

Esercizio 3:

Gestione Documenti

Tecnologie Informatiche per il Web – AA2021/22

Alessandro Cogollo – 10571078

Lorenzo Franzè - 10679981



POLITECNICO
MILANO 1863

Project Description

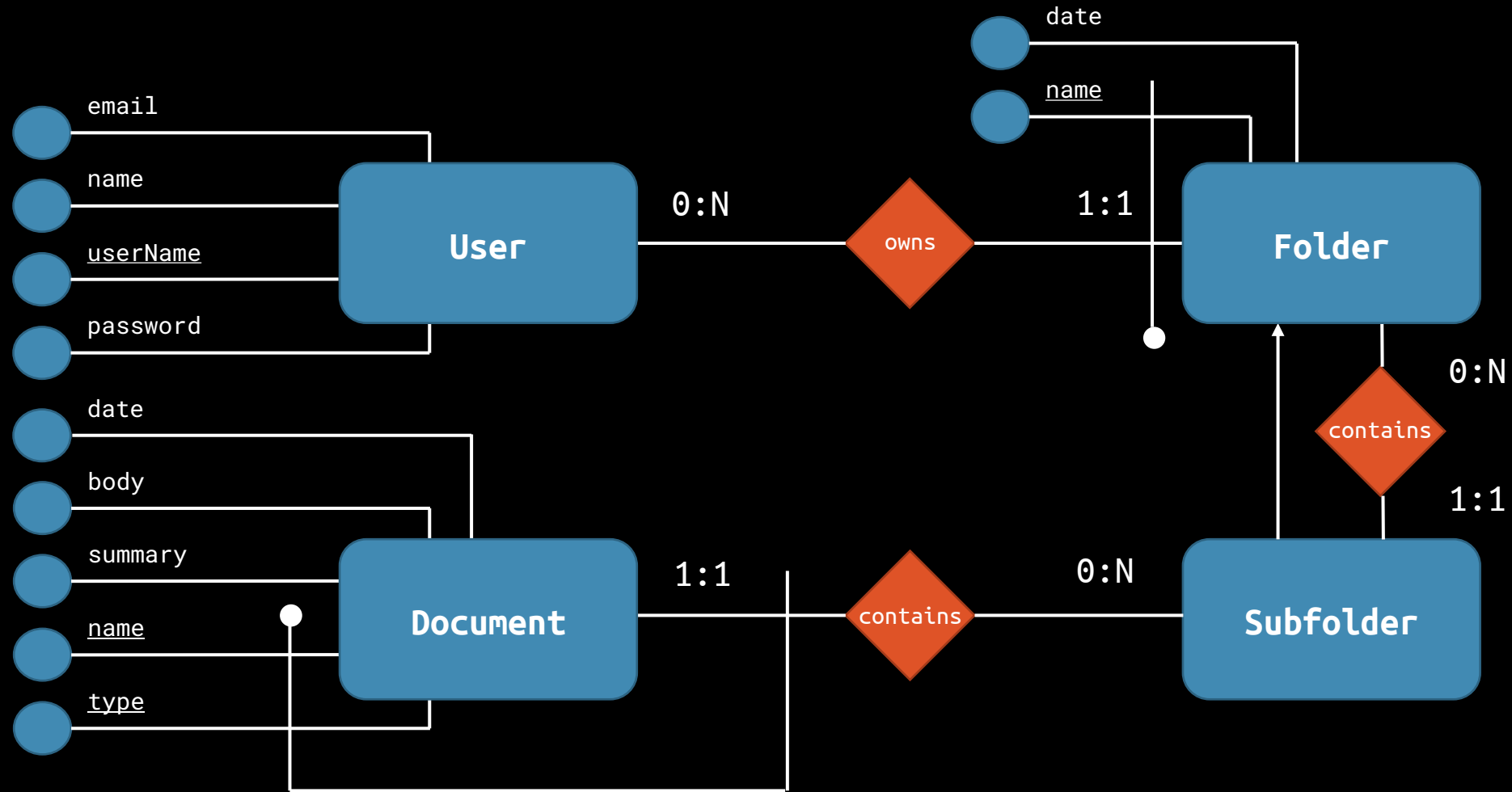
Un'applicazione web consente la gestione di cartelle, sottocartelle e documenti online. L'applicazione supporta registrazione e login di utenti mediante una pagina pubblica con opportune form. La registrazione controlla l'unicità dello username. Una cartella ha un proprietario, un nome e una data di 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.

