Progetto TIW 2022

Gruppo 34

Andrea Alari(10707336)
Giovanni Barbiero(10722159)

Versione PureHTML

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.

Aggiunta alle specifiche

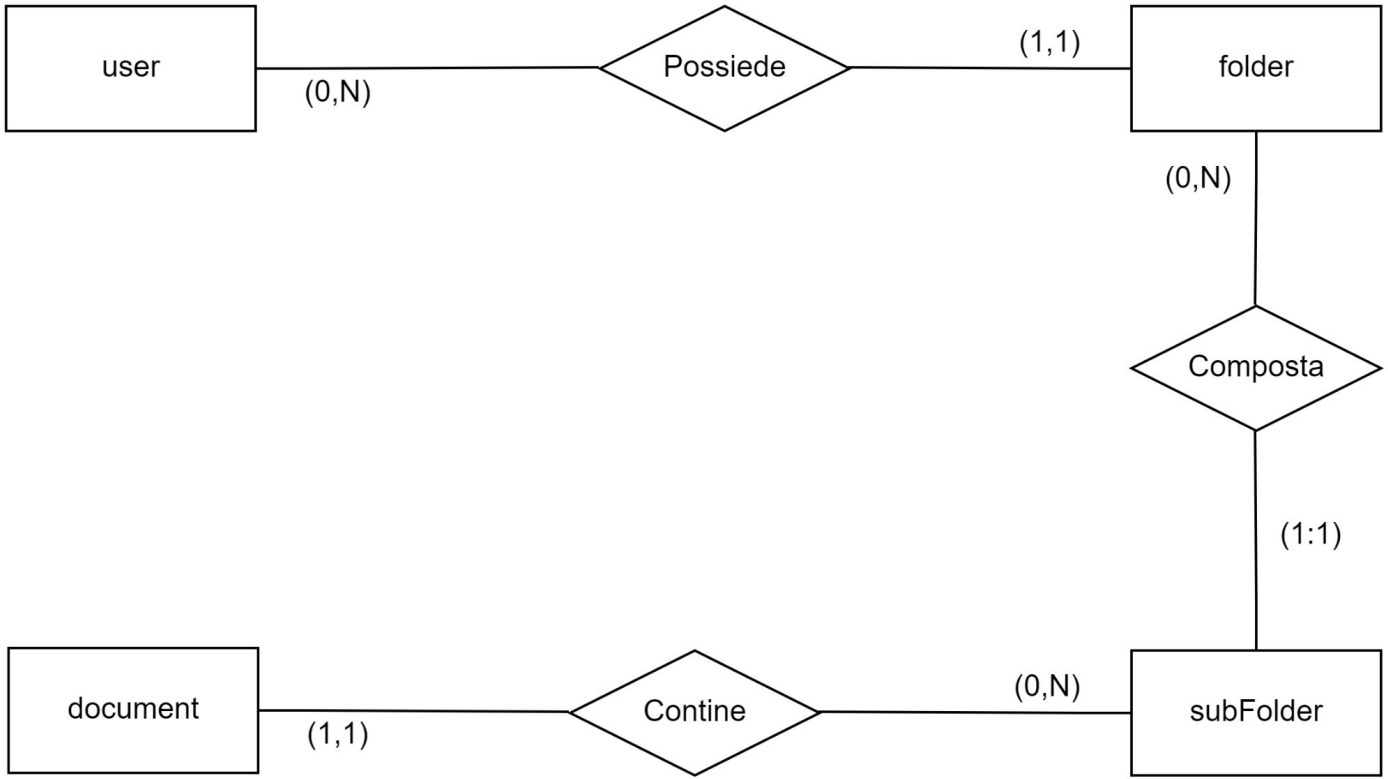
- È stato deciso di aggiungere il campo email e di lasciare la possibilità di effettuare il login sia tramite username che tramite email.
- La gestione del tasto back per tornare alla pagina precedente è stata implementata in modo tale che segua la gerarchia delle pagine.
- Si è deciso di implementare il controllo di avvenuto login tramite dei Filtri.
- Non è stato aggiunto un attributo per il proprietario di un documento per evitare ridondanza nel database in quanto si è assunto che il proprietario di cartelle, sottocartelle e documenti sia lo stesso utente.

Analisi del Testo

Entities, Attributes, Relationship

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. Albero delle cartelle e sottocartelle nella HOME PAGE Contenuto della pagina DOCUMENTI di una sottocartella 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

DataBase Design



Local DataBase Schema

```
CREATE TABLE `document` (
  `iddocument` int NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `format` varchar(10) NOT NULL,
  `summary` text NOT NULL,
  `creationDate` date NOT NULL,
  `subfolder_idsubfolder` int NOT NULL,
  PRIMARY KEY (`iddocument`),
  KEY `fk_document_subfolder1_idx`
(`subfolder_idsubfolder`),
  CONSTRAINT `fk_document_subfolder1` FOREIGN
KEY (`subfolder_idsubfolder`) REFERENCES
`subfolder` (`idsubfolder`) ON DELETE CASCADE ON
UPDATE CASCADE
)
```

```
CREATE TABLE `folder` (
  `idfolder` int NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `creationDate` date NOT NULL,
  `user_iduser` int NOT NULL,
  PRIMARY KEY (`idfolder`),
  KEY `fk_folder_user_idx` (`user_iduser`),
  CONSTRAINT `fk_folder_user` FOREIGN KEY
(`user_iduser`) REFERENCES `user` (`iduser`)
ON DELETE CASCADE ON UPDATE CASCADE
)
```

```
CREATE TABLE `subfolder` (
  `idsubfolder` int NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `creationDate` date NOT NULL,
  `folder_idfolder` int NOT NULL,
  PRIMARY KEY (`idsubfolder`),
  UNIQUE KEY `unique_name_per_folder` (`name`,`folder_idfolder`),
  KEY `fk_subforlder_folder1_idx` (`folder_idfolder`),
  CONSTRAINT `fk_subforlder_folder1` FOREIGN KEY
(`folder_idfolder`) REFERENCES `folder` (`idfolder`)
)

CREATE TABLE `user` (
  `iduser` int NOT NULL AUTO_INCREMENT,
  `username` varchar(20) NOT NULL,
  `email` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `name` varchar(20) NOT NULL,
  `surname` varchar(20) NOT NULL,
  PRIMARY KEY (`iduser`),
  UNIQUE KEY `username_UNIQUE` (`username`),
  UNIQUE KEY `email_UNIQUE` (`email`)
)
```

Application requirements analysis

Page, viewComponent, event, action

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 (submit)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(utente preme il link)(action). L'applicazione consente il (bottone logout)logout dell'utente. Una pagina GESTIONE CONTENUTI raggiungibile dalla HOME PAGE permette all'utente di (form creazione)creare (creazione)una cartella, una sottocartella di una cartella esistente e un documento all'interno di una sottocartella.

Viste e Componenti

Controllers:
-Login
-Register
-Logout
-HomePage
-MoveDocument
-CreateFolder
-CreateSubFolder
-CreateDocument
-Documents
-DocumentDetails
-ContentManagement

Model objects Beans (Beans)
-User
-Folder
-SubFolder
-Document

Views:
-Login
-Register
-HomePage
-Documents
-Document
-ContentManagement

Filters:
LoggedInChecker
LoggedOutChecker

Data Access Object(Classes)

UserDAO

- -checkUsernameCredentials (username, password)
- -checkEmailCredentials (email, password)
- -addUser (username, name, surname, password, email)
- -doesUsernameExists (username)
- -doesEmailExists (email)

FolderDAO

- -checkOwner (folderId, userId)
- -getFolder (folderId)
- -createFolder (username, folderName)
- -getFoldersWithSubFolders (username)

