



# **TIW – GESTIONE DOCUMENTI**

**ANDREA RUBAGOTTI** (10780379 – 982036)

**MARIA CONCETTA SANTAGATA** (10799787 – 987495)

IL PROGETTO È STATO REALIZZATO CON UN  
FOCUS SPECIFICO SU TRE MACROAREE CHIAVE:



BASE DI DATI



HTML



RIA - Javascript



# BASE DI DATI



# ANALISI DEI DATI

## Esercizio 2: Gestione documenti

### Versione pure HTML

Entità

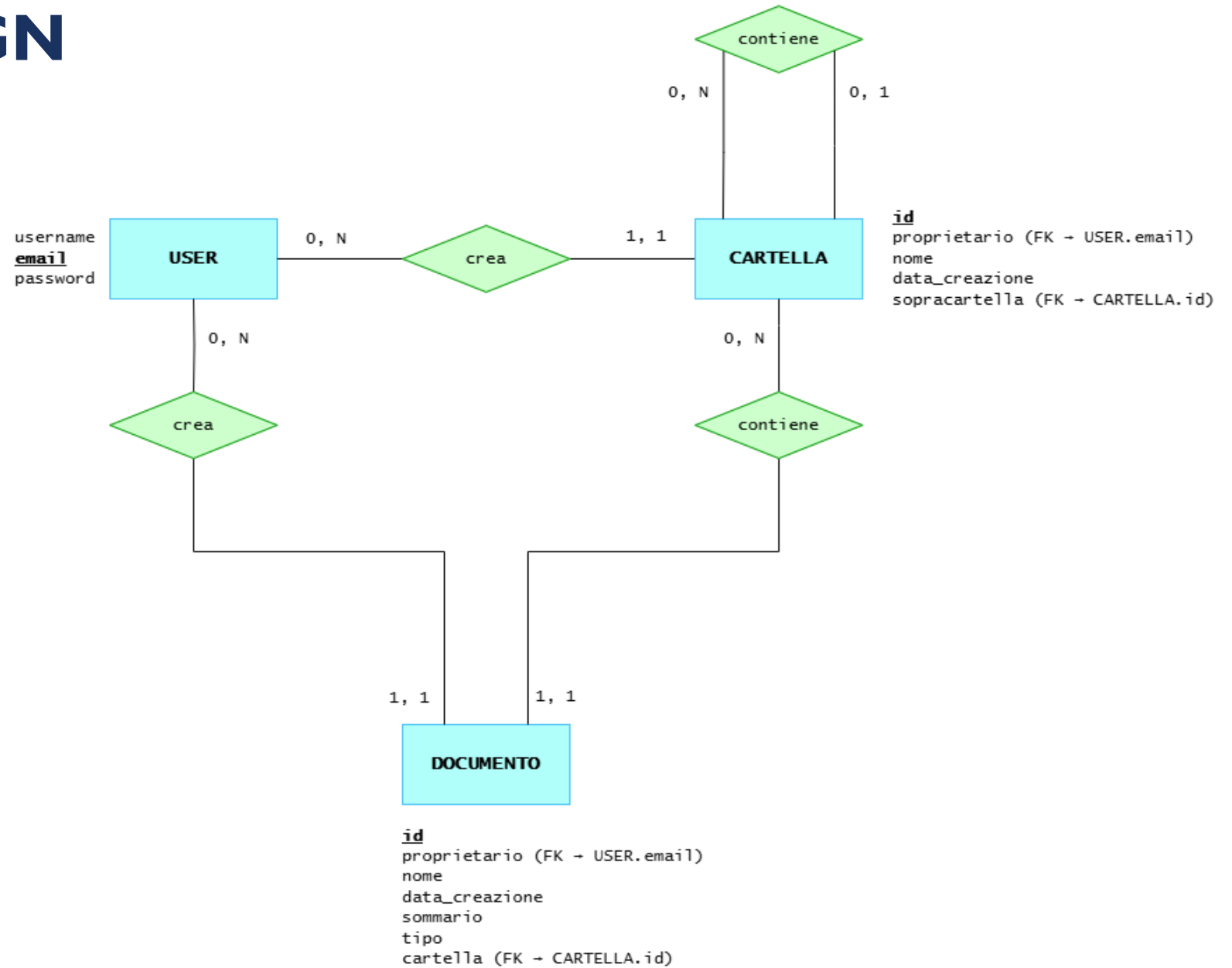
Relazioni

Attributi

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. (sopracartella)

# DATABASE DESIGN



# LOCAL DATABASE SCHEMA

## USER

```
CREATE TABLE `user` (  
  
  `username` varchar(50) NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `password` varchar(50) NOT NULL,  
  
  PRIMARY KEY (`email`),  
  
  UNIQUE KEY `username_UNIQUE` (`username`)  
  
)
```

## CARTELLA

```
CREATE TABLE `cartella` (  
  
  `id` int NOT NULL AUTO_INCREMENT,  
  `proprietario` varchar(255) NOT NULL,  
  `nome` varchar(50) NOT NULL,  
  `data_creazione` date NOT NULL,  
  `sopracartella` int DEFAULT NULL,  
  
  PRIMARY KEY (`id`),  
  
  CONSTRAINT `proprietario` FOREIGN KEY (`proprietario`) REFERENCES `user` (`email`) ON DELETE  
  CASCADE ON UPDATE CASCADE,  
  
  CONSTRAINT `sopracartella` FOREIGN KEY (`sopracartella`) REFERENCES `cartella` (`id`) ON DELETE  
  CASCADE ON UPDATE CASCADE  
  
)
```

## DOCUMENTO

```
CREATE TABLE `documento` (  
  
  `id` int NOT NULL AUTO_INCREMENT,  
  `proprietario` varchar(255) NOT NULL,  
  `nome` varchar(50) NOT NULL,  
  `data_creazione` date NOT NULL,  
  `sommario` varchar(300) DEFAULT NULL,  
  `tipo` varchar(50) NOT NULL,  
  `cartella` int NOT NULL,  
  
  PRIMARY KEY (`id`),  
  
  CONSTRAINT `cartella` FOREIGN KEY (`cartella`) REFERENCES `cartella` (`id`) ON DELETE CASCADE  
  ON UPDATE CASCADE,  
  
  CONSTRAINT `utente` FOREIGN KEY (`proprietario`) REFERENCES `user` (`email`) ON DELETE  
  CASCADE ON UPDATE CASCADE  
  
)
```



# HTML

```
96     <h1>One more for good measure.</h1>
97     <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit sed purus consectetur elit.
98     .</p>
99     <p><a class="btn btn-lg btn-primary" href="#" role="button">Browse gallery</a>
100     </div>
101     </div>
102     <a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev">
103         <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
104     </a>
105     <a class="right carousel-control" href="#myCarousel" role="button" data-slide="next">
106         <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
107     </a>
108     </div><!-- /.carousel -->
109
110 <!--Featured Content Section-->
111
112 <div class="container">
113     <div class="row">
114         <div class="col-md-4"></div>
115         <div class="col-md-4"></div>
116         <div class="col-md-4"></div>
117     </div>
118     <div class="row">
119         <div class="col-md-4"></div>
120         <div class="col-md-4"></div>
121         <div class="col-md-4"></div>
122     </div>
123     </div>
124 </div><!-- /container -->
125
126 <div class="text-center">
127     <h2>FEATURED CONTENT</h2>
128     <div class="featurette-divider">
129         <div class="row">
130             <div class="col-md-6">
131                 <h3>First featurette</h3>
132                 <p>...</p>
133             </div>
134             <div class="col-md-6">
135                 <h3>Second featurette</h3>
136                 <p>...</p>
137             </div>
138         </div>
139     </div>
140 </div>
```

# ANALISI DEI DATI

## Esercizio 2: Gestione documenti

### Versione 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.

- Cartella 1
  - Cartella 11
    - ★ Cartella 111
  - Cartella 12
  - Cartella 13
- Cartella 2
  - Cartella 21
- Cartella 3
  - Cartella 31

Figura 1: Esempio di albero delle cartelle nella HOME PAGE

Cartella 111

- Documento 1 > accedi > sposta
- Documento 2 > accedi > sposta
- Documento 3 > accedi > sposta

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.



# COMPONENTI

## BEANS

- File.java
- Folder.java

## DAO

- CartellaDao.java
- DocumentoDao.java
- UserDao.java

## CONTROLLERS

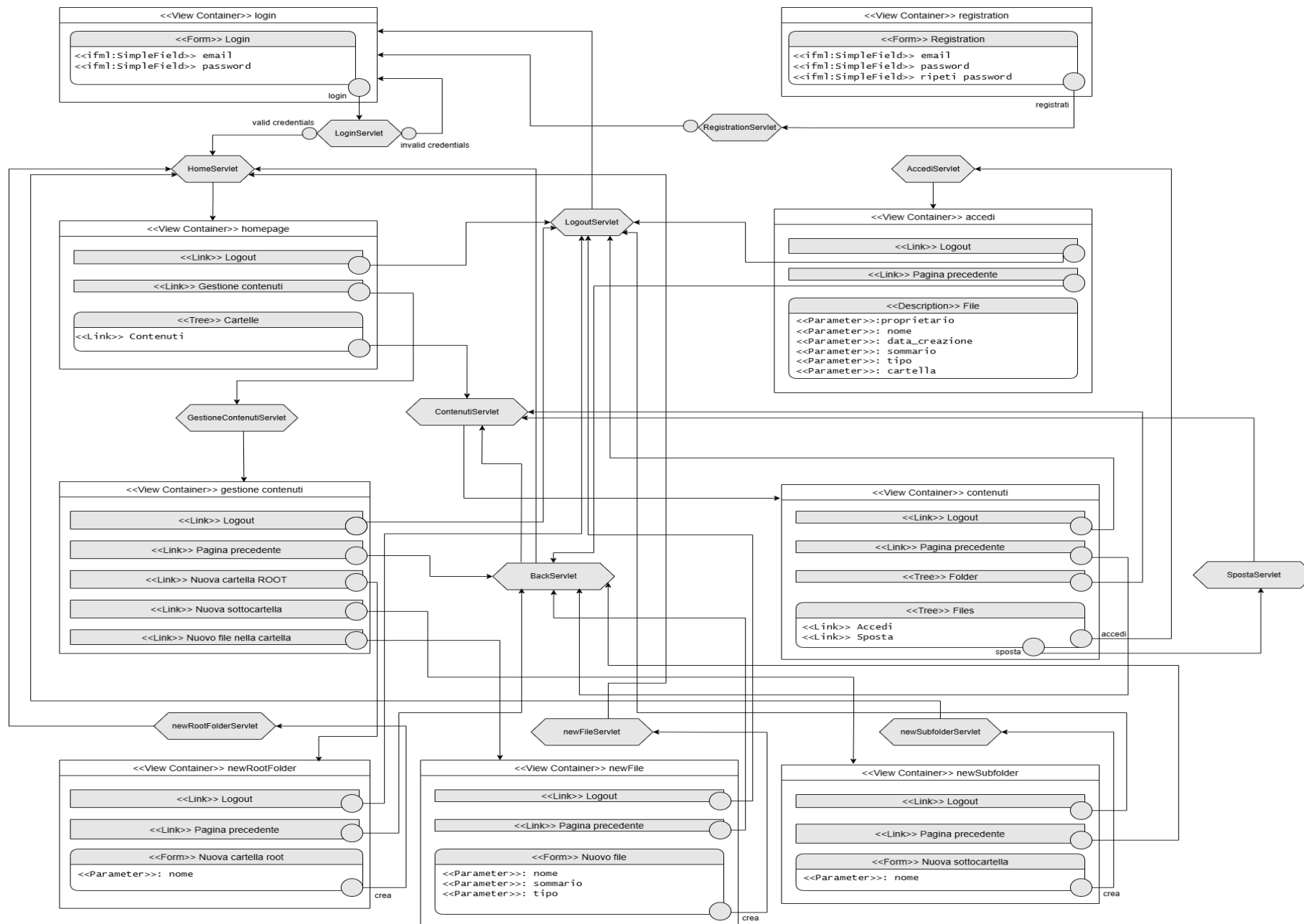
- AccediServlet.java
- BackServlet.java
- ContenutiServlet.java
- GestioneContenutiServlet.java
- HomeServlet.java
- LoginServlet.java
- LogoutServlet.java

