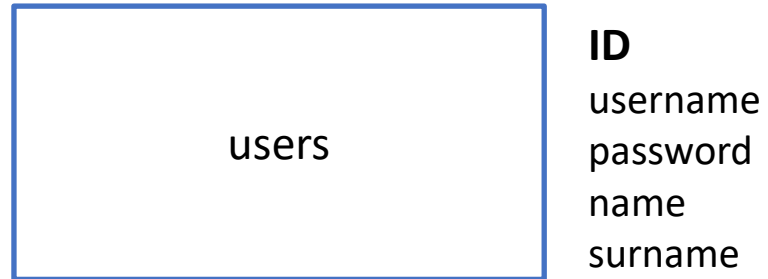
Gaia Locchi – 10750598

# Analisi dei dati

Un'applicazione permette all'**utente** (ad esempio il curatore di un catalogo online di immagini) di gestire una tassonomia di classificazione utile per etichettare immagini allo scopo di consentire la ricerca in base alla **categoria**. Dopo il login, l'utente accede a una pagina HOME in cui compare un albero gerarchico di categorie. Le categorie non dipendono dall'utente e sono in comune tra tutti gli utenti. Le categorie hanno **nomi** distinti. L'utente può inserire una nuova categoria nell'albero. Per fare ciò usa una form nella pagina HOME in cui specifica il nome della nuova categoria e sceglie la categoria padre. L'invio della nuova categoria comporta l'aggiornamento dell'albero: la nuova categoria **è appesa alla categoria padre** come ultimo sottoelemento. Alla nuova categoria viene assegnato un **codice numerico** che ne riflette la posizione. Per semplicità si ipotizzi che per ogni categoria il numero massimo di sottocategorie sia 9, numerate da 1 a 9. Dopo la creazione di una categoria, la pagina HOME mostra l'albero aggiornato. L'utente può spostare di posizione una categoria: per fare ciò clicca sul link "sposta" associato alla categoria da spostare. A seguito di tale azione l'applicazione mostra, sempre nella HOME page, l'albero con evidenziato il sotto albero attestato sulla categoria da spostare: tutte le altre categorie hanno un link "sposta qui". La selezione di un link "sposta qui" comporta l'inserimento della categoria da spostare come ultimo figlio della categoria destinazione. Le modifiche effettuate da un utente e salvate nella base di dati diventano visibili agli altri utenti. Lo spostamento potrebbe creare un vuoto nella numerazione delle categorie figlie dello stesso padre. La funzione di ricalcolo della numerazione delle categorie figlie dello stesso padre di quella spostata non è richiesta ma è lasciata come opzionale.

**Entities, attributes, relationships**

# Database design



# Local database schema

```
CREATE TABLE `users`  
(  
  `id` int NOT NULL AUTO_INCREMENT,  
  `username` varchar(45) NOT NULL  
  UNIQUE,  
  `password` varchar(45) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `surname` varchar(45) NOT NULL,  
  PRIMARY KEY (`id`)  
)
```

```
CREATE TABLE `categories`  
(  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) NOT NULL,  
  `code` varchar(255) NOT NULL,  
  `father` int,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`father`) REFERENCES  
  `categories` (`id`) ON DELETE CASCADE  
  ON UPDATE CASCADE  
)
```

# Analisi dei requisiti

Un'applicazione permette all'utente (ad esempio il curatore di un catalogo online di immagini) di gestire una tassonomia di classificazione utile per etichettare immagini allo scopo di consentire la ricerca in base alla categoria. Dopo il **login**, l'utente **accede a** una pagina **HOME** in cui compare **un albero gerarchico di categorie**. Le categorie non dipendono dall'utente e sono in comune tra tutti gli utenti. Le categorie hanno nomi distinti. L'utente può **inserire una nuova categoria** nell'albero. Per fare ciò usa una **form** nella pagina HOME in cui specifica il nome della nuova categoria e sceglie la categoria padre. L'**invio della nuova categoria** comporta l'**aggiornamento dell'albero**: la nuova categoria è appesa alla categoria padre come ultimo sottoelemento. Alla nuova categoria viene assegnato un codice numerico che ne riflette la posizione. Per semplicità si ipotizzi che per ogni categoria il numero massimo di sottocategorie sia 9, numerate da 1 a 9. Dopo la creazione di una categoria, la pagina HOME mostra l'albero aggiornato. L'utente può **spostare di posizione una categoria**: per fare ciò **clicca sul link "sposta"** associato alla categoria da spostare. A seguito di tale azione l'applicazione mostra, sempre nella HOME page, l'albero con evidenziato il sotto albero attestato sulla categoria da spostare: tutte le altre categorie hanno un **link "sposta qui"**. La **selezione di un link "sposta qui"** comporta l'inserimento della categoria da spostare come ultimo figlio della categoria destinazione. Le modifiche effettuate da un utente e salvate nella base di dati diventano visibili agli altri utenti. Lo spostamento potrebbe creare un vuoto nella numerazione delle categorie figlie dello stesso padre. La funzione di ricalcolo della numerazione delle categorie figlie dello stesso padre di quella spostata non è richiesta ma è lasciata come opzionale.

