



**POLITECNICO**  
**MILANO 1863**

**TECNOLOGIE  
INFORMATICHE PER IL WEB**

**ESERCIZIO 2: GESTIONE DOCUMENTI**

---

**Giovanni Ni**

**Docente: Piero Fraternali**

## Esercizio 2 : Gestione Documenti – PURE HTML

L'applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione richiede l'inserimento di username, indirizzo di email e password e controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password". La registrazione controlla l'unicità dello username. Una cartella ha un proprietario, un nome e una data di creazione e può contenere altre cartelle e/o documenti. Un documento ha un proprietario, nome, una data di creazione, un sommario e un tipo. Quando l'utente accede all'applicazione appare una HOME PAGE che contiene un albero delle proprie cartelle e delle sottocartelle.

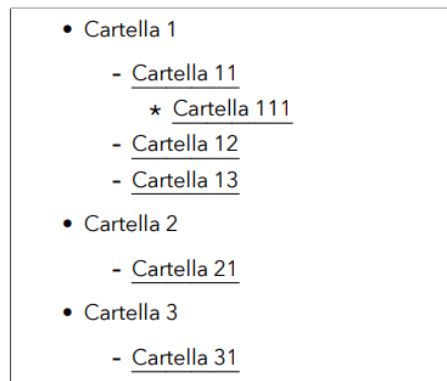


Figura 1: Esempio di albero delle cartelle nella HOME PAGE

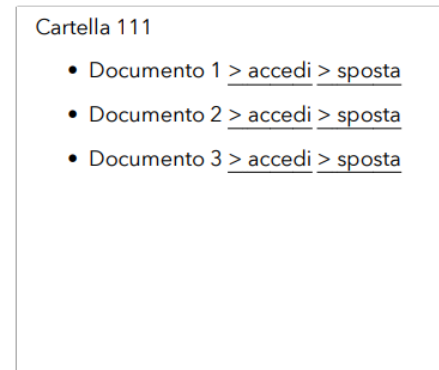


Figura 2: Contenuto della pagina CONTENUTI di una cartella

Nell'HOME PAGE l'utente può selezionare una cartella e accedere a una pagina CONTENUTI che mostra l'elenco delle cartelle e dei documenti di una cartella. Ogni documento in elenco ha due link: accedi e sposta. Quando l'utente seleziona il link accedi, appare una pagina DOCUMENTO (nella stessa finestra e tab del browser) che mostra tutti i dati del documento selezionato. Quando l'utente seleziona il link sposta, appare la HOME PAGE con l'albero delle cartelle; in questo caso la pagina mostra il messaggio "Stai spostando il documento X dalla cartella Y. Scegli la cartella di destinazione", la cartella a cui appartiene il documento da spostare NON è selezionabile e il suo nome è evidenziato (per esempio con un colore diverso). Quando l'utente seleziona la cartella di destinazione, il documento è spostato dalla cartella di origine a quella di destinazione e appare la pagina CONTENUTI che mostra il contenuto aggiornato della cartella di destinazione. Ogni pagina, tranne la HOME PAGE, contiene un collegamento per tornare alla pagina precedente. L'applicazione consente il logout dell'utente da qualsiasi pagina. Una pagina GESTIONE CONTENUTI raggiungibile dalla HOME PAGE permette all'utente di creare una cartella di primo livello, una cartella all'interno di una cartella esistente e un documento all'interno di una cartella. L'applicazione non richiede la gestione dell'upload dei documenti.

## **VERSIONE JAVASCRIPT**

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

- L'applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password", anche a lato client. La registrazione controlla l'unicità dello username.
- Dopo il login dell'utente, l'intera applicazione è realizzata con un'unica pagina.
- 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.
- Errori a lato server devono essere segnalati mediante un messaggio di allerta all'interno della pagina.
- La funzione di spostamento di un documento è realizzata mediante drag and drop.
- La funzione di creazione di una sottocartella è realizzata nella pagina HOME mediante un bottone AGGIUNGI SOTTOCARTELLA posto di fianco ad ogni cartella. La pressione del bottone fa apparire un campo di input per l'inserimento del nome della cartella da inserire.
- La funzione di creazione di un documento è realizzata nella pagina HOME mediante un bottone AGGIUNGI DOCUMENTO posto di fianco ad ogni cartella. La pressione del bottone fa apparire una form di input per l'inserimento dei dati del documento.
- Si aggiunge una cartella denominata "cestino". Il drag and drop di un documento o di una cartella nel cestino comporta la cancellazione. Prima di inviare il comando di cancellazione al server l'utente vede una finestra modale di conferma e può decidere se annullare l'operazione o procedere. La cancellazione di una cartella comporta la cancellazione integrale e ricorsiva del contenuto dalla base di dati (documenti e cartelle)

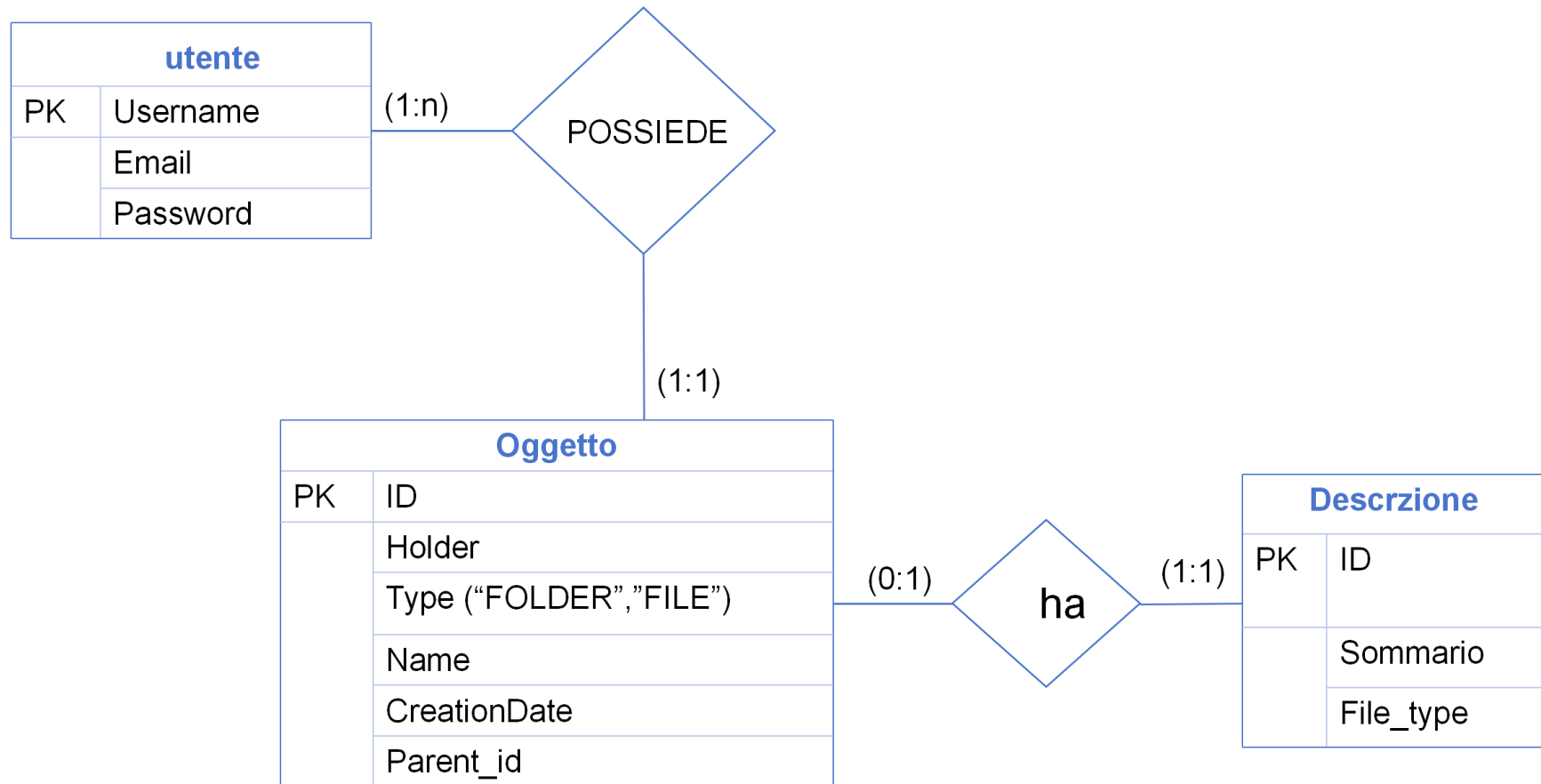
## Analisi dati per Database

L'applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione richiede l'inserimento di **username**, **indirizzo di email** e **password** e controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password". La registrazione controlla l'unicità dello username. **Una cartella** ha un **proprietario**, un **nome** e una **data di creazione** e può contenere altre **cartelle e/o documenti**. **Un documento** ha un **proprietario**, **nome**, **una data di creazione**, **un sommario** e un **tipo**. Quando l'**utente** accede all'applicazione appare una HOME PAGE che contiene **un albero delle proprie cartelle e delle sottocartelle**.

