

Esercizio 3 - pure HTML

Luca Maestri - Virginia Longo

Esercizio 3: gestione documenti

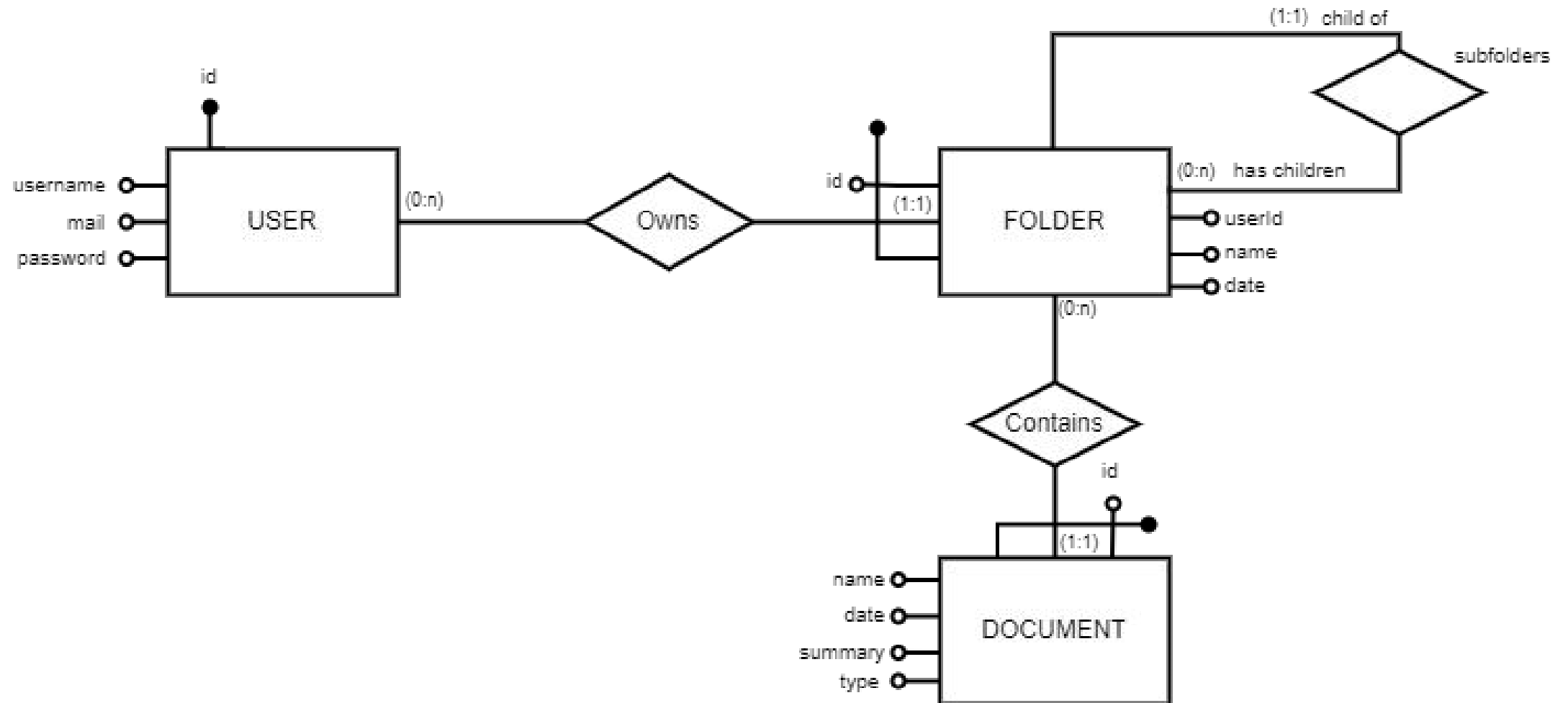
Un'applicazione web consente la gestione di cartelle, sottocartelle e documenti online. L'applicazione supporta registrazione e login di utenti mediante una pagina pubblica con opportune form. La registrazione controlla l'unicità dello username. Una cartella ha un proprietario, un nome e una data di creazione e può contenere (solo) sottocartelle. Una sottocartella può contenere (solo) dei 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 sottocartella e accedere a una pagina DOCUMENTI che mostra l'elenco dei documenti di una sottocartella. Ogni documento in elenco ha due link: accedi e sposta. Quando l'utente seleziona il link accedi, appare una pagina DOCUMENTO che mostra tutti i dati del documento selezionato. Quando l'utente seleziona il link sposta, appare la HOME PAGE con l'albero delle cartelle e delle sottocartelle; in questo caso la pagina mostra il messaggio "Stai spostando il documento X dalla sottocartella Y. Scegli la sottocartella di destinazione", la sottocartella 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 sottocartella di destinazione, il documento è spostato dalla sottocartella di origine a quella di destinazione e appare la pagina DOCUMENTI che mostra il contenuto aggiornato della sottocartella di destinazione. Ogni pagina, tranne la HOME PAGE, contiene un collegamento per tornare alla pagina precedente. L'applicazione consente il logout dell'utente. Una pagina GESTIONE CONTENUTI raggiungibile dalla HOME PAGE permette all'utente di creare una cartella, una sottocartella di una cartella esistente e un documento all'interno di una sottocartella.

Analisi dati per database

Un'applicazione web consente la gestione di cartelle, sottocartelle e documenti online. L'applicazione supporta registrazione e login di **utenti** mediante una pagina pubblica con opportune form. La registrazione controlla l'unicità dello **username**. Una **cartella** ha un **proprietario**, un **nome** e una **data di creazione** e può **contenere (solo) sottocartelle**. Una **sottocartella** può **contenere (solo) dei 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 sottocartella e accedere a una pagina DOCUMENTI che mostra l'elenco dei documenti di una sottocartella. Ogni documento in elenco ha due link: accedi e sposta. Quando l'utente seleziona il link accedi, appare una pagina DOCUMENTO che mostra tutti i dati del documento selezionato. Quando l'utente seleziona il link sposta, appare la HOME PAGE con l'albero delle cartelle e delle sottocartelle; in questo caso la pagina mostra il messaggio "Stai spostando il documento X dalla sottocartella Y. Scegli la sottocartella di destinazione", la sottocartella 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 sottocartella di destinazione, il documento è spostato dalla sottocartella di origine a quella di destinazione e appare la pagina DOCUMENTI che mostra il contenuto aggiornato della sottocartella di destinazione. Ogni pagina, tranne la HOME PAGE, contiene un collegamento per tornare alla pagina precedente. L'applicazione consente il logout dell'utente. Una pagina GESTIONE CONTENUTI raggiungibile dalla HOME PAGE permette all'utente di creare una cartella, una sottocartella di una cartella esistente e un documento all'interno di una sottocartella.

Entità, **attributi**, **relazioni**

Database Design



Database Schema (1/2)

```
CREATE TABLE `user` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(45) NOT NULL,  
  `pwd` varchar(45) NOT NULL,  
  `email` varchar(45) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB
```

```
CREATE TABLE `folder` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `userId` int DEFAULT NULL,  
  `name` varchar(45) DEFAULT NULL,  
  `date` date DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `userId_idx` (`userId`),  
  CONSTRAINT `userId` FOREIGN KEY (`userId`) REFERENCES `user` (`id`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB
```

Database Schema (2/2)

```
CREATE TABLE `subfolder` (  
  `idsubfolder` int NOT NULL,  
  `idFather` int DEFAULT NULL,  
  PRIMARY KEY (`idsubfolder`),  
  KEY `idFather_idx` (`idFather`),  
  CONSTRAINT `idFather` FOREIGN KEY (`idFather`) REFERENCES `folder` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `idSubFolder` FOREIGN KEY (`idsubfolder`) REFERENCES `folder` (`id`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB
```

```
CREATE TABLE `document` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(45) DEFAULT NULL,  
  `date` date DEFAULT NULL,  
  `summary` varchar(160) DEFAULT NULL,  
  `type` varchar(45) DEFAULT NULL,  
  `folderId` int DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `folderId_idx` (`folderId`),  
  CONSTRAINT `folderId` FOREIGN KEY (`folderId`) REFERENCES `folder` (`id`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB
```

Analisi Requisiti Applicazione

Un'applicazione web consente la gestione di cartelle, sottocartelle e documenti online. L'applicazione supporta registrazione e login di utenti mediante una **pagina pubblica** con opportune **form**. La registrazione **controlla l'unicità** dello username. Una cartella ha un proprietario, un nome e una data di creazione e può contenere (solo) sottocartelle. Una sottocartella può contenere (solo) dei 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 sottocartella e accedere** a una **pagina DOCUMENTI** che mostra **l'elenco dei documenti di una sottocartella**. Ogni documento in elenco ha due link: accedi e sposta. Quando l'utente **seleziona il link accedi**, appare una **pagina DOCUMENTO** che mostra tutti i **dati del documento selezionato**. Quando l'utente **seleziona il link sposta**, appare la HOME PAGE con **l'albero delle cartelle e delle sottocartelle**; in questo caso la pagina mostra il **messaggio** "Stai spostando il documento X dalla sottocartella Y. Scegli la sottocartella di destinazione", la sottocartella 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 sottocartella di destinazione**, il **documento è spostato** dalla sottocartella di origine a quella di destinazione e **appare la pagina DOCUMENTI che mostra il contenuto aggiornato** della sottocartella di destinazione. Ogni pagina, tranne la HOME PAGE, contiene un **collegamento per tornare alla pagina precedente**. L'applicazione consente il **logout** dell'utente. Una **pagina GESTIONE CONTENUTI** raggiungibile dalla HOME PAGE permette all'utente **di creare una cartella, una sottocartella di una cartella esistente e un documento all'interno di una sottocartella**.