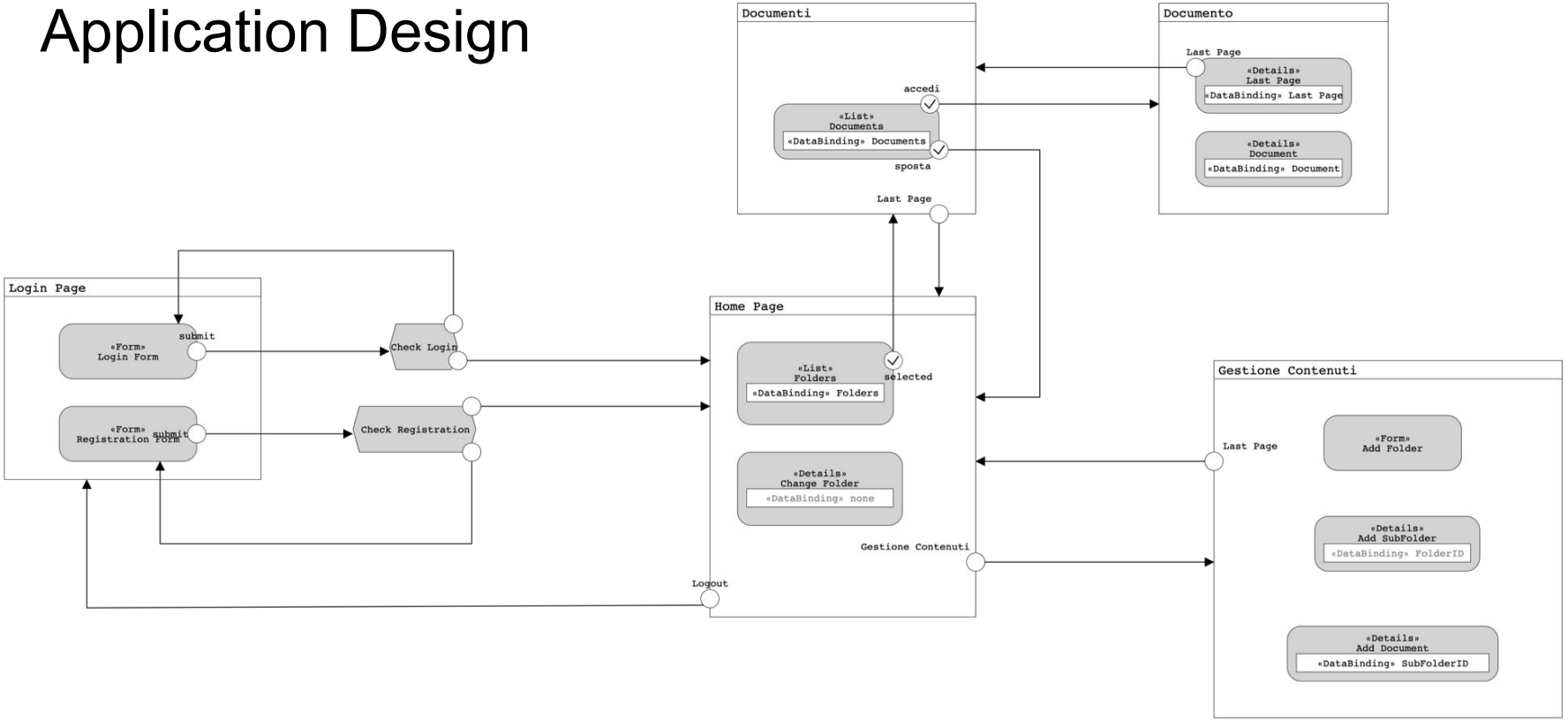
SubFolderDAO

- -checkOwner(username, subFolderId)
- -getSubFolder (subFolderId)
- -createSubFolder (username, folderId,subFolderName)
- -getDocuments (username, subFolderId)

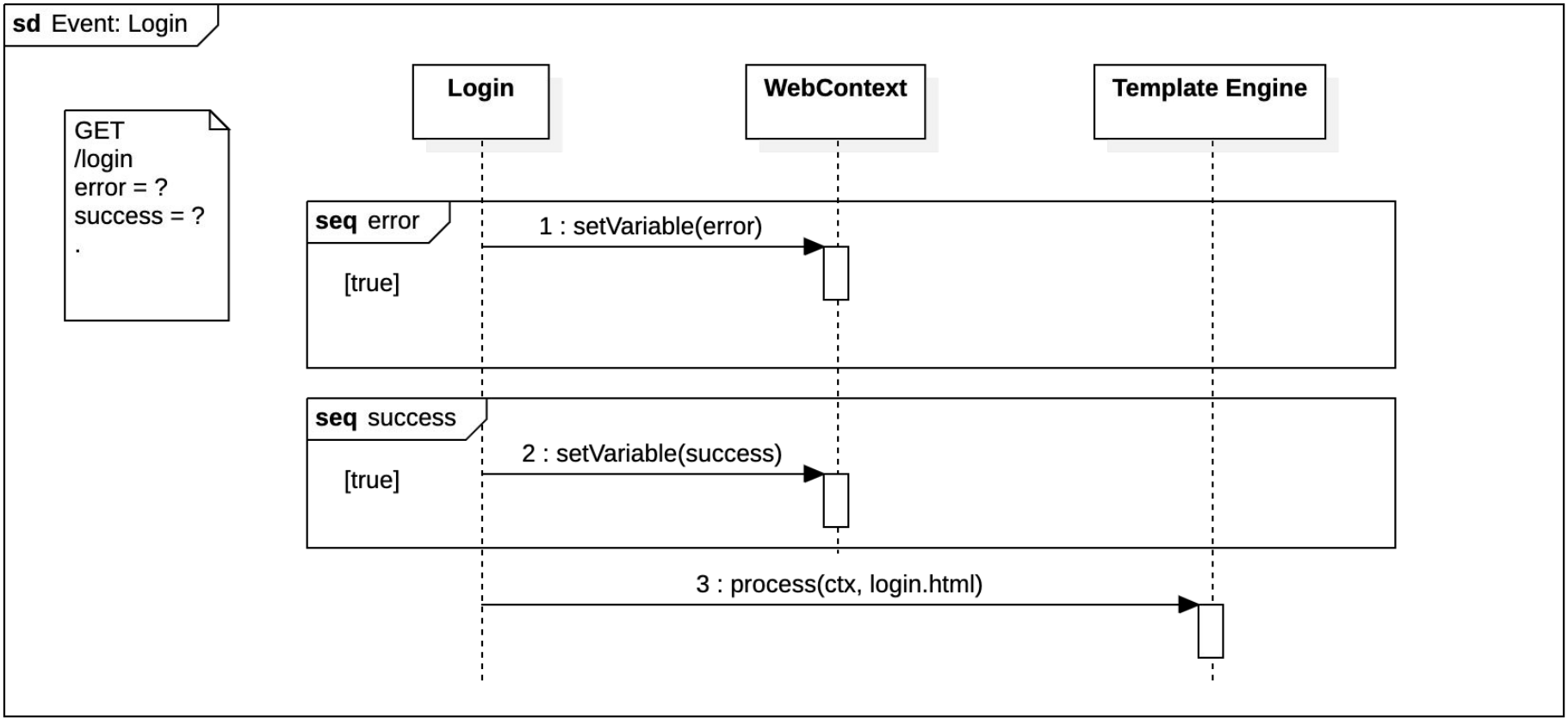
DocumentDAO

- -checkOwner(username, documentId)
- -getDocument (documentId)
- -moveDocument (username, subFolderId, documentId)
- -createDocument (name, format, summary, subfolderId)