Nell'HOME PAGE l'utente può selezionare una cartella e accedere a una pagina CONTENUTI che mostra l'elenco delle cartelle e dei documenti di una cartella. Ogni documento in elenco ha due link: accedi e sposta. Quando l'utente seleziona il link accedi, appare una pagina DOCUMENTO (nella stessa finestra e tab del browser) che mostra tutti i dati del documento selezionato. Quando l'utente seleziona il link sposta, appare la HOME PAGE con l'albero delle cartelle; in questo caso la pagina mostra il messaggio "Stai spostando il documento X dalla cartella Y. Scegli la cartella di destinazione", la cartella a cui appartiene il documento da spostare NON è selezionabile e il suo nome è evidenziato (per esempio con un colore diverso). Quando l'utente seleziona la cartella di destinazione, il documento è spostato dalla cartella di origine a quella di destinazione e appare la pagina CONTENUTI che mostra il contenuto aggiornato della cartella di destinazione. Ogni pagina, tranne la HOME PAGE, contiene un collegamento per tornare alla pagina precedente. L'applicazione consente il logout dell'utente da qualsiasi pagina. Una pagina GESTIONE CONTENUTI raggiungibile dalla HOME PAGE permette all'utente di creare una cartella di primo livello, una cartella all'interno di una cartella esistente e un documento all'interno di una cartella. L'applicazione non richiede la gestione dell'upload dei documenti.

- **Entities**, **attributes**, **relationships**

## Database Design: concettuale



## Database Design: logico

Utente(Username, Email, Password)

Oggetto(ID, Holder, Tipo, Nome)

Descrizione(ID, Sommario, Tipo)



## Database Schema (1/2)

```
create table `foldersystem` (  
    'id' int not null auto_increment,  
    'holder' varchar(40) not null,  
    'name' varchar(40) not null,  
    'creation_date' varchar(40) not null,  
    'tipo' ENUM('FOLDER', 'FILE'),  
    'parent_id' int,  
    primary key (`id`),  
    unique(`email`)  
);
```

```
create table `tb_user` (  
    'username' varchar(50) not null,  
    'email' varchar(50) not null,  
    'password' varchar(50) not null,  
    primary key (`username`),  
    unique(`email`)  
);
```

## Database Schema (2/2)

```
create table `tb_document` (  
  
    'id' int not null,  
    'sommario' varchar(100) not null,  
    'tipo' varchar(20) not null,  
  
    primary key (`id`),  
  
);
```



## Analisi Requisiti Applicazione – pure HTML

L'applicazione supporta **registrazione** e **login** mediante una pagina pubblica con **opportune form**. La registrazione richiede l'inserimento di username, indirizzo di e-mail e password e **controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password"**. La registrazione **controlla l'unicità dello username**. Una cartella ha un proprietario, un nome e una data di creazione e può contenere altre cartelle e/o documenti. Un documento ha un proprietario, nome, una data di creazione, un sommario e un tipo. Quando l'utente accede all'applicazione appare una **HOME PAGE** che **contiene un albero delle proprie cartelle e delle sottocartelle**.

Nell'**HOME PAGE** l'utente può **selezionare una cartella e accedere** a una **pagina CONTENUTI** che **mostra l'elenco delle cartelle e dei documenti di una cartella**. Ogni documento in elenco **ha due link: accedi e sposta**. Quando l'utente **seleziona il link accedi**, appare una **pagina DOCUMENTO** (nella stessa finestra e tab del browser) che **mostra tutti i dati del documento selezionato**. Quando l'utente **seleziona il link sposta**, appare la **HOME PAGE** con l'**albero delle cartelle**; in questo caso la pagina **mostra il messaggio "Stai spostando il documento X dalla cartella Y. Scegli la cartella di destinazione"**, la cartella a cui appartiene il documento da spostare NON è selezionabile e il suo nome è evidenziato (per esempio con un colore diverso). Quando l'utente **seleziona** la cartella di destinazione, il documento **è spostato dalla cartella di origine a quella di destinazione** e appare la **pagina CONTENUTI** che **mostra il contenuto aggiornato della cartella di destinazione**. Ogni pagina, tranne la **HOME PAGE**, contiene un **collegamento** per tornare alla pagina precedente. L'applicazione consente il **logout** dell'utente da qualsiasi pagina. Una **pagina GESTIONE CONTENUTI** raggiungibile dalla **HOME PAGE** permette all'utente di **creare** una cartella di primo livello, una cartella all'interno di una cartella esistente e un documento all'interno di una cartella. L'applicazione non richiede la gestione dell'upload dei documenti.

- **pages (views)**, **view components**, **events**, **actions**

## **Analisi Requisiti Applicazione – versione Javascript**

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

- L'applicazione supporta **registrazione** e **login** mediante una **pagina pubblica** con opportune **form**. La registrazione **controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password"**, anche a lato client. La registrazione **controlla l'unicità dello username**.
  - Dopo il **login** dell'utente, l'intera applicazione è realizzata con un'**unica pagina**.
  - 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**.
  - Errori a lato server devono essere segnalati mediante un **messaggio di allerta** all'interno della pagina.
  - **La funzione di spostamento di un documento è realizzata mediante drag and drop**.
  - La funzione di **creazione di una sottocartella** è realizzata nella **pagina HOME** mediante un **bottone AGGIUNGI SOTTOCARTELLA** posto di fianco ad ogni cartella. La **pressione del bottone** fa apparire un **campo di input** per l'inserimento del nome della cartella da inserire.
  - La funzione di **creazione di un documento** è realizzata nella **pagina HOME** mediante un **bottone AGGIUNGI DOCUMENTO** posto di fianco ad ogni cartella. La **pressione del bottone** fa apparire una **form di input** per l'inserimento dei dati del documento.
  - Si aggiunge una cartella denominata "cestino". Il drag and drop di un documento o di una cartella nel cestino comporta la **cancellazione**. Prima di **inviare** il comando di cancellazione al server l'utente vede una **finestra modale di conferma** e può decidere se **annullare l'operazione o procedere**. La **cancellazione** di una cartella comporta la cancellazione integrale e ricorsiva del contenuto dalla base di dati (documenti e cartelle)
- 
- **pages (views), view components, events, actions**

## **Completamento delle Specifiche**

- Ogni utente è identificato dal suo username o dalla sua mail.
- Per login servono username e password.
- Se l'utente non è loggato e tenta l'accesso a zone protette, verrà reindirizzato a una pagina di errore.
- Se in un qualsiasi momento si verificano azioni impreviste, l'utente viene indirizzato a una pagina di ERRORE contenente il motivo dell'errore.
- Dalla pagina di login, è possibile spostarsi nella pagina di REGISTRAZIONE, dove inserendo nome, email e password, conferma password in una form, è possibile creare un utente; se la mail o username fornita è già in uso l'utente viene riportato alla pagina di REGISTRAZIONE in cui gli si mostra un messaggio di errore, altrimenti viene indirizzato alla pagina di HomePage.
- Tutti gli oggetti nella stessa cartella devono avere nomi diversi.

