

# Tecnologie Informatiche per il Web

AA. 2020/21

Docente: Piero Fraternali

Studente: Zheng Maria Yu  
(Gruppo 88)

# Catalogazione di immagini – Versione HTML pura

Un'applicazione permette all'utente 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 ogni categoria il numero massimo di sottocategorie è 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.

# Analisi dei dati

Un'applicazione permette all'utente 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 ogni categoria il numero massimo di sottocategorie è 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.

(**Entità**, **attributi**, **relazioni**)

# Esempio dell'interfaccia

**Welcome, test A!**

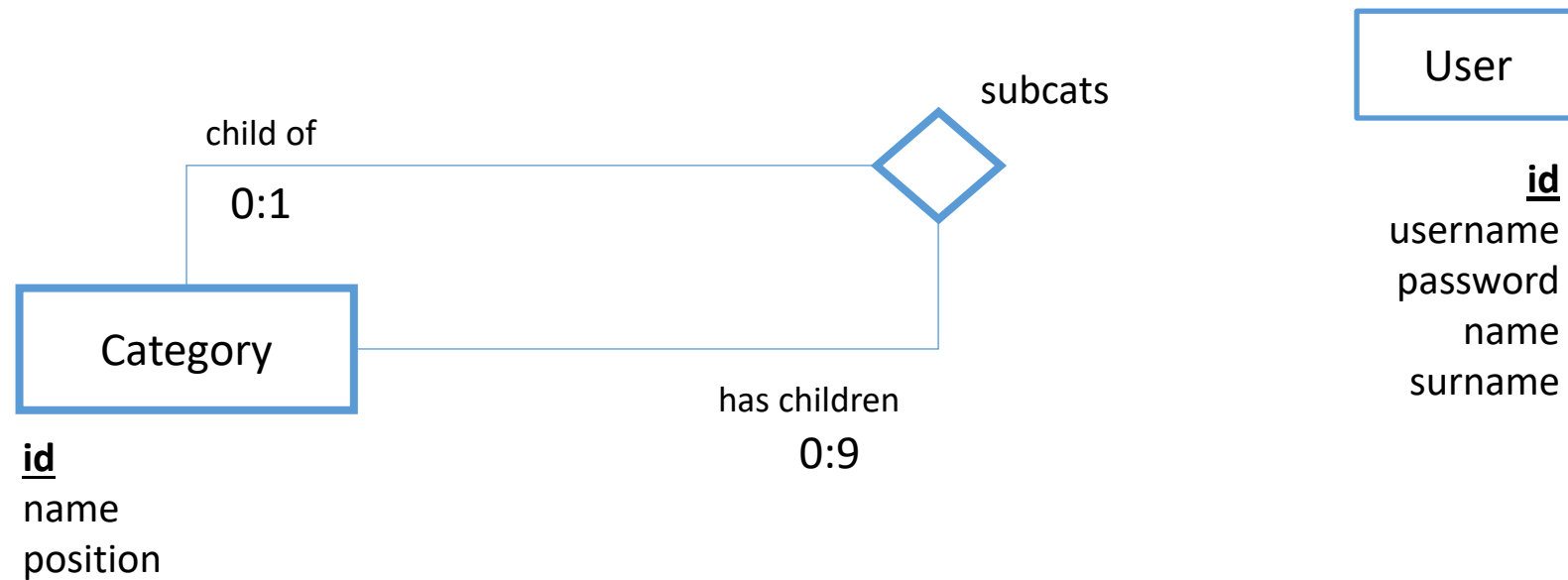
- └ 1 Mitologia classica e storia antica [Sposta](#)
  - └ 11 Divinità della mitologia classica [Sposta](#)
    - └ 111 Divinità del cielo [Sposta](#)
      - └ 1111 Giove [Sposta](#)
        - └ 11111 Attributi di Giove [Sposta](#)
      - └ 1112 Giunone [Sposta](#)
    - └ 112 Divinità degli inferi [Sposta](#)
      - └ 1121 Plutone [Sposta](#)
      - └ 1122 Ecate [Sposta](#)

Complete the form to create a category

Name:

Choose a father category:

# Progettazione del database



Ogni categoria può avere al massimo 9 categorie figlie, ed è figlio di una categoria (tranne root che non ha categoria padre).