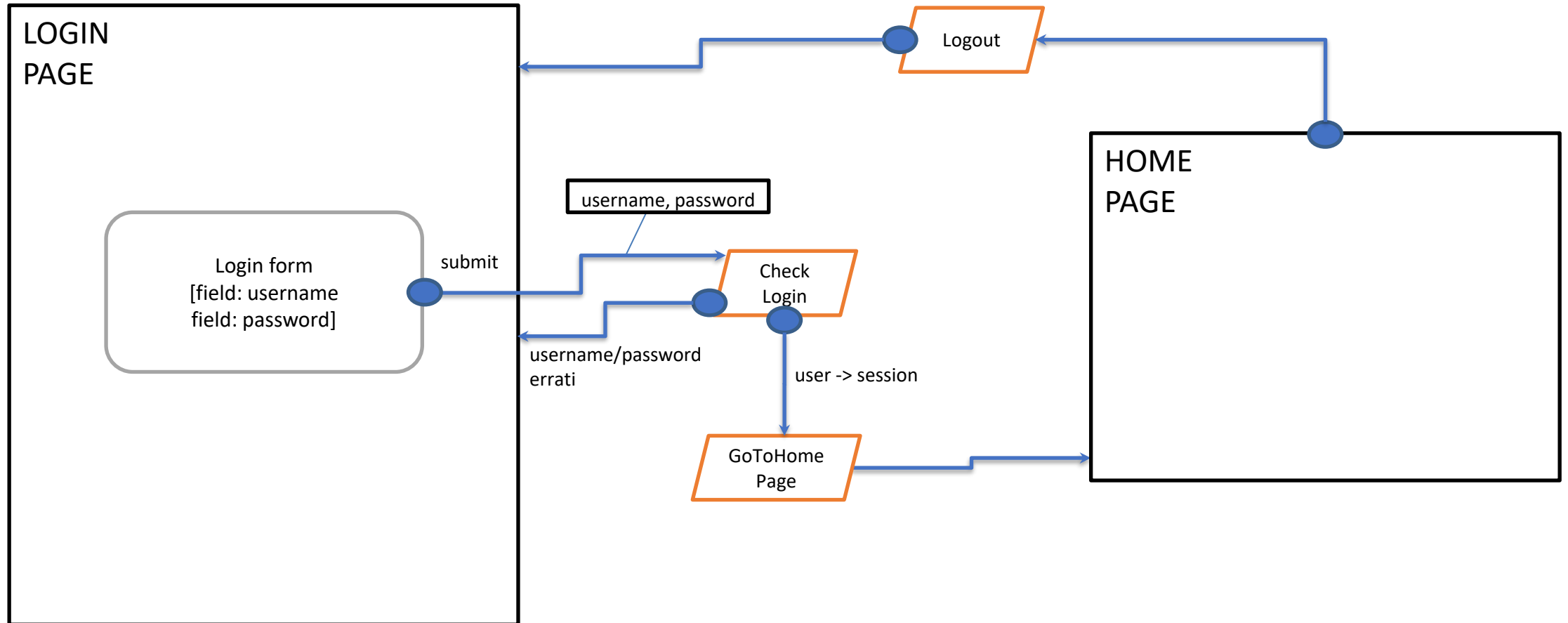
**Pages (views), view components, events, actions**