## **Componenti – pure HTML (1/2)**

### **Model Objects (Beans)**

- User
- File
- Folder

### **Data access objects(DAO)**

- UserDao
  - checkCredential(String username, String password): User
  - findUser(String userName): Boolean
  - addUser(User u): int
  - findUserEmail(String email): Boolean
- FileDAO
  - checkFileName(String name): Boolean
  - add(File file): void
  - deleteFile(String id): void
  - moveFile(String id, String destination): void
  - getFile(String id): File
- FolderDAO
  - checkFolderName(String name): Boolean
  - add(Folder folder): void
  - deleteFolder(String id): void
  - getFolderName(String folderId): String
- ObjectDAO
  - delete(String id): void
  - generateFileSystemHtml(..): void

## **Componenti – pure HTML (2/2)**

### **Filters**

- LoginFilter

### **Controllers (servlets) [access rights]**

- AddFile [logged user]
- AddFolder [logged user]
- Contenuto [logged user]
- Document [logged user]
- HomePage [logged user]
- Login [not logged user]
- Logout [all]
- Registration [not logged user]
- Sposta [logged user]

### **Views(Templates)**

- login.html
- AddFile.html
- AddFolder.html
- Contenuto.html
- Documento.html
- Gestione.html
- HomePage.html
- Register.html

### **Utils**

- JdbcHelper

## **Componenti – RIA (1/2)**

### **Model Objects (Beans)**

- User
- File
- Folder

### **Data access objects(DAO)**

- UserDao
  - checkCredential(String username, String password): User
  - findUser(String userName): Boolean
  - addUser(User u): int
  - findUserEmail(String email): Boolean
- FileDAO
  - checkFileName(String name): Boolean
  - add(File file): void
  - deleteFile(String id): void
  - moveFile(String id, String destination): void
  - getFile(String id): File
- FolderDAO
  - checkFolderName(String name): Boolean
  - add(Folder folder): void
  - deleteFolder(String id): void
  - getFolderName(String folderId): String
- ObjectDAO
  - delete(String id): void
  - generateFileSystemHtml(..): void

## **Componenti – RIA (2/2)**

### **Filters**

- LoginFilter

### **Controllers (servlets) [access rights]**

- AddFile [logged user]
- AddFolder [logged user]
- VisitaFile [logged user]
- HomePage [logged user]
- CheckLogin [not logged user]
- Logout [all]
- Registration [not logged user]
- MoveFile [logged user]
- Delete [logged user]

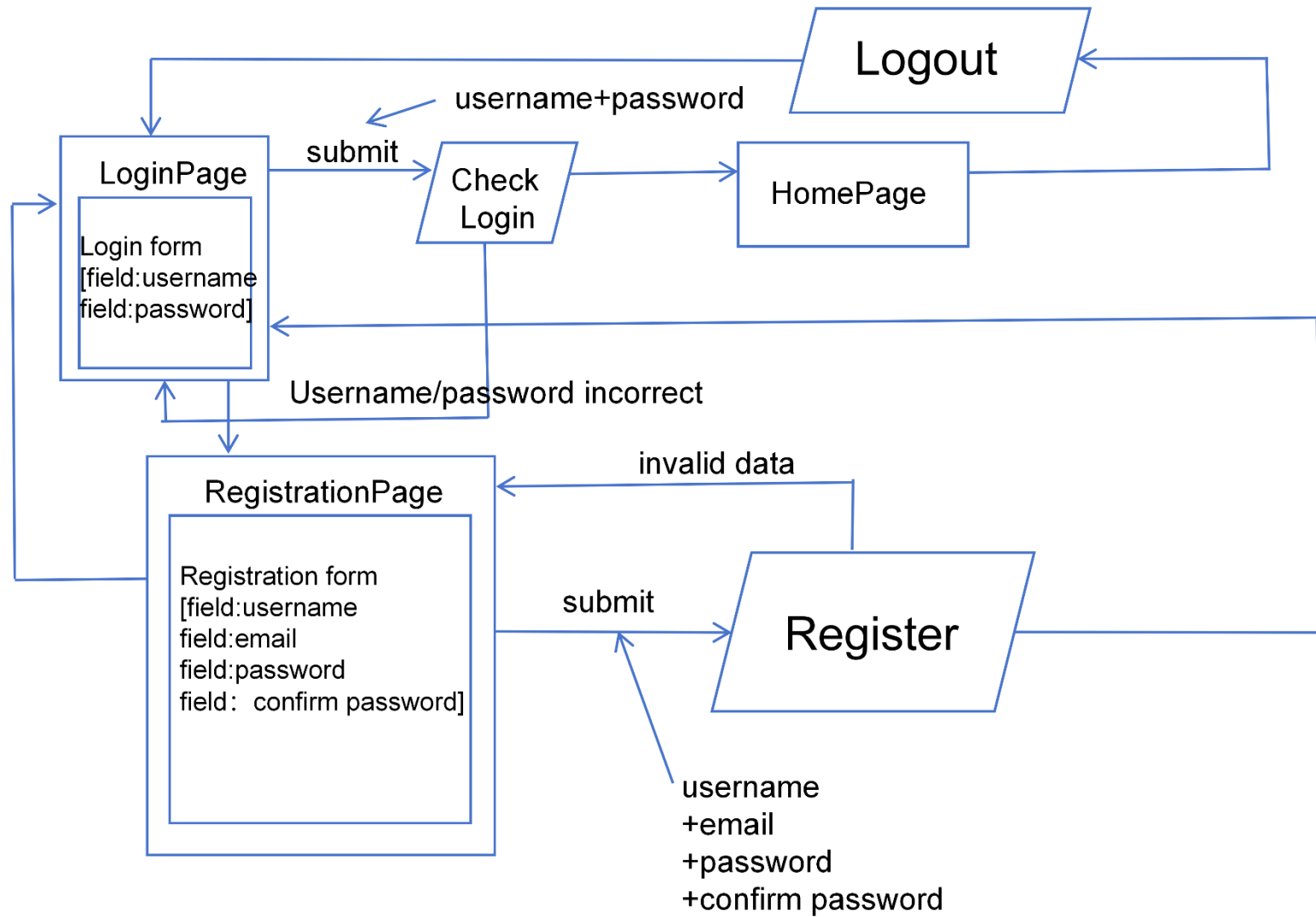
### **Views(Templates) and .js**

- index.html
- HomeCS.html
- Register.html
- loginManagement.js
- registration.js
- script.js
- utils.js