# Schema database locale

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

# Schema database locale

```
CREATE TABLE `category` (  
    `id` int NOT NULL AUTO_INCREMENT,  
    `name` varchar(45) NOT NULL,  
    `position` varchar(45) NOT NULL,  
    PRIMARY KEY (`id`),  
    UNIQUE KEY `name_UNIQUE` (`name`)  
)
```

# Schema database locale

```
CREATE TABLE `subcats` (  
    `father` int NOT NULL,  
    `child` int NOT NULL,  
    PRIMARY KEY (`father`,`child`),  
    KEY `childtcategory_idx` (`child`),  
    CONSTRAINT `childtcategory` FOREIGN KEY (`child`) REFERENCES  
    `category` (`id`),  
    CONSTRAINT `fathertcategory` FOREIGN KEY (`father`) REFERENCES  
    `category` (`id`)  
)
```



# Analisi dei requisiti dell'applicazione

Un'applicazione permette all'utente 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 ogni categoria il numero massimo di sottocategorie è 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.

(**Pagine**, **componenti**, **eventi**, **azioni**)

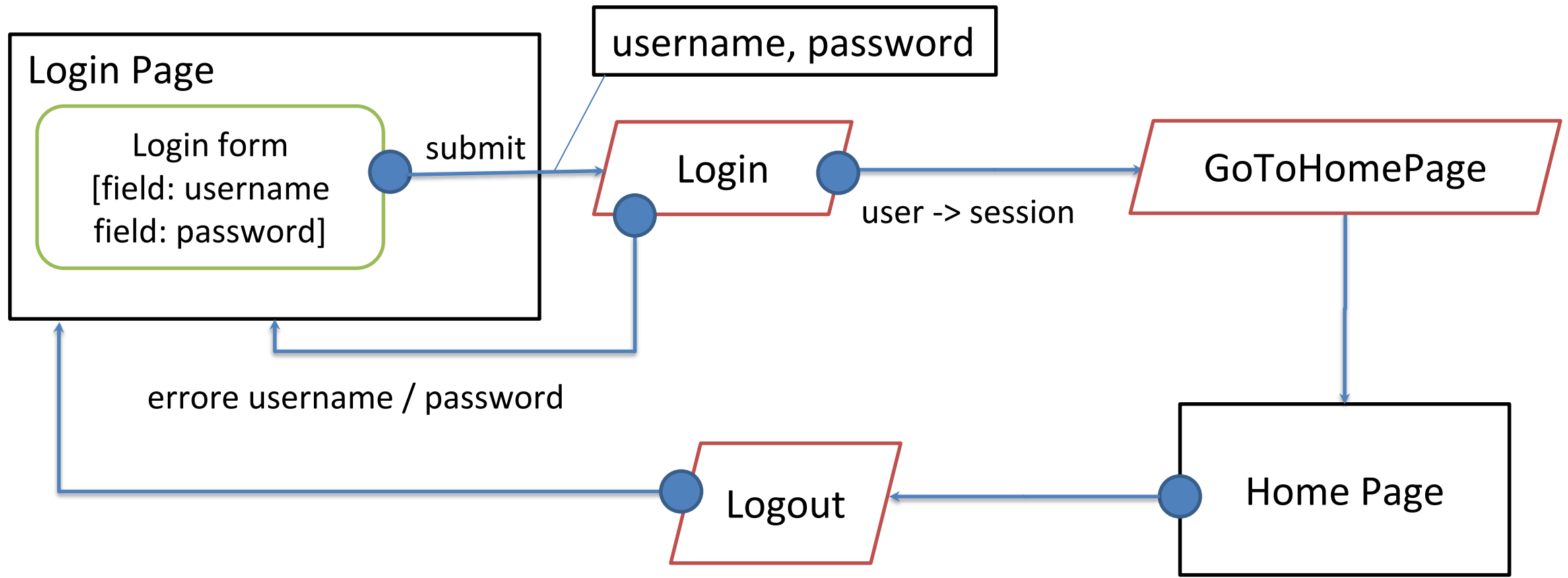
# Completamento delle specifiche

- Si effettua il login nella **pagina di default** tramite una **form**.
- Le categorie non possono avere nomi nulli.
- I dati dell'utente non possono essere nulli.
- Non è possibile spostare una categoria in una delle sue sottocategorie.
- Dalla Home è possibile effettuare **logout cliccando** su **"Logout"** , tornando quindi alla pagina di default.

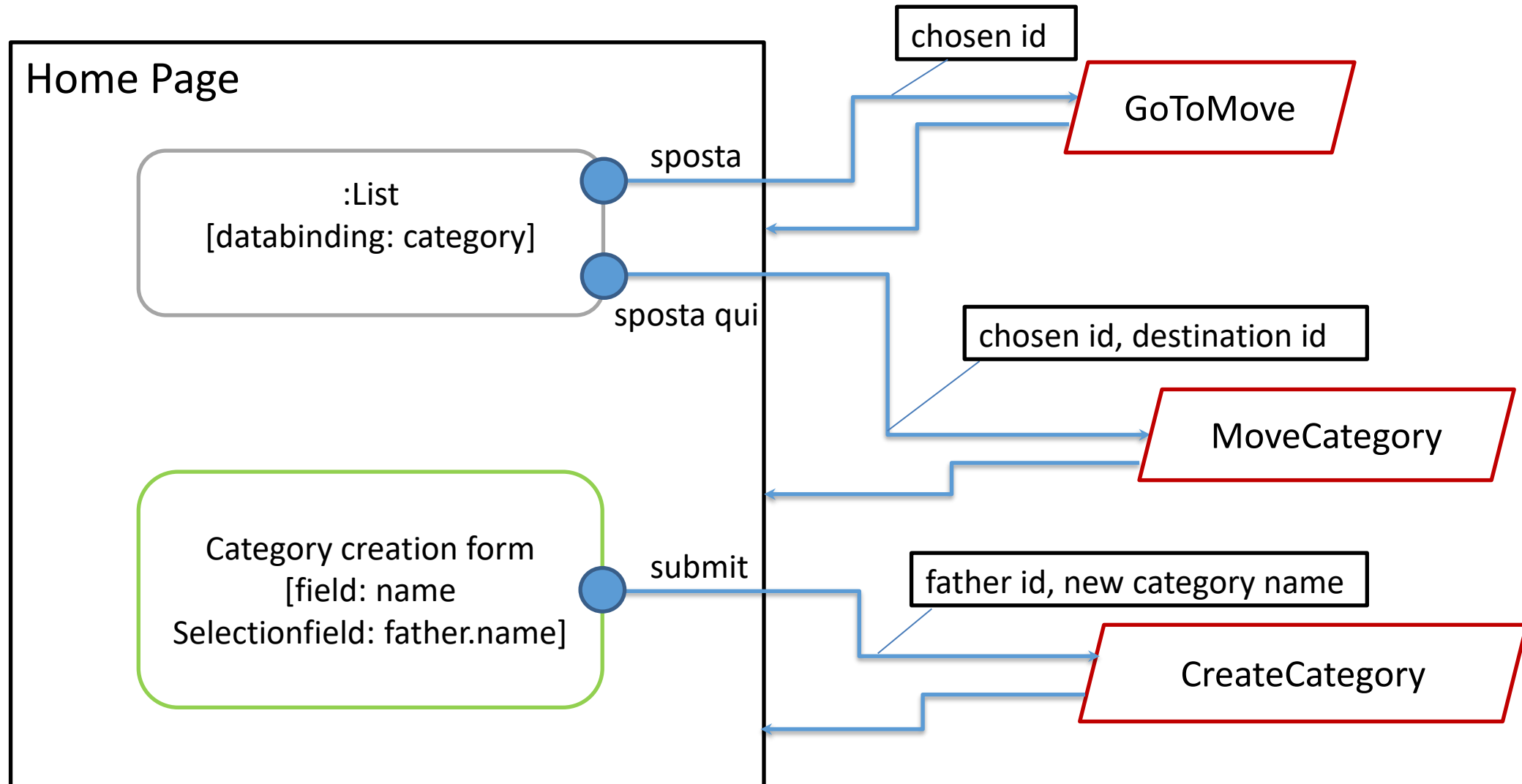
# Tabella di requisiti dell'applicazione