Data Analysis

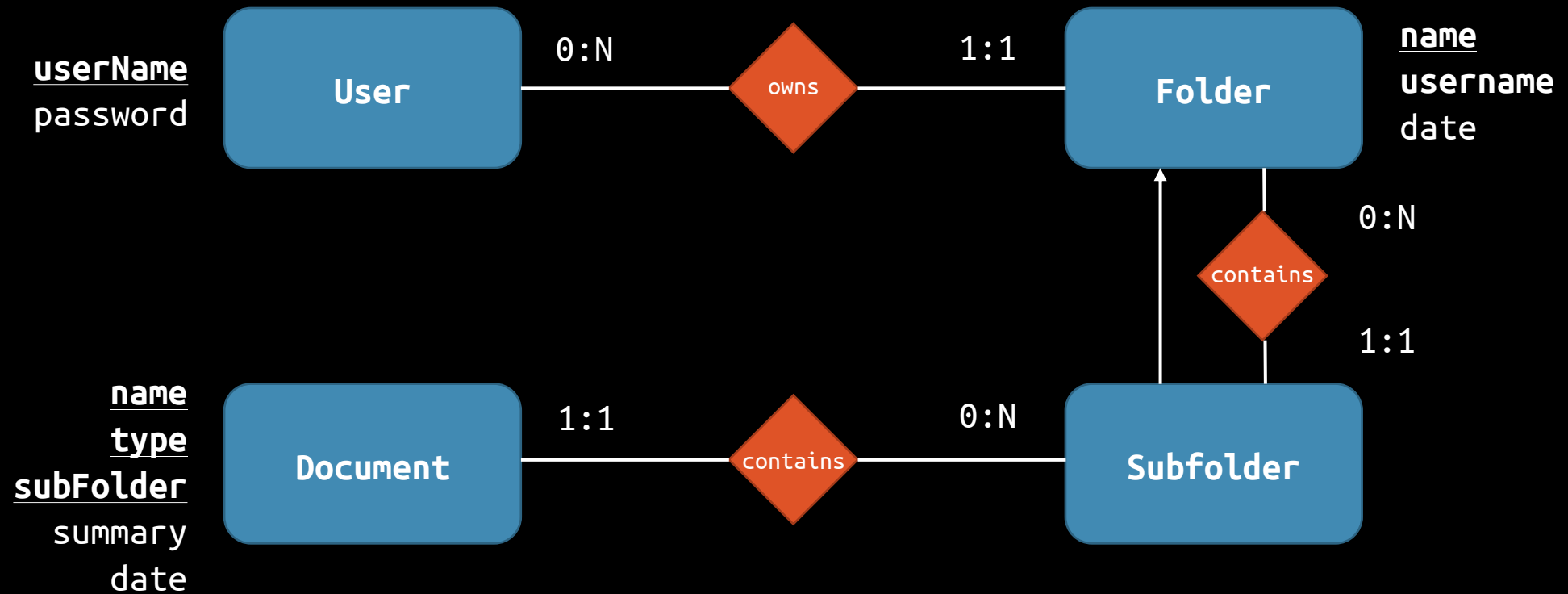
Un'applicazione web consente la gestione di **cartelle**, **sottocartelle** e **documenti** online. L'applicazione supporta registrazione e login di **utenti** mediante una pagina pubblica con opportune form. La registrazione controlla l'unicità dello **username**. Una **cartella** ha un **proprietario**, 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**.

Entità, Attributi, Relazioni

Database Design



Database Design



Logical Database Schema 1/4

```
CREATE TABLE `document` (  
  `username` varchar(40) NOT NULL,  
  `foldername` varchar(40) NOT NULL,  
  `subfoldername` varchar(40) NOT NULL,  
  `documentname` varchar(40) NOT NULL,  
  `type` varchar(40) NOT NULL,  
  `summary` varchar(255) NOT NULL,  
  `date` date NOT NULL,  
  `body` medium blob NOT NULL,  
  PRIMARY KEY  
    (`username`, `foldername`, `subfoldername`, `documentname`, `type`),  
  FOREIGN KEY (`username`, `foldername`, `subfoldername`) REFERENCES  
    `subfolder` (`username`, `foldername`, `subfoldername`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
)
```

Logical Database Schema 2/4

```
CREATE TABLE `folder` (  
  `username` varchar(40) NOT NULL,  
  `foldername` varchar(40) NOT NULL,  
  `date` date NOT NULL,  
  PRIMARY KEY (`username`, `foldername`),  
  FOREIGN KEY (`username`) REFERENCES `user` (`username`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
)
```

Logical Database Schema 3/4

```
CREATE TABLE `subfolder` (  
  `username` varchar(40) NOT NULL,  
  `foldername` varchar(40) NOT NULL,  
  `subfoldername` varchar(40) NOT NULL,  
  `date` date NOT NULL,  
  PRIMARY KEY (`username`, `foldername`, `subfoldername`),  
  FOREIGN KEY (`username`, `foldername`)  
  REFERENCES `folder` (`username`, `foldername`)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
)
```


Logical Database Schema 4/4

```
CREATE TABLE `user` (  
  `username` varchar(40) NOT NULL,  
  `email` varchar(40) NOT NULL,  
  `name` varchar(40) NOT NULL,  
  `password` varchar(40) NOT NULL,  
  PRIMARY KEY (`username`),  
  UNIQUE KEY `email` (`email`))
```

Application Requirements Analysis

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.

Pages (views), View Components, Events, Actions

Components

- **Data Access Objects (Classes)**

- UserDao
 - *checkUser*
 - *existsUser, existsEmail*
 - *createUser*
- FolderDAO
 - *getAllFolderOfUser*
 - *getAllChildOfFolder*
(utility: uses a SubFolderDAO method)
 - *insertFolder*
 - *existsFolder*
- SubFolderDAO
 - *getAllSubFolderOfFolder*
 - *insertSubFolder*
 - *existsSubFolder*
- DocumentDAO
 - *exists*
 - *insertDocument*
 - *moveDocumentFromSubFolder*
 - *getAllDocumentsOfSubFolder*