# Completamento delle specifiche

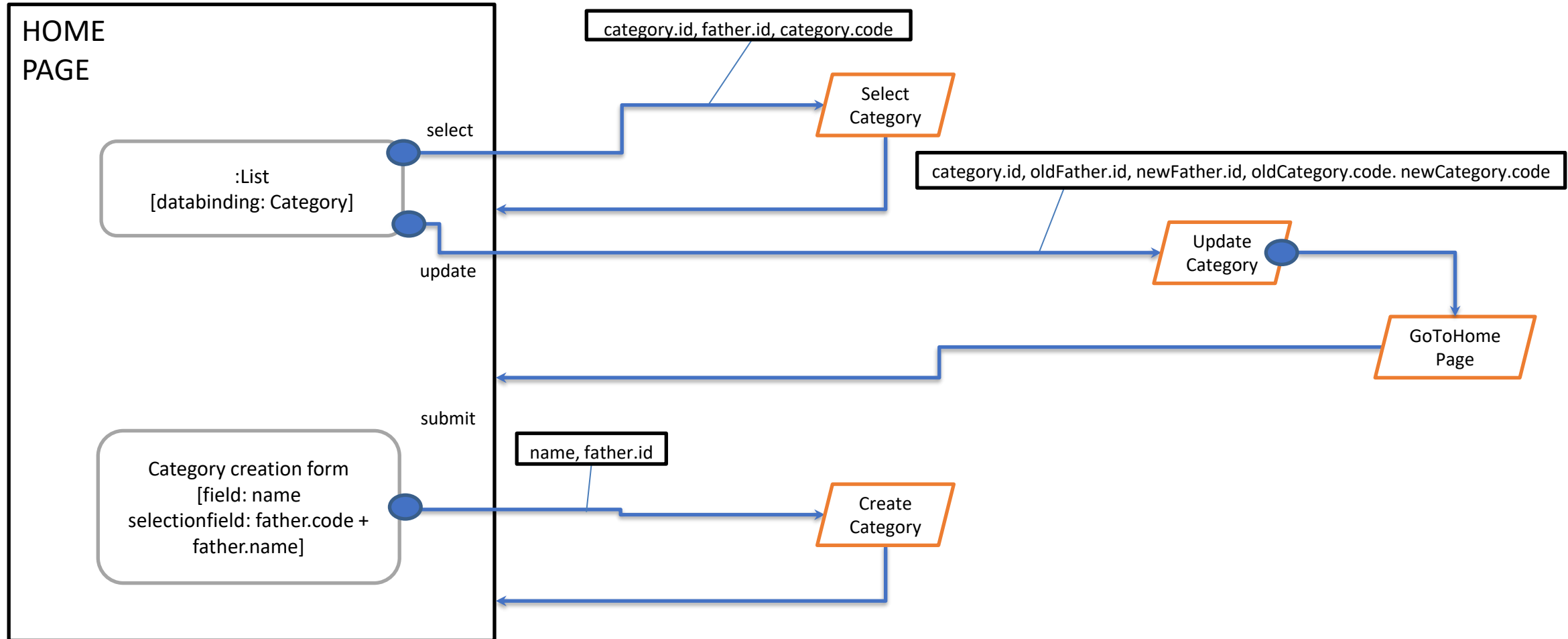
- La **pagina di default** contiene la **form di login**
- Nella form di login sia username che password sono obbligatori. Nella creazione di una nuova categoría, solo il nome della nuova categoría è obbligatorio.
- Le categorie “radice” (il cui codice è composto da una sola cifra) non hanno una categoria padre.
- Una categoría non può essere spostata come sotto-categoría dello stesso padre di partenza (quindi, dopo aver premuto “Sposta” in corrispondenza di una categoría che non sia radice, il padre di quella categoría non presenterà lo “Sposta qui”), nè di uno dei suoi figli (anche indiretti).
- **Funzione di ricalcolo:** se la categoría spostata è un “figlio intermedio” di una categoria (quindi ha “fratelli” con codice maggiore del suo), dopo il suo spostamento, per occupare il vuoto lasciato, il fratello con codice maggiore viene spostato (con tutto il suo albero) al suo posto, con conseguente aggiornamento di tutti i codici interessati.