Viste (Pagine)	Componenti	Eventi	Azioni
Login	Form per l'accesso	Inviare credenziali	Indirizzamento al HomePage
HomePage	<ul style="list-style-type: none"><li>- Albero delle categorie</li><li>- Scelta di "Sposta" (oppure "Sposta qui")</li><li>- Form per creazione categorie</li><li>- Bottone per logout</li></ul>	<ul style="list-style-type: none"><li>- Scegliere categoria esistente da spostare</li><li>- Inviare informazioni per creare una nuova categoria</li><li>- Fare logout</li></ul>	<ul style="list-style-type: none"><li>- Spostamento categoria gestito da GoToMove / MoveCategory</li><li>- Creazione categoria gestita da CreateCategory</li><li>- Indirizzamento alla pagina di login (Logout)</li></ul>

# Progettazione dell'applicazione



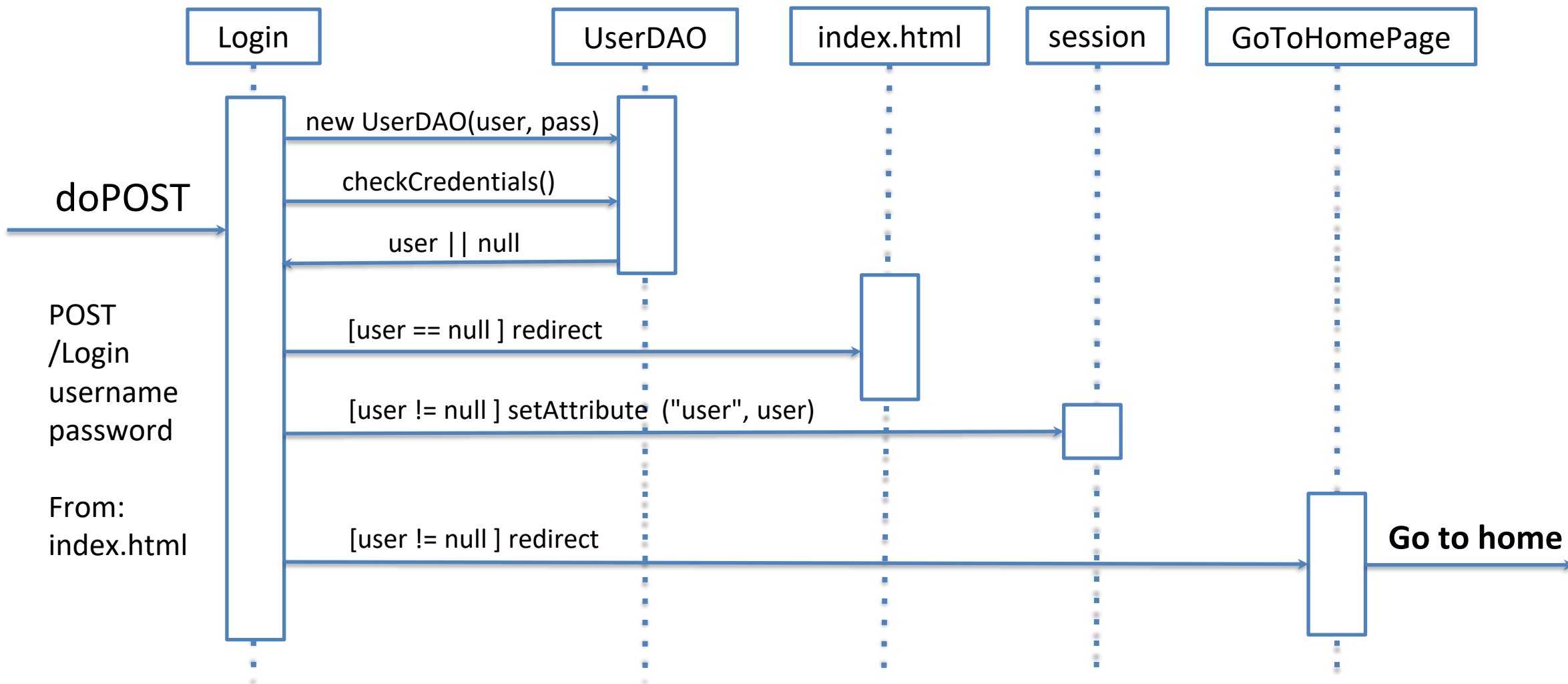