Application Design

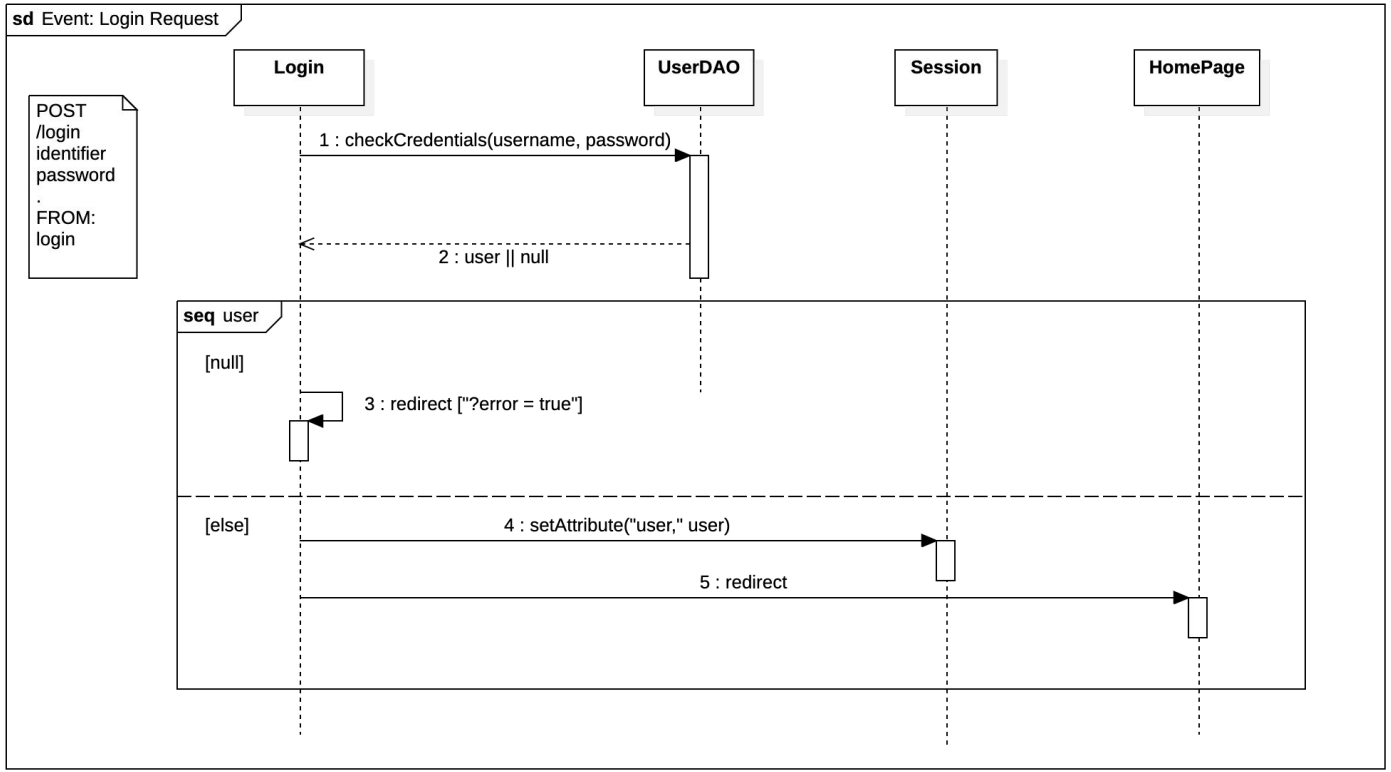


Event Login

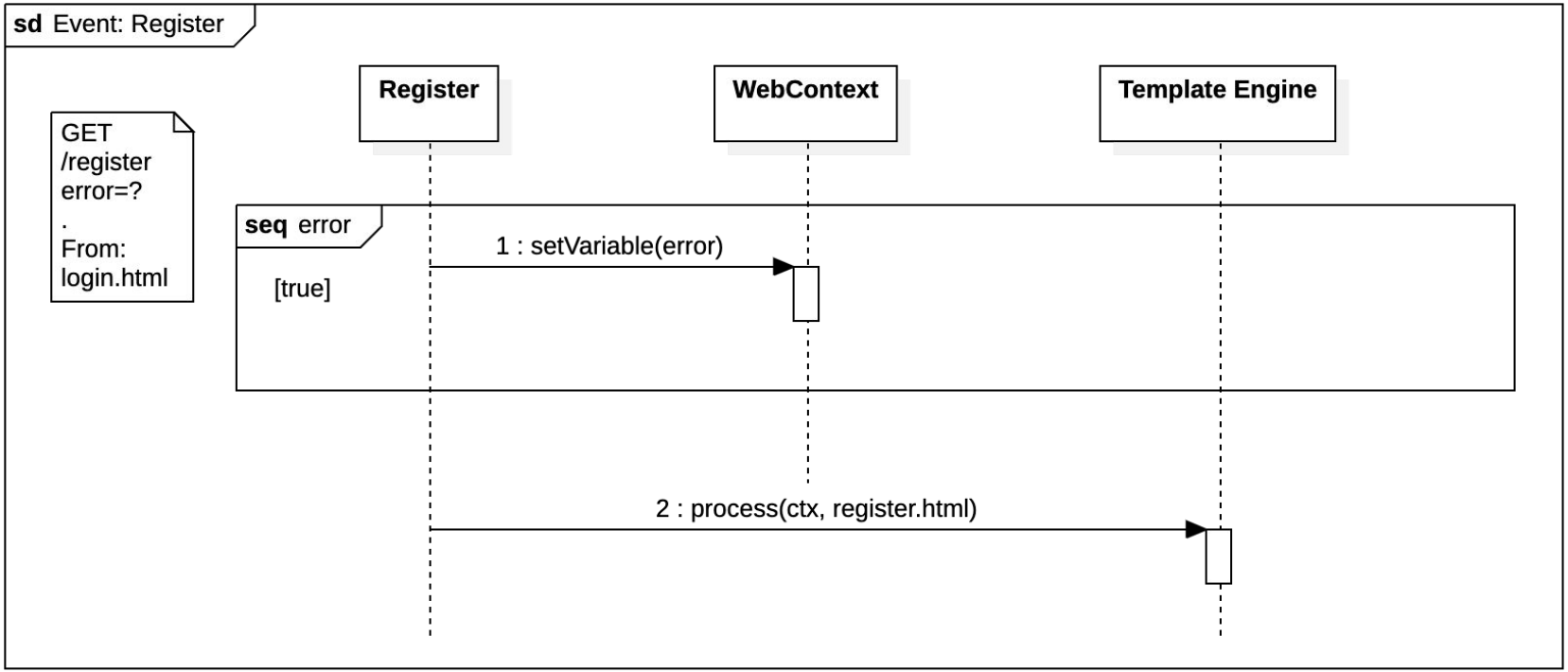

sd Event: Login
GET
/login
error = ?
success = ?
.
Login
WebContext
Template Engine
seq error
1 : setVariable(error)
[true]
seq success
2 : setVariable(success)
[true]
3 : process(ctx, login.html)

Event Login Request

in caso di eccezione SQL viene inviato l'errore INTERNAL_SERVER_ERROR(500)