Pagine (viste), componenti vista, eventi, azioni.

Components

- **Data Access Objects (DAO)**

- **UserDAO**
 - findUser(email, password) : user
 - registerUser(name, email, pwd)
 - getUserByEmail(email) : user
 - getUserByName(name) : user
 - getUserById(id) : user
- **FolderDAO**
 - findAllFolders() : List<Folder>
 - findAllFoldersById(userId) : List<Folder>
 - findFolderById(folderId) : Folder
 - findTopFolderAndSubfoldersById(userId) : List<Folder>
 - findTopFolderAndSubfolders() : List <Folder>
 - findSubparts(Folder p)
 - createFolder(userId, name, date)
 - createSubFolder(userId, name, date, idFather)
 - findSubfoldersById(id) : List<Folder>
- **DocumentDAO**
 - getDocumentsByIdFolder(folderId) : List<Document>
 - getDocumentById(id) : Document
 - createDocument(name, date, summary, type, folderId)
 - updateDocument(id, folderId)

- **Model Objects (Beans)**

- User
- Folder
- Document

- **Filters**

- CheckLoggedUser
- CheckNotLoggedUser

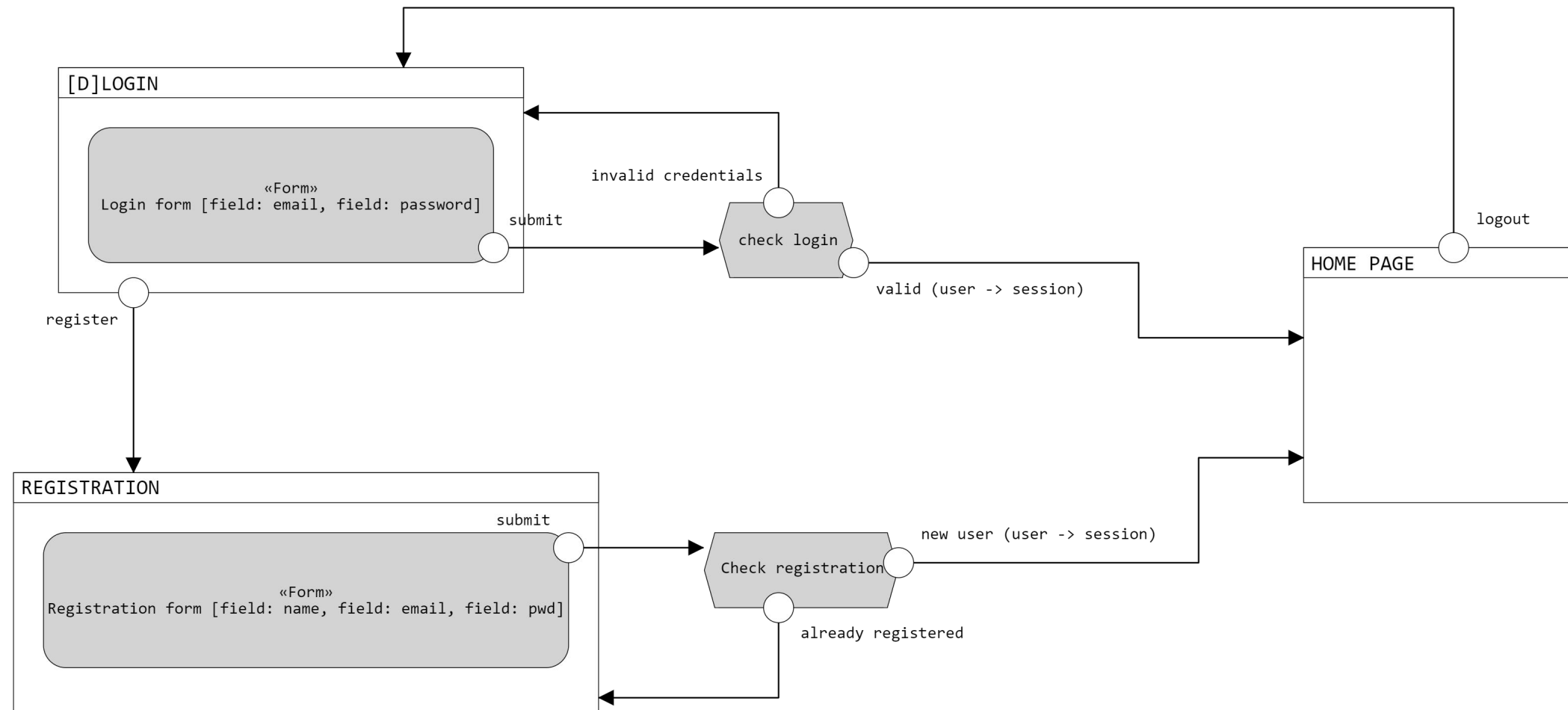
- **Controllers (Servlets) [access rights]**

- Login
- Logout
- Access
- Register
- GoToRegisterPage
- GoToLoginPage
- GoToHome
- GoToHomePage
- GoToDocument
- GoToManagePage
- CreateFolder
- CreateSubFolder
- CreateDocument
- Move
- UpdateDocument

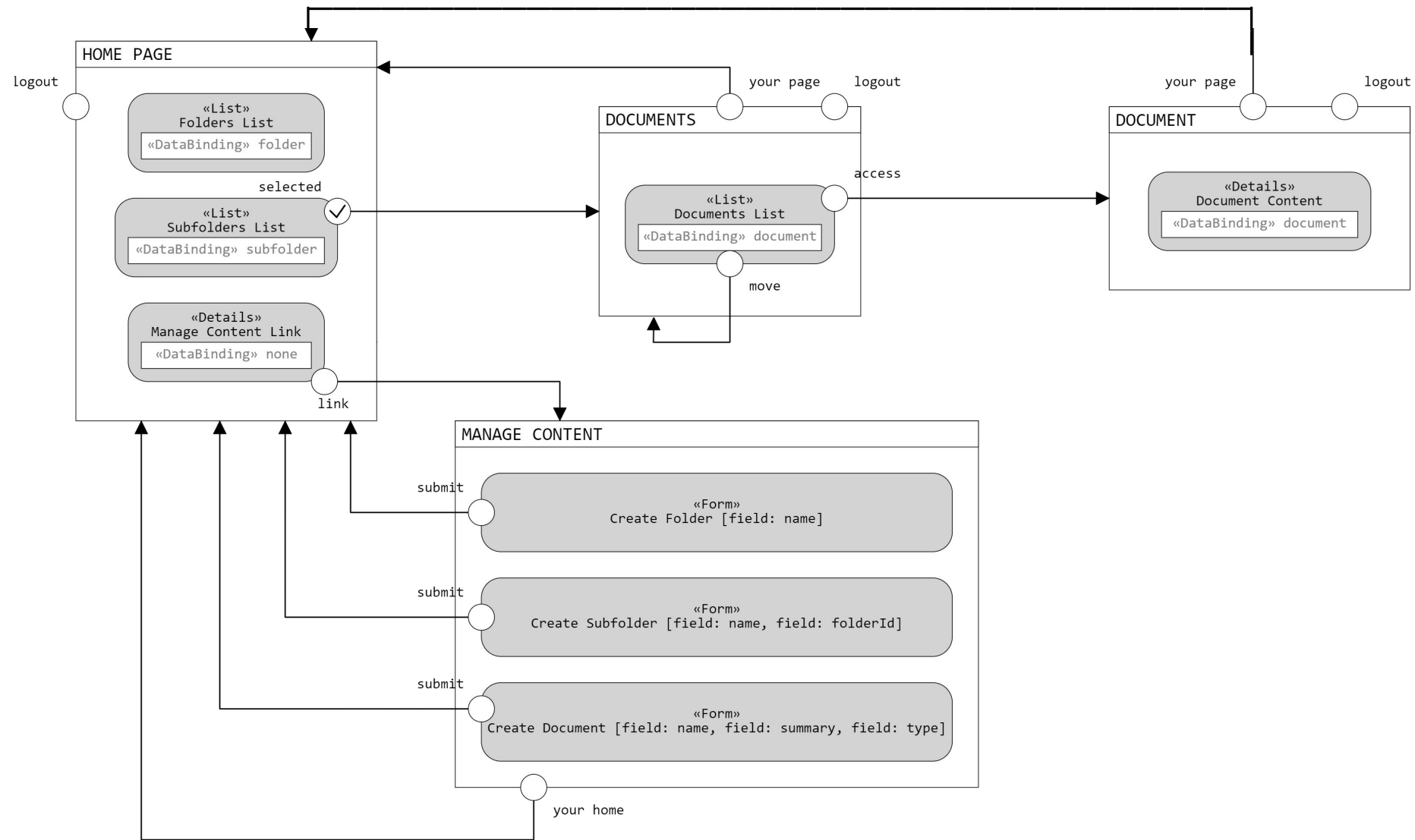
- **View (Templates)**

- document.html
- documentFolder.html
- error.html
- fragment.html
- contentFolder.html
- header.html
- home.html
- login.html
- manage.html
- register.html

Design Applicativo (IFML)



Design Applictivo (IFML)



Sequence Diagrams

I seguenti diagrammi vogliono descrivere quanto più coerentemente possibile i principali eventi dell'applicazione web realizzata.