# Progettazione dell'applicazione



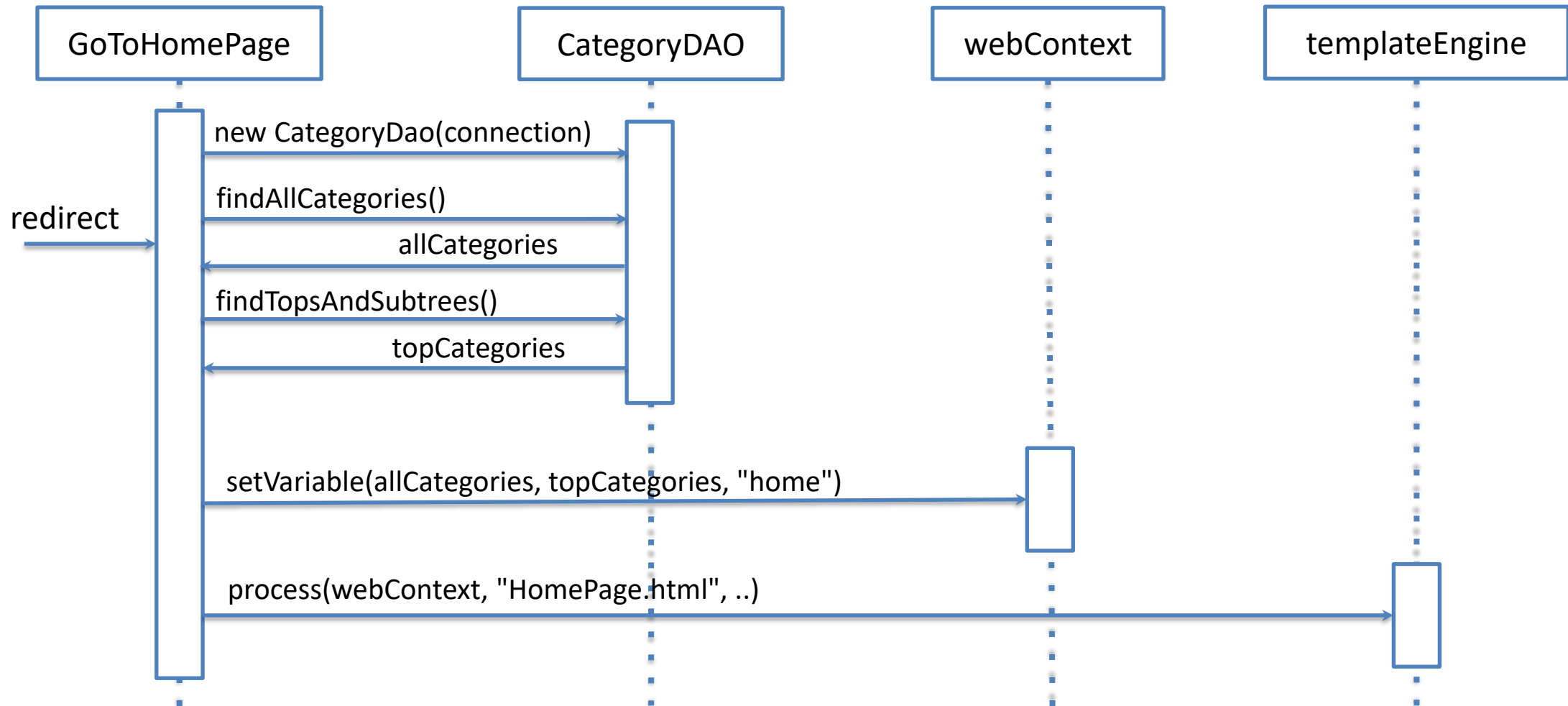
# Componenti

- Model Objects (Beans)
  - User
  - Category
- Data Access Objects (Classes)
  - UserDao
    - checkCredentials(username,password)
  - CategoryDao
    - findAllCategories()
    - findTopsAndSubtrees()
    - findSubclasses(category)
    - createCategory(name,fid)
    - moveCategory(cid,destid)
    - findNumChild(father)
    - updatePosition(position,id)
    - insertNewCategory(name,position)
    - addLink(fid,cid)
- Controllers (Servlets)
  - Login
  - GoToHomePage
  - CreateCategory
  - GoToMove
  - MoveCategory
  - Logout
- Views (Templates)
  - Login Page
  - Home Page
  - Category tree: fragment
  - Category tree: movefragment