## CSS COMPONENTS

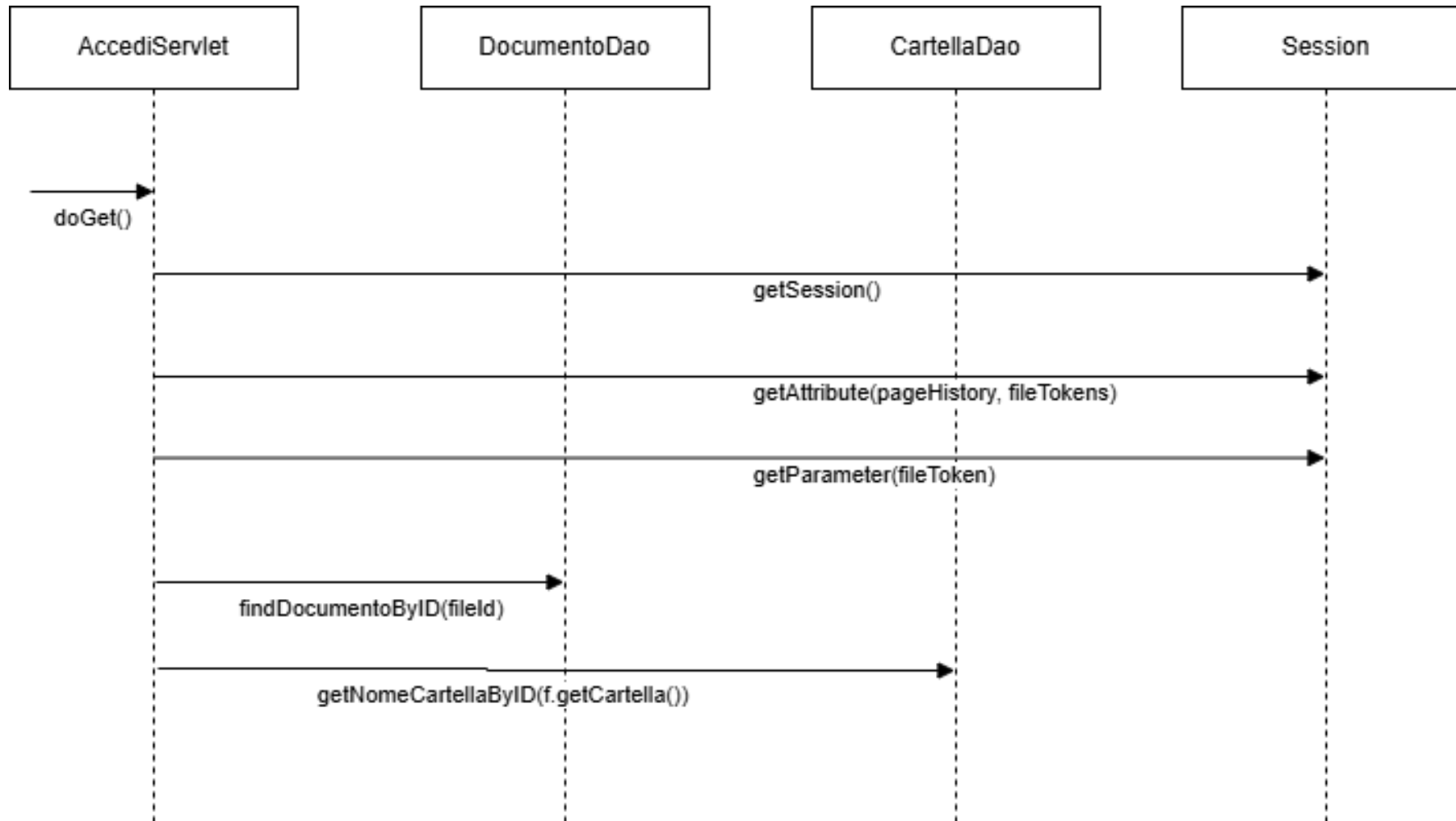
- ContenutiStyle.css
- FolderStyle.css
- highlightOnClick.css

- NewFileServlet.java
- NewRootFolderServlet.java
- NewSubfolderServlet.java
- RegistrationServlet.java
- SpostaServlet.java VIEW
- login.html
- registration.html