**Pages (views), view components, events, actions**

# Application design



# Application design

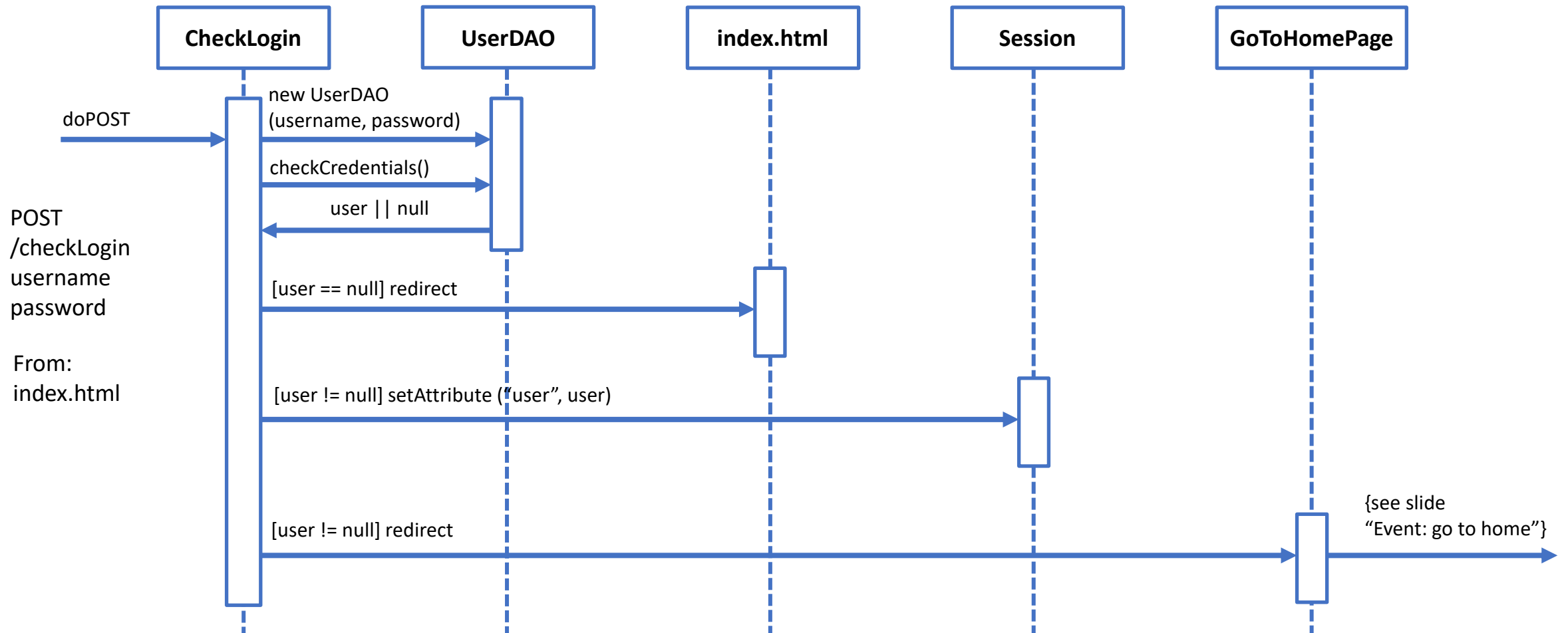




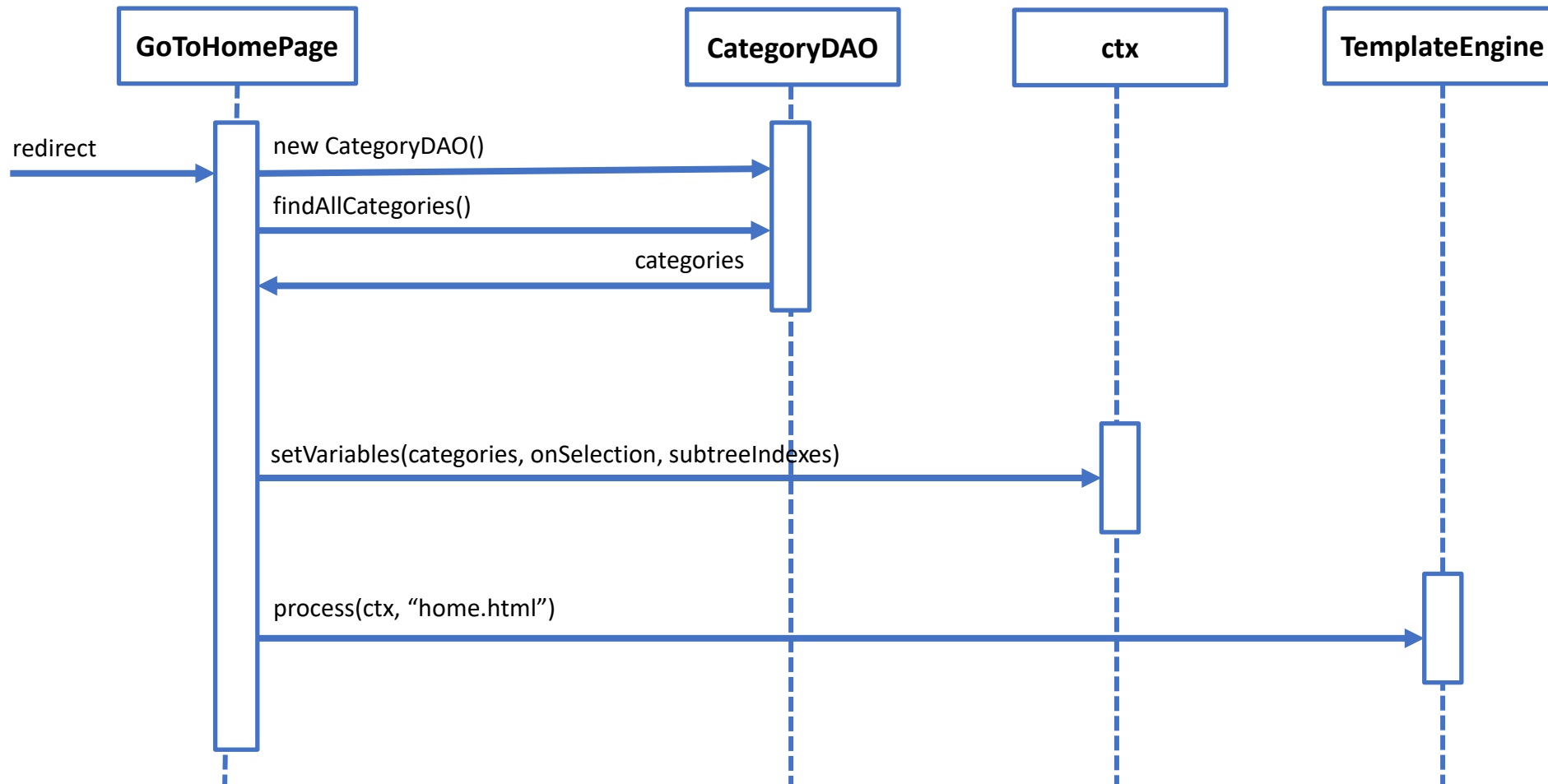
# Components

- Model objects (Beans)
  - User
  - Category
- Data Access Objects (Classes)
  - UserDao
    - checkCredentials(username, password)
  - CategoryDAO
    - createCategory(name, code, fatherId)
    - findLastChildCode(categoryId)
    - findCategoriesByFather(fatherId)
    - findAllCategories()
    - getNumOfRoots()
    - getCategorySubtree(categoryId)
    - findCategoryCode(categoryId)
    - findMaxRootCode()
    - updateCategory(categoryId, oldFatherId, newFatherId, oldCategoryCode, newCategoryCode)
    - existsCategory(name)
    - legitCategory(id, code, father)
    - legitDestinationCategory(id, oldFatherId, newFatherId)
- Controllers (Servlets)
  - CheckLogin
  - CreateCategory
  - GoToHomePage
  - SelectCategory
  - UpdateCategory
  - Logout
- Views (Templates)
  - index
  - home