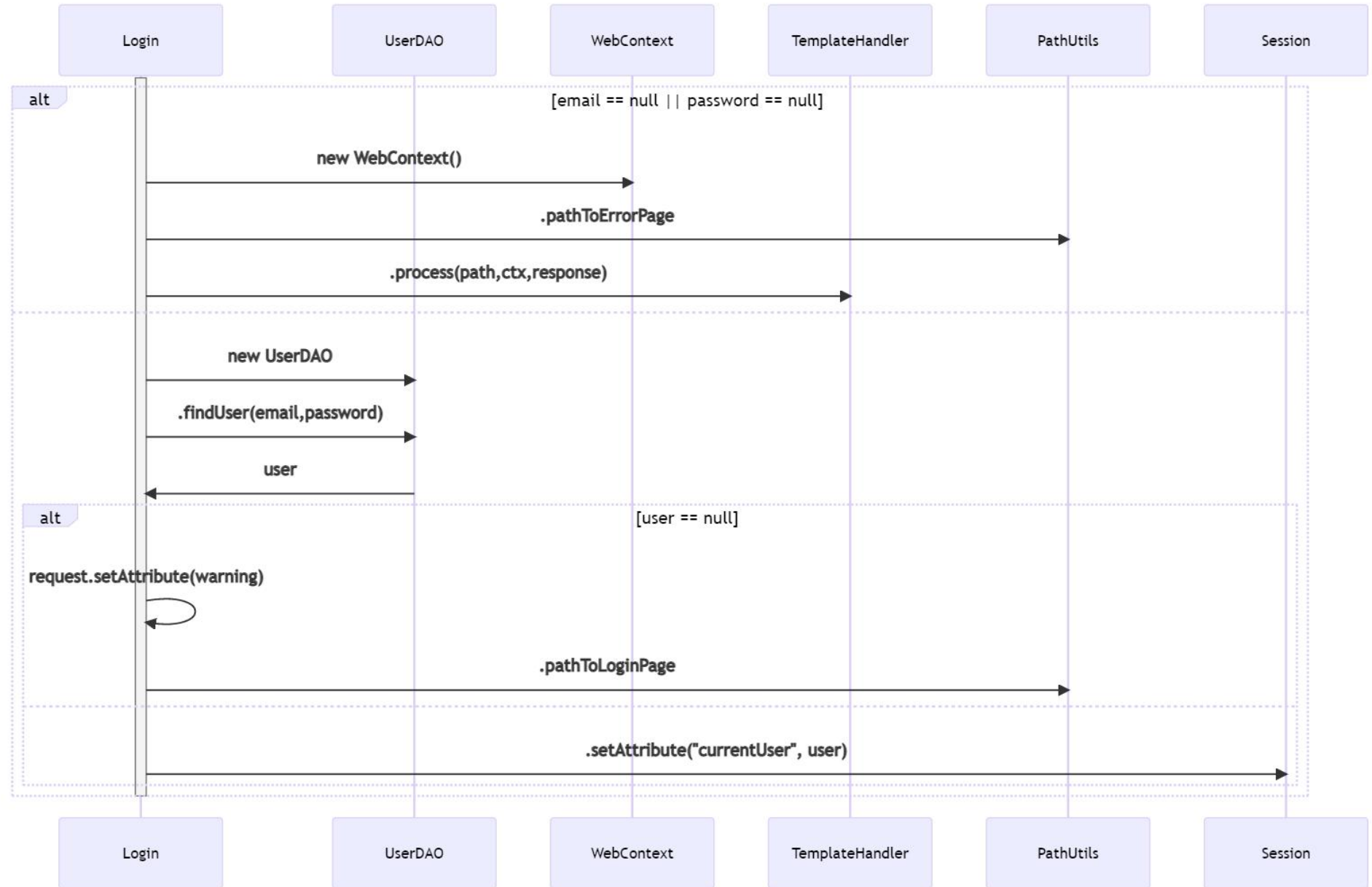
Alcuni dettagli sono talvolta omessi per mantenere i diagrammi chiari e puliti.

Ogni errore interno del server o di accesso e interazione col database è sottointeso nei diagrammi ma sempre trattato in pratica con la visualizzazione di una pagina d'errore.

A tutti gli eventi rappresentati precede una chiamata dal Browser con una POST o una GET.