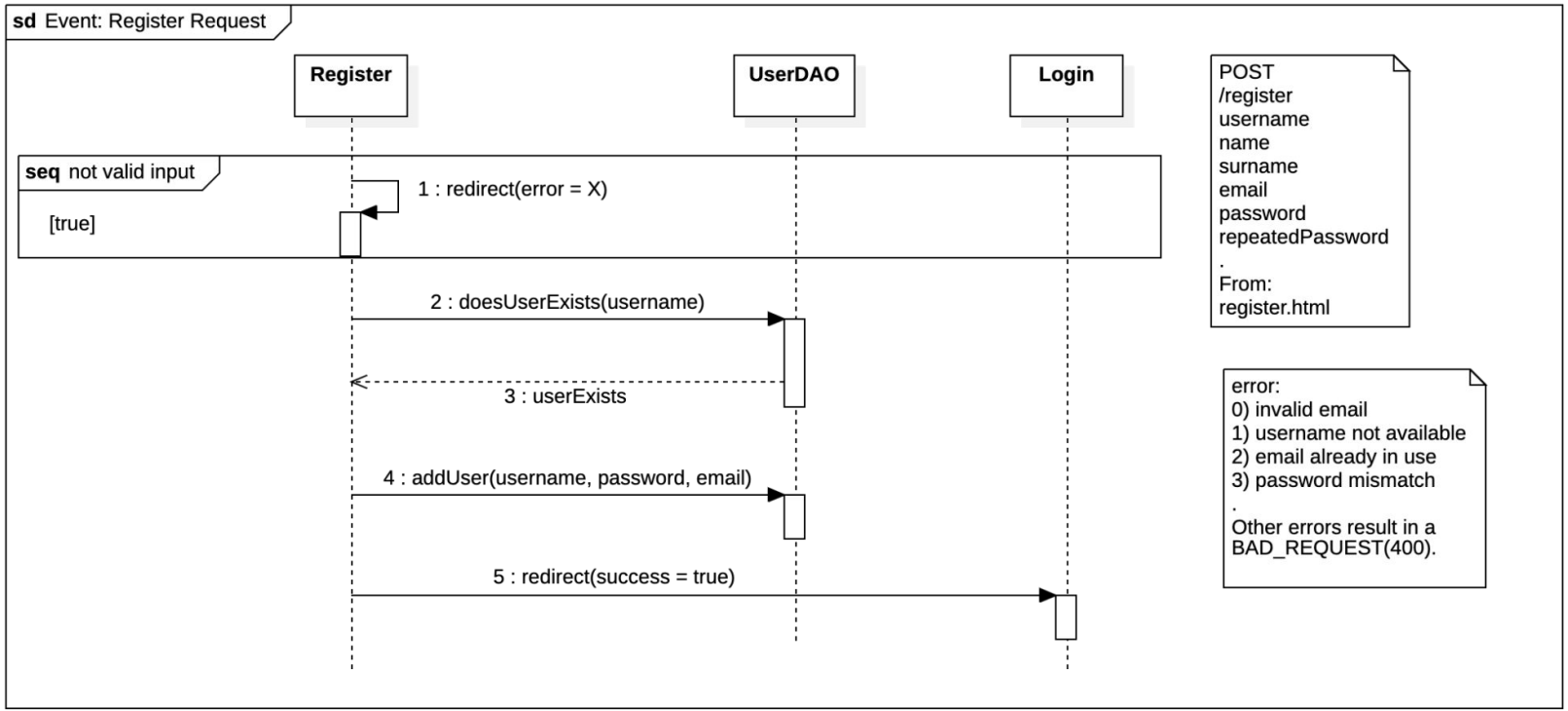


Event Register



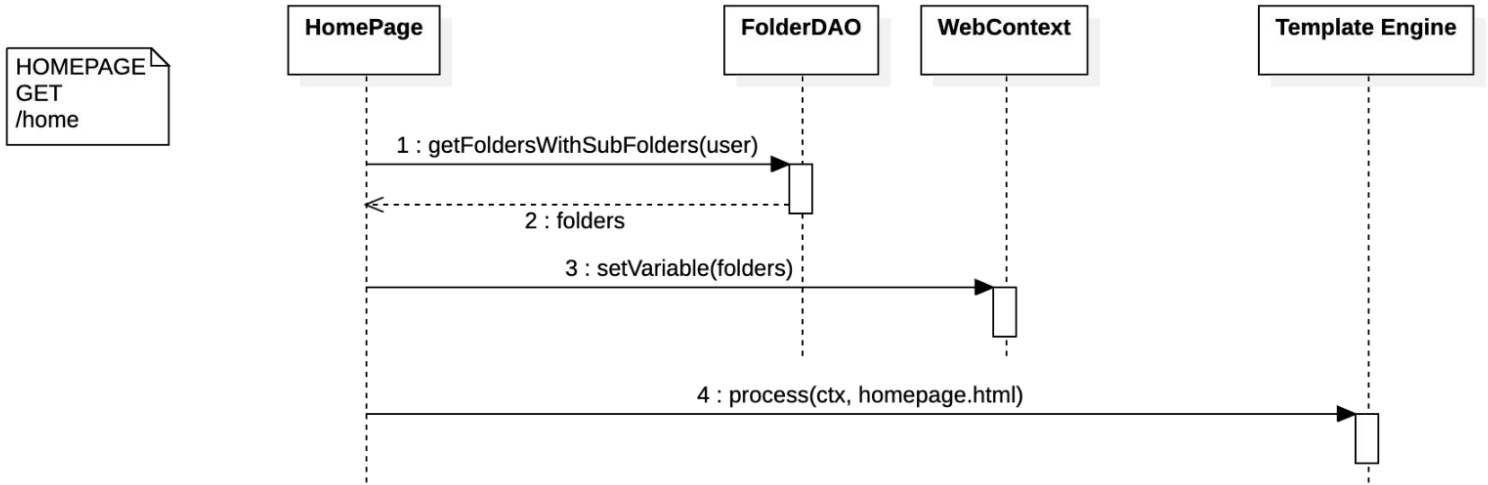
Event Register Request

in caso di eccezione SQL viene inviato l'errore INTERNAL_SERVER_ERROR(500)



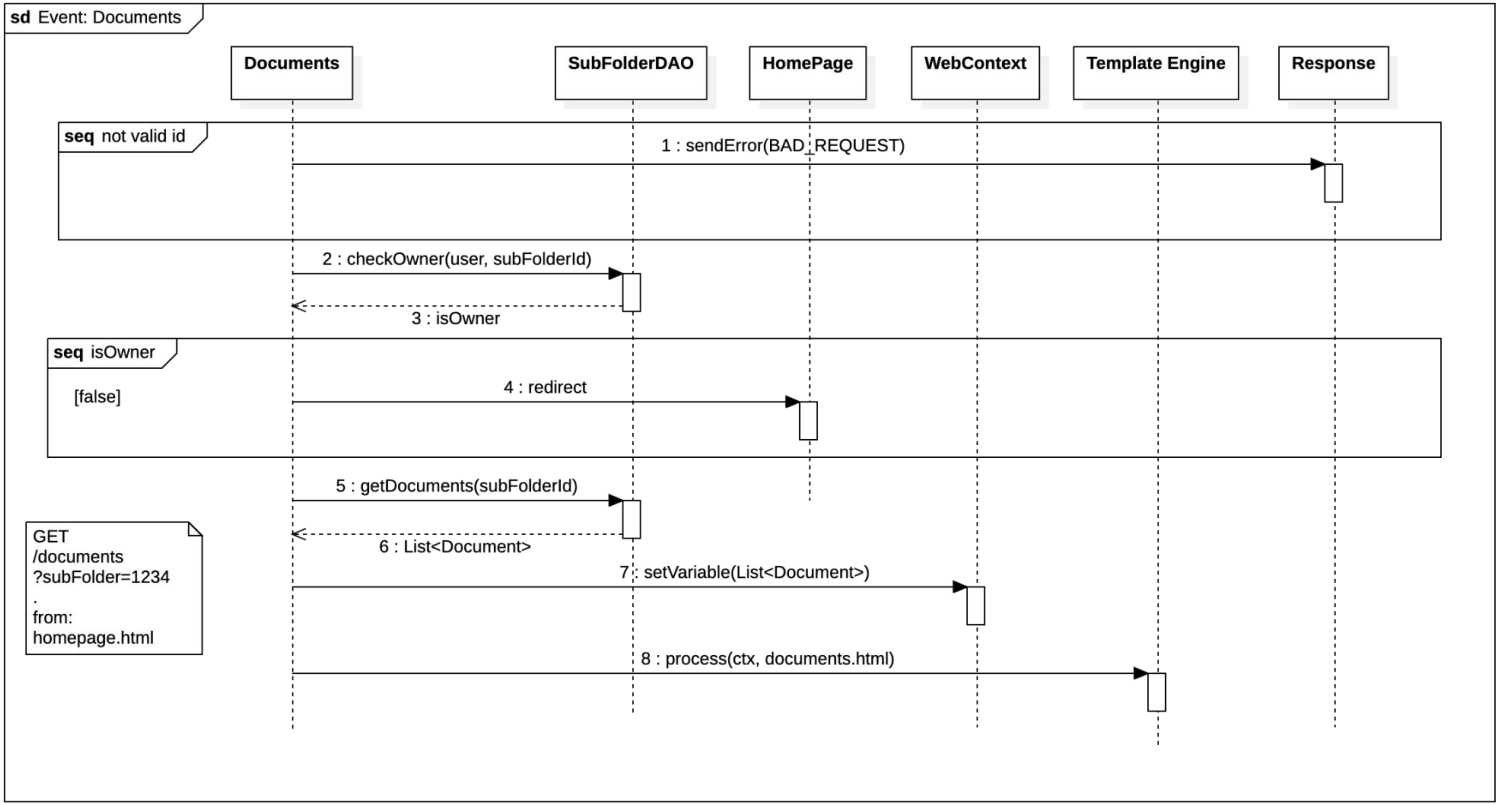
Event Home Page

in caso di eccezione SQL viene inviato l'errore INTERNAL_SERVER_ERROR(500)