# Event: login

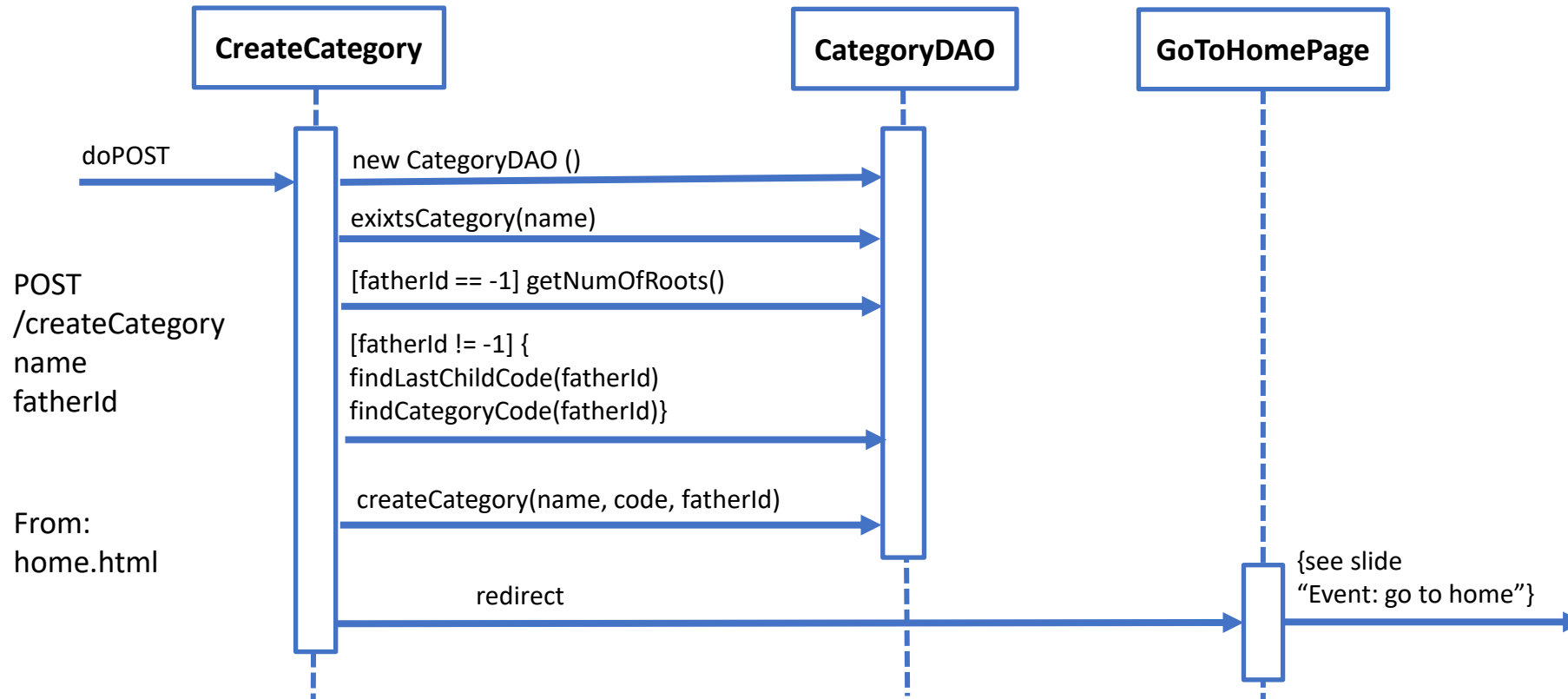


# Event: go to home



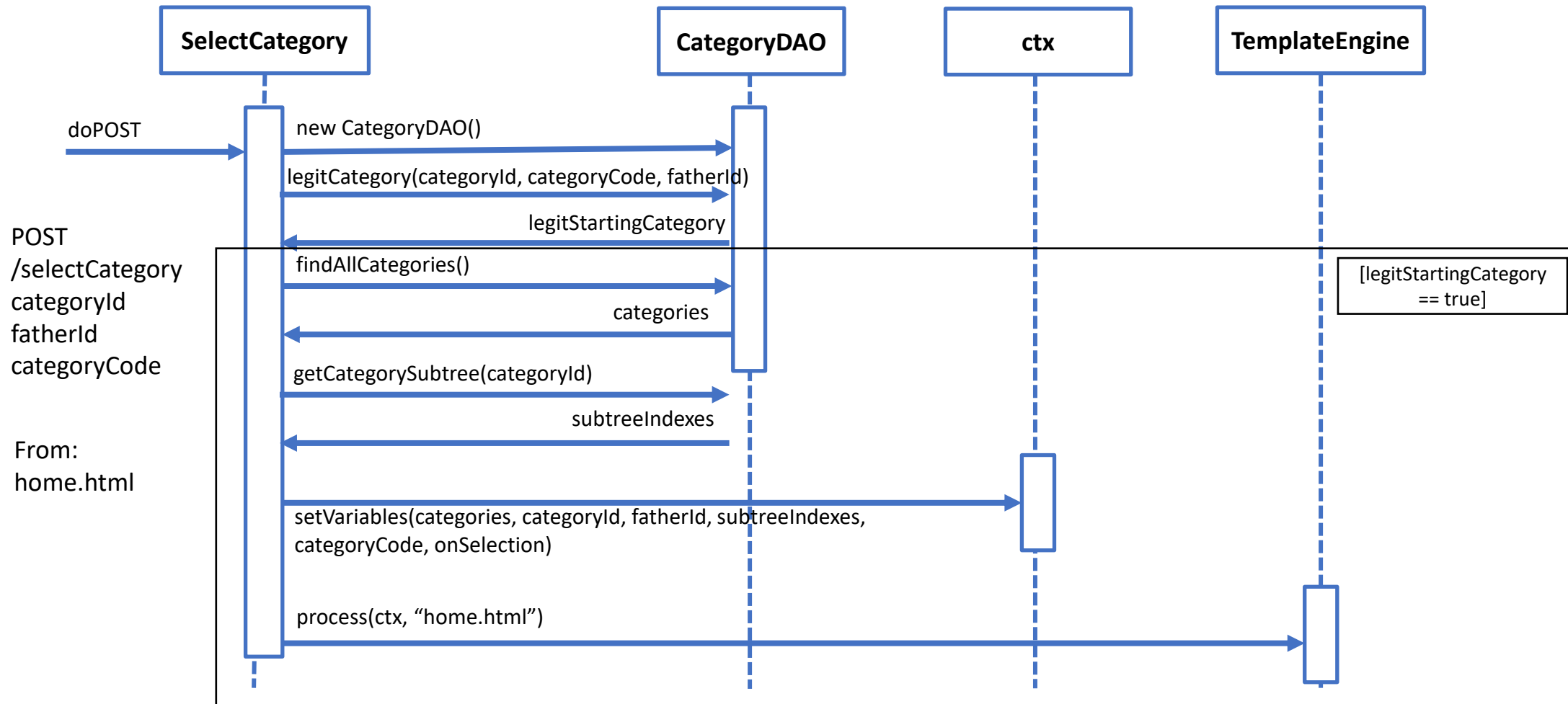
# Event: create category

→→→ I valori di ritorno delle funzioni invocate sul DAO