- *getDocumentByKey*
- *getDocumentData*

- **Model objects (Beans)**

- User
- Folder
- SubFolder
- Document

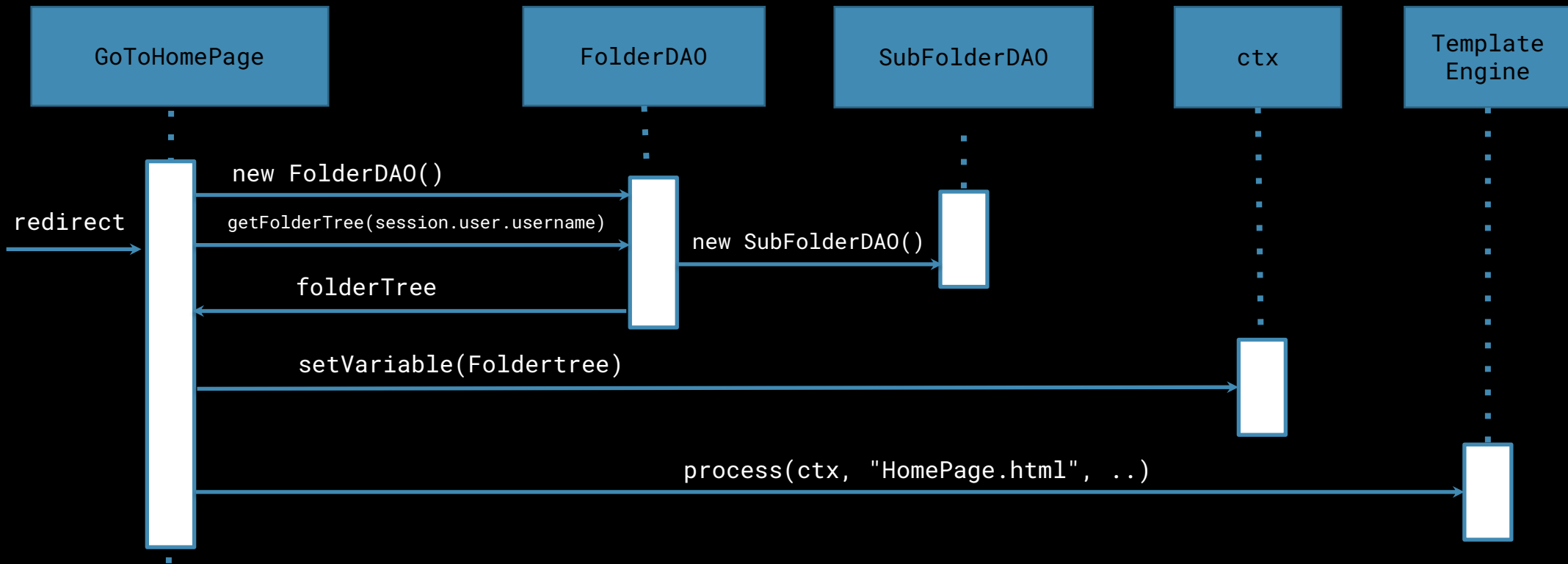
- **Controllers (servlets)**

- Login
- Logout
- Subscribe
- GoToHomePage
- Documents
- Move
- Access
- GetFile
- ContentManager
- CreateDocument
- CreateFolder
- CreateSubFolder

- **Views (Templates) & components**

- Login
 - *Login form*
- HomeAdmin (admin)
 - *Folder – SubFolder list*
 - *Open SubFolder (go to DocumentsPage)*
 - *GoToContentManager*
- ContentManager
 - *Folder – SubFolder list*
 - *Create: folder, subfolder, document forms*
- DocumentsPage
 - *Move (Open moveToHomePage)*
 - *Access (go to DocumentPage)*
- moveToHomePage
 - *Select target SubFolder*
- DocumentPage
 - *Document details*
 - *Download*