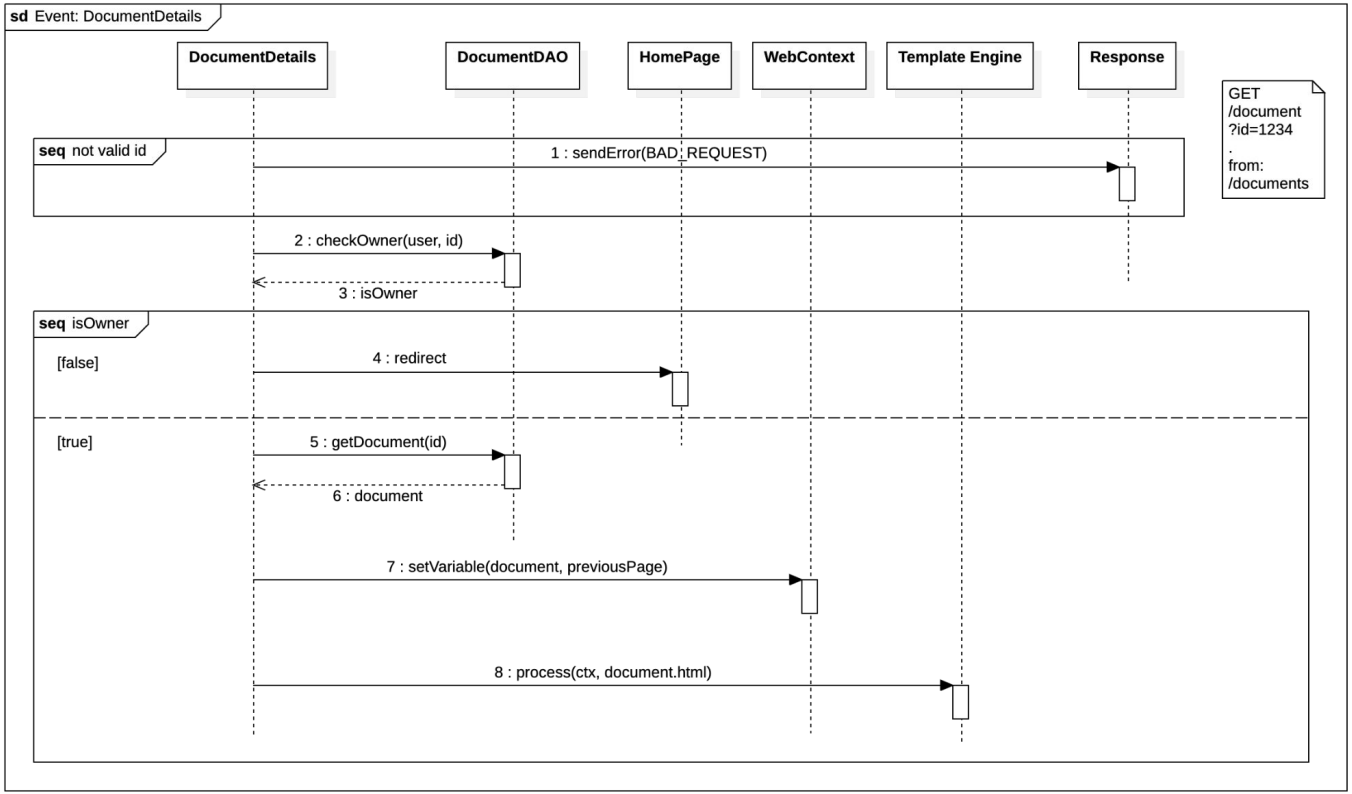
Event Documents

in caso di eccezione SQL viene inviato l'errore INTERNAL_SERVER_ERROR(500)



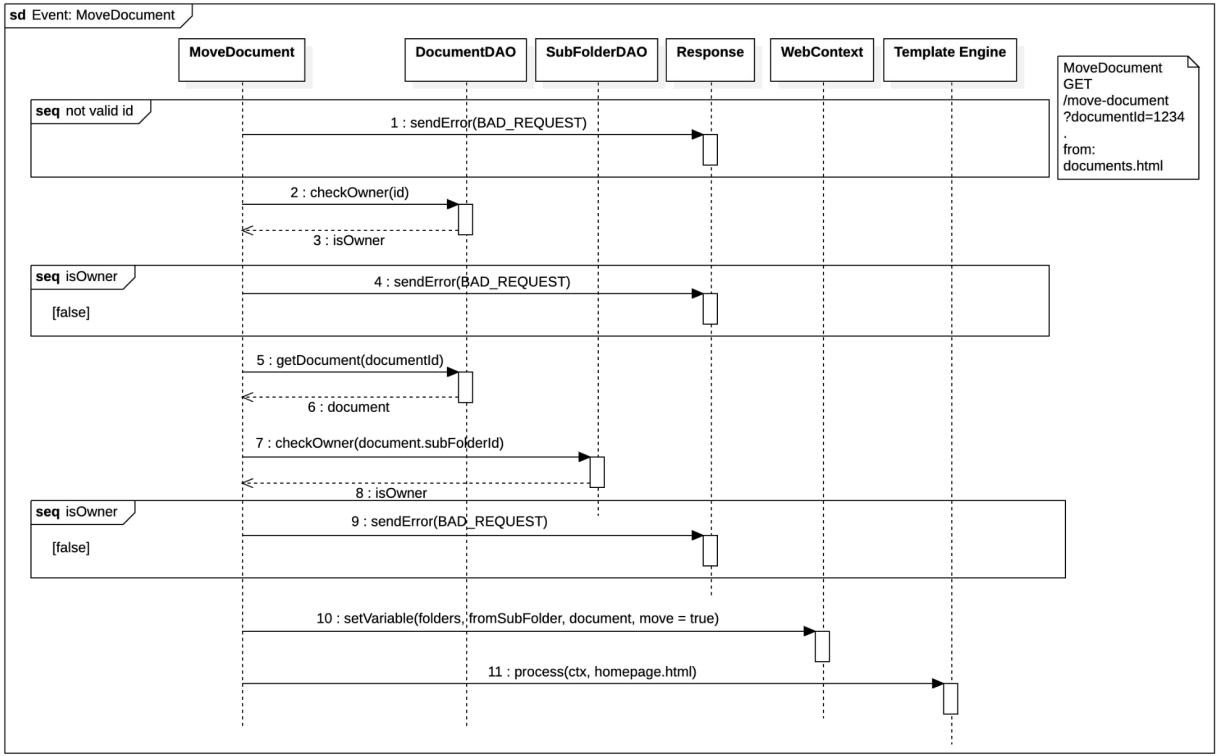
Event Document Details

in caso di eccezione SQL viene inviato l'errore INTERNAL_SERVER_ERROR(500)