non sono riportati per questioni di leggibilità



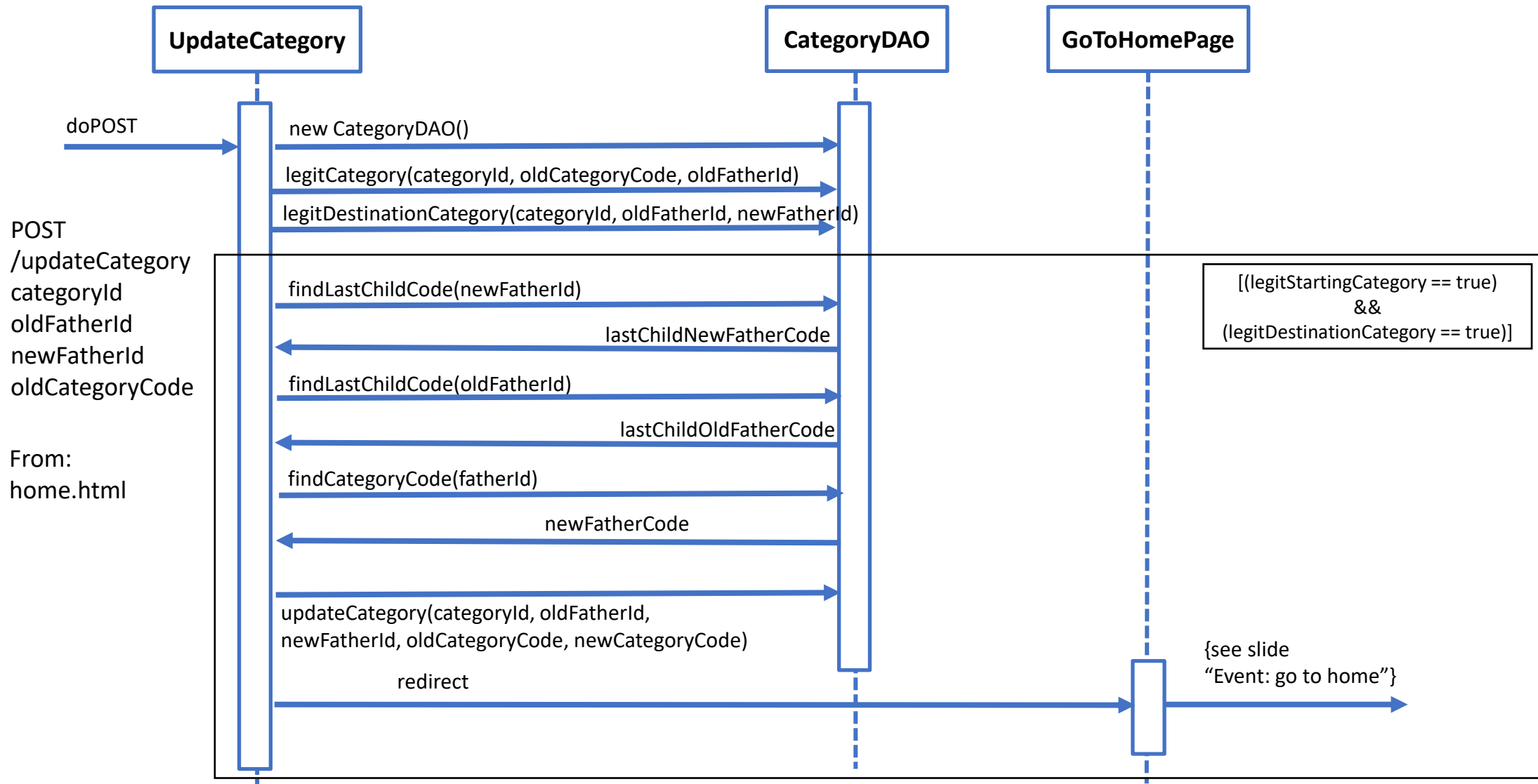
# Event: select category

→→→ Se legitStartingCategory  
è false, viene segnalato l'errore  
nella risposta HTTP