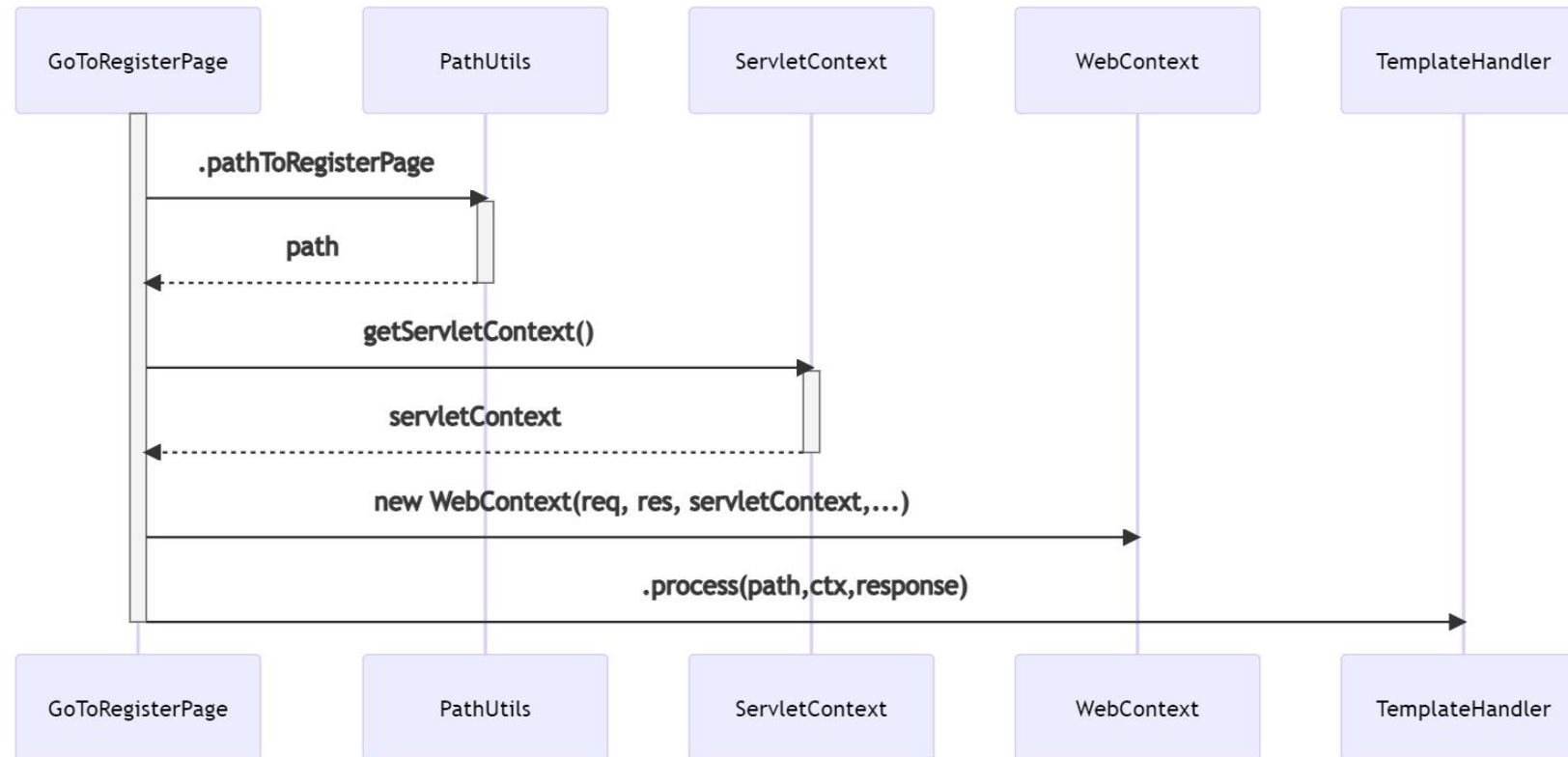
Event: Login

(Browser)
POST
email
password



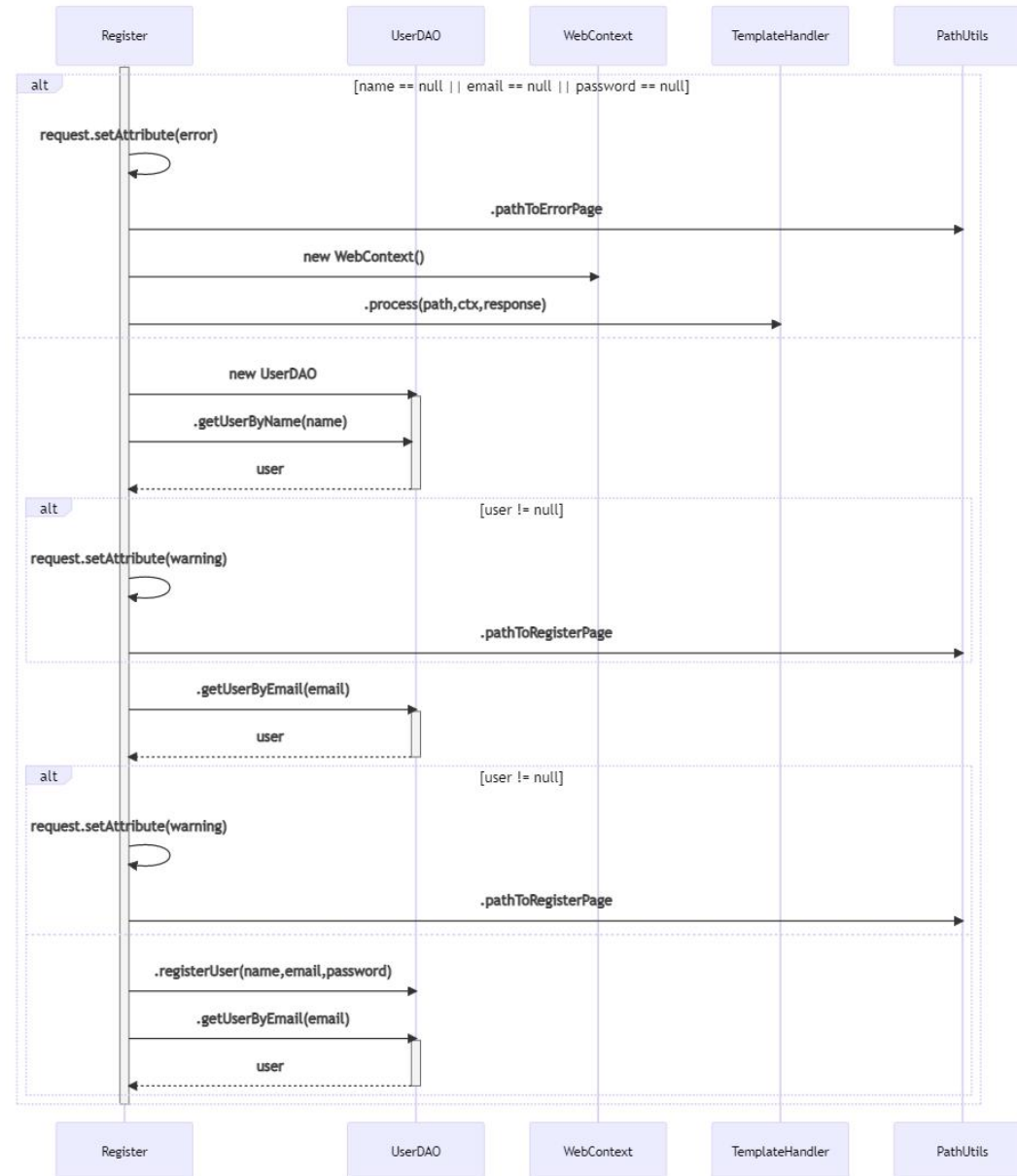
Event: GoToRegisterPage

(Browser)
GET

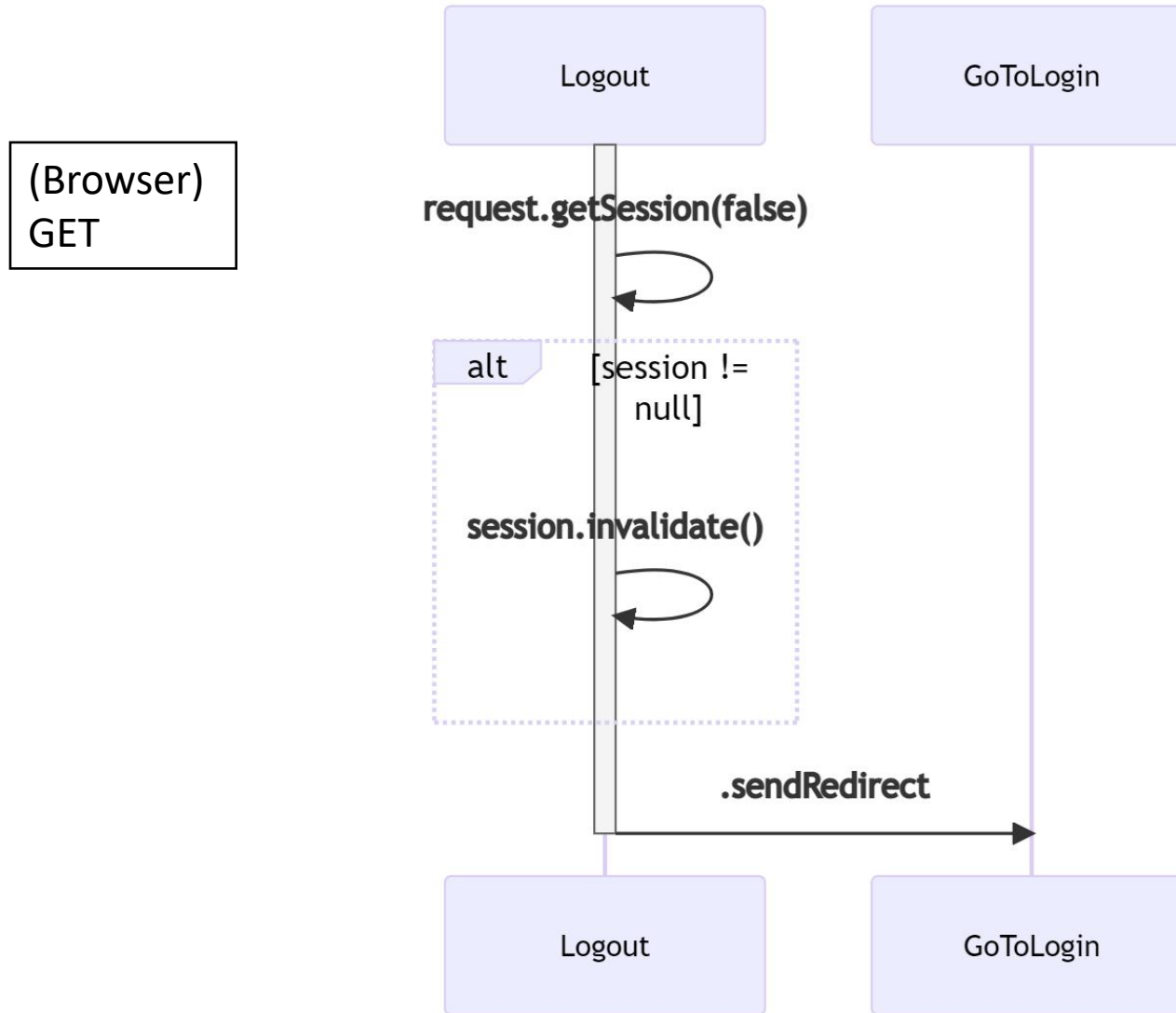


Event: Registration