### **Utilis**

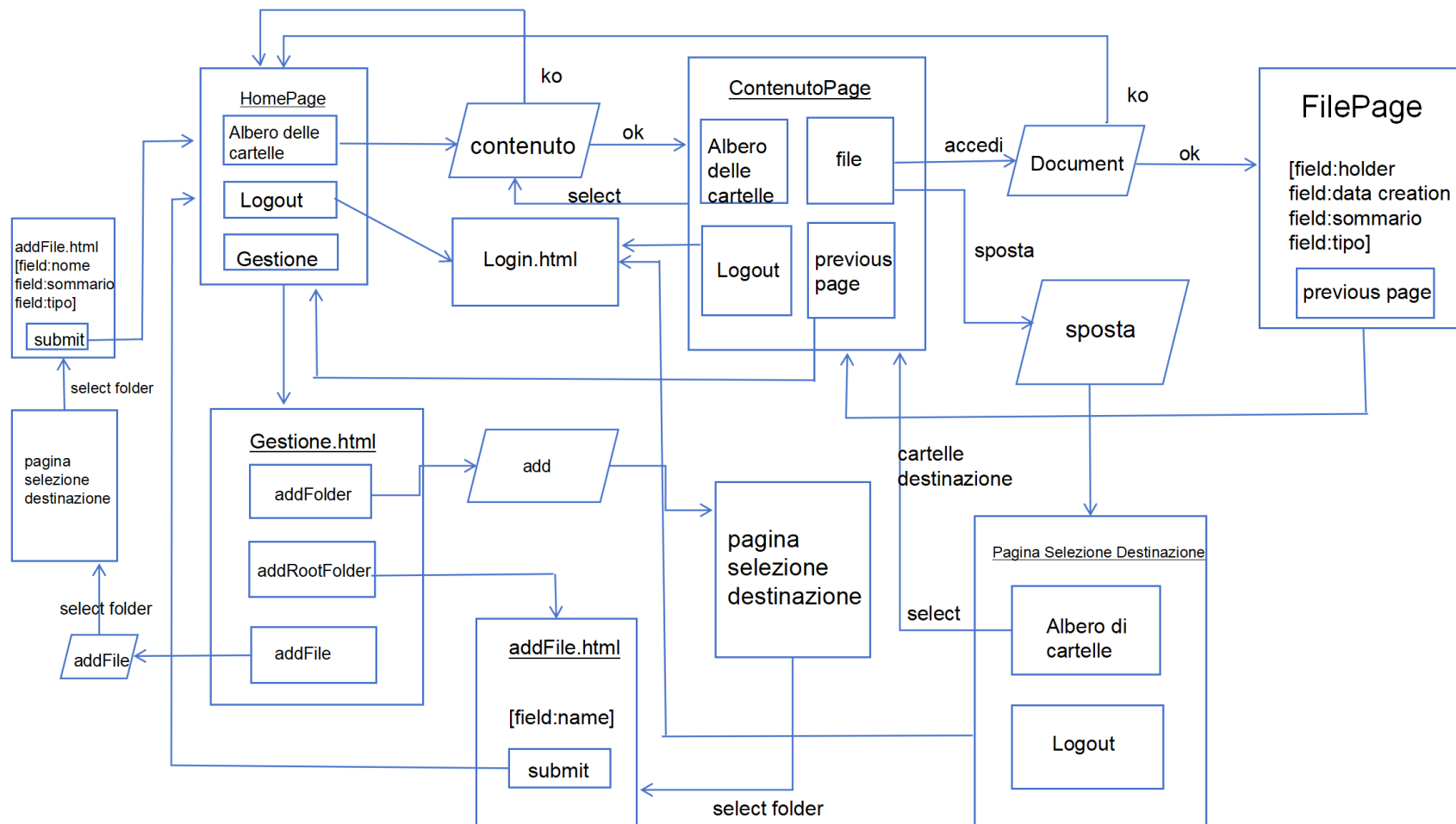
- JdbcHelper

## Application Design – HTML (1/2)

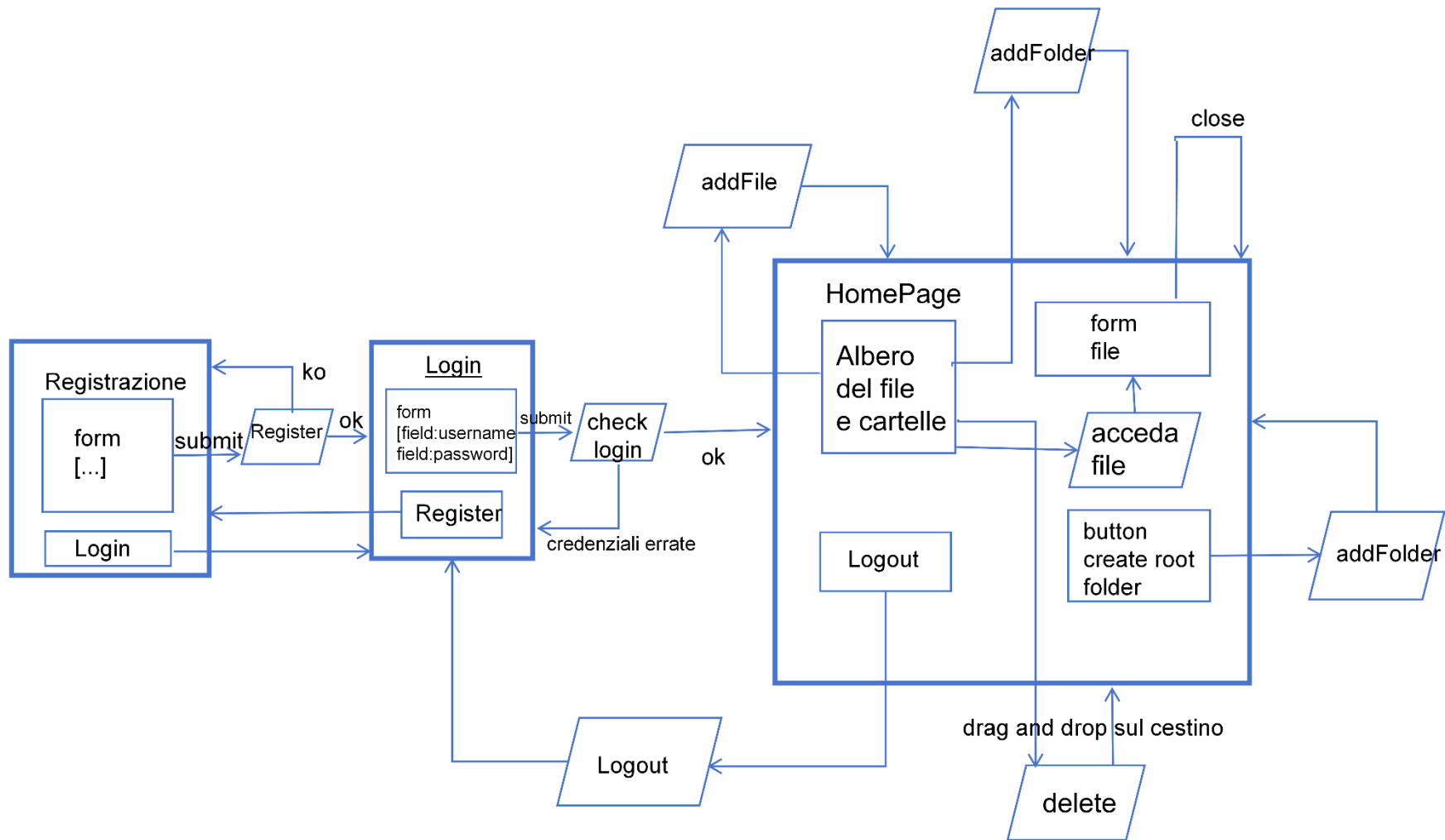




## Application Design – HTML (2/2)



## Application Design – RIA

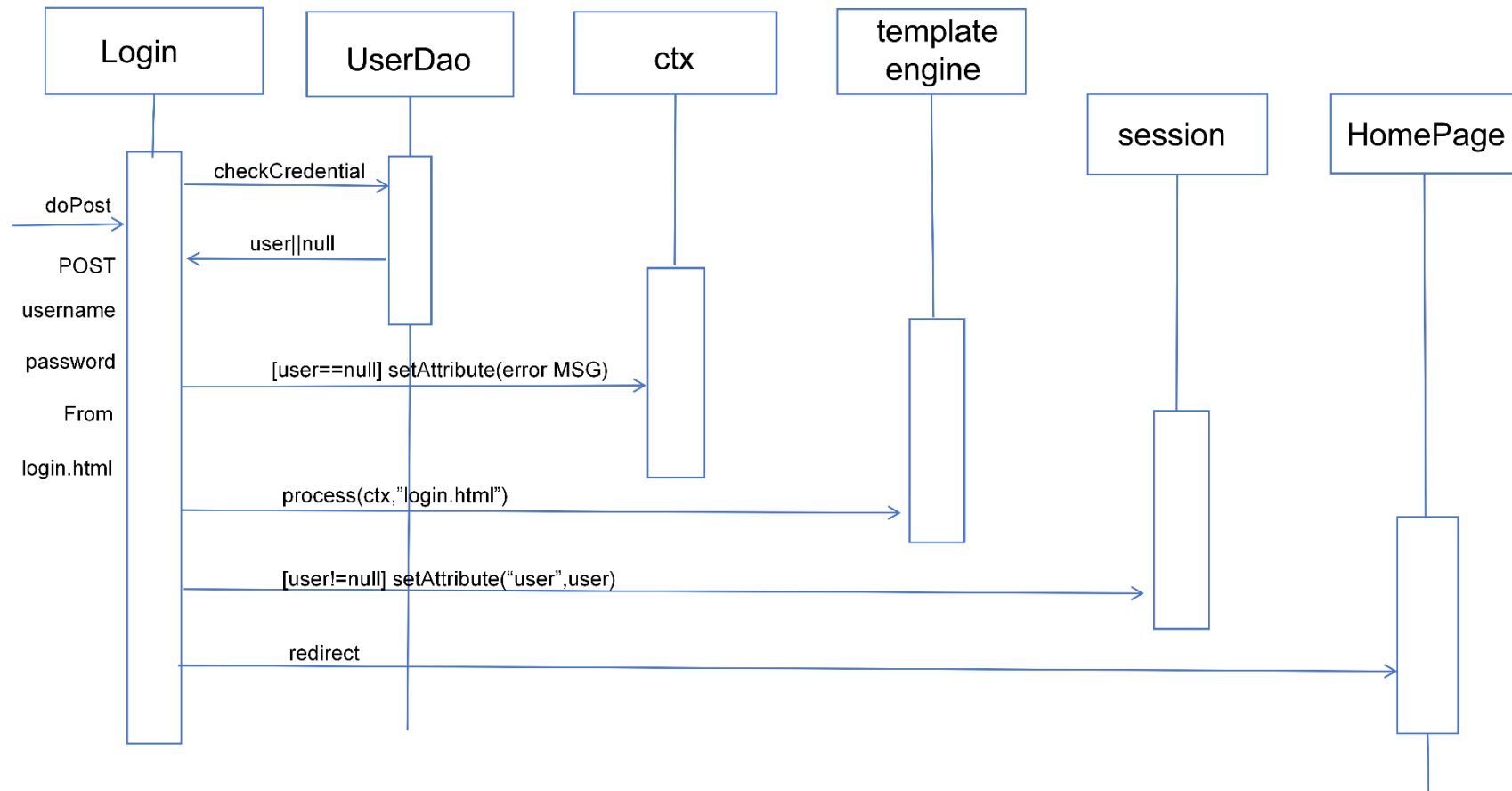


# Sequence Diagram

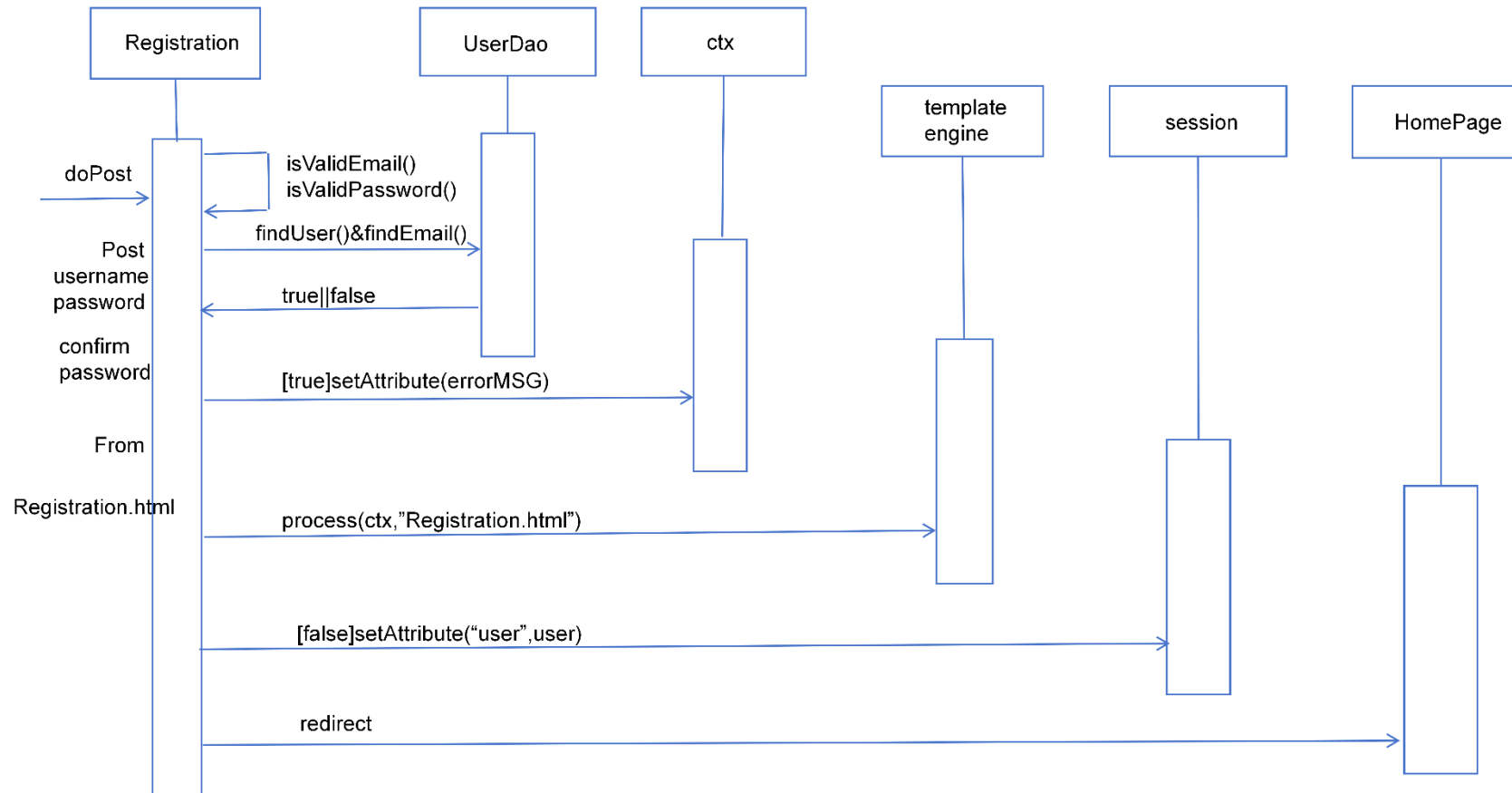
I seguenti diagrammi di sequenza mirano a descrivere gli eventi principali dell'applicazione web. alcuni dettagli minori sono stati omessi a causa della loro ripetitività (ad esempio errori interni del server, errori di accesso al database, parametri nulli).

In particolare, gli errori di accesso al database portano sempre alla pagina di errore.