# IFML

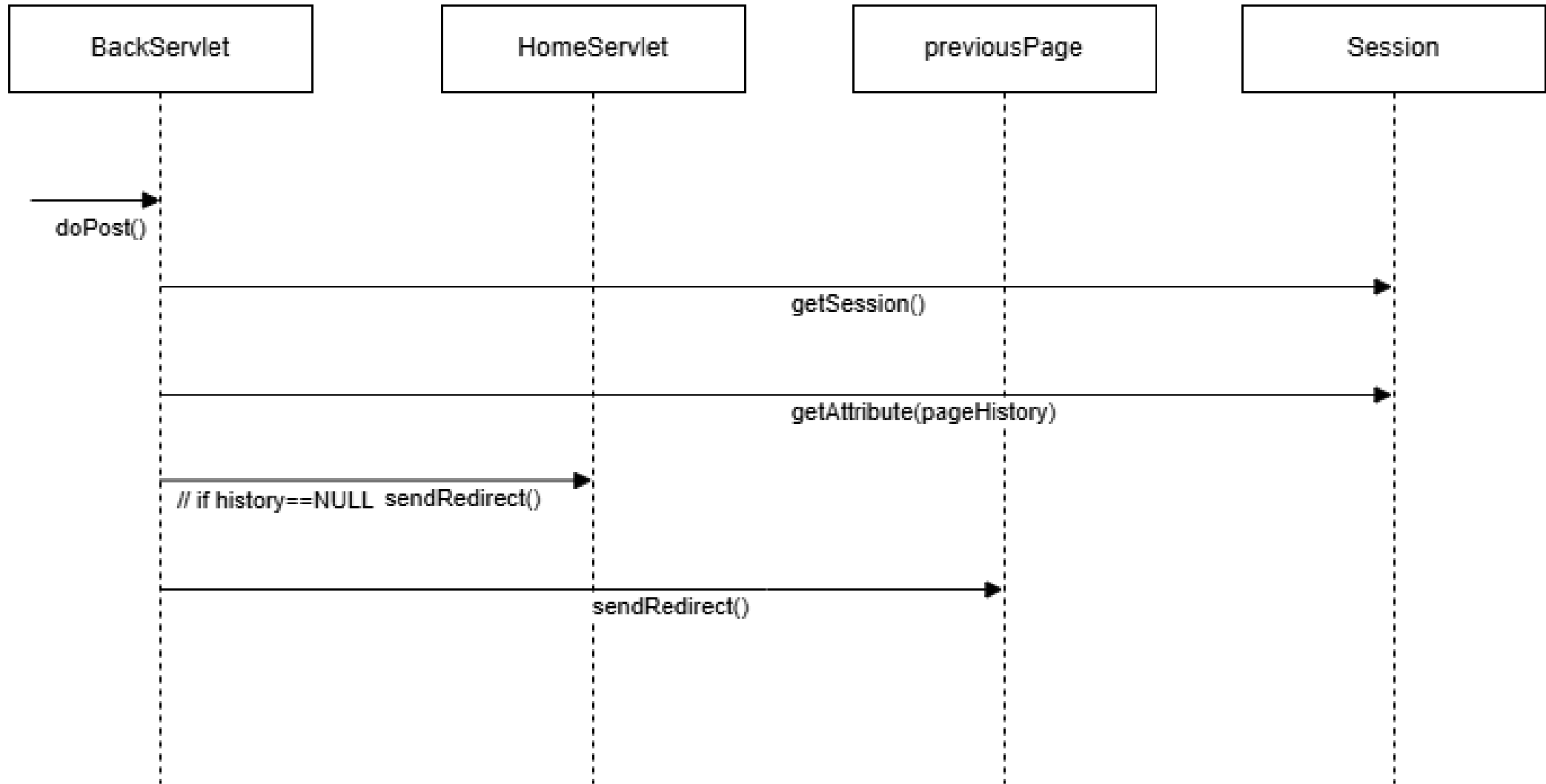


# ACCEDERE AI DETTAGLI DI UN FILE

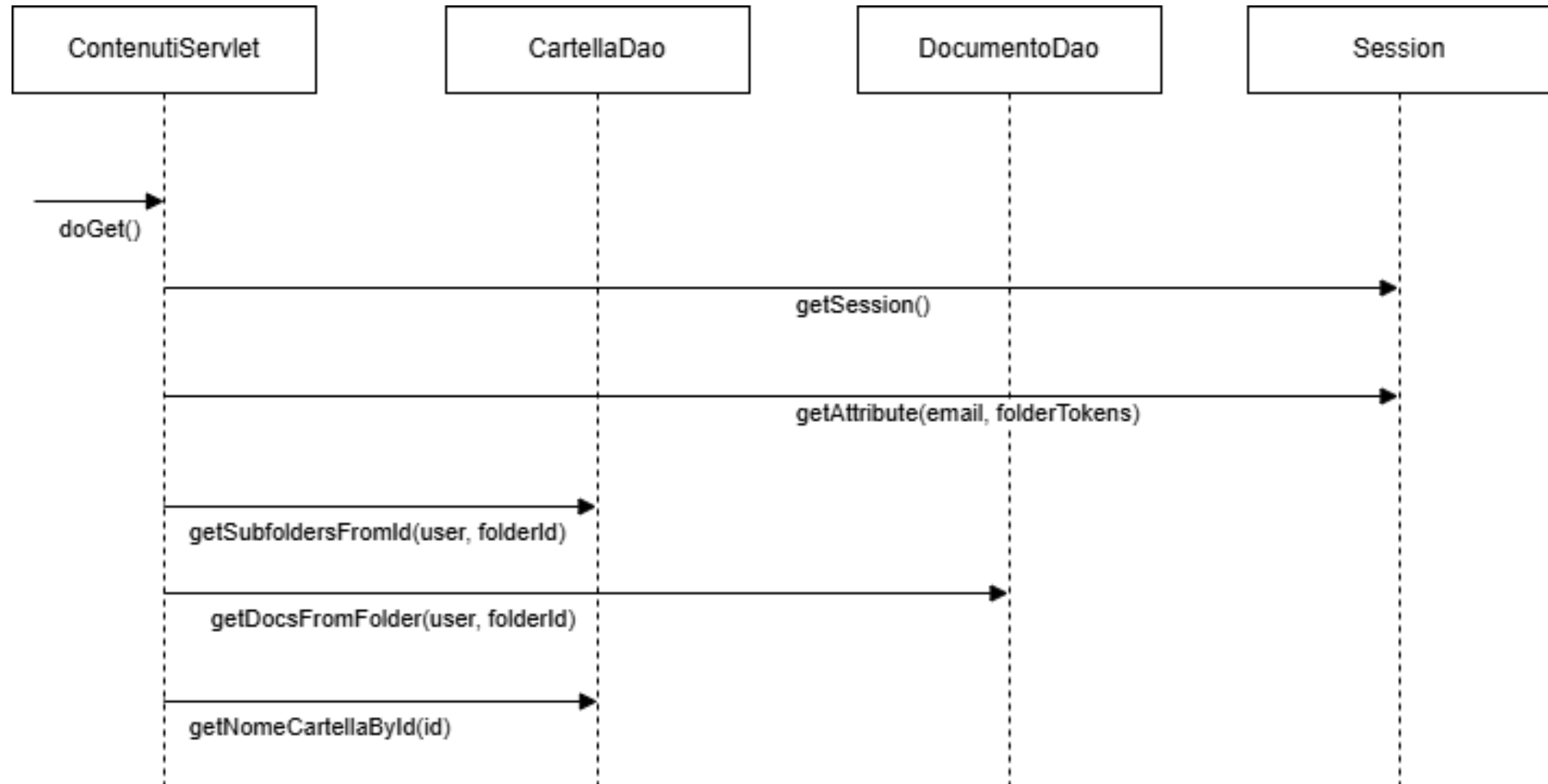




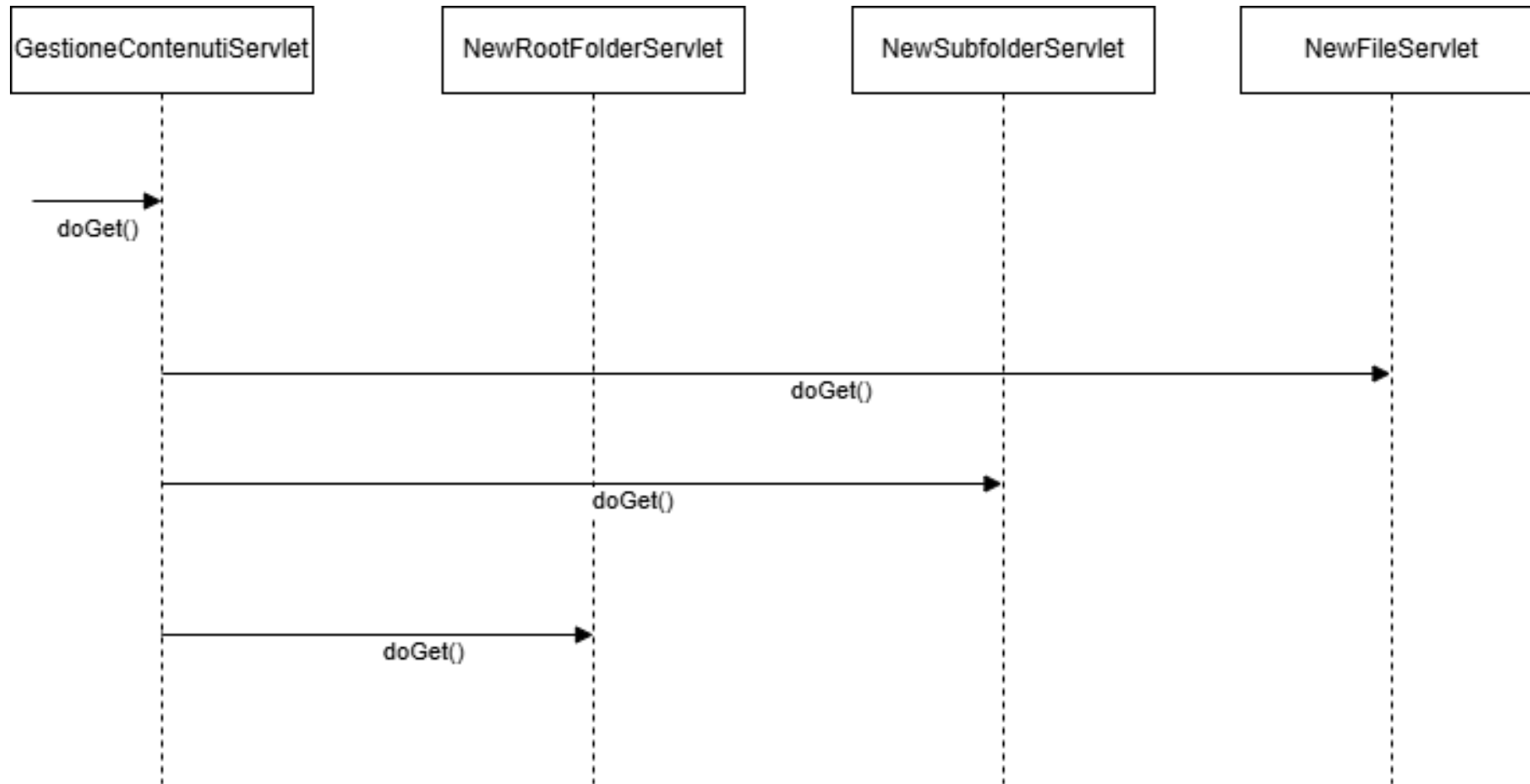
# TORNARE ALLA PAGINA PRECEDENTE



# ACCEDERE AI CONTENUTI DI UNA CARTELLA

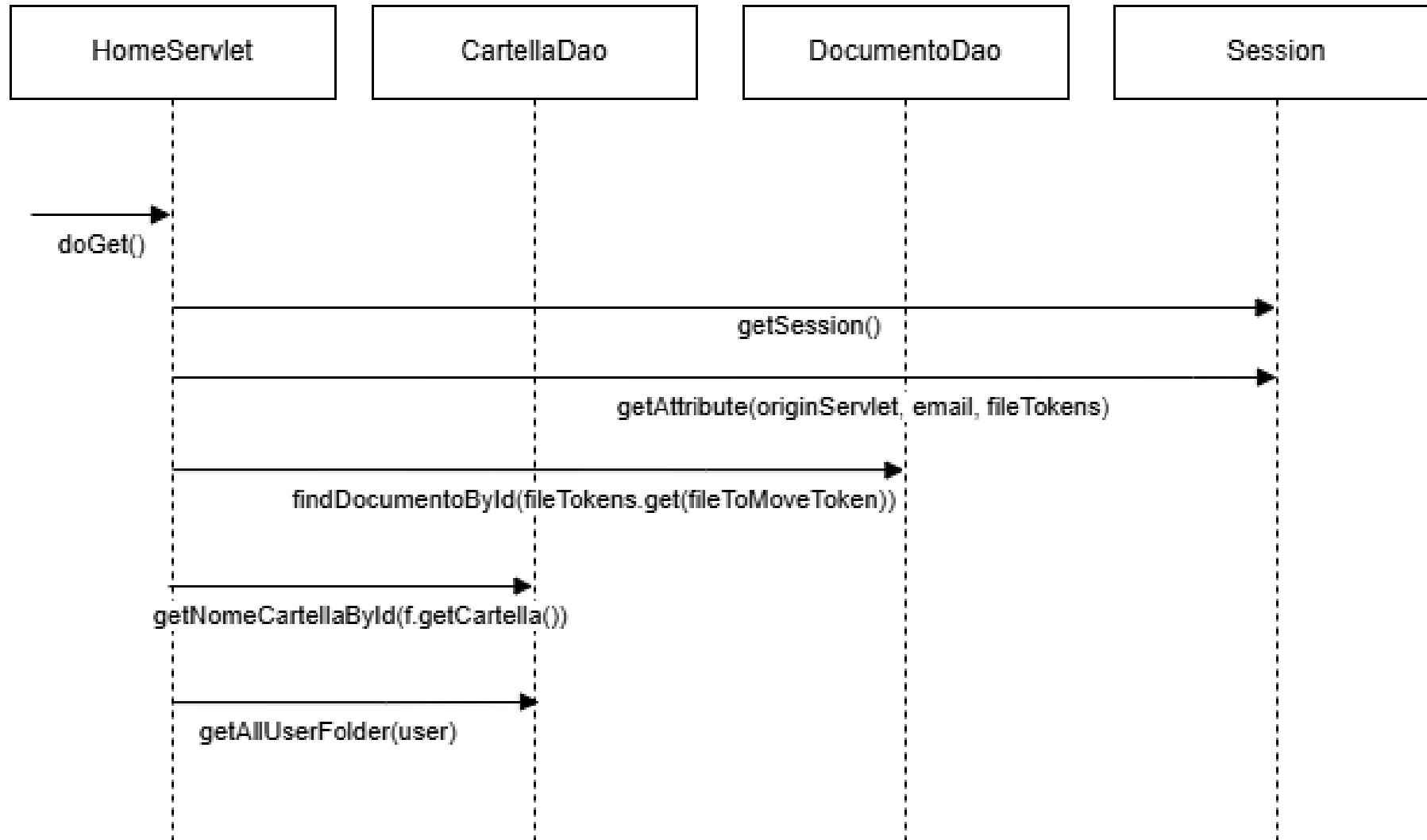


# ACCEDERE ALLA PAGINA DI GESTIONE CONTENUTI

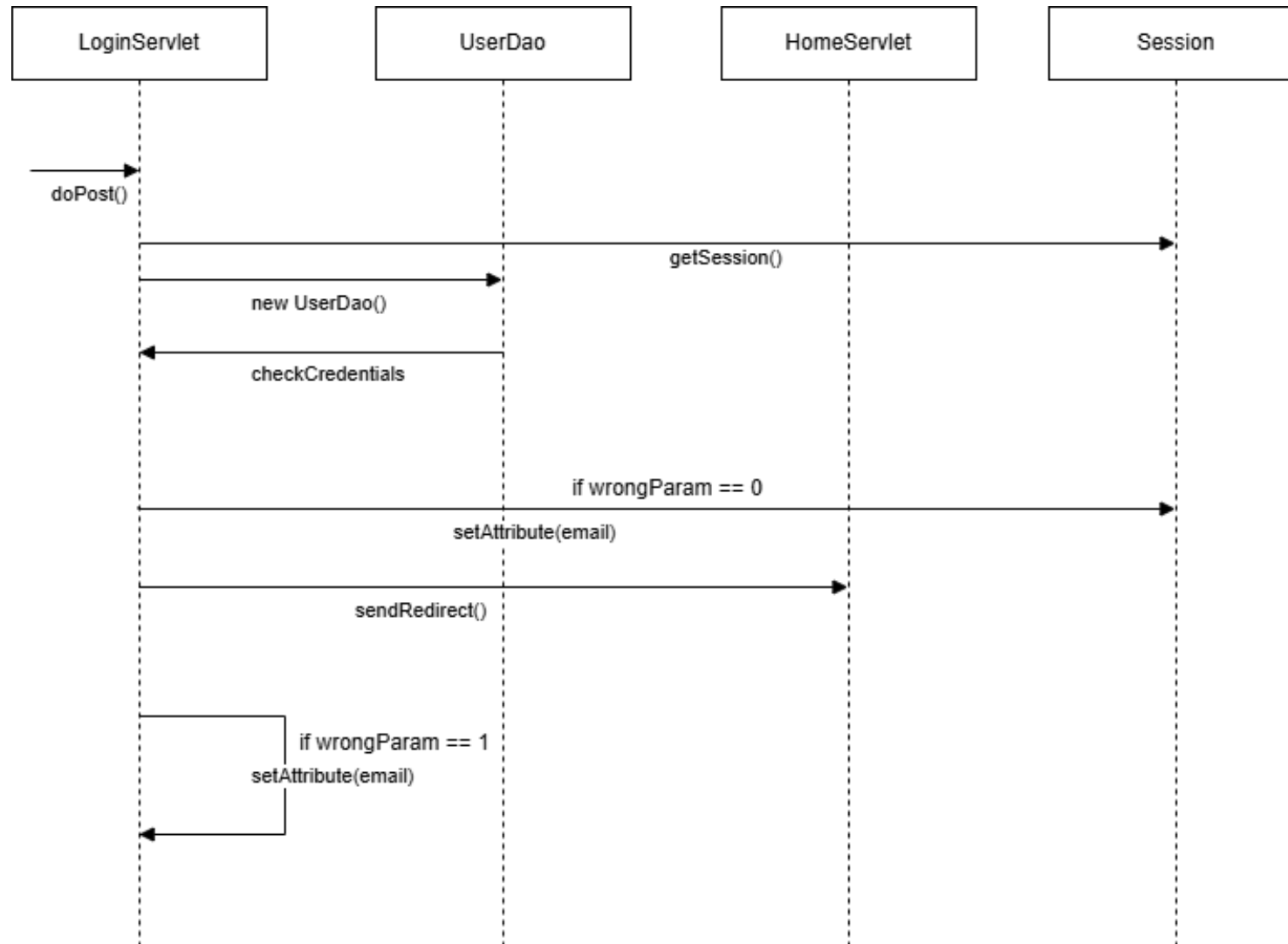




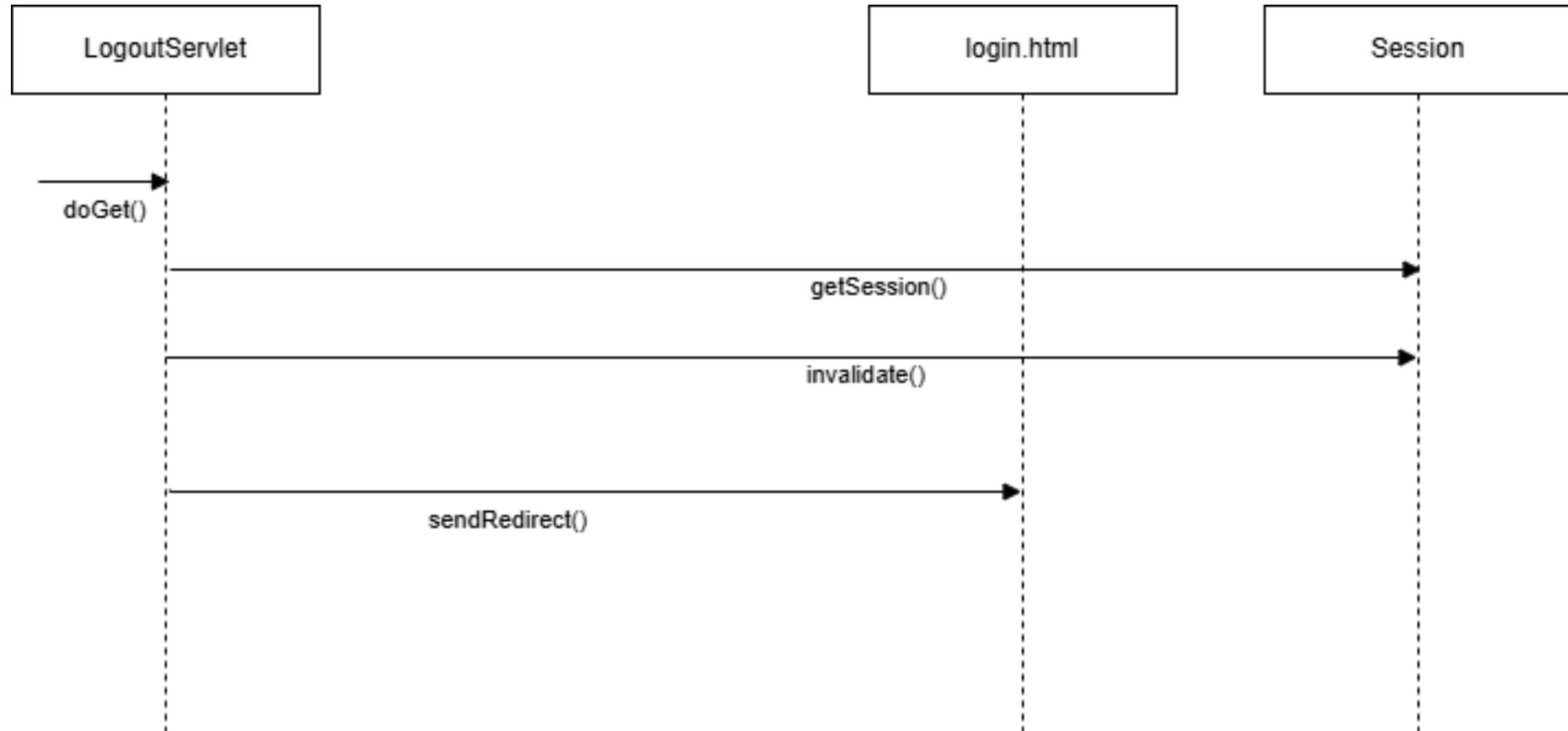
# APERTURA DELLA HOME PAGE



# ACCESSO DI UN UTENTE

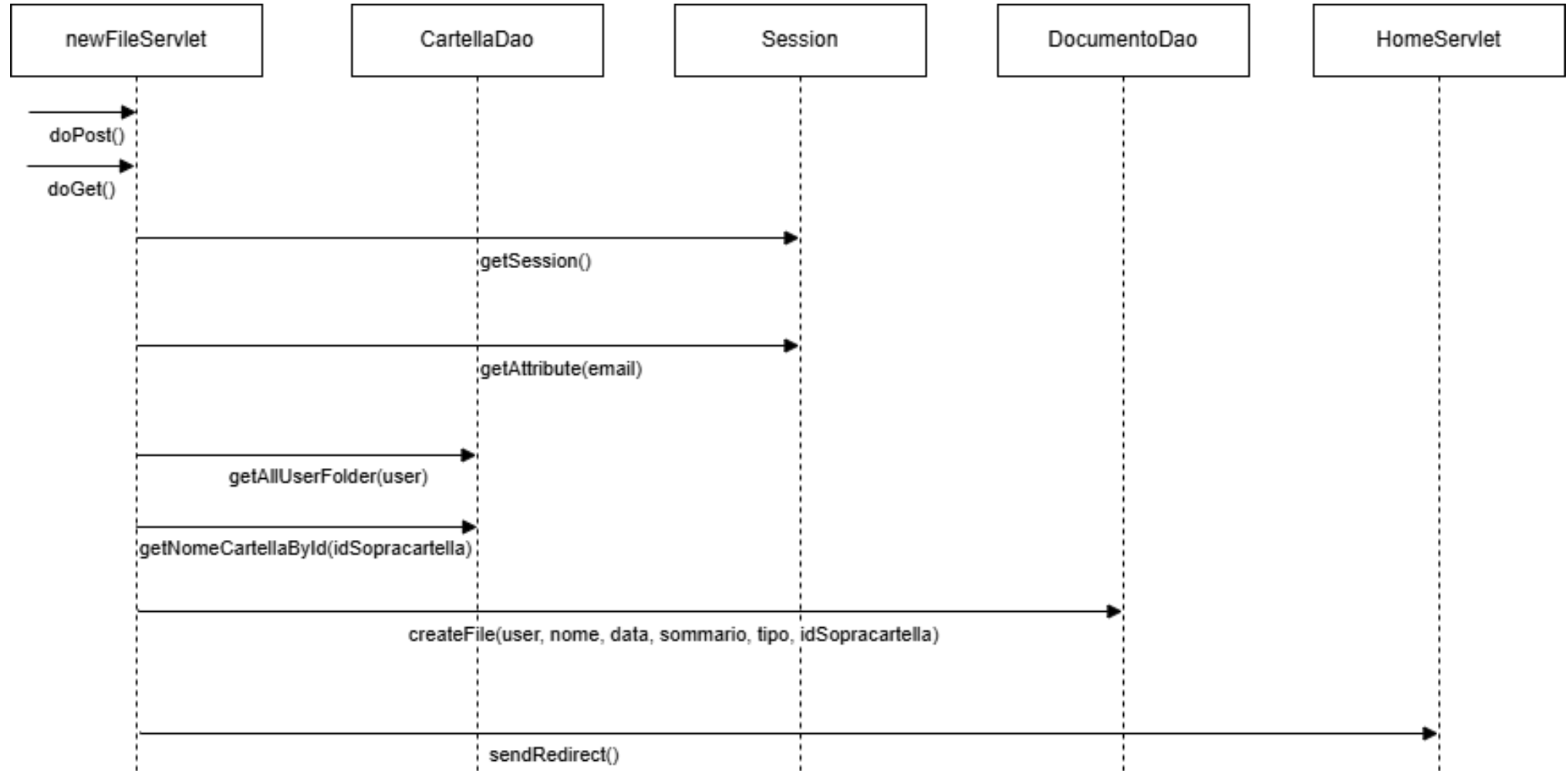


# LOGOUT DI UN UTENTE

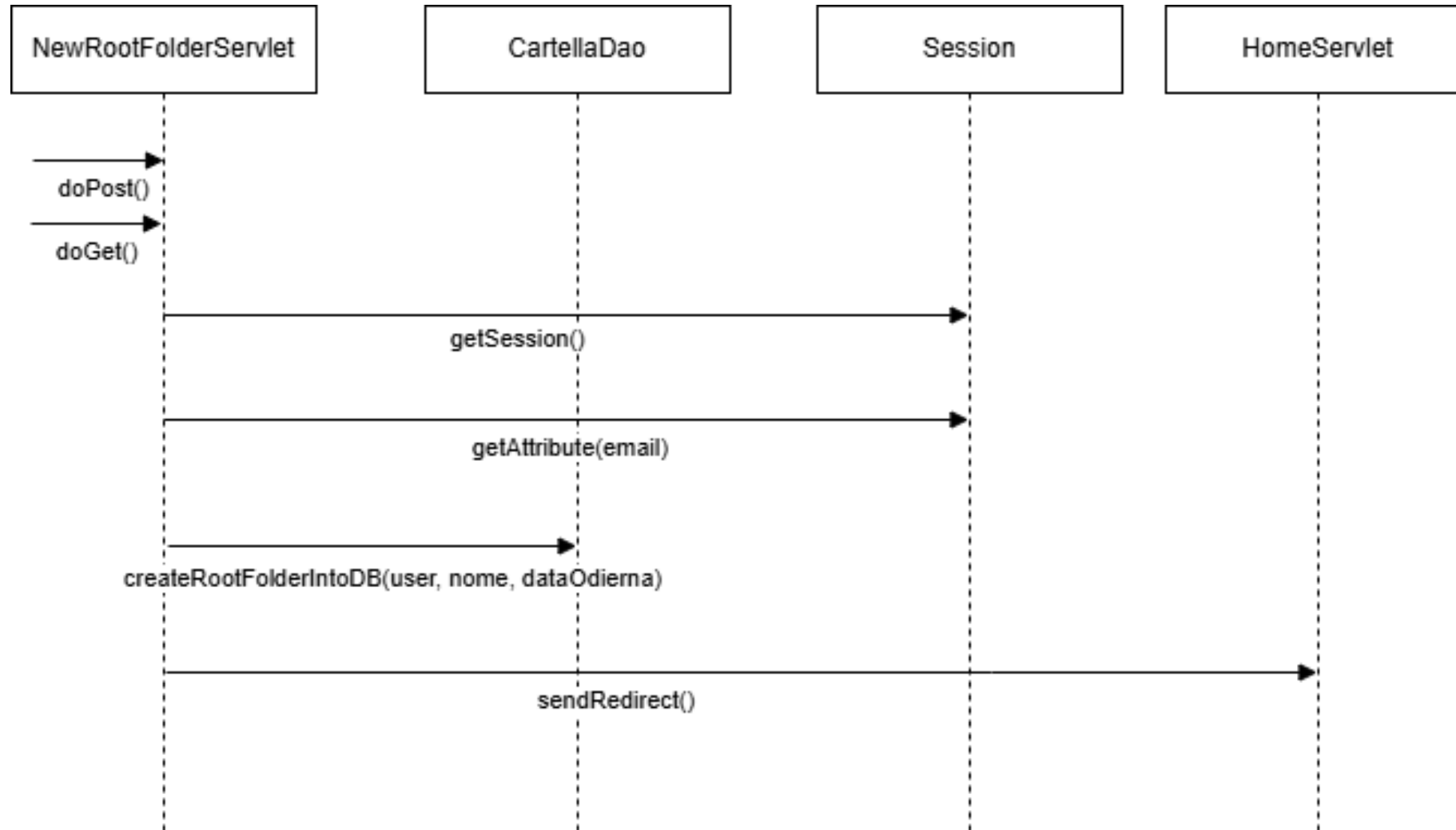




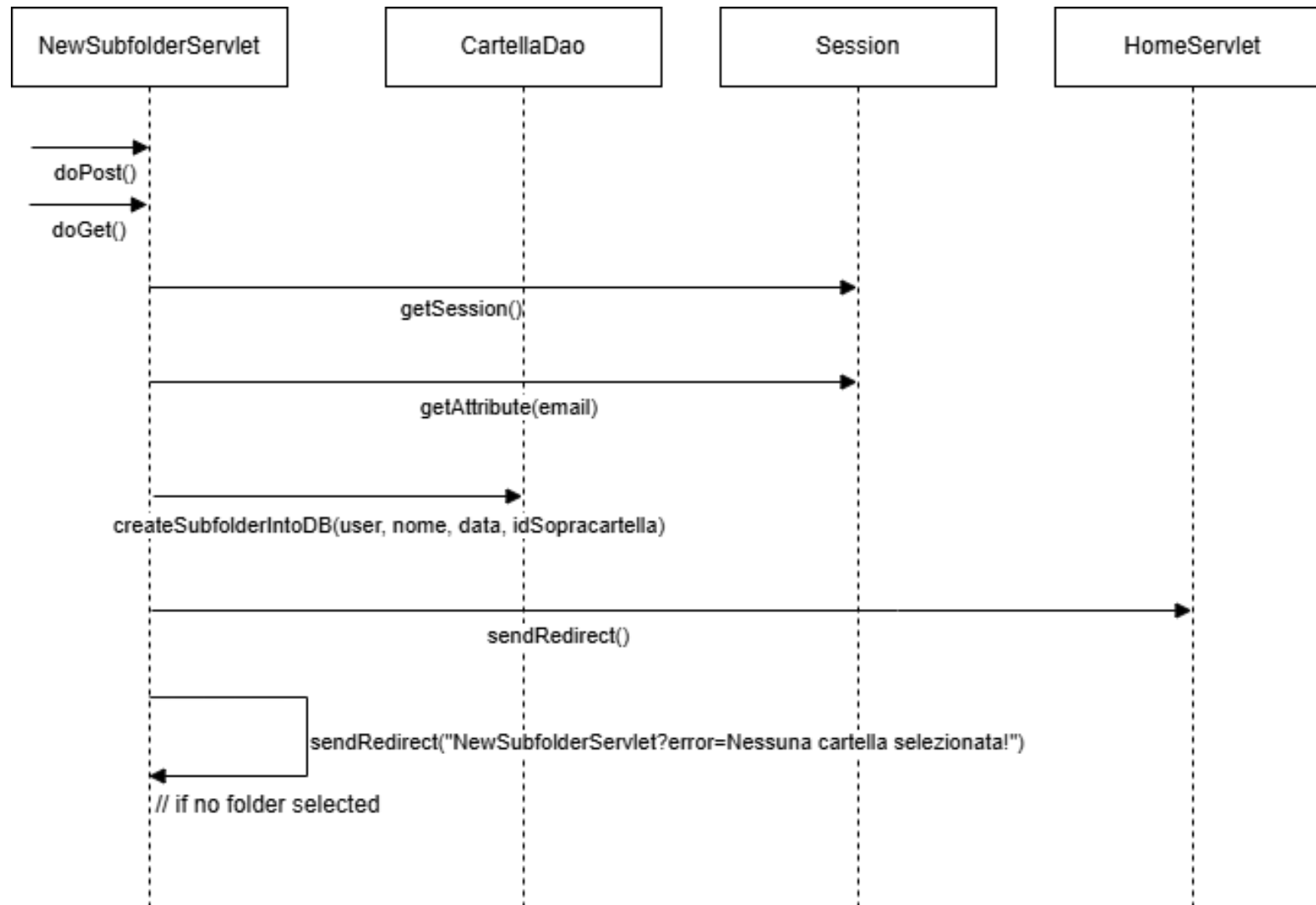
# CREAZIONE DI UN NUOVO FILE



# CREAZIONE DI UNA NUOVA CARTELLA ROOT

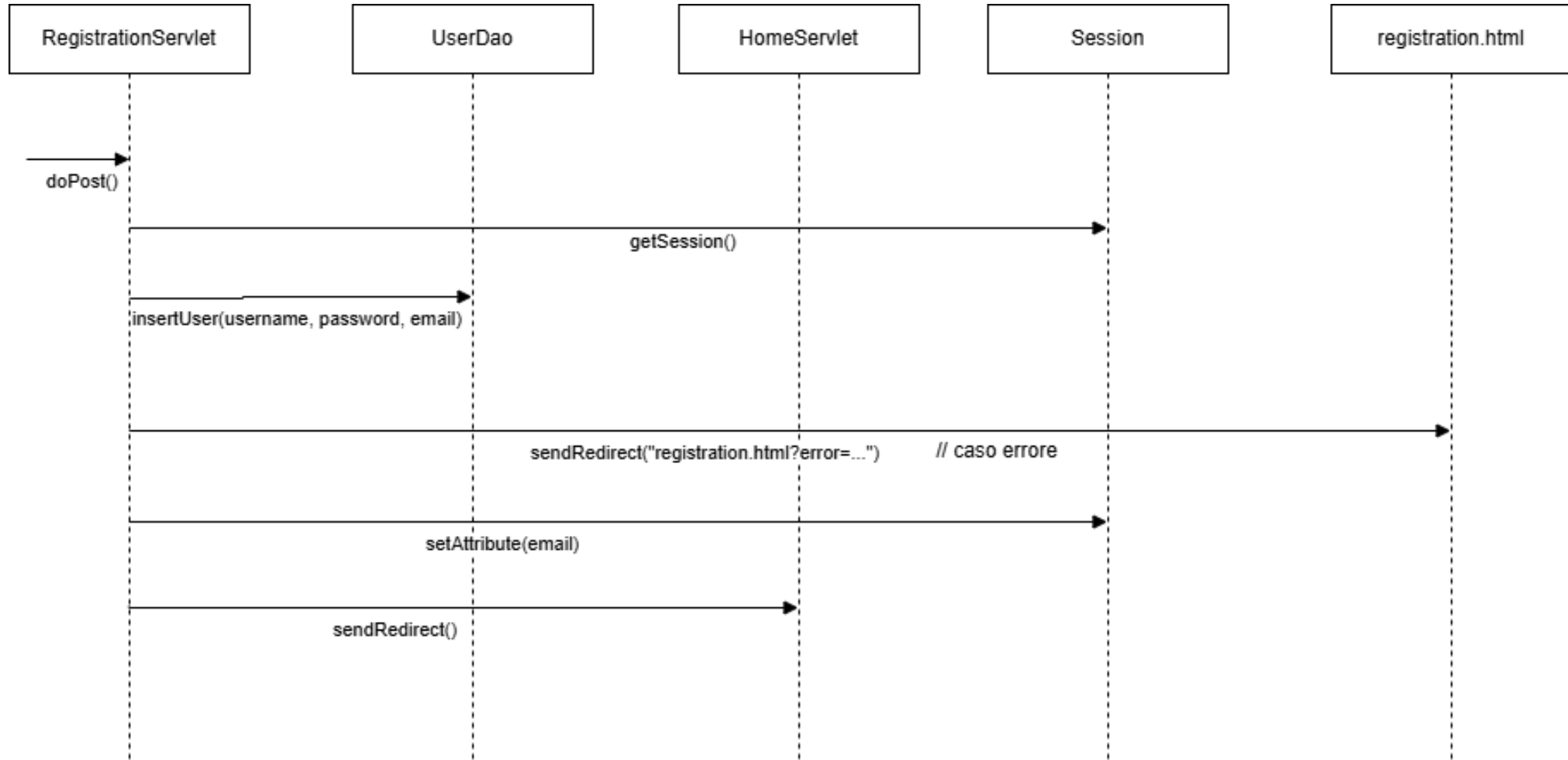


# CREAZIONE DI UNA NUOVA CARTELLA INTERNA

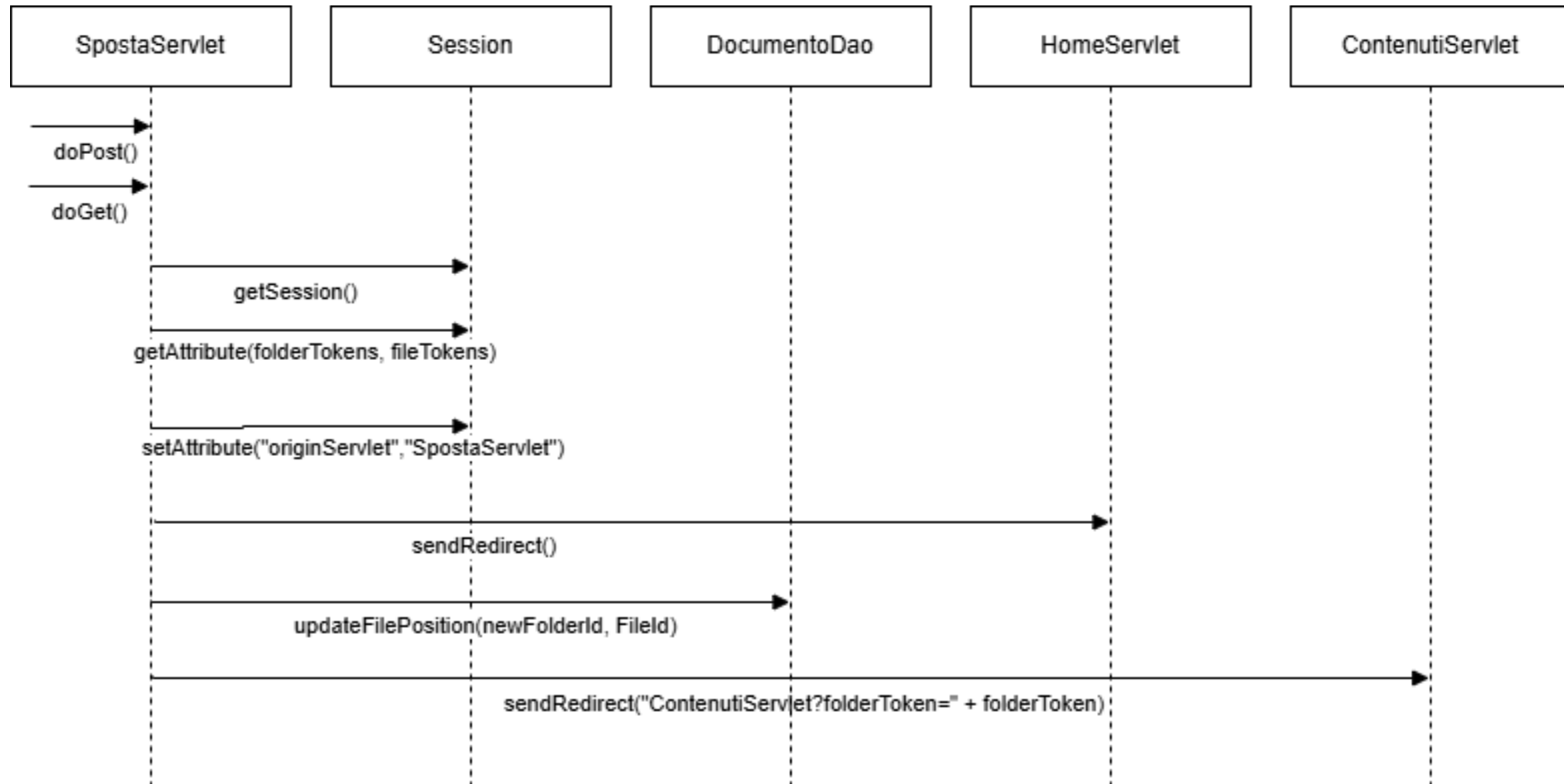




# REGISTRAZIONE DI UN NUOVO UTENTE



# SPOSTAMENTO DI UN FILE



# RIA

```

96     <h1>One more for good measure.</h1>
97     <p>Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.
98     .</p>
99     <p><a class="btn btn-lg btn-primary" href="#" role="button">Browse gallery</a>
100   </div>
101 </div>
102 <a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev">
103   <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
104 </a>
105 <a class="right carousel-control" href="#myCarousel" role="button" data-slide="next">