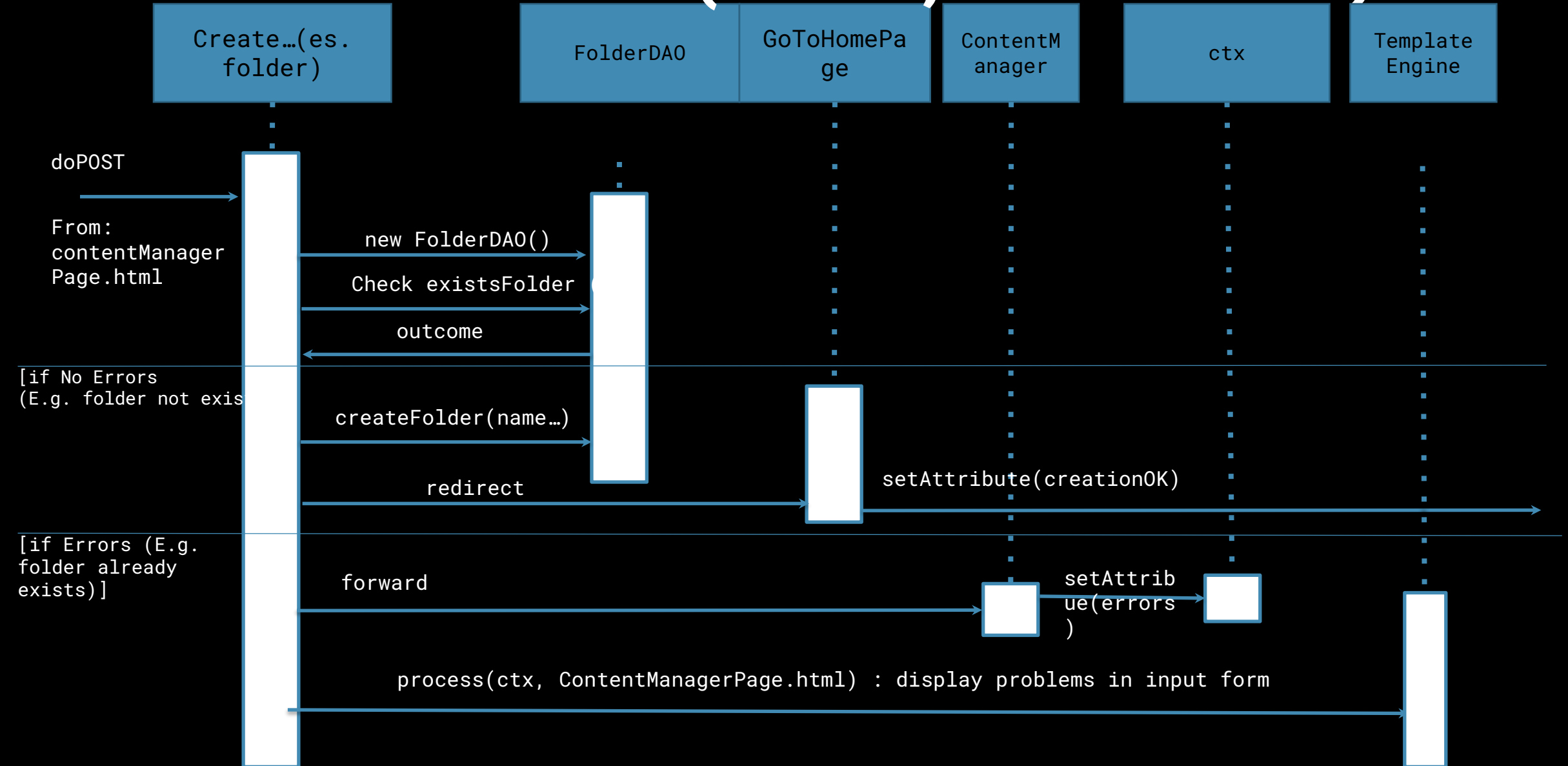
Event: GoToHomePage



Note:

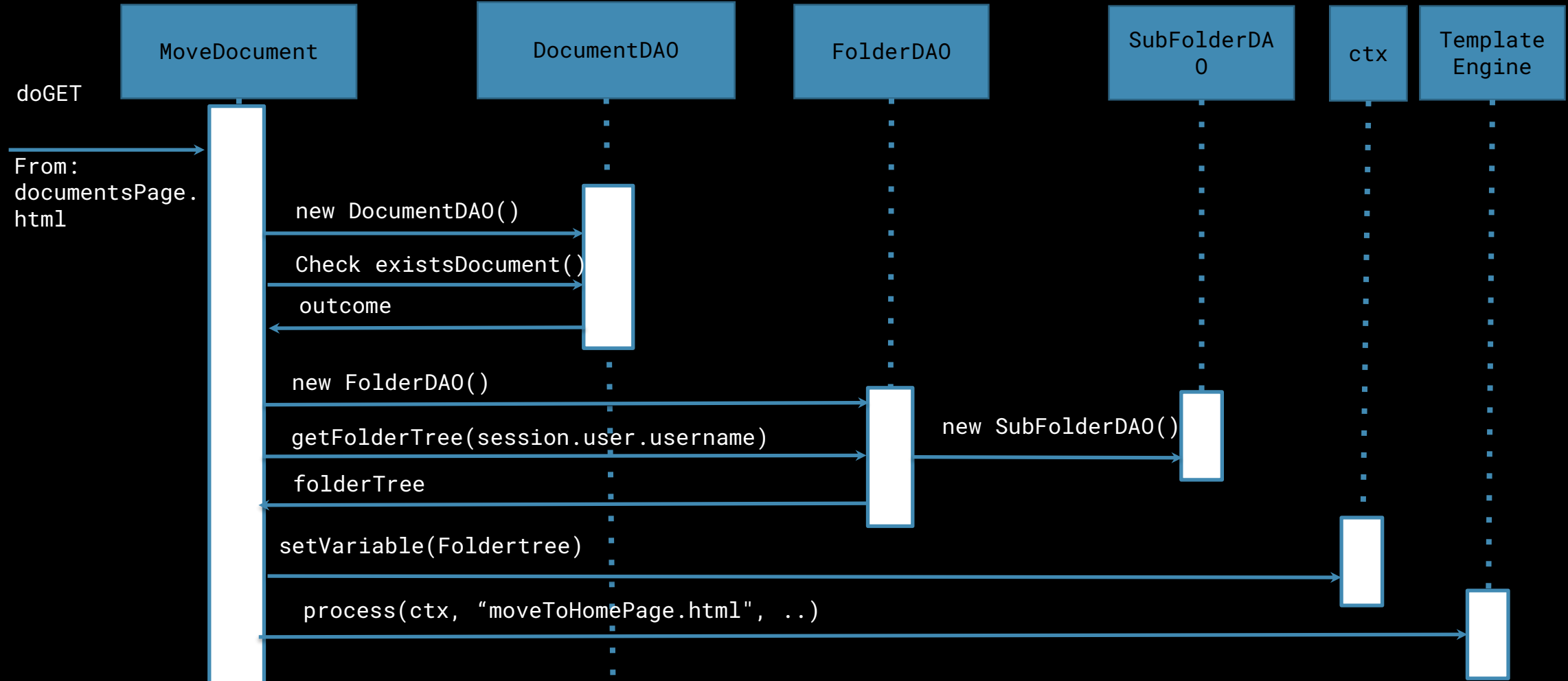
- event Documents is analogous to this event, the only difference is `getDocumets` (documents in the selected subFolder) instead of `getFolderTree` and the processed page is `DocumetsPage.html`
- event Contentmanager is analogous to this event, the only difference is that the processed page is `contentManagerPage.html` and the presence of 3 forms to add new elements
- event MoveDocument is analogous to this event (explained later)

Event: Create(folder, document..)