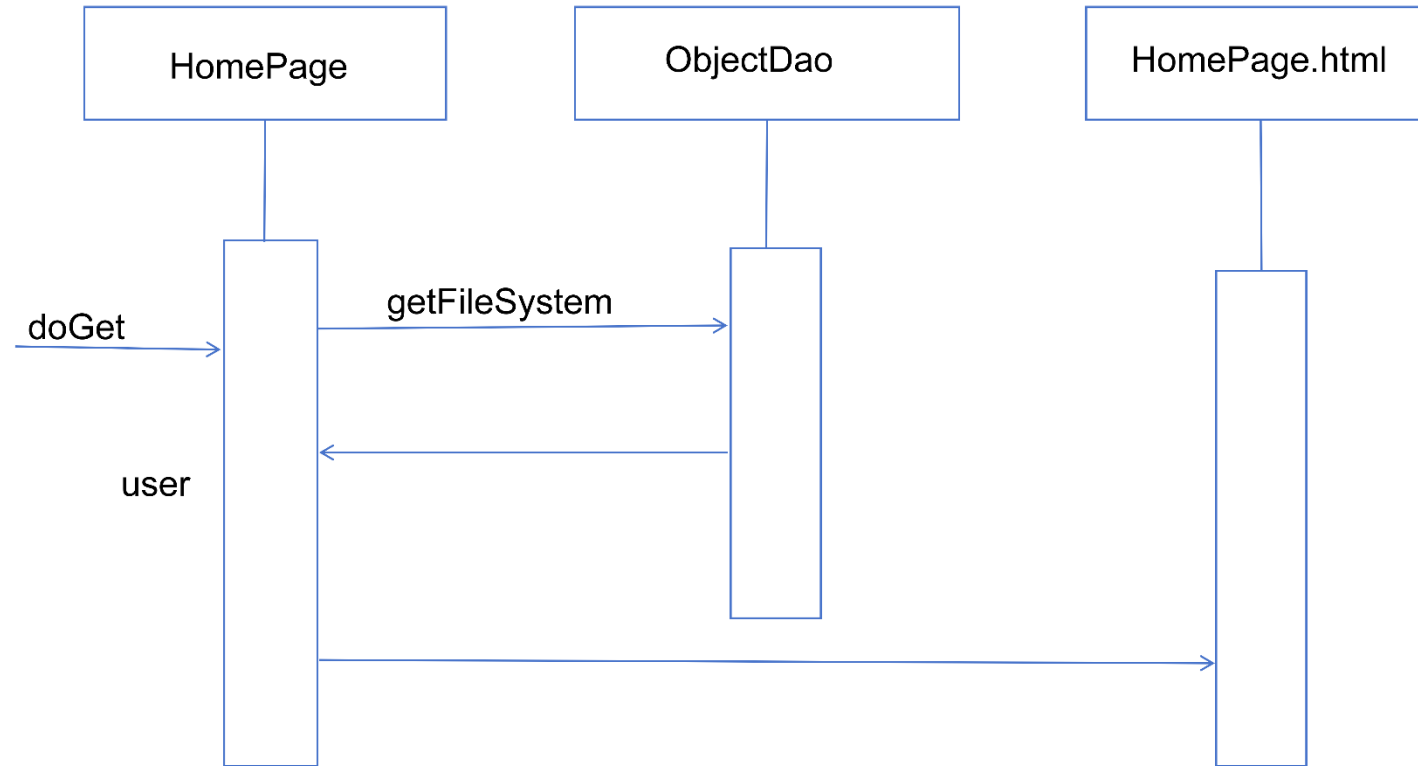
## Event: Login



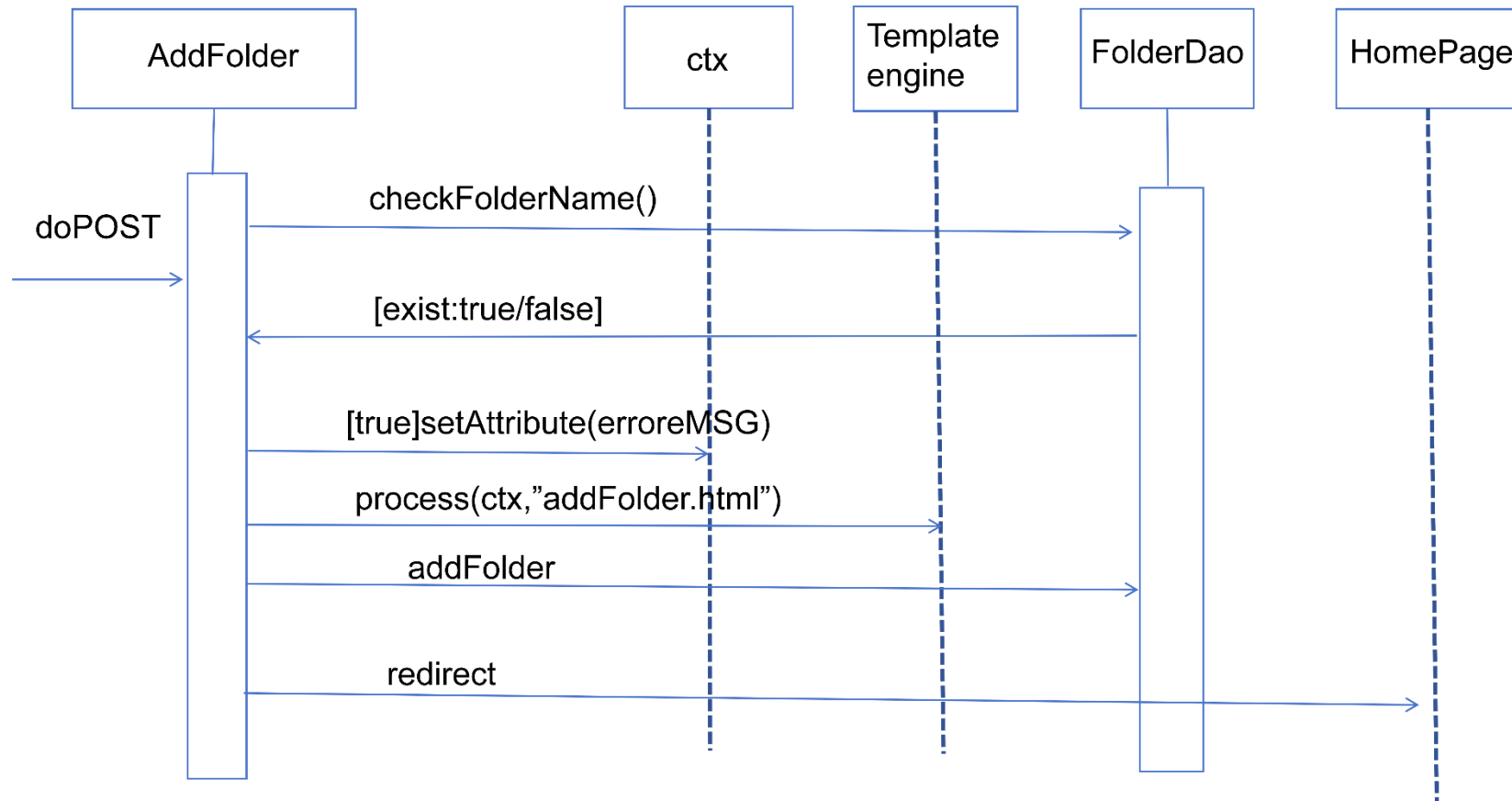
## Event: Registration



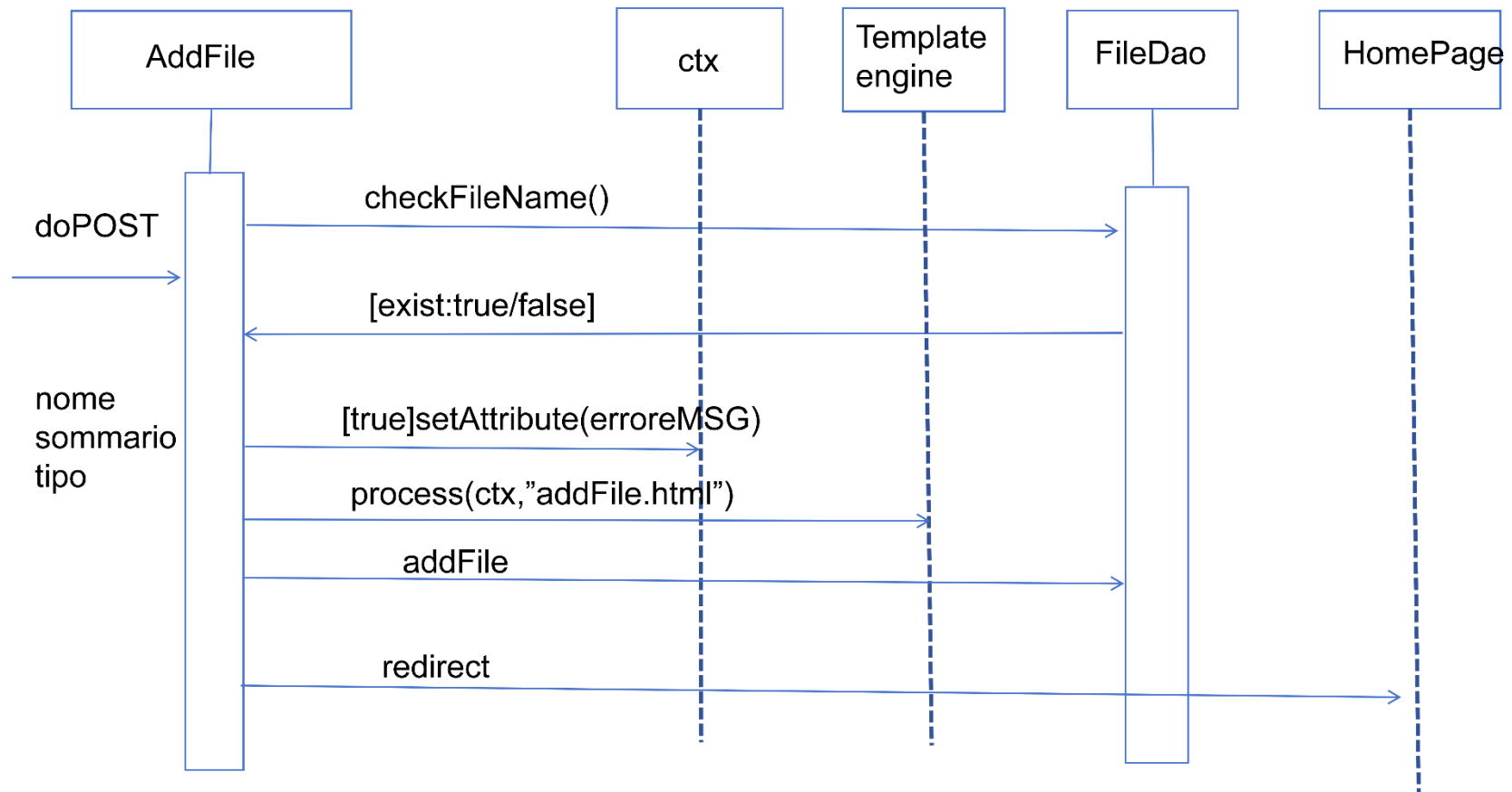