106   <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
107 </a>
108 </div><!-- /.carousel -->
109
110 <!--Featured Content Section-->
111
112 <div class="container">
113   <div class="row">
114     <div class="col-md-4"></div>
115     <div class="col-md-4"></div>
116     <div class="col-md-4"></div>
117   </div>
118 </div>

```

# ANALISI DEI DATI

## Versione con 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)**.

**Pagine**

**Eventi**

**Componenti di vista**

**Azioni**



# COMPONENTI

## BEANS

- File.java
- Folder.java

## DAO

- CartellaDao.java
- DocumentoDao.java
- UserDao.java

## CONTROLLERS

- AccediServlet.java
- DeleteServlet.java
- HomeServlet.java
- LoginServlet.java
- LogoutServlet.java
- NewFileServlet.java
- NewRootFolderServlet.java
- NewSubfolderServlet.java
- RegistrationServlet.java
- SpostaServlet.java

## VIEW

- login.html
- registration.html

## JAVASCRIPT COMPONENTS

- homeManager.js
- loginManager.js
- registrationManager.js

## CSS COMPONENTS

- ContenutiStyle.css
- Home.css

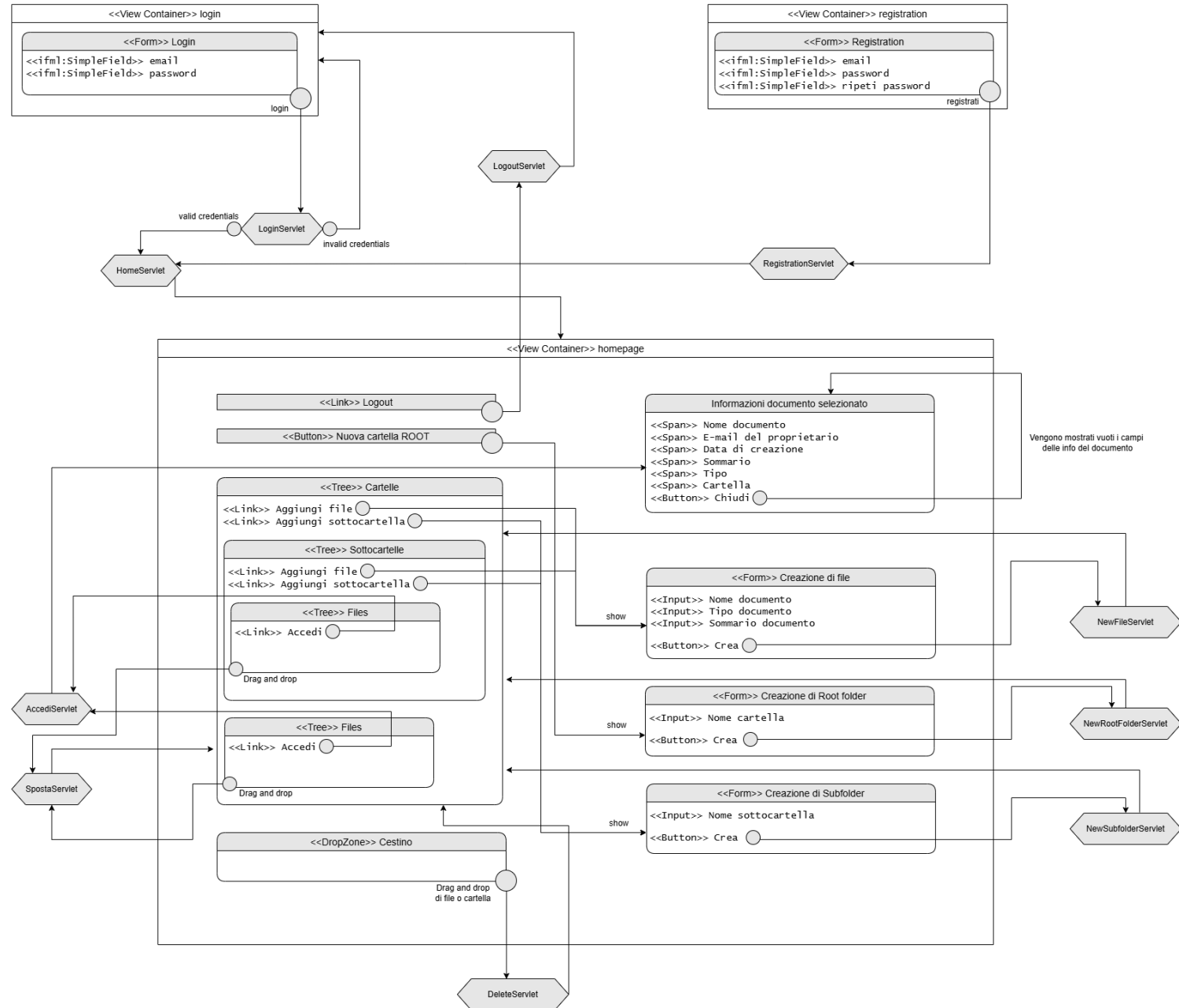
# EVENTI E AZIONI

CLIENT SIDE			SERVER SIDE		
EVENTI	AZIONI	CONTROLLER	EVENTI	AZIONI	CONTROLLER
login utente	controllo del formato corretto email, controllo campi non vuoti, invio dei dati al server	loginManager.js	POST (email, password)	controllo della correttezza dei campi (tramite DB)	LoginServlet ⇒ UserDao