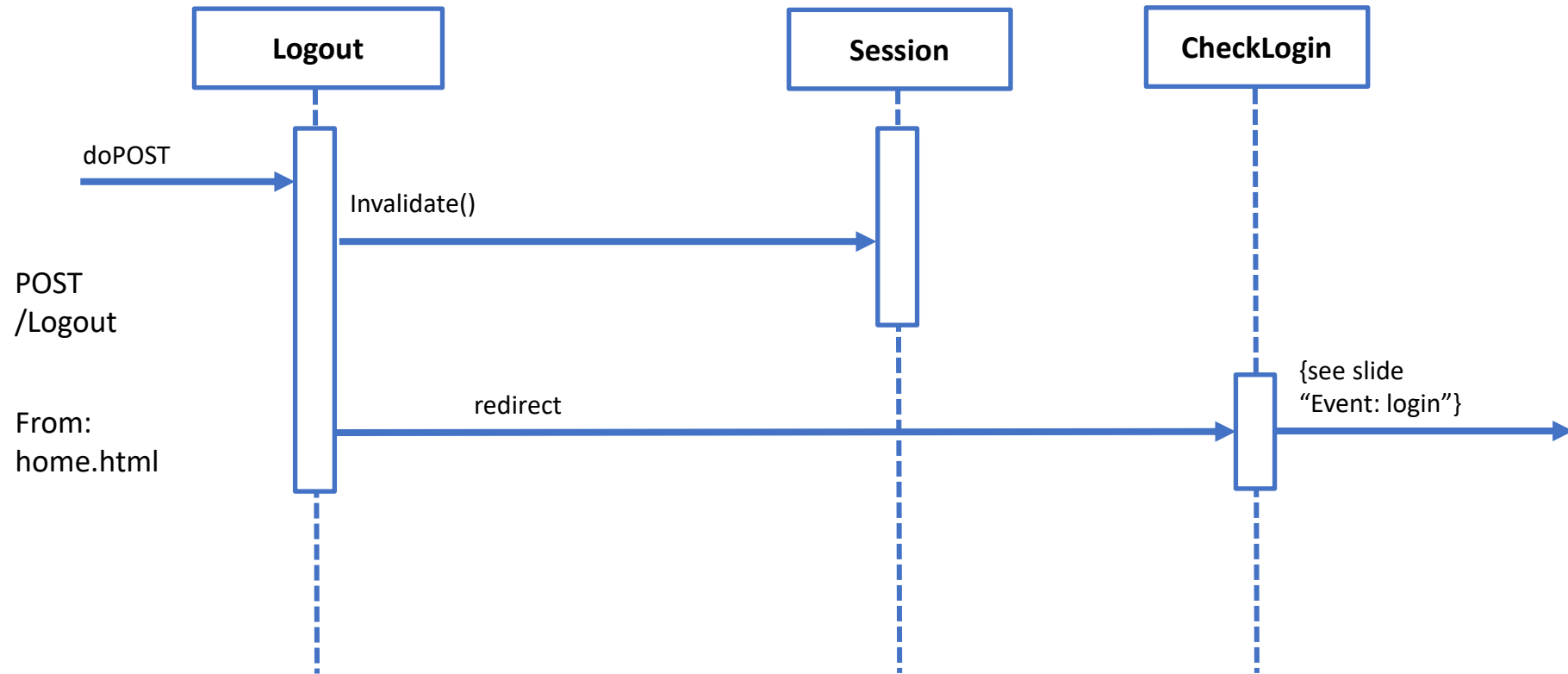


# Event: update category

→→→ Se almeno uno tra legitStartingCategory e legitDestinationCategory sono false, viene segnalato l'errore nella risposta HTTP



# Event: logout



# Packages Utils e Filters

- Package Utils

- ConnectionHandler

- Si occupa di istanziare e chiudere la Connection

- SharedPropertyMessageResolver

- Classe utilizzata nel metodo init() di ogni servlet per inizializzare o richiamare le proprietà di internazionalizzazione della rispettiva servlet

- Package Filters

- LoginChecker

- Classe utilizzata ogni volta in cui un utente realizza un'azione nella pagina web (GET/POST), che si occupa di controllare la validità della sessione



# Internazionalizzazione

- Il progetto è stato sviluppato sfruttando il supporto di Thymeleaf per l'internazionalizzazione.
- Le lingue supportate sono:
  - italiano
  - inglese
  - spagnolo
  - francese