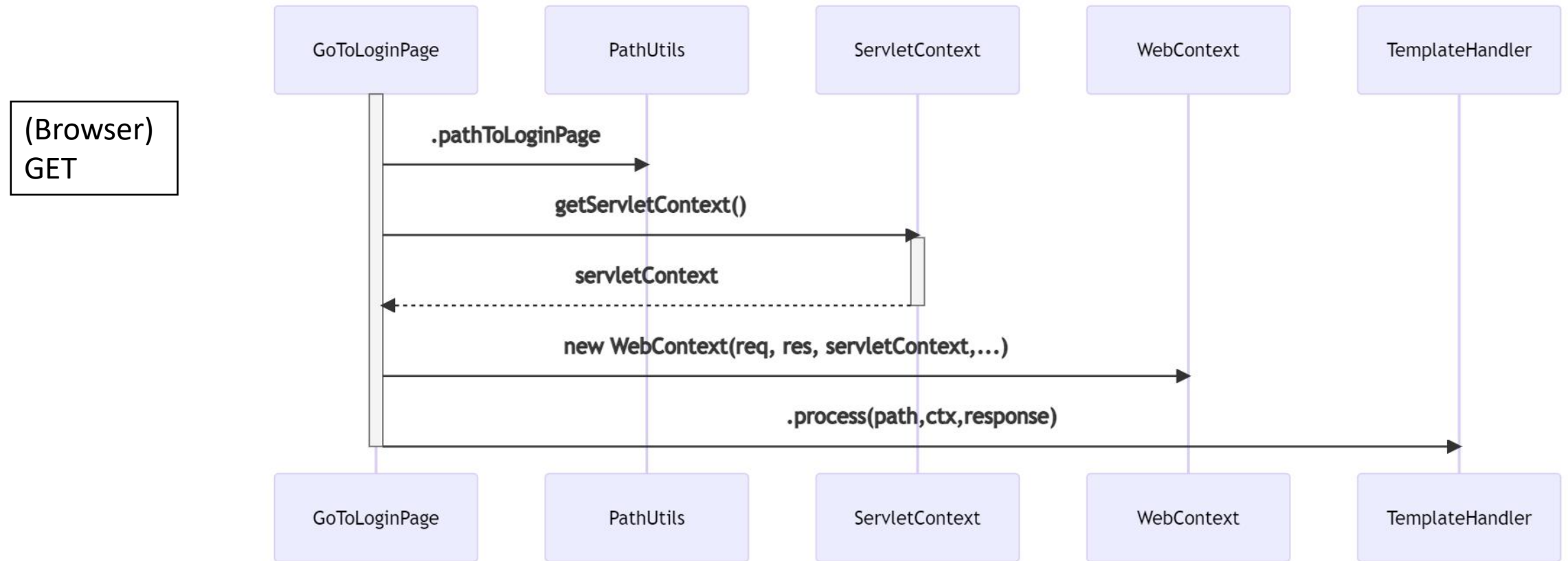
(Browser)
POST
name
email
password



Event: Logout

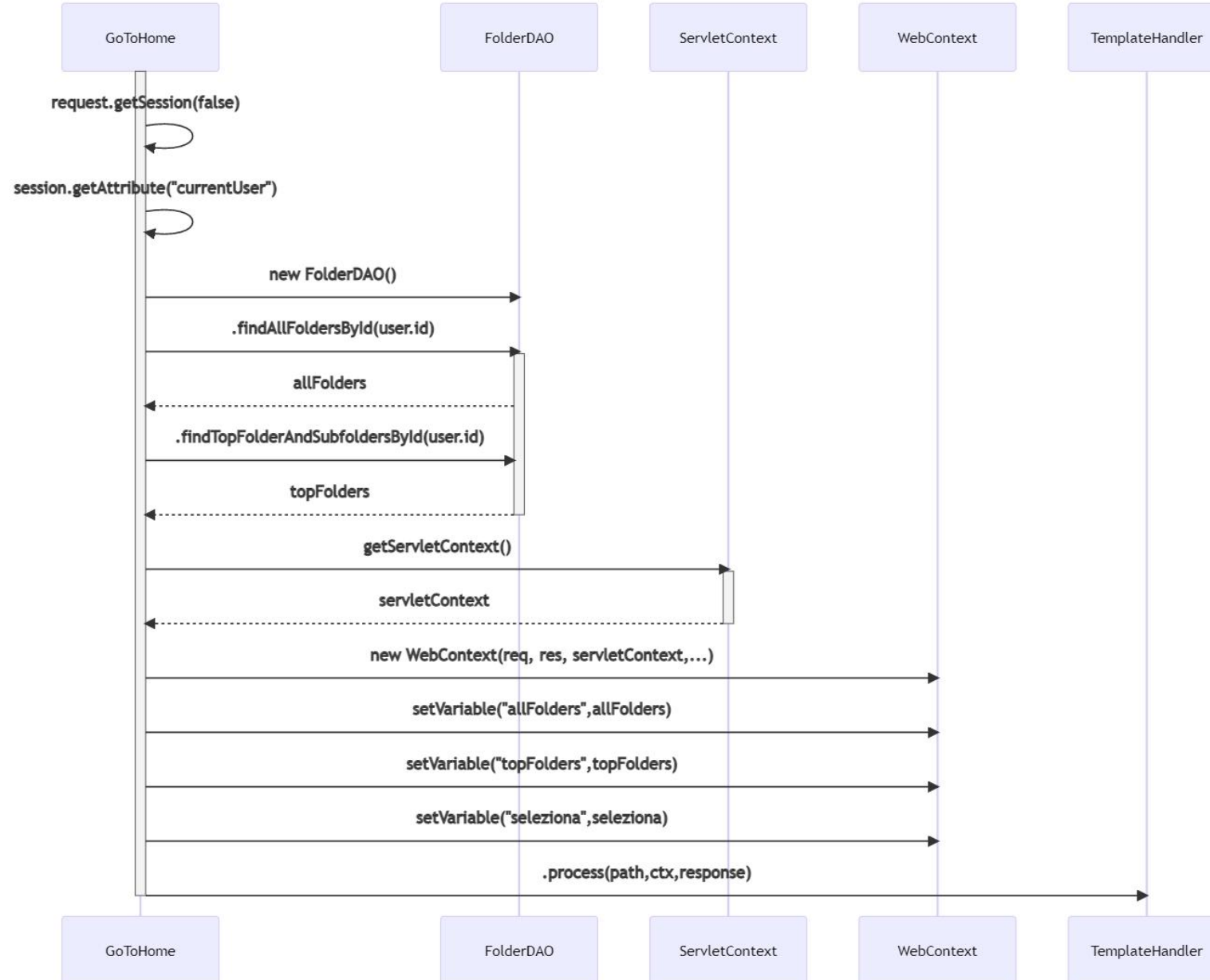


Event: GoToLoginPage

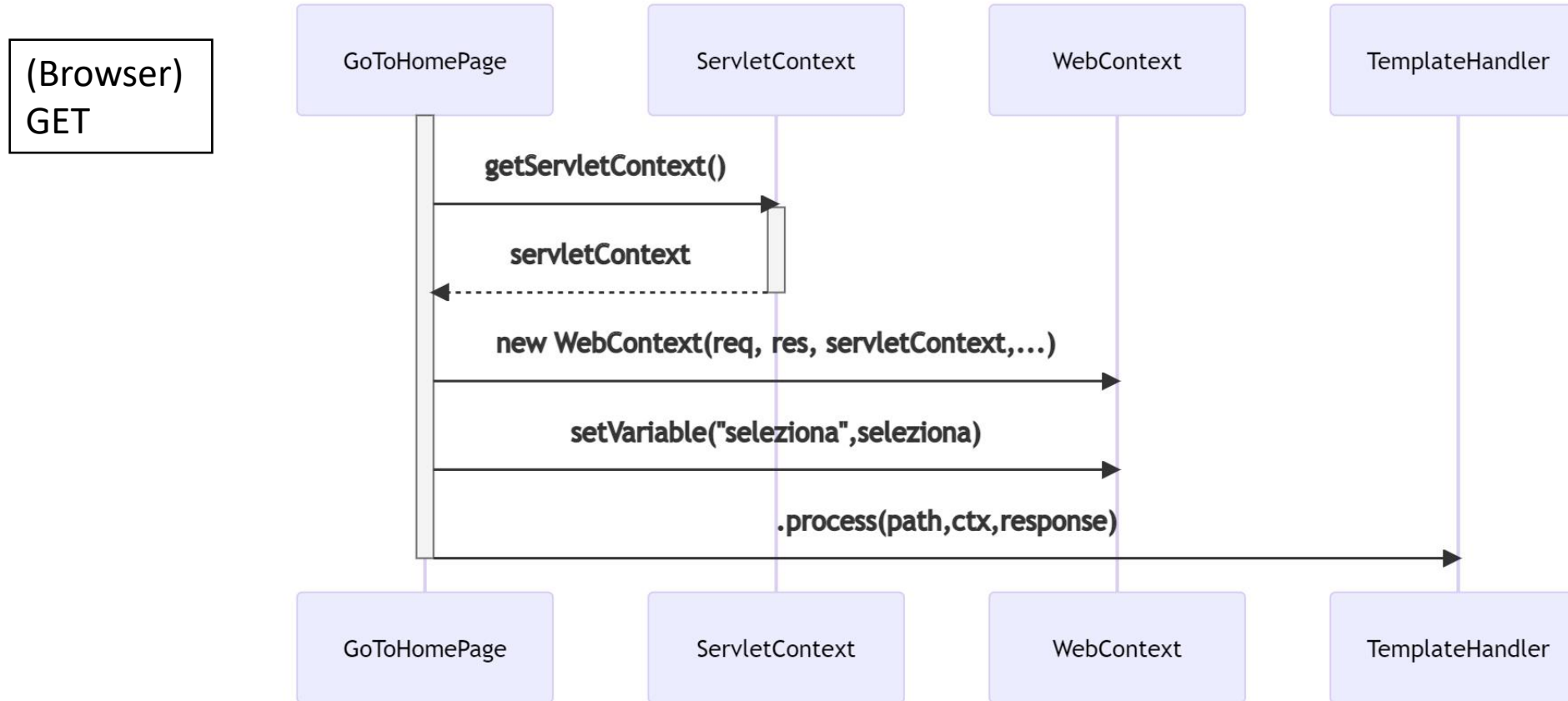


Event: GoToHome

(Browser)
GET