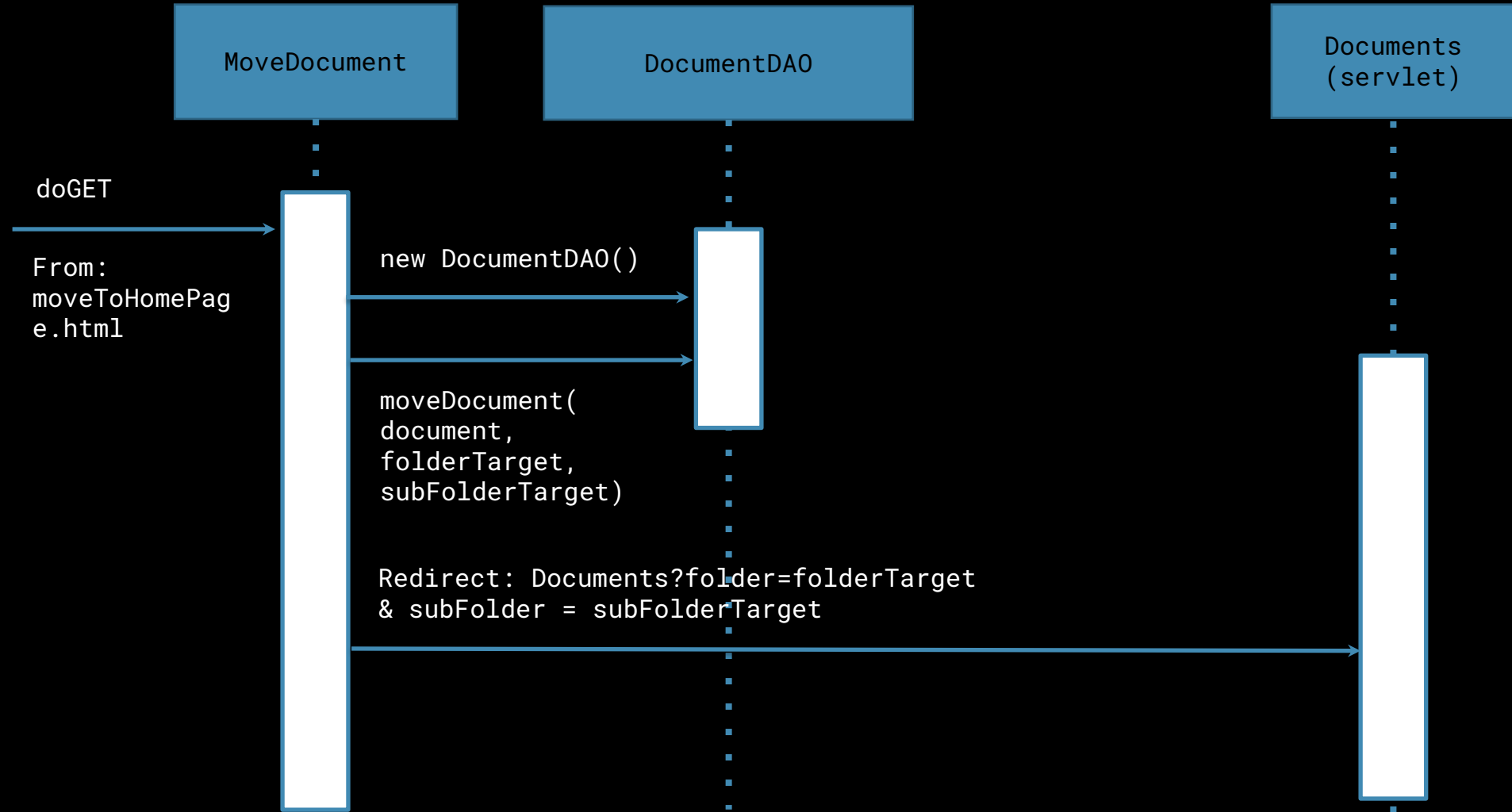
Event: MoveToHomePage

(action:chooseMovement)



Event: MoveToHomePage

(action:updateFolder)



RIA Version

Summary

- **Viste e componenti**
 - Login
 - Subscribe
 - Pagina Home
 - Tree folder - subFolder
 - Wizard
 - Form create folder
 - Form create sub folder
 - Form create document and upload
 - Trash Bin
 - Show sub folder details: documents
 - Documents action: move
- **Eventi e azioni**
 - Login
 - Credential check
 - **Subscribe**
 - Check and subscribe
 - **Click «+» button : add element in selected position**
 - Show correct form wizard
 - **Click «View content» link**
 - Show sub folder details: documents
- **«Move» on document**
 - Move document in new sub folder with drag & drop
- **«Move» each element in BIN to delete**
 - Drag & drop to delete
- **Click on «View» button**
 - Show document details
- **Click on «Download» button**
 - Start download
- **Click bottone chiusura**
 - Cambio stato missione
- **Logout**
 - logout

Server side: DAO & model objects

(more details in components for HTML version)

- **Controllers**

- Login
- Logout
- Subscribe
- GoToHomePage
- Documents
- Move
- Access
- GetFile
- ContentManager
- CreateDocument
- CreateFolder
- CreateSubFolder
- DeleteItem