# Evento: Login

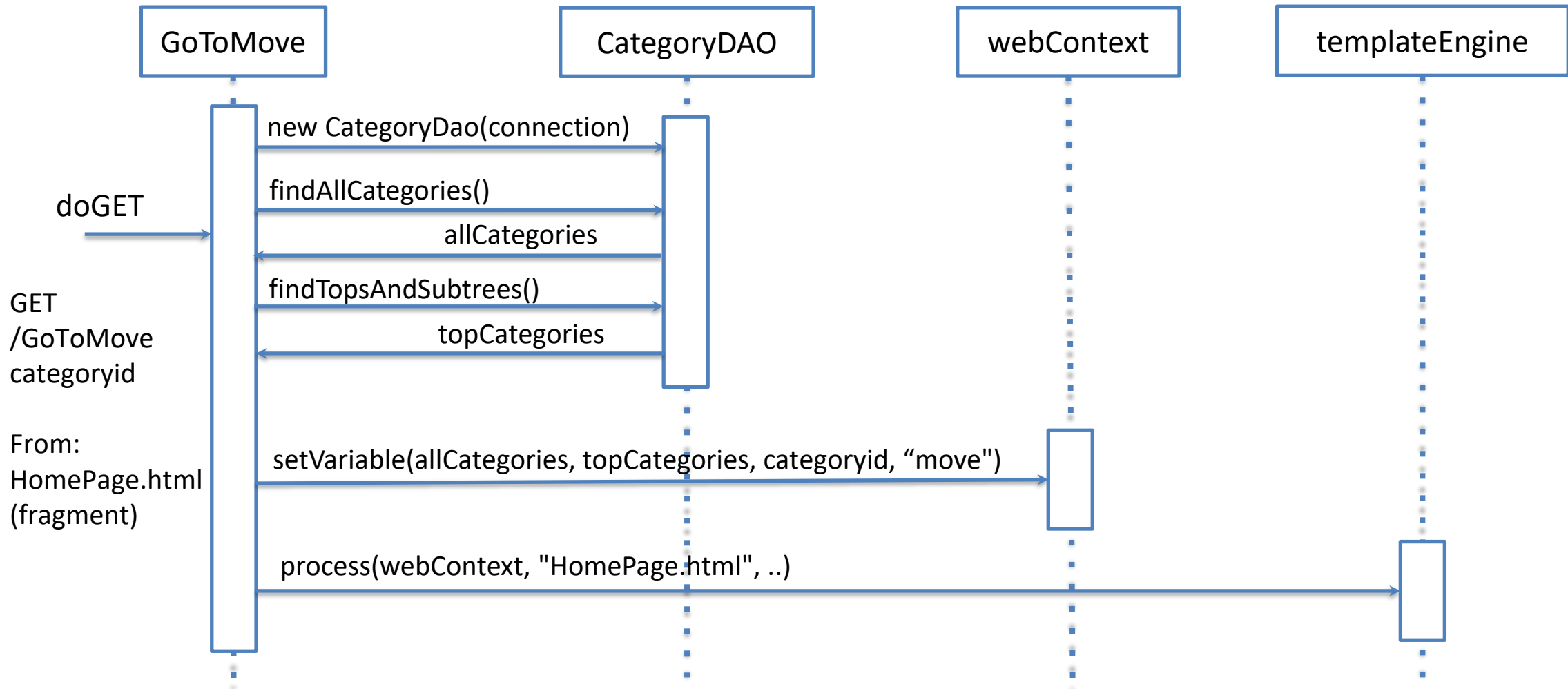


# Evento: Go to home