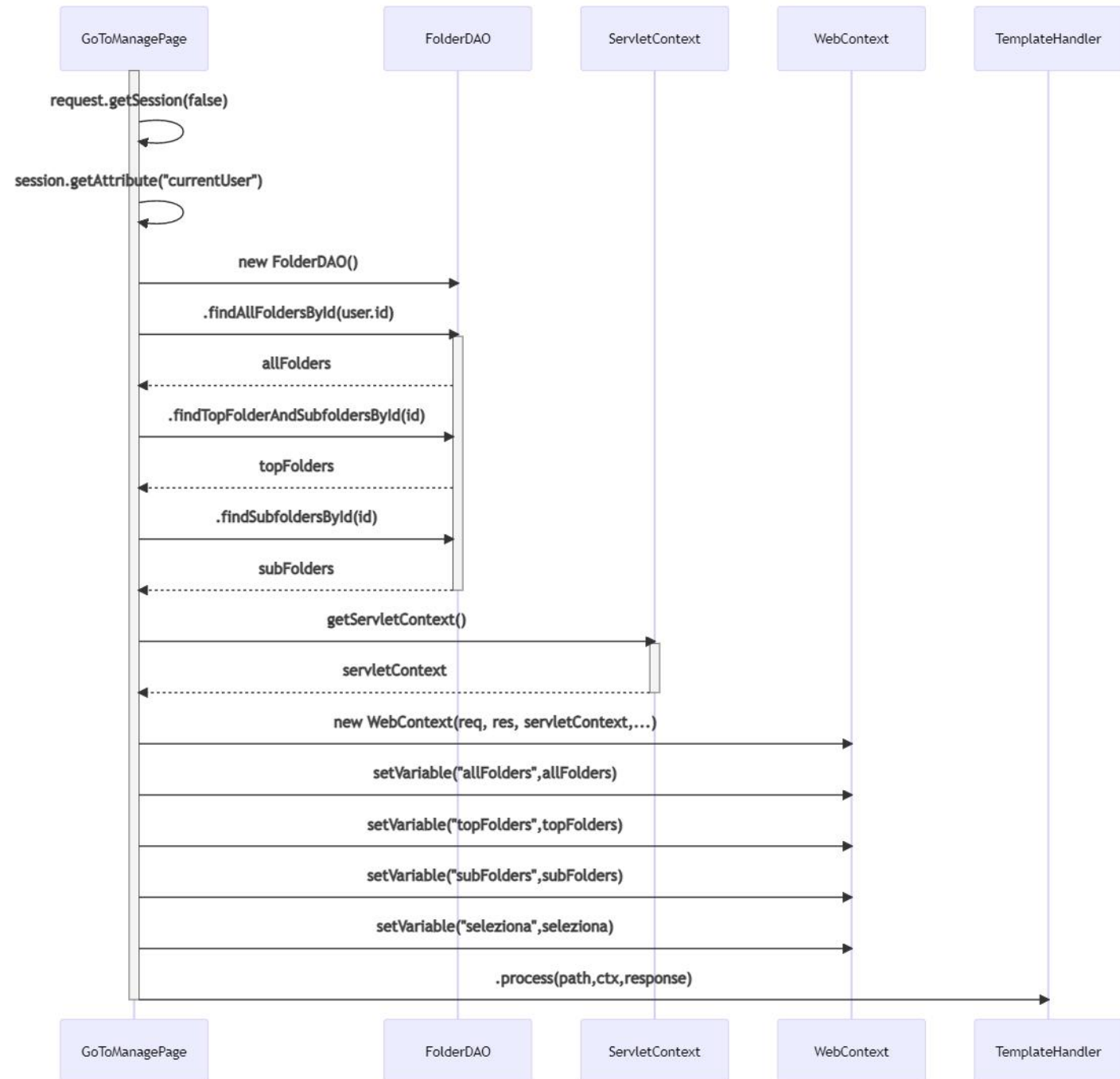


Event: GoToHomePage



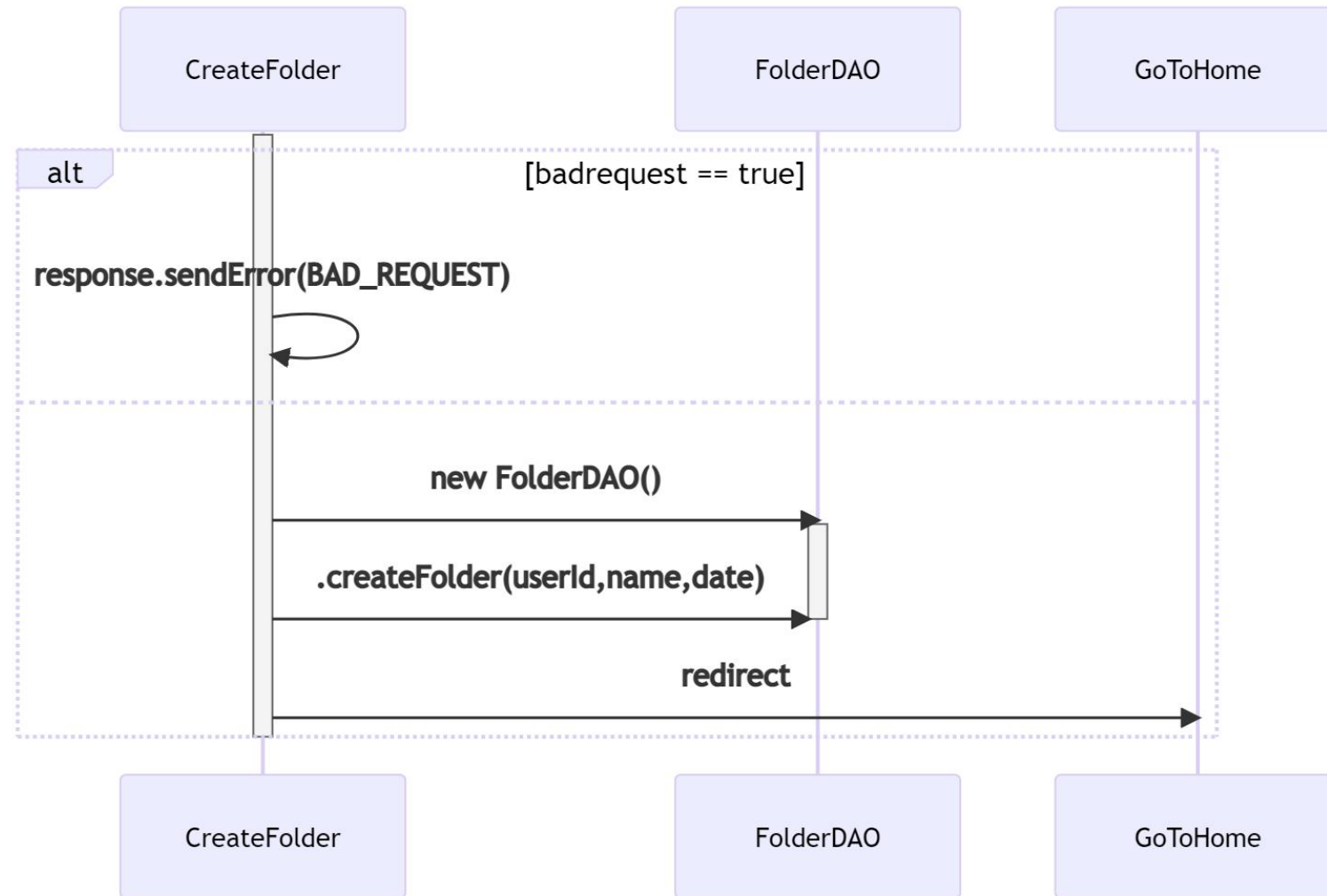
Event: GoToManagePage

(Browser)
GET



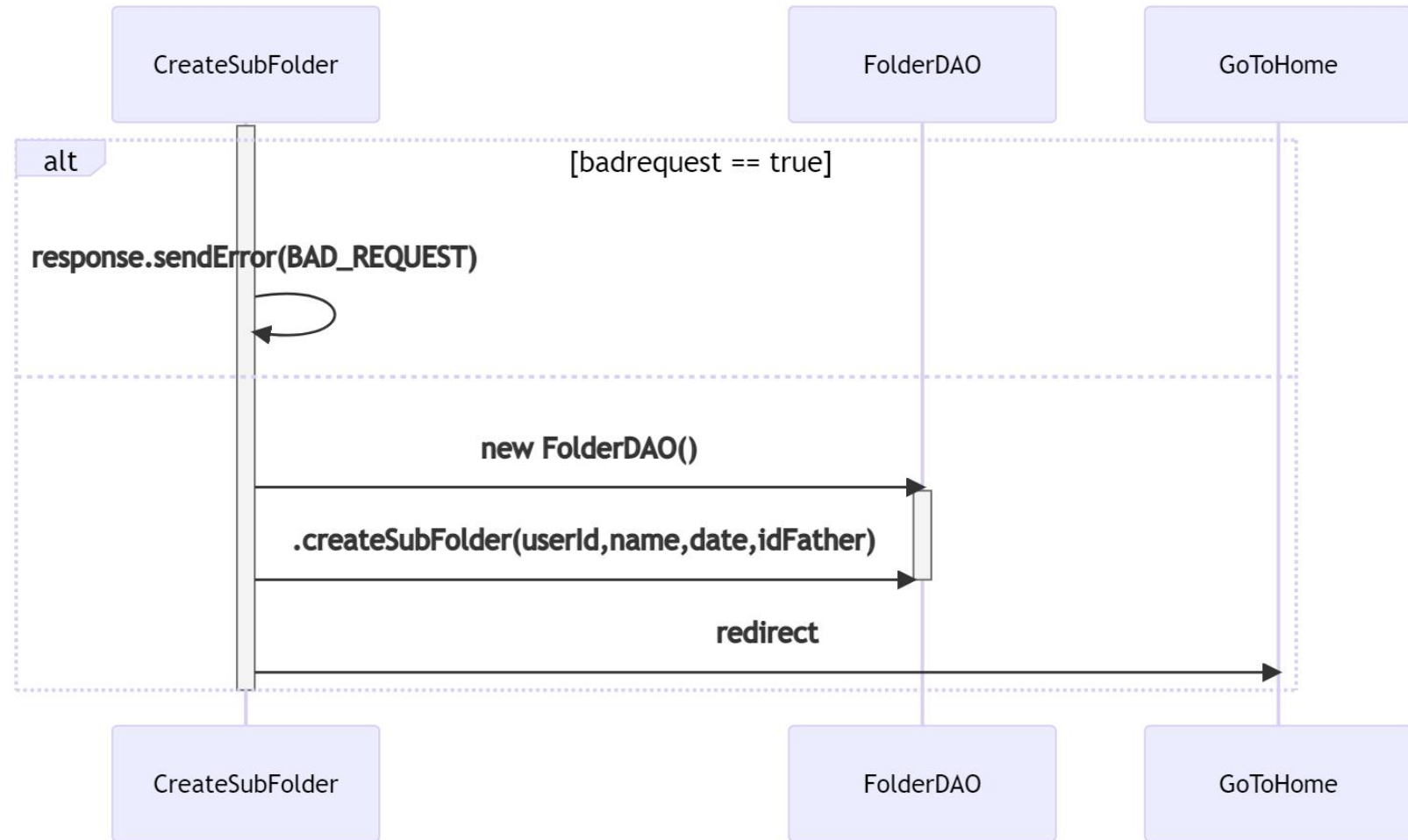
Event: CreateFolder

(Browser)
POST
name
userId