- **Model objects (Beans)**

- User
- Folder
- SubFolder
- Document

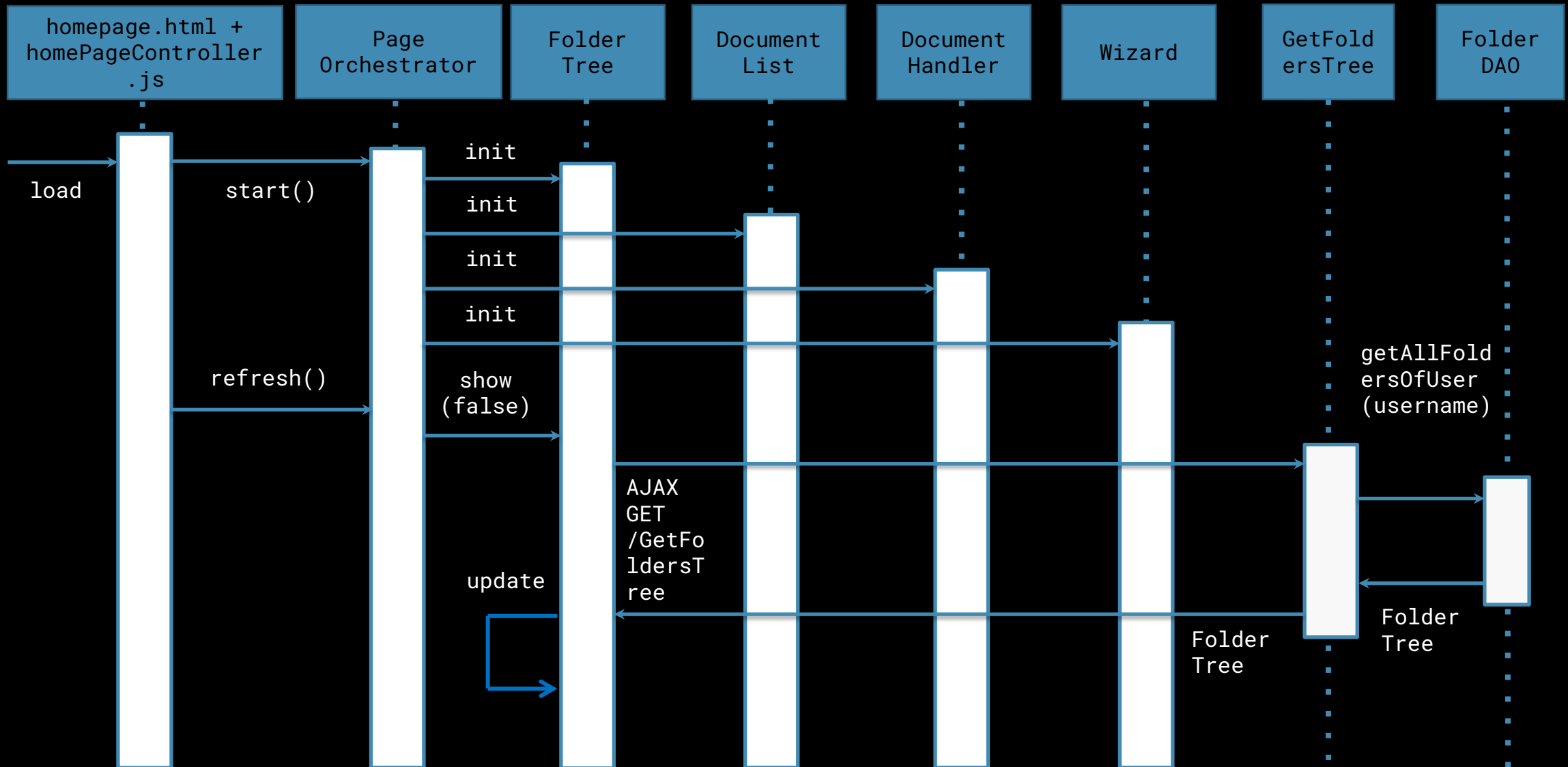
- **Data Access Objects (Classes)**

- UserDAO
- FolderDAO
- SubFolderDAO
- DocumentDAO

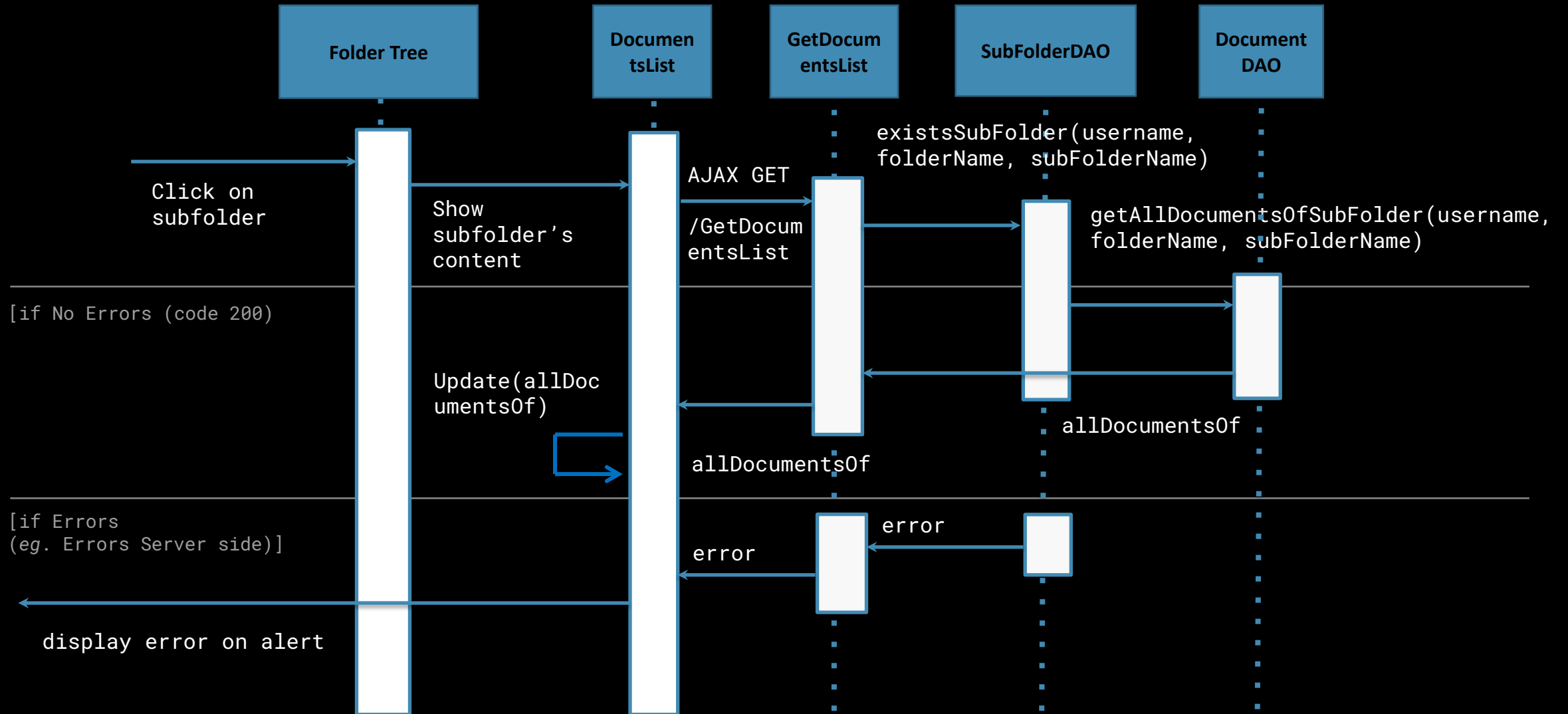
Client side: view & view component

- **Login**
 - Login form
 - Gestione del submit e degli errori
- **Subscribe**
 - Login form
 - Gestione del submit e degli errori
- **Home**
 - PageOrchestrator
 - start()
 - refresh()
 - FolderTree
 - show() and update(): ask the server for folderTree and show folder tree, create BIN and handle D&D
 - reset()
 - DocumentsList
 - show() and update(): ask the server for documents in a sub folder
- reset()
- DocumentHandler
 - show() and update(): ask the server for document's details and show details box, link for download
 - reset()
- Wizard
 - showWizard(): show correct form and callBak to handler function
 - reset()
 - changeVisibility(): hide other form and make visibile the correct form