registrazione nuovo utente	controllo del formato corretto email, controllo campi non vuoti, controllo uguaglianza campi password e ripeti password, controllo vincoli password, invio dei dati al server	registrationManager.js	POST (username, email, password, password_confirm)	stessi controlli del client + controllo dell'unicità dei campi username ed email	RegistrationServlet ⇒ UserDao
creazione di una cartella ROOT	controllo che il campo "nome cartella" non sia vuoto, invio dati al server	homeManager.js	POST (nomecartella)	controllo che i campi non siano nulli, inserisco la cartella nel DB	NewRootFolderServlet ⇒ CartellaDao
creazione di una sottocartella	controllo che il campo "nome cartella" non sia vuoto, invio dati al server	homeManager.js	POST (nomecartella, IDSopracartella)	controllo che i campi non siano nulli, inserisco la cartella nel DB	NewSubfolderServlet ⇒ CartellaDao
creazione di un file	controllo che i campi "nome file" e "tipo file" non siano vuoti, invio dati al server	homeManager.js	POST (nomefile, tipofile, sommariofile, IDSopracartella)	controllo che i campi non siano nulli, inserisco il file nel DB	NewFileServlet ⇒ DocumentoDao
spostamento di un file (drag and drop)	controllo che l'elemento spostato sia un singolo file e non una cartella	homeManager.js	POST (ID file, oldFolder, newFolder)	spostamento del file dal DB	SpostaServlet ⇒ DocumentoDao