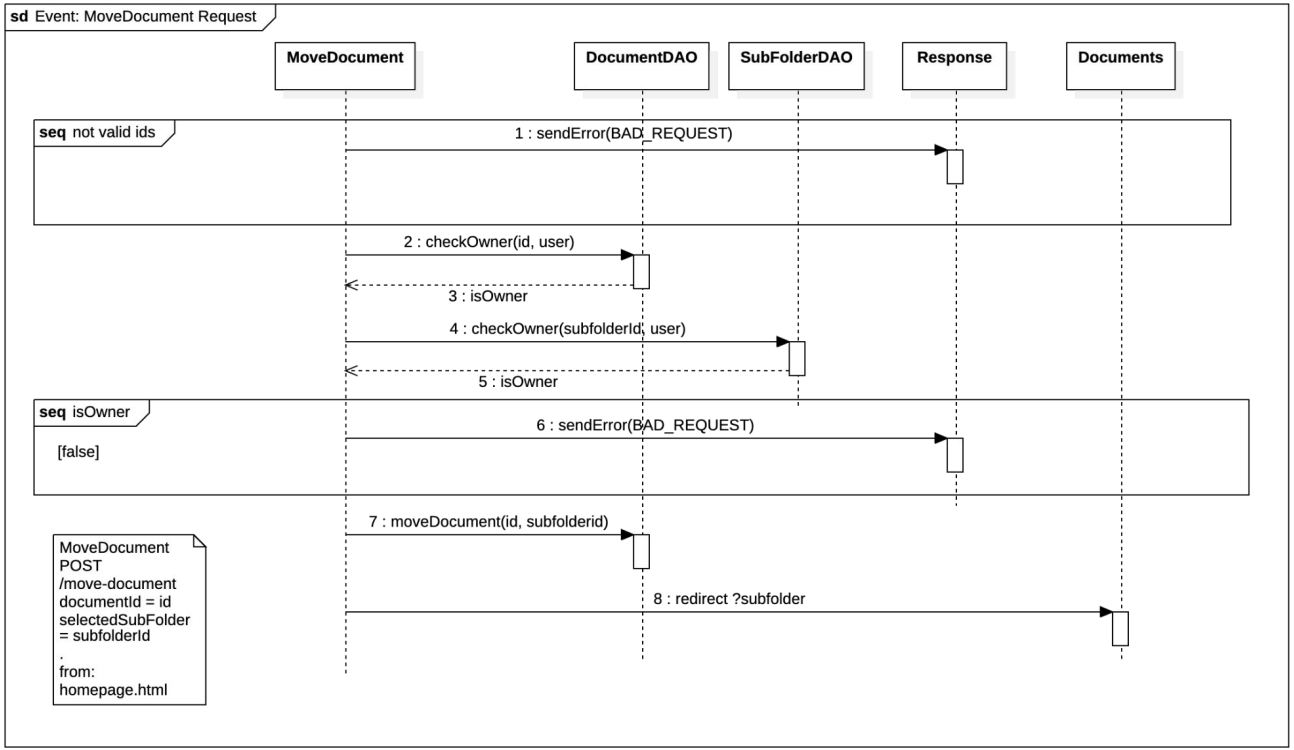
Event Move Document

in caso di eccezione SQL viene inviato l'errore INTERNAL_SERVER_ERROR(500)



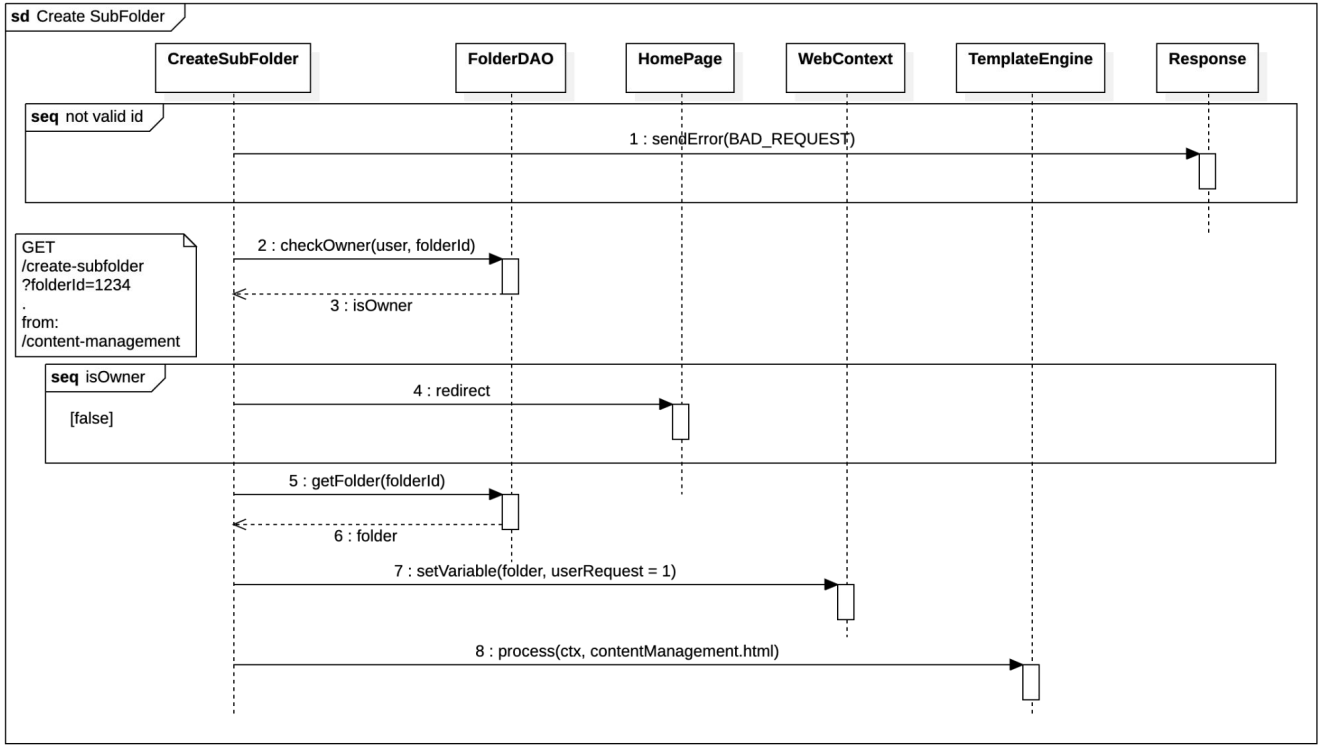
Event Move Document Request

in caso di eccezione SQL viene inviato l'errore INTERNAL_SERVER_ERROR(500)



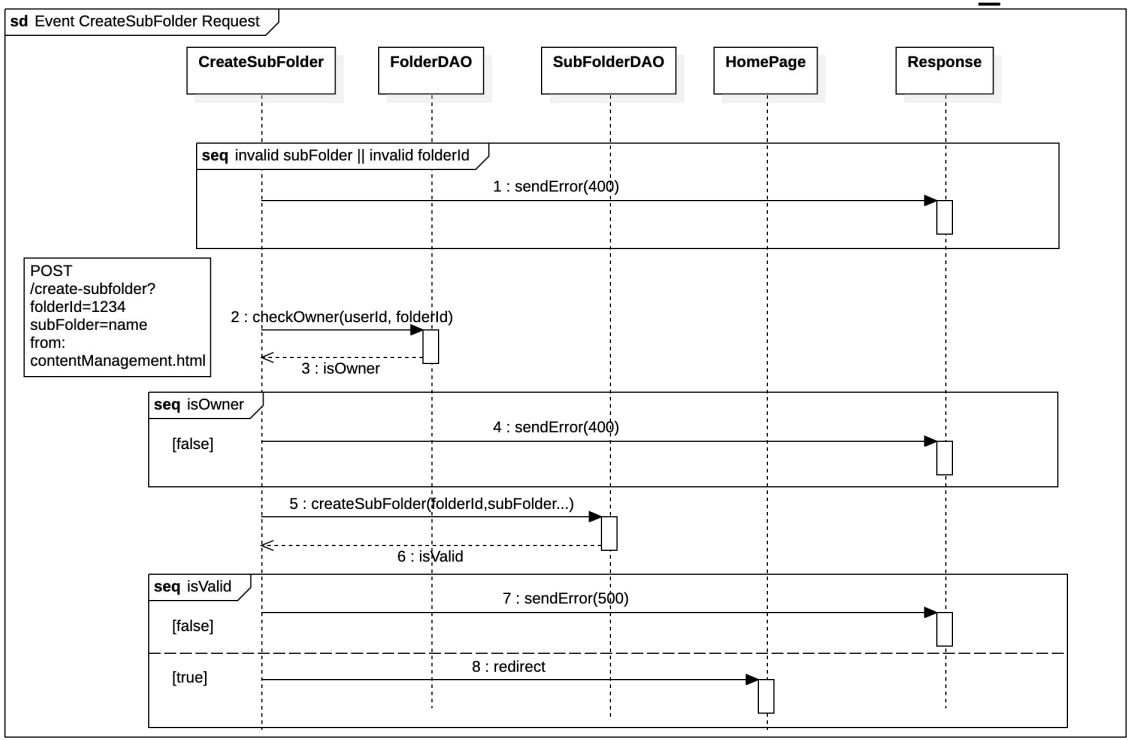
Event Create SubFolder

in caso di eccezione SQL viene inviato l'errore INTERNAL_SERVER_ERROR(500)