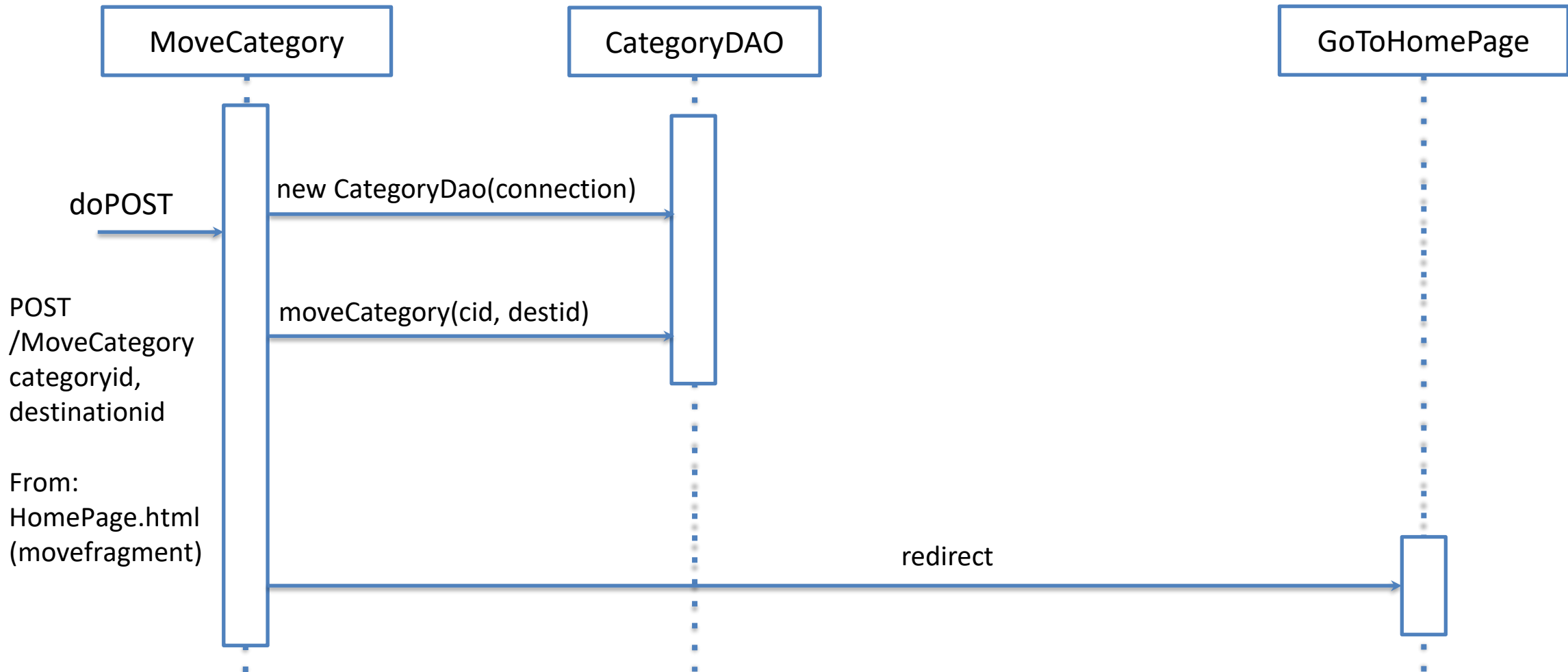




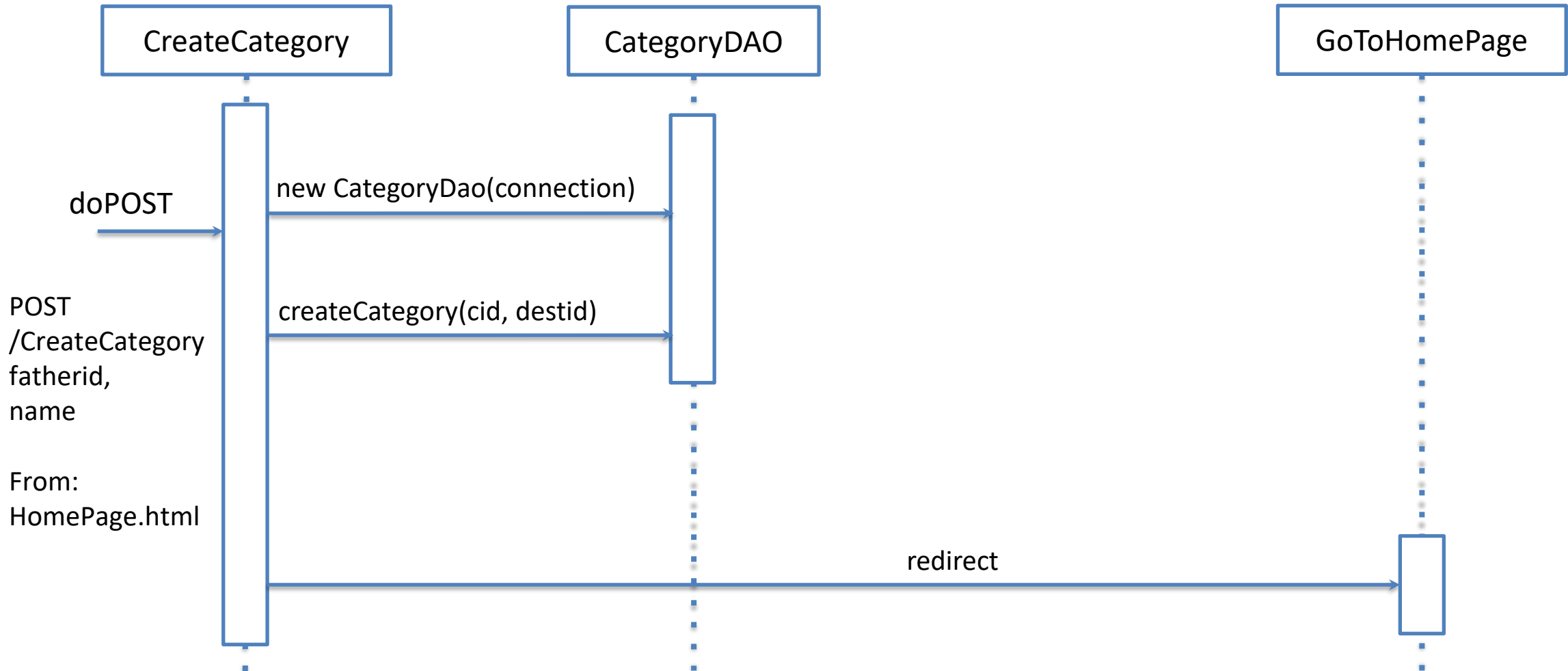
# Evento: Go to move



# Evento: Move category



# Evento: Create category



# Evento: Logout