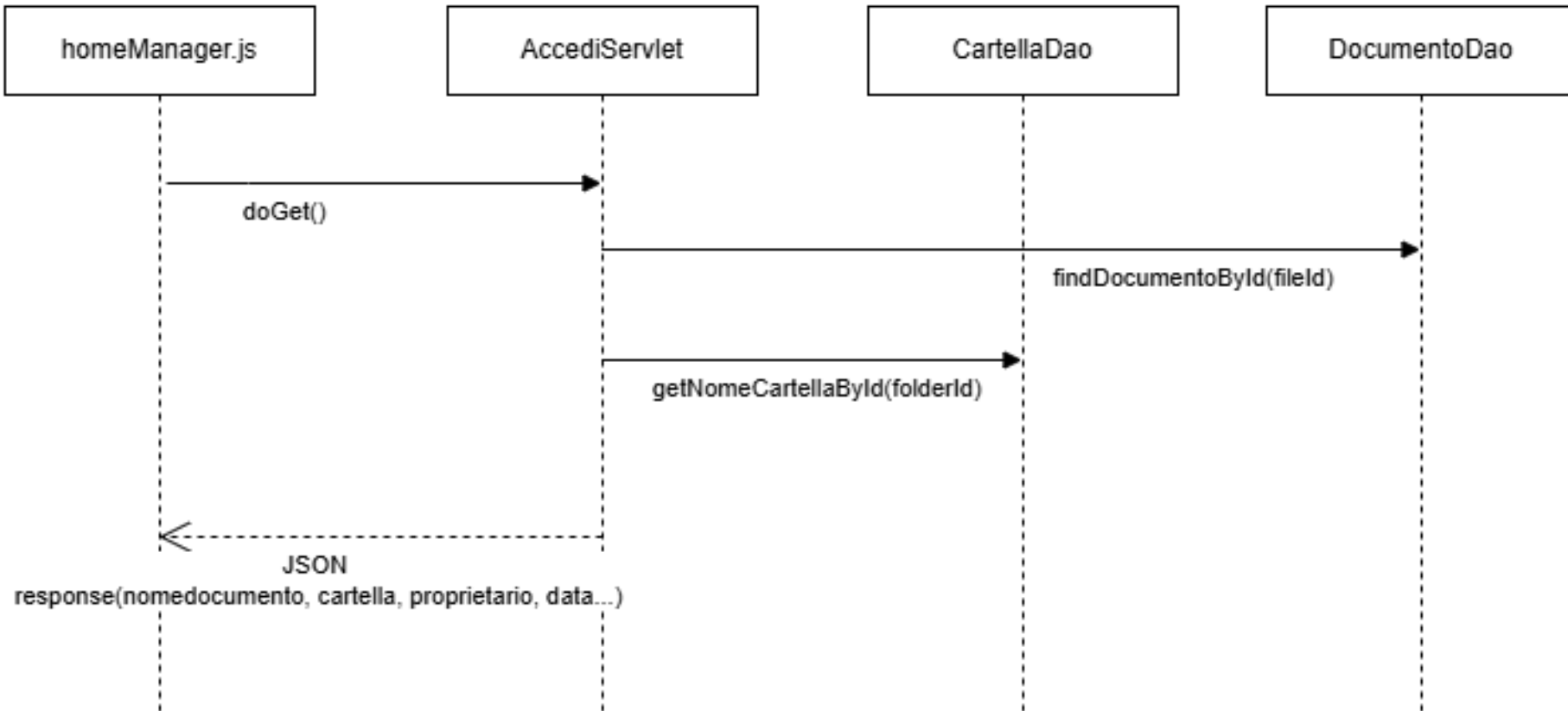
# EVENTI E AZIONI

eliminazione di un elemento (file o cartella)	controllo che l'elemento trascinato sia valido, mostro alert di eliminazione, invio dati al server	homeManager.js	DELETE (IdToDelete)	eliminazione del file o della cartella dal DB	DeleteServlet ⇒ DocumentoDao CartellaDao
accesso ad un file esistente all'interno di una cartella	invio al server ID del file di cui voglio le informazioni	homeManager.js	GET (IDfile)	raccolte informazioni dal DB e mandate al client (che le mostra nello span pre impostato)	AccediServlet ⇒ DocumentoDao CartellaDao
operazione di chiudi (pulitura dei campi di info file)	i campi dello span del file vengono puliti, e vengono cancellati	homeManager.js	-	-	-
logout utente	invio al server	-	GET	la sessione viene invalidata e l'utente viene reindirizzato alla pagina di login.html	LogoutServlet ⇒ DocumentoDao CartellaDao