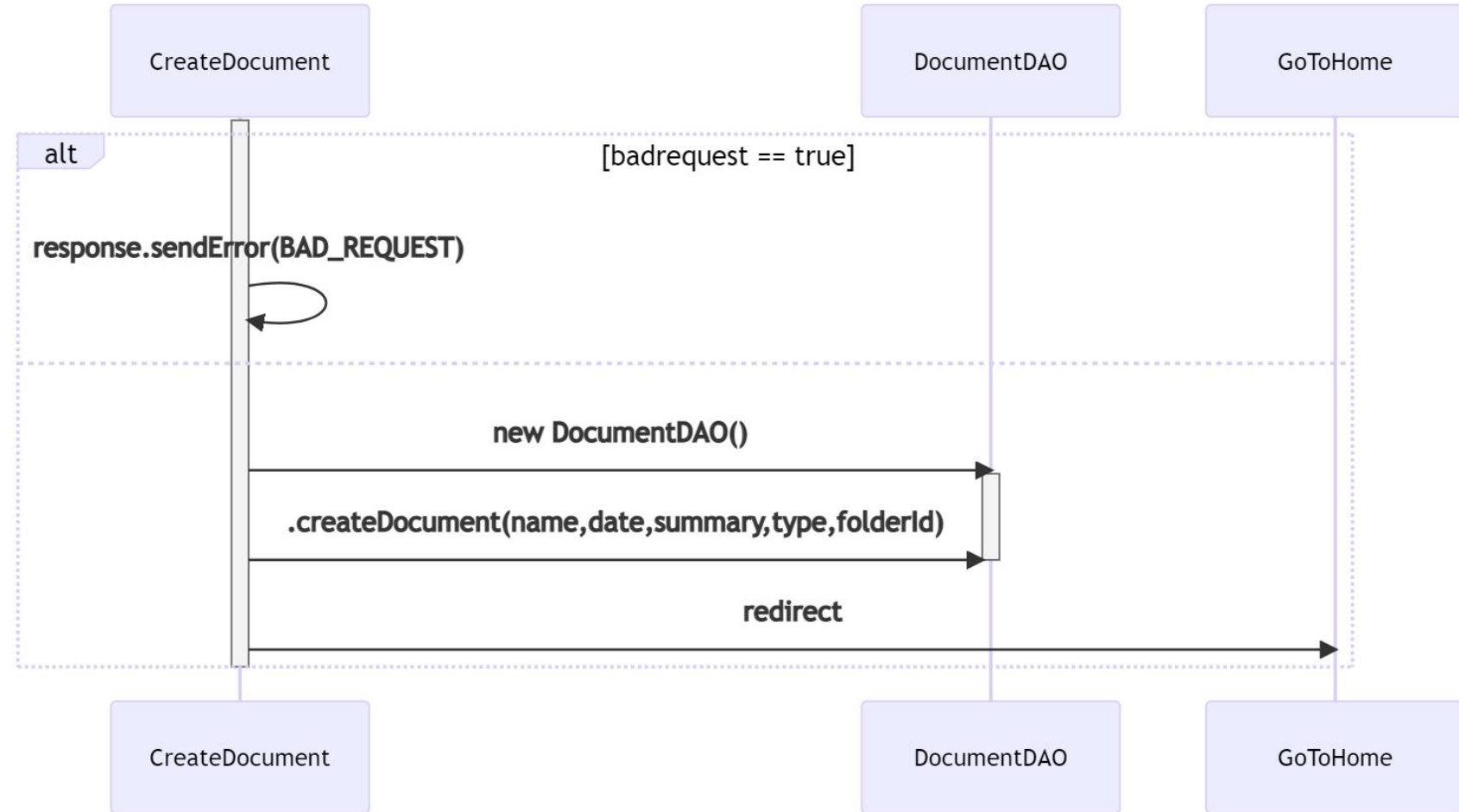
Event: CreateSubFolder

(Browser)
POST
nameS
userIdS
idFatherS



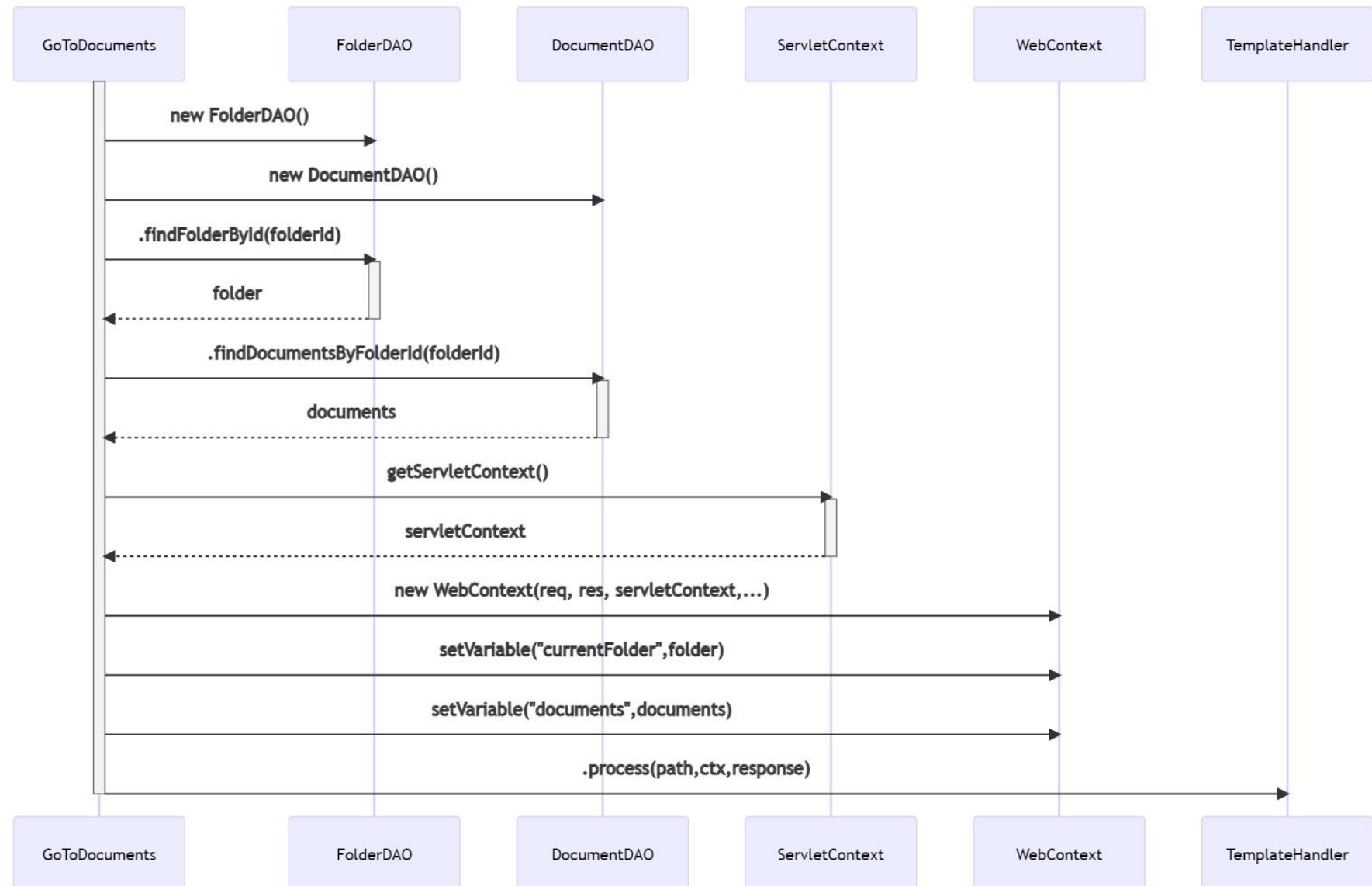
Event: CreateDocument

(Browser)P
OST
named
summaryD
typeD
folderIdD



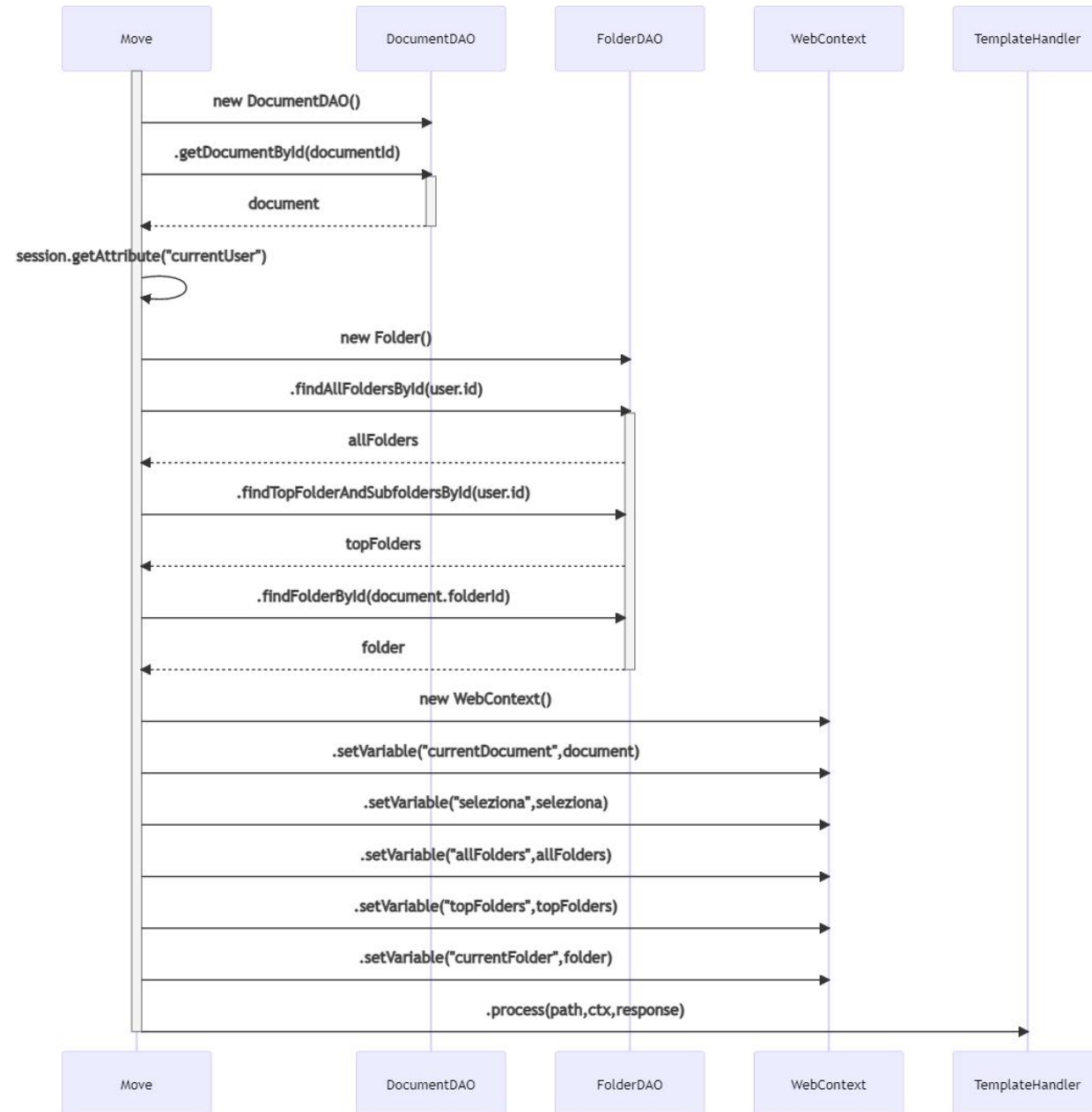