## Event: HomePage



## Event: AddFolder

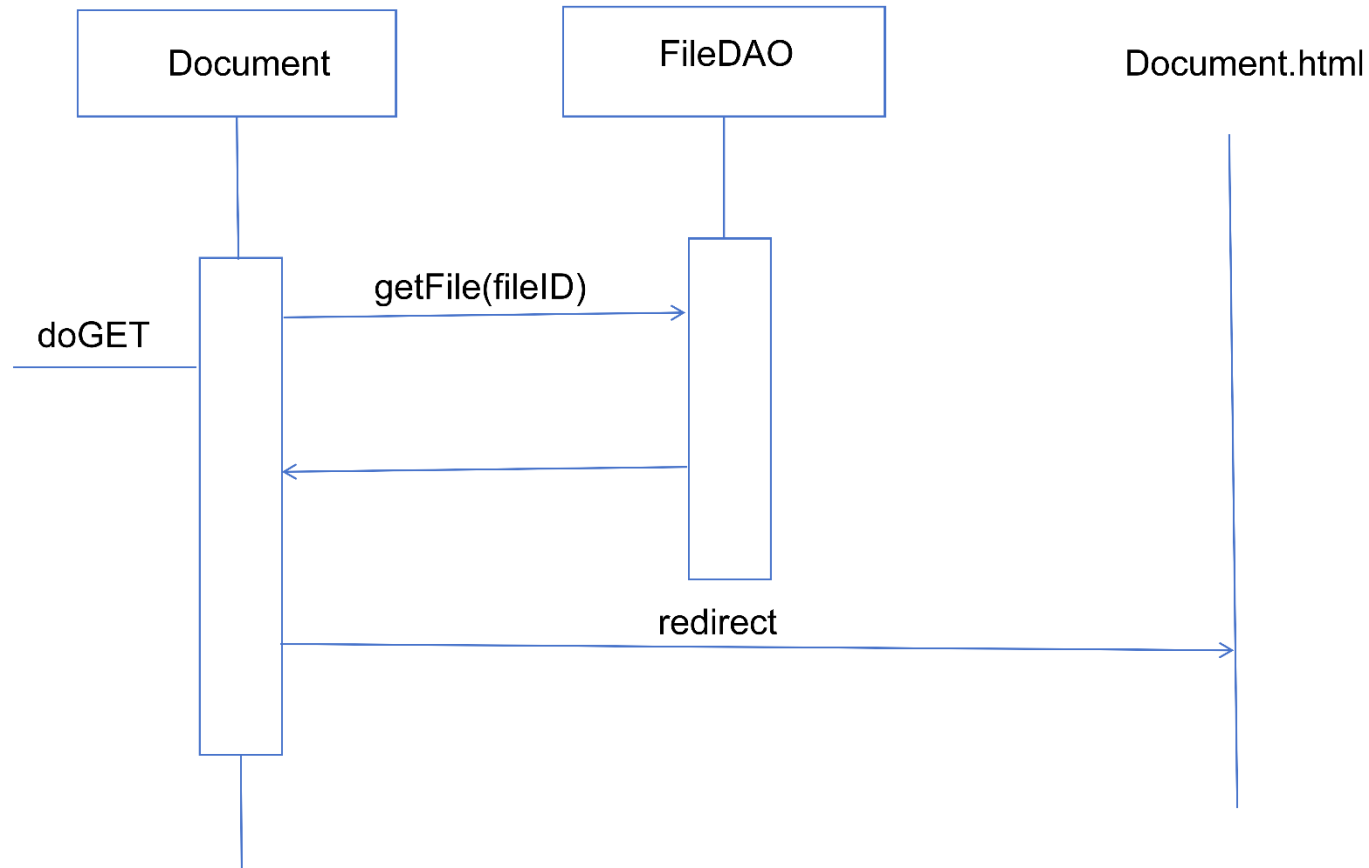


## Event: AddFile



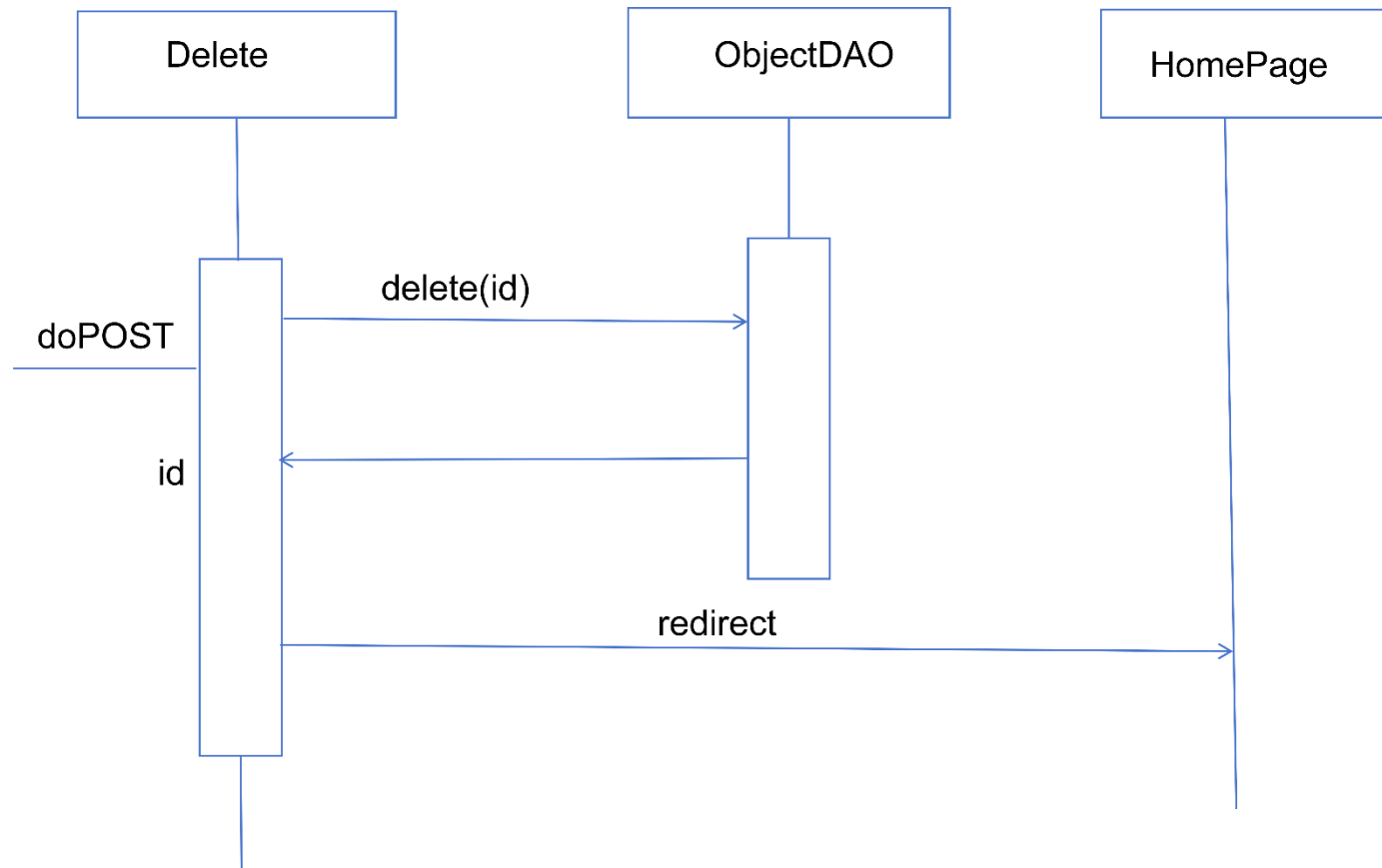