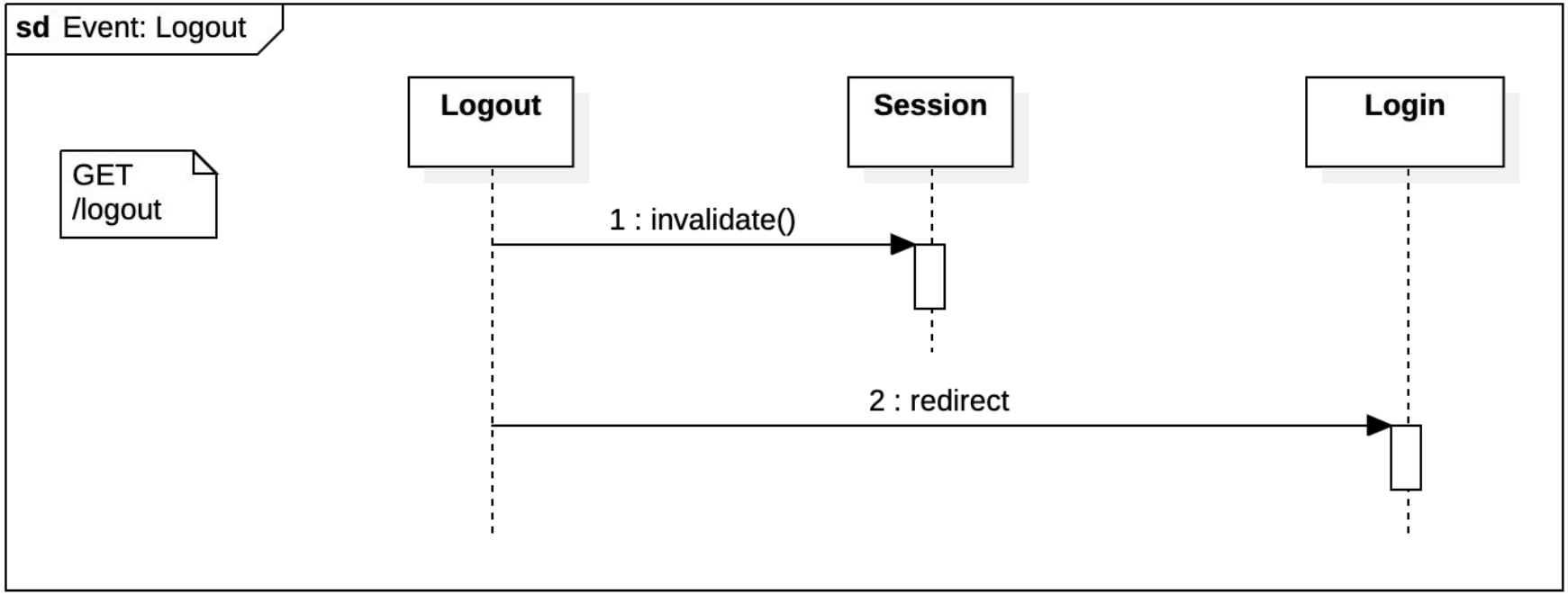
Event Create SubFolder Request

in caso di eccezione SQL viene inviato l'errore INTERNAL_SERVER_ERROR(500)



Per evitare ripetizioni sono stati omessi i sequence diagram della create Folder e create Document in quanto simili a create SubFolder

Event Logout



Filtri

LoggedInChecker: il filtro controlla che l’utente abbia effettuato il login, altrimenti reindirizza l’utente alla pagina di login.

LoggedOutChecker: il filtro controlla che l’utente non abbia effettuato il login, altrimenti reindirizza l’utente alla home page.

Versione RIA

Completamento delle specifiche

Page, viewComponent, event, action

- La registrazione (submit) controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password", anche a lato client.
- Dopo il login, 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 sottocartella.
- La funzione di creazione di un documento è realizzata nella pagina HOME mediante un bottone AGGIUNGI DOCUMENTO posto di fianco ad ogni sottocartella. 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 & 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

Aggiunta alle specifiche

- Abbiamo dato la possibilità di effettuare il login tramite username o email.
- Per una resa grafica più gradevole è stato introdotto nella home page un bottone denominato EDIT, alla pressione di quest'ultimo vengono mostrati i bottoni per creare nuovi contenuti e se viene nuovamente premuto questi ultimi vengono nascosti.
- Non è stato aggiunto un attributo per il proprietario di un documento per evitare ridondanza nel database in quanto si è assunto che il proprietario di cartelle, sottocartelle e documenti sia lo stesso utente.
- Abbiamo dato la possibilità di creare anche le cartelle.

Sommario

Viste e componenti

- Home
  - -Lista Folders
  - -Bottoni Show Details
  - -Bottone Hide Details
  - -Dettagli document
  - -Bottone EDIT
  - -Bottoni creazione
  - -Form creazione folder
  - -Form creazione subfolder
  - -Form creazione document
  - -Logout
  - -Cestino
- Login
  - -Login Form
  - -Register Link
- Register
  - -Subscription Form
  - -Login Link