# IFML

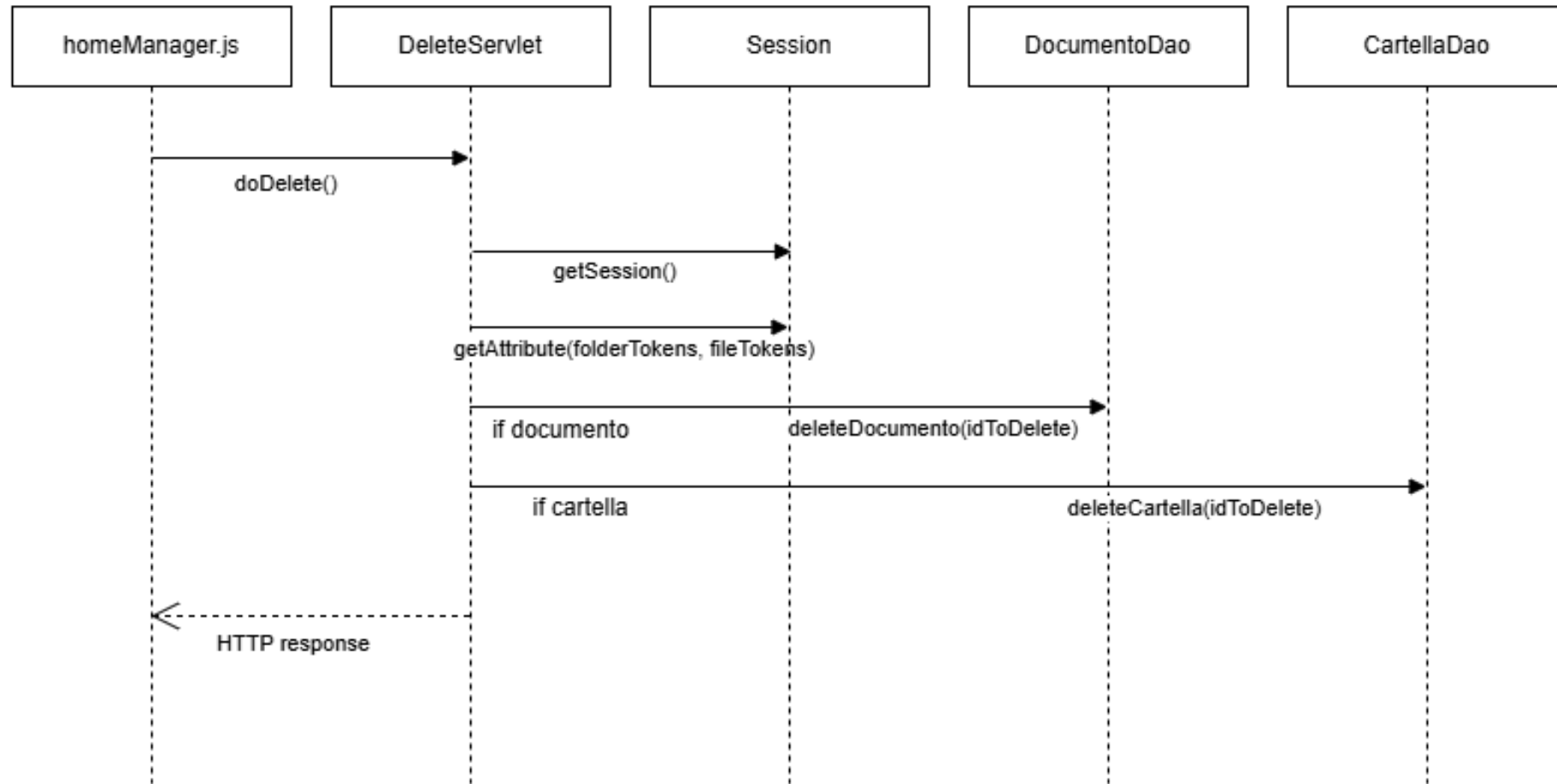


# ACCEDERE AI DETTAGLI DI UN FILE

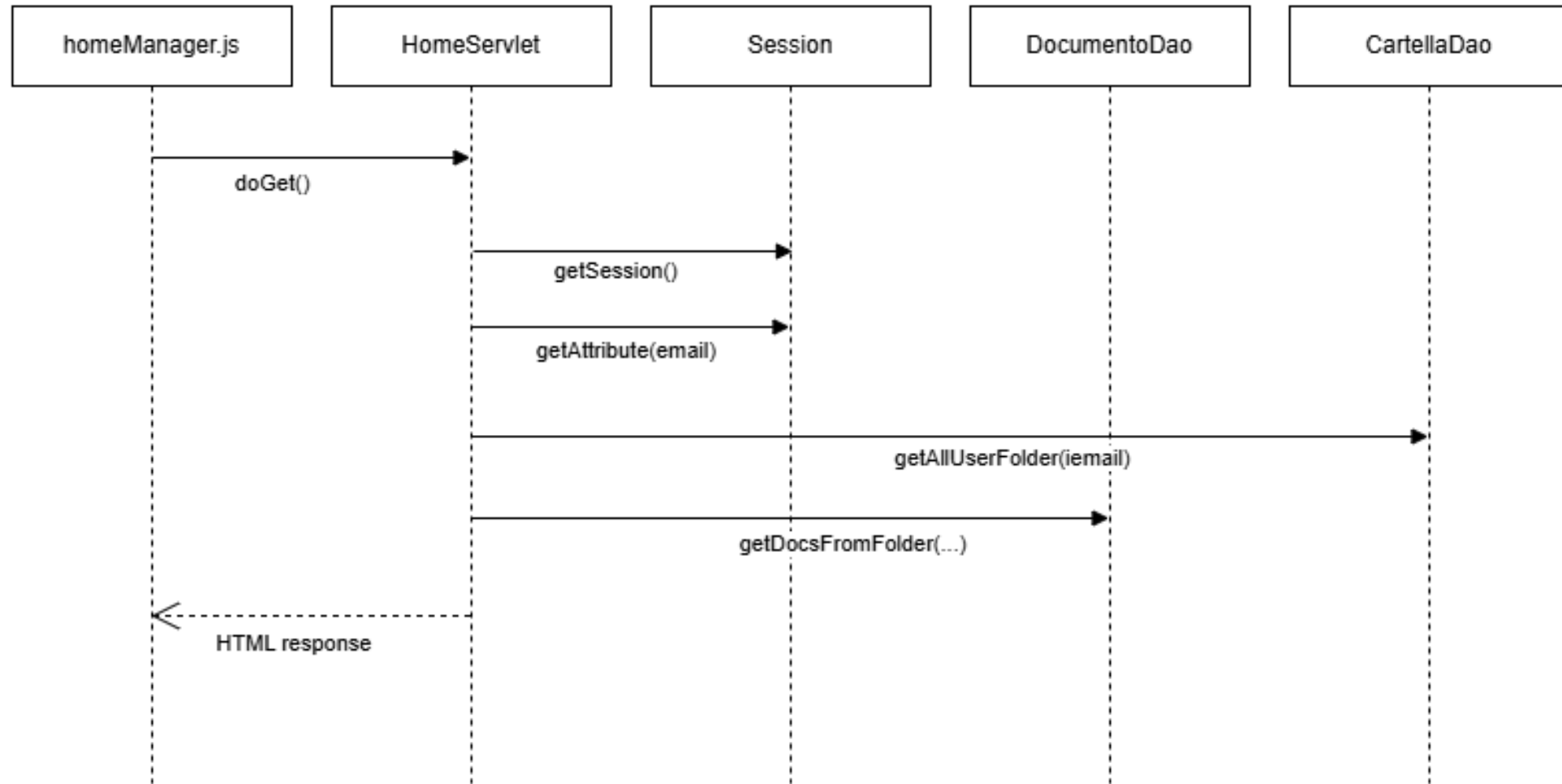




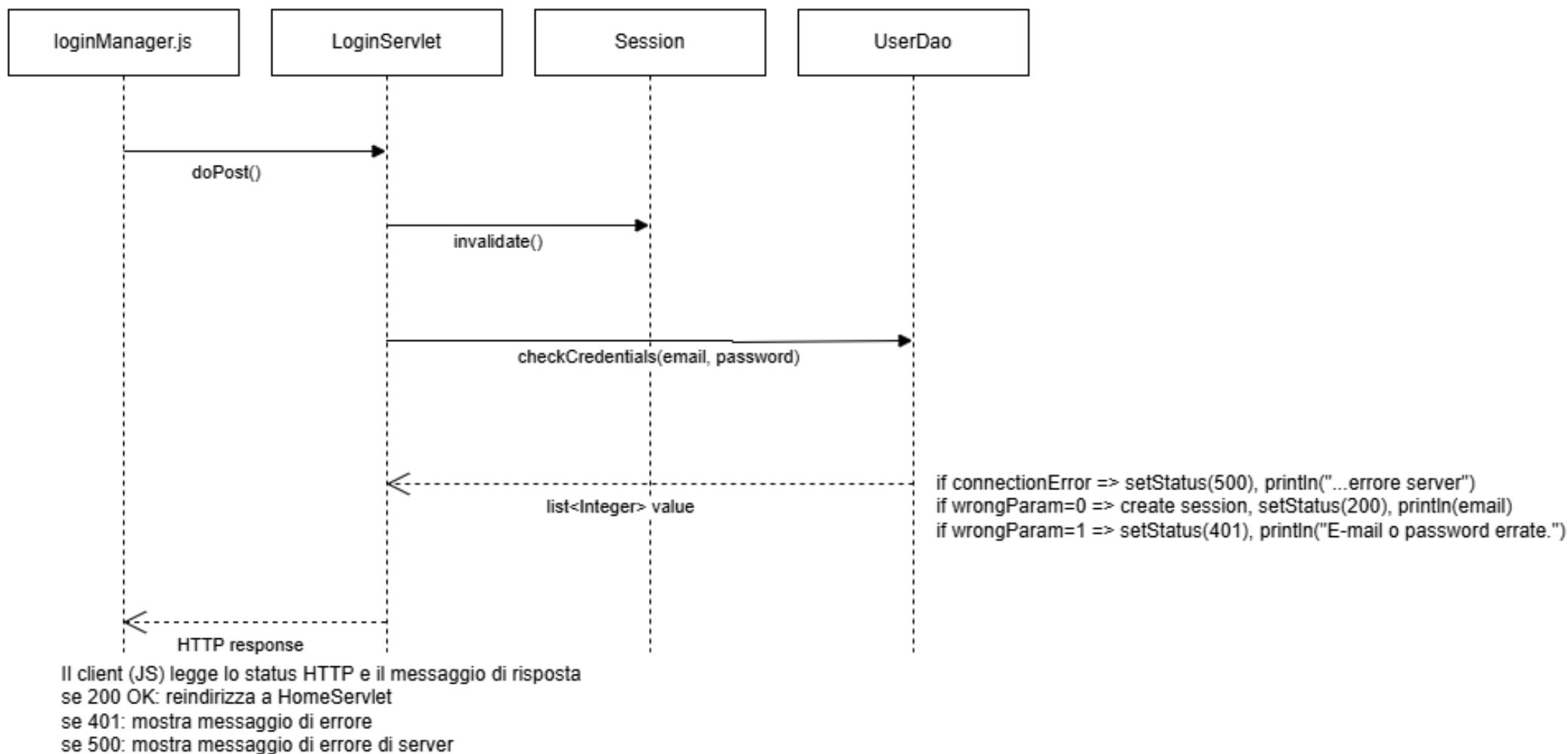
# CANCELLAZIONE DI UN FILE O DI UNA CARTELLA