## Event: accediFile

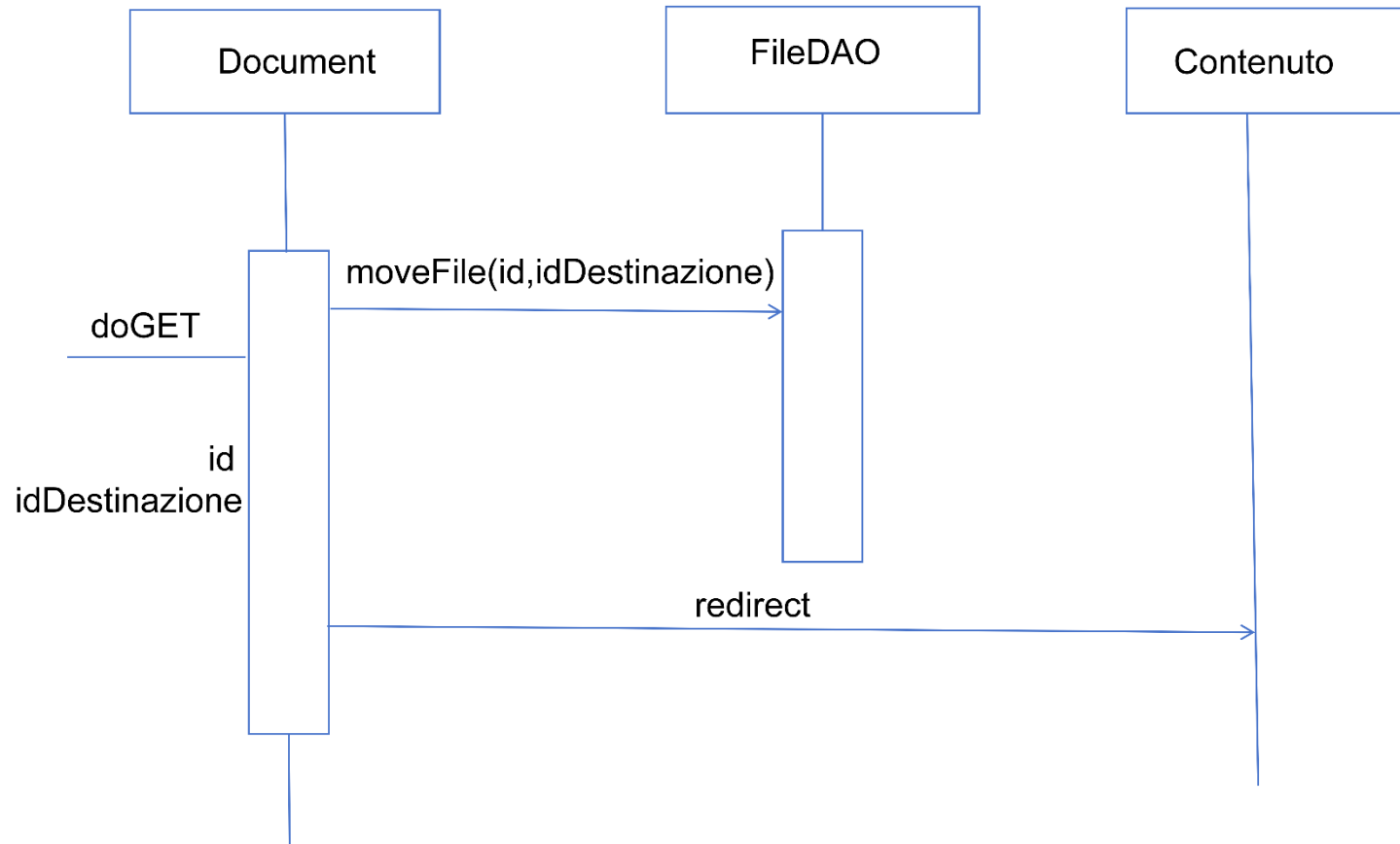


(Solo RIA)

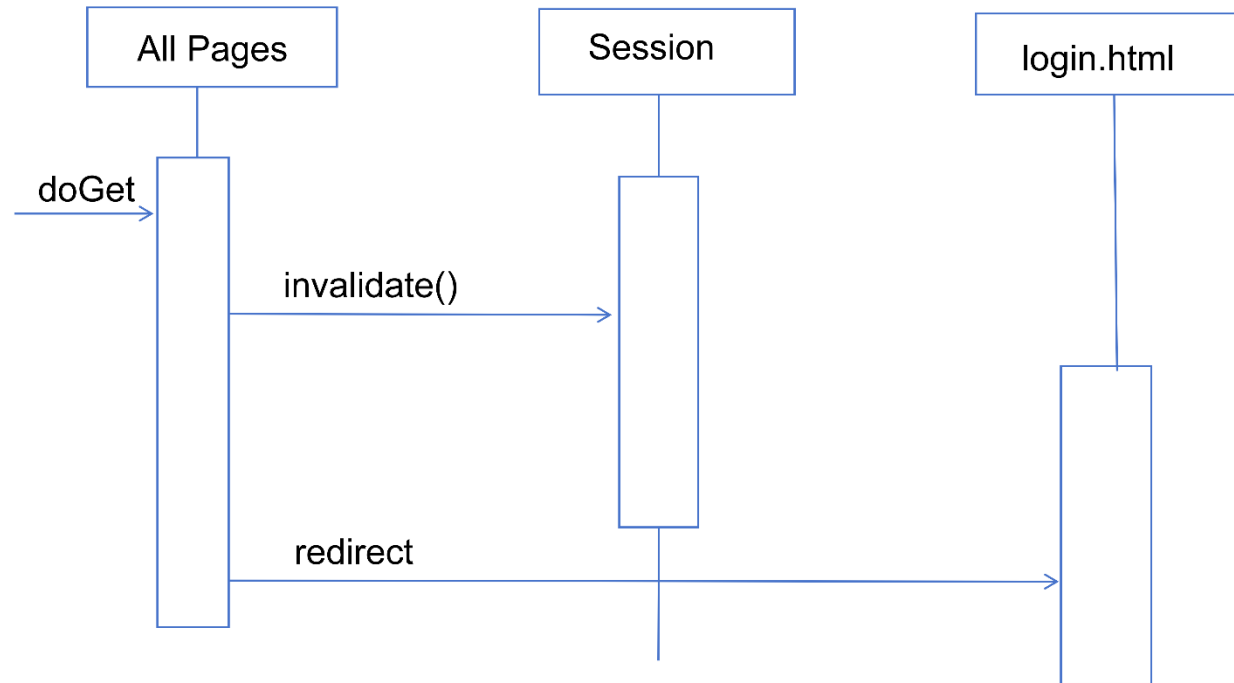
## Event: Delete



## Event: sposta File



## Event: Logout



**FINE**