Event: GoToDocuments

(Browser)
POST
folderId



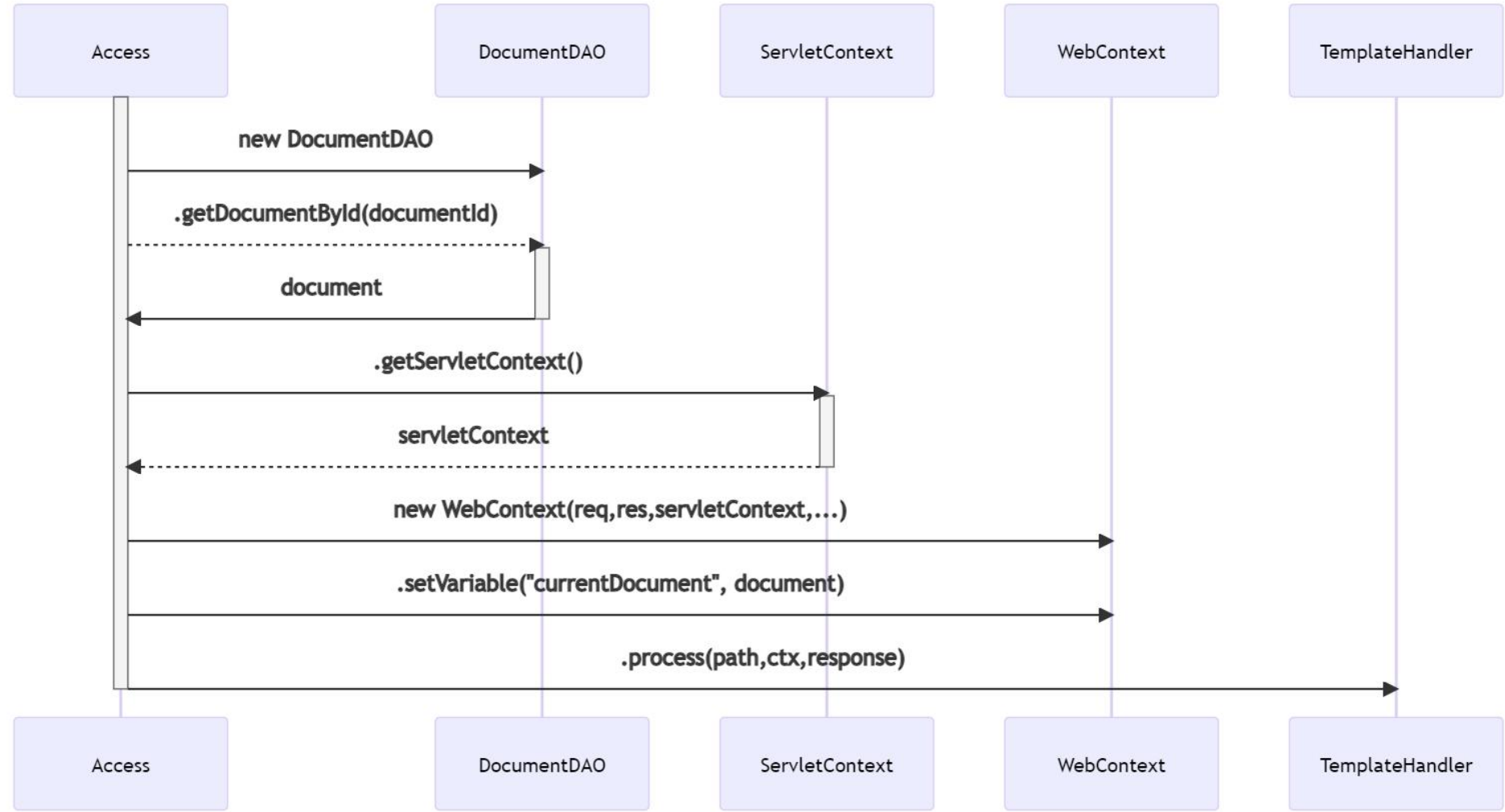
Event: Move

(Browser)
POST
documentId



Event: Access

(Browser)
POST
documentId



Event: UpdateDocument

(Browser)
POST
documentId
folderId