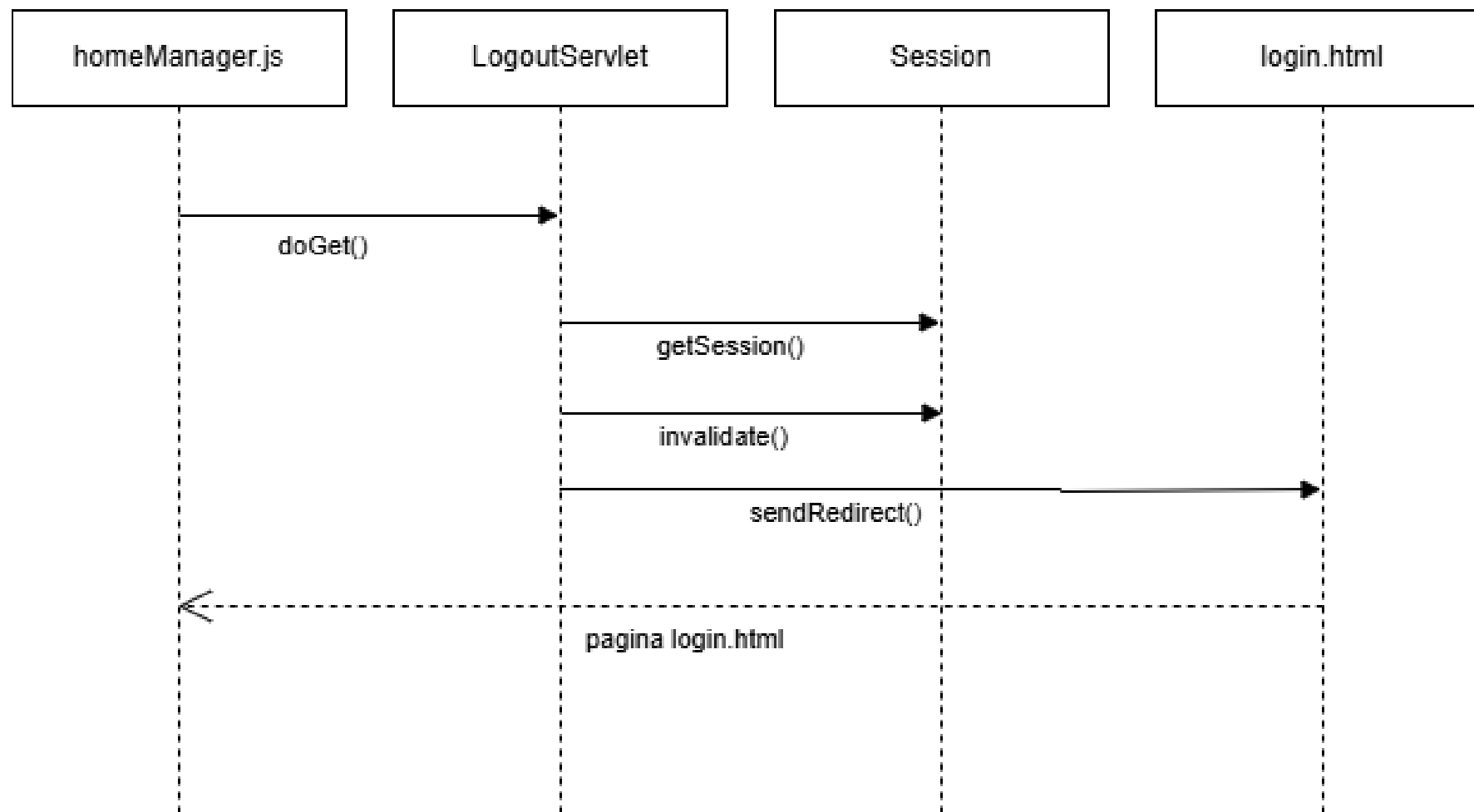
# ACCESSO ALLA HOME PAGE



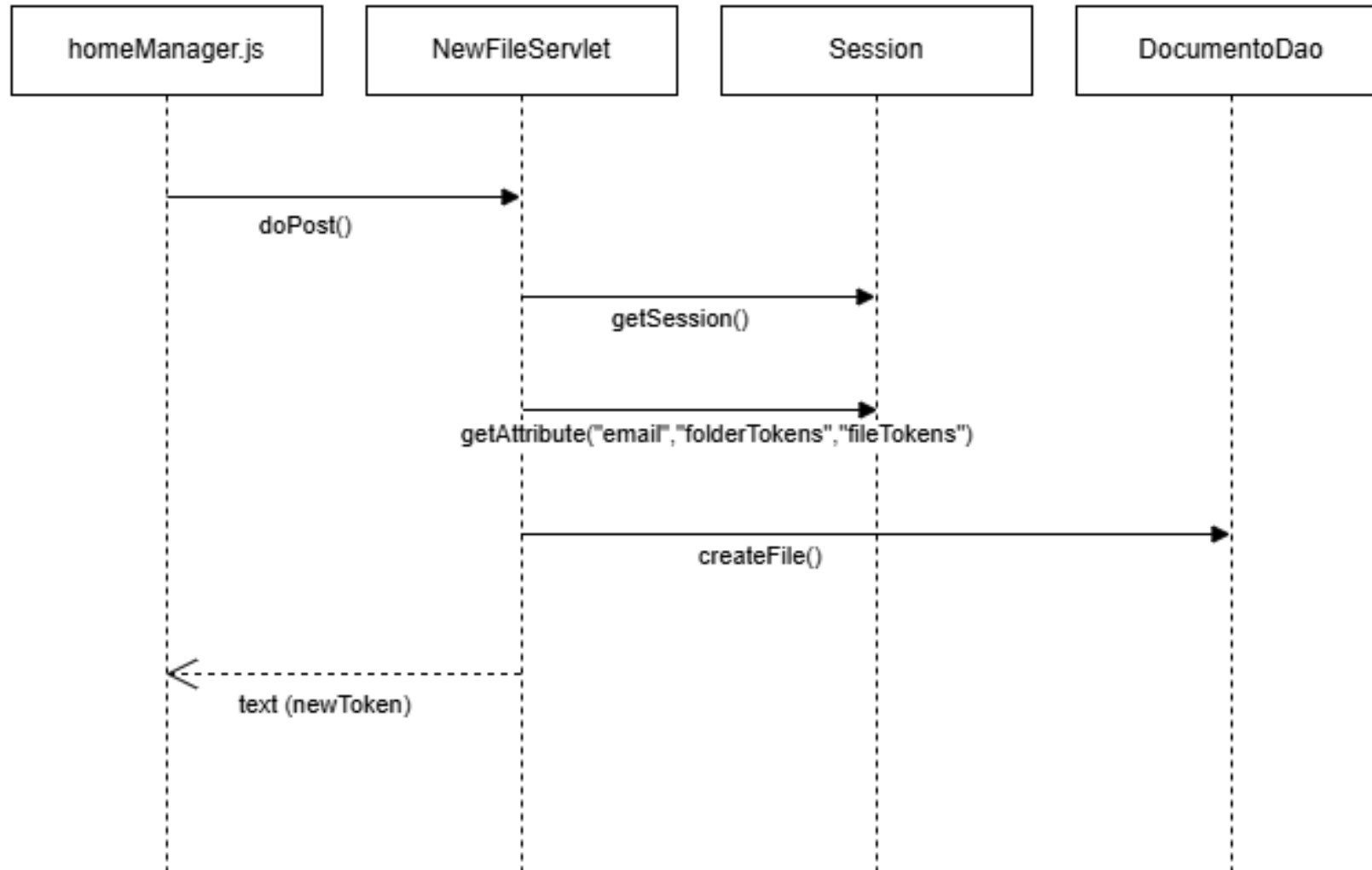
# LOGIN DI UN UTENTE



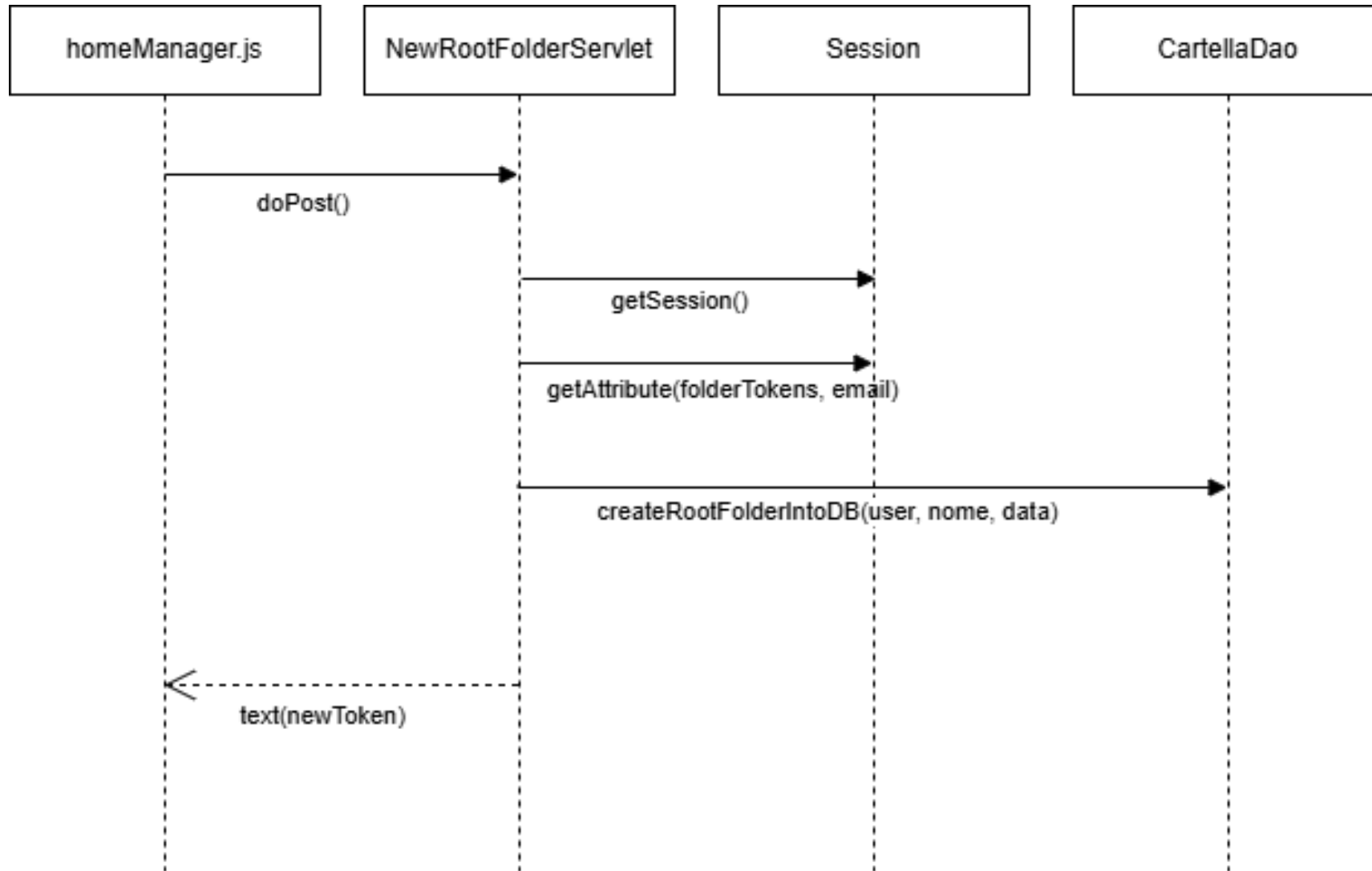
# LOGOUT DI UN UTENTE



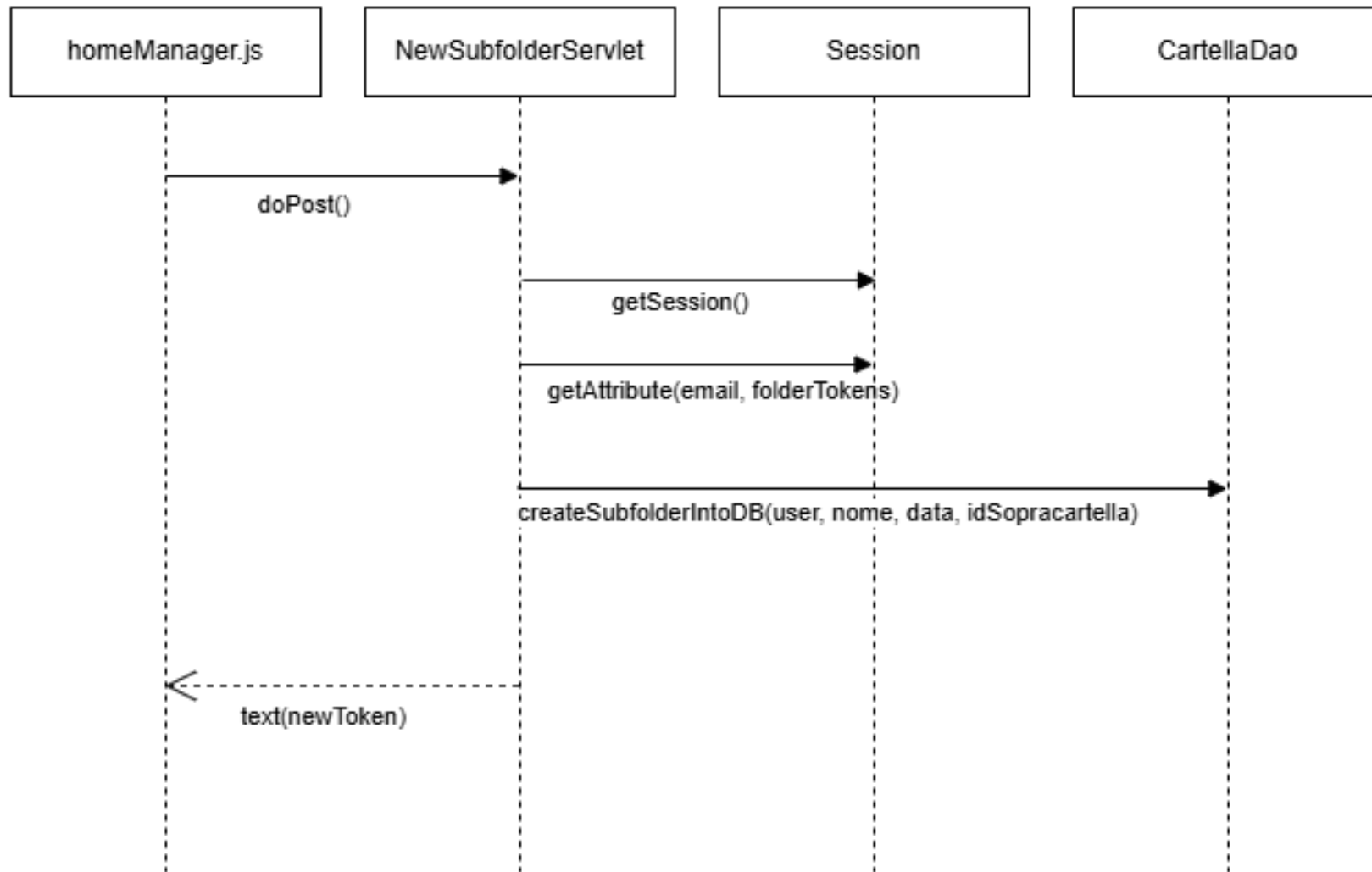
# CREAZIONE DI UN NUOVO FILE



# CREAZIONE DI UNA NUOVA CARTELLA ROOT

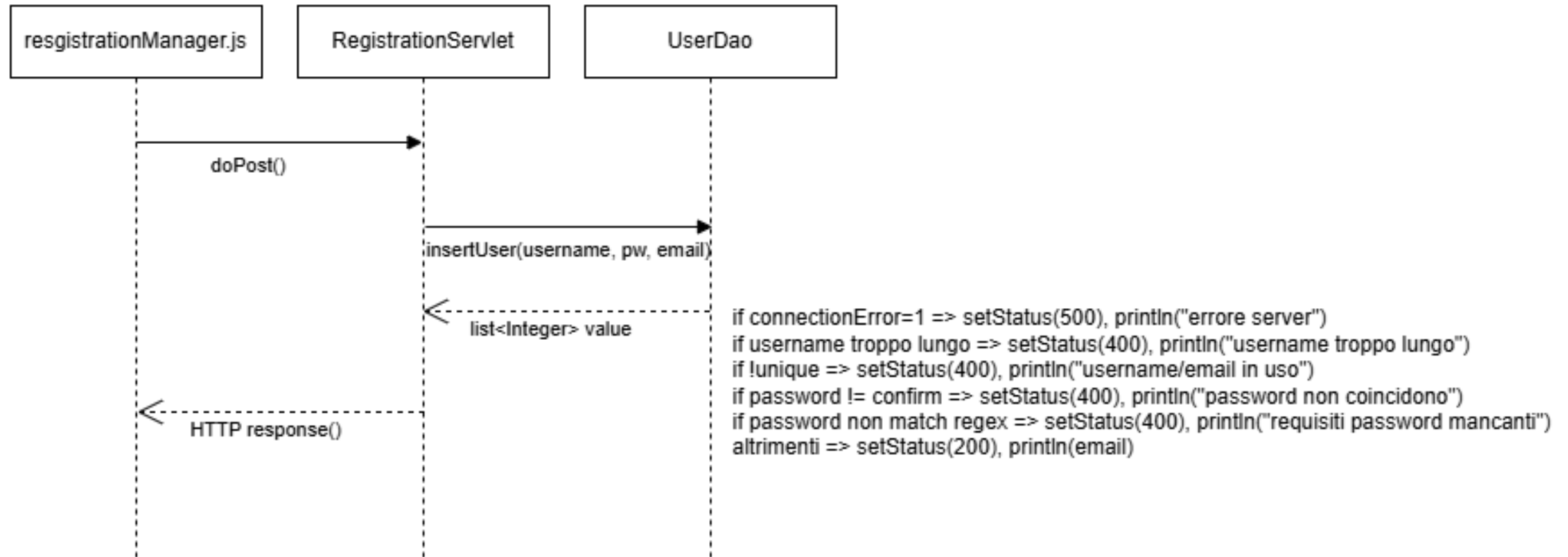


# CREAZIONE DI UNA NUOVA SOTTOCARTELLA

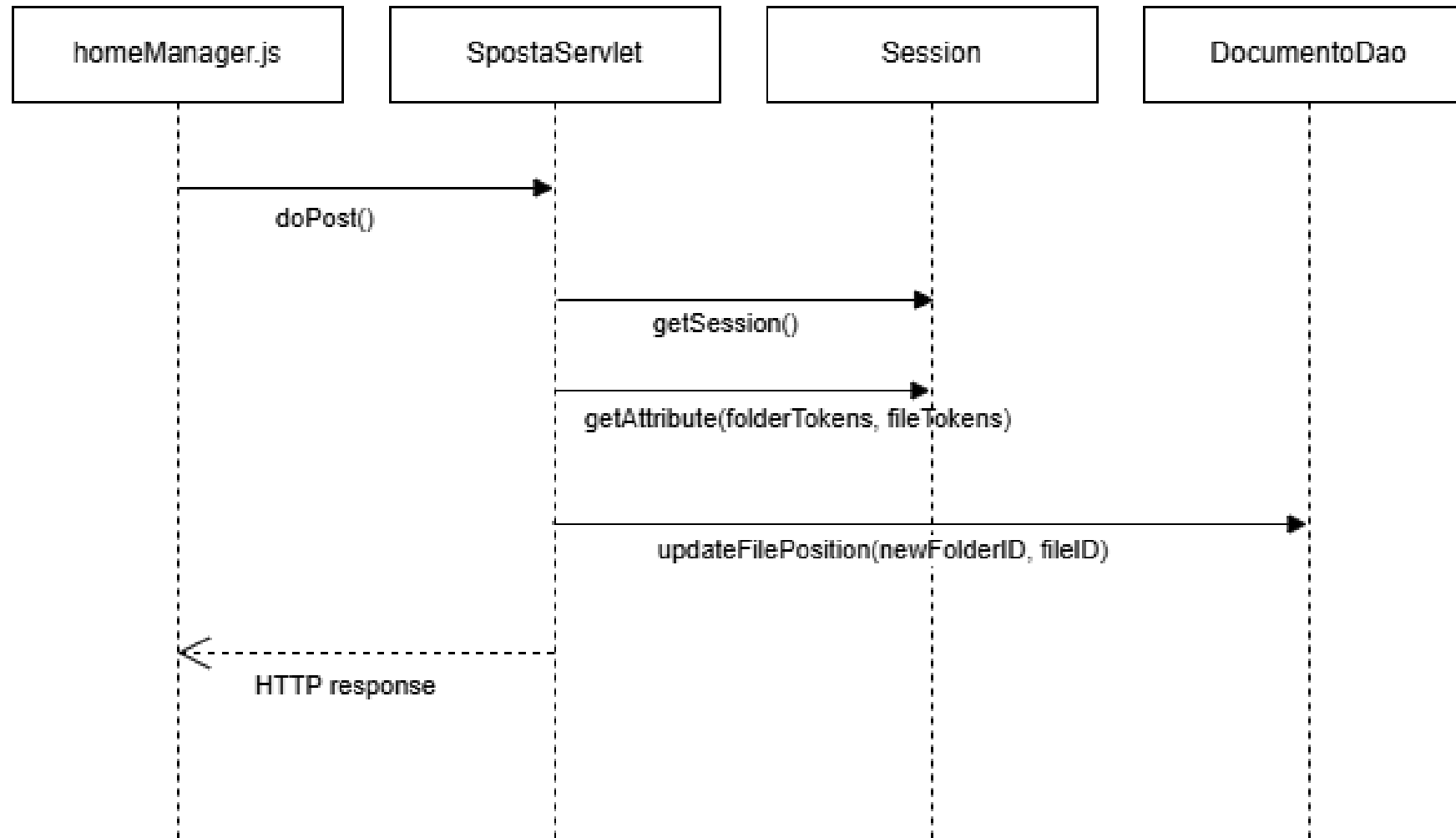




# REGISTRAZIONE DI UN NUOVO UTENTE



# SPOSTAMENTO DI UN FILE





GRAZIE PER  
L'ATTENZIONE