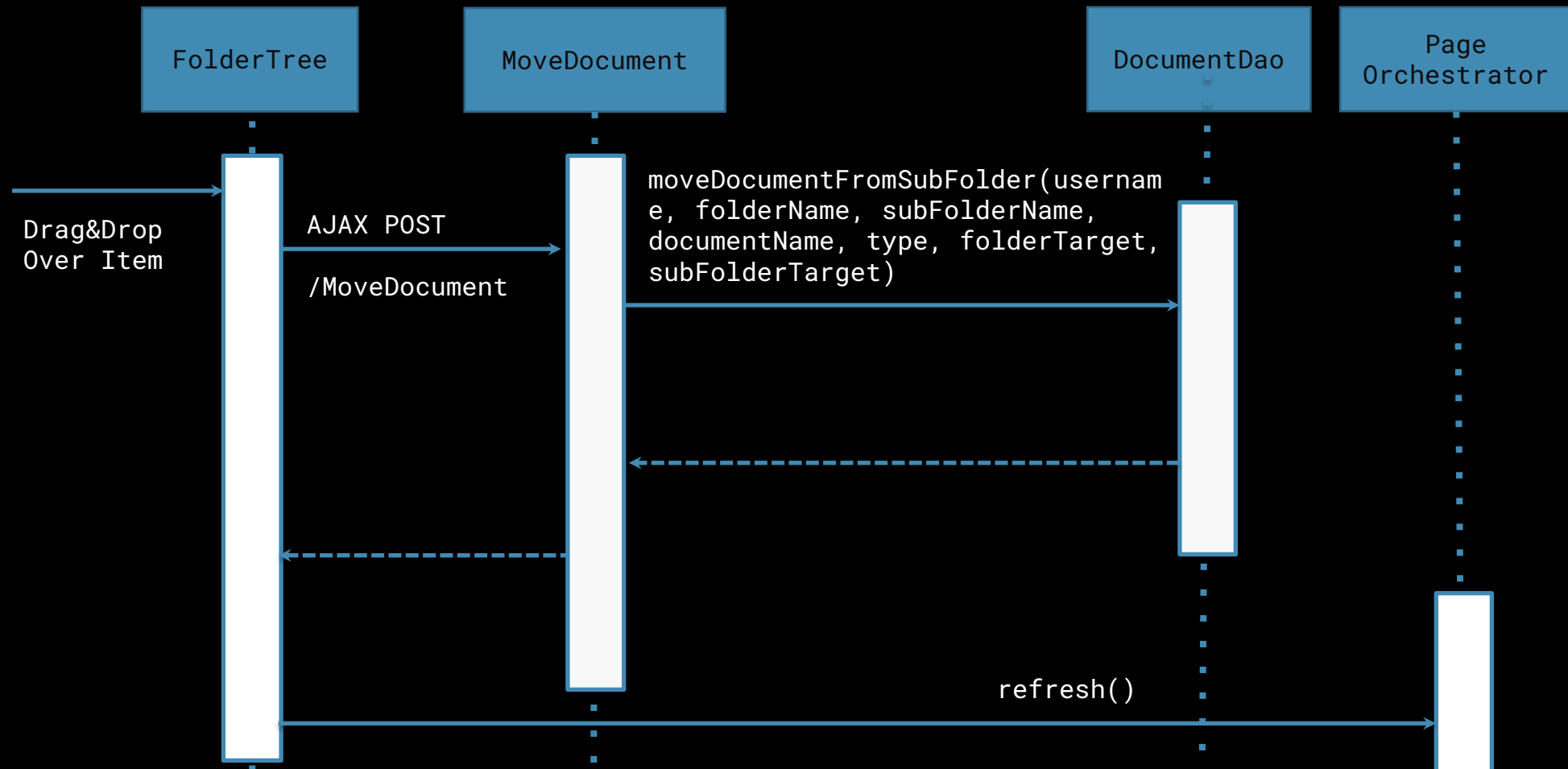
Event: first homepage load



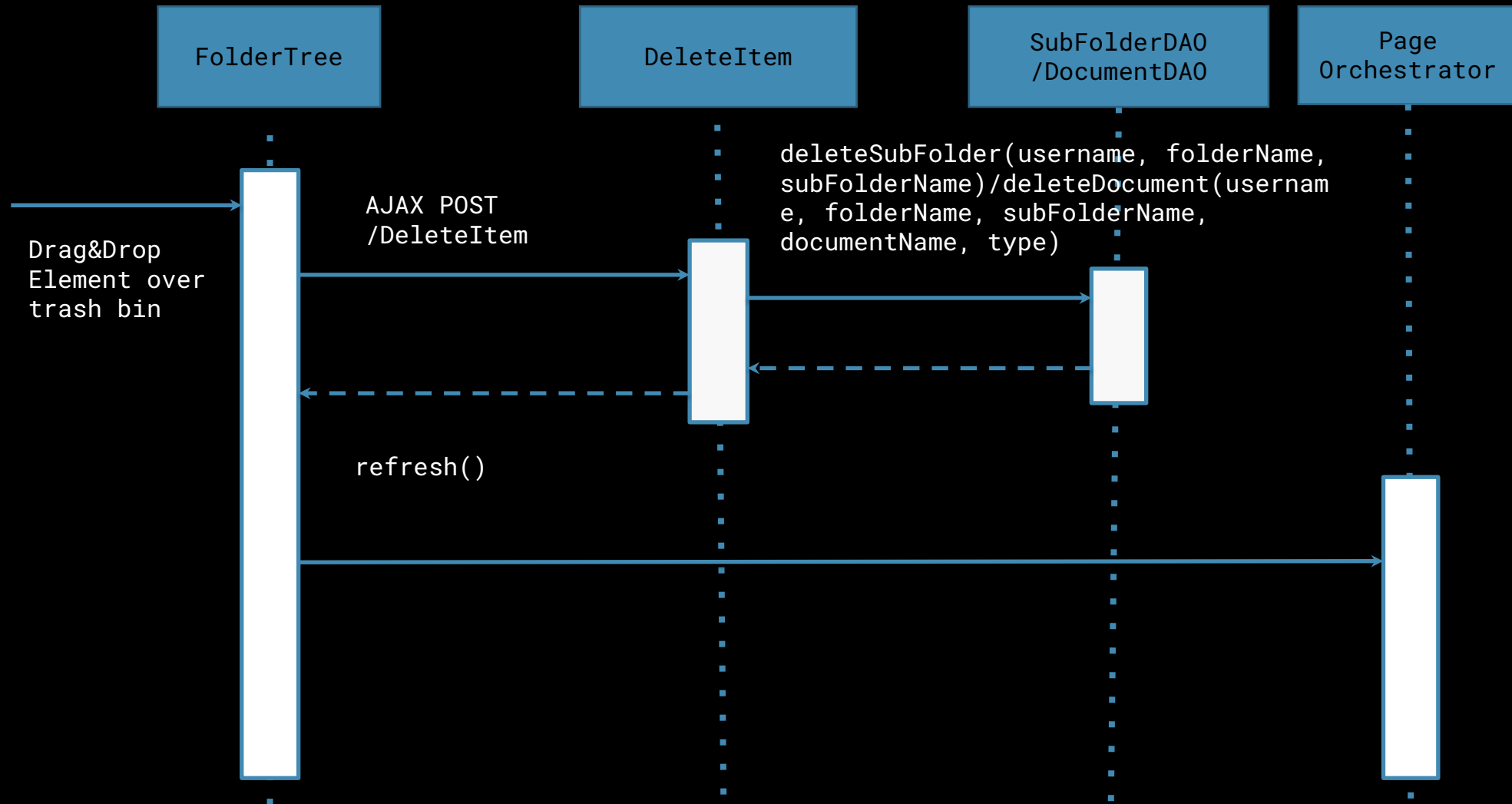
Event: show documentList



Event: move



Event: delete



Thanks! 😊



POLITECNICO
MILANO 1863