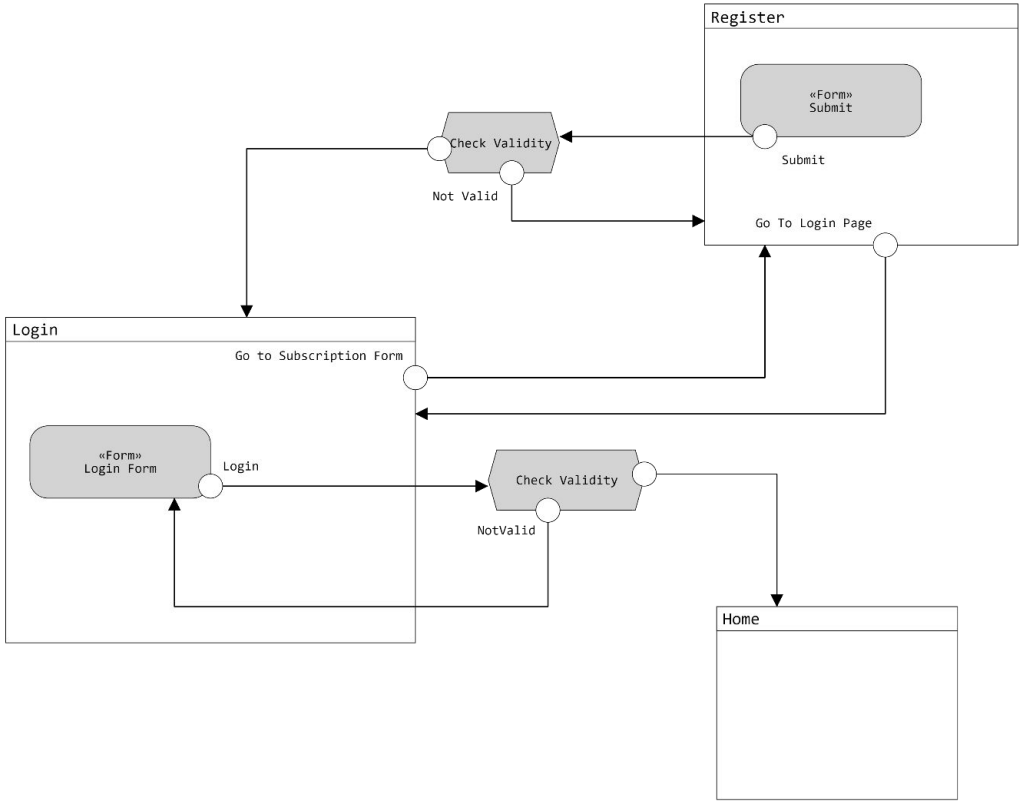
Eventi e azioni

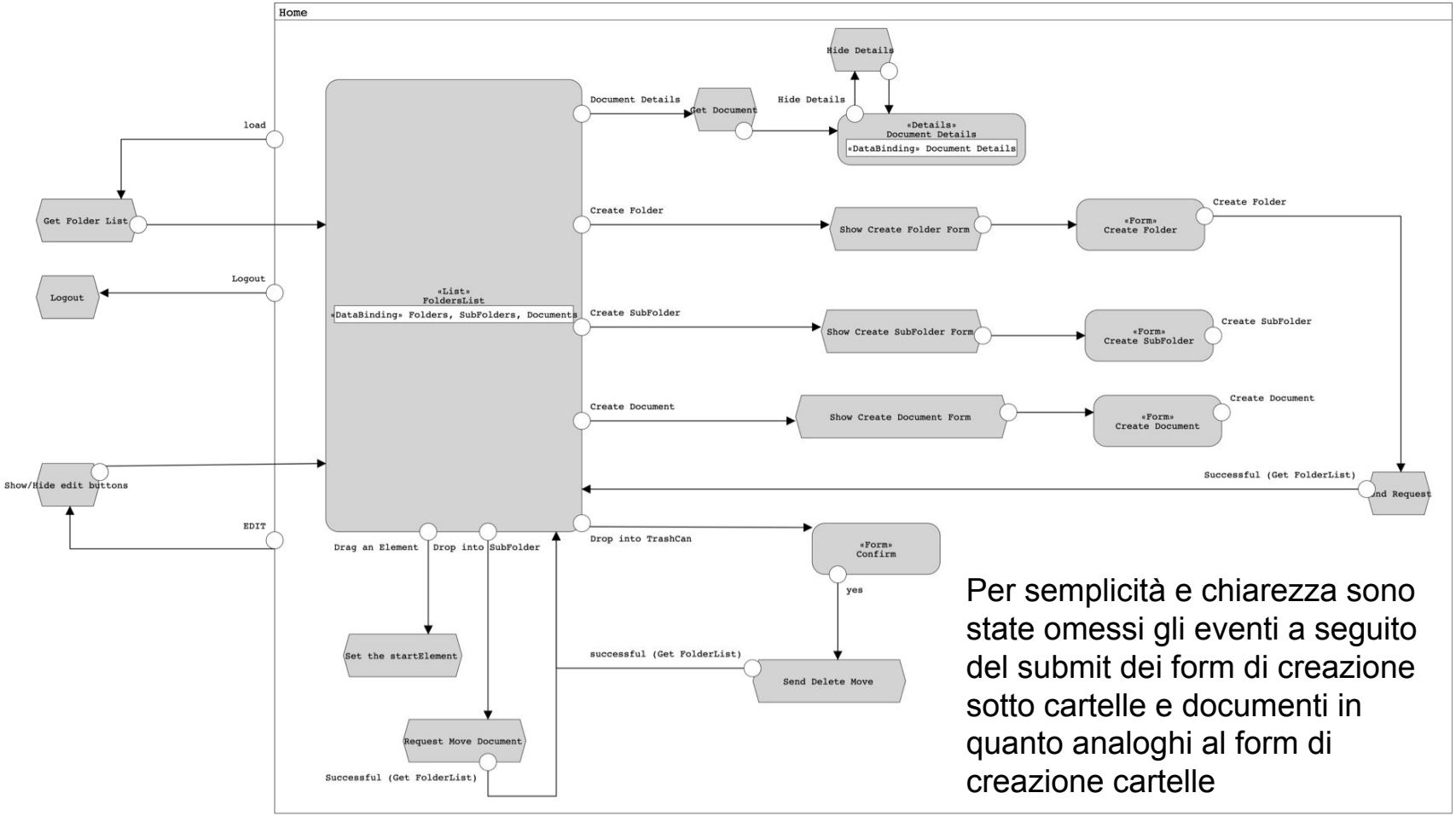
Login
- -Click su bottone Login
  invio richiesta di login
- -Click link subscription Form
  redirect to register page

Register
- -Click su bottone Submit
  check inputs e invio richiesta registrazione
- -Click link Login Page
  redirect to login page

Home
- -Click bottone Logout
  richiesta logout e redirect to login
- -Click bottone Show Details
  richiesta dettagli documento e show contenuto
- -Click bottone Hide Details
  hide dettagli documento
- -Drag and Drop Document in a Subfolder
  richiesta di spostamento documento
- -Drag and Drop in TrashCan
  richiesta eliminazione contenuto
- -Click bottone EDIT
  show/hide bottoni creazione
- -Click bottone Create a Folder/SubFolder/Document
  show form creazione Folder/SubFolder/Document
- -Click bottone Create a Folder/SubFolder/Document
  (nel form di creazione)
  richiesta creazione Folder/SubFolder/Document

Application Design





Per semplicità e chiarezza sono state omessi gli eventi a seguito del submit dei form di creazione sotto cartelle e documenti in quanto analoghi al form di creazione cartelle

Eventi e Azioni

Login

Evento	Azione	Evento	Azione
login	controllo dati	POST(username, password)	controllo credenziali
go to subscription form	open link	GET	-

Register

Evento	Azione	Evento	Azione
submit	controllo dati	POST (username, name, surname, email, password)	creazione utente
go to login page	open link	GET	-

Home

Evento	Azione	Evento	Azione
Logout	logout makeCall	Get	logout
Show Details	show document Details ShowDocument.showDocument()	Get (documentId)	get Document
Hide Details	hide detail	-	-
Drag and Drop Document in a Subfolder	move a document drop event defined in DragAndDropManager	POST(documentId, subFolderId)	move document
Drag and Drop in TrashCan	delete a content drop event defined in DragAndDropManager	POST(id)	delete a folder, subfolder or document

Evento	Azione	Evento	Azione
EDIT	show/hide create buttons FolderList.edit()	-	-
Create Folder	show create Folder Form CreateFolder.enableForm()	-	-
Create SubFolder	show create SubFolder CreateSubFolder.enableForm()	-	-
Create Document	show create Folder CreateFolder.enableForm()	-	-
Create Folder (in form)	create a Folder makeCall()	POST(folderName)	create a new folder
Create SubFolder (in form)	create a SubFolder sendFormData()	POST(subFolderName)	create a new subfolder
Create Document (in form)	create a document sendFormData()	POST(name, format, summary)	create a new document

Server Side: View & View components

Controllers:
-CreateDocument
-CreateFolder
-CreateSubFolder
-DeleteDocument
-DeleteFolder
-DeleteSubFolder
-DocumentDetails
-GetFolder
-Login
-Logout
-MoveDocument
-Register

Model Objects:
-Document
-Folder
-SubFolder
-User

Data Access Object(Classes)

UserDAO
- -checkUsernameCredentials (username, password)
- -checkEmailCredentials (email, password)
- -addUser (username, name, surname, password, email)
- -doesUsernameExists (username)
- -doesEmailExists (email)

FolderDAO
- -checkOwner (folderId, userId)
- -createFolder (username, folderName)
- -getFoldersWithSubFoldersAndDocuments (username)
- -deleteFolder (folderId)

SubFolderDAO
- -checkOwner(username, subFolderId)
- -createSubFolder (username, folderId,subFolderName)
- -deleteSubFolder (subFolderId)

DocumentDAO
- -checkOwner(username, documentId)
- -getDocument (documentId)
- -moveDocument (username, subFolderId, documentId)
- -createDocument (name, format, summary, subfolderId)
- -deleteDocument(documentId)

Filters:
-LoggedInChecker
-LoggedOutChecker
-LoggedOutCheckerRedirect

Filtri

LoggedInChecker: il filtro controlla che l’utente abbia effettuato il login, altrimenti reindirizza l’utente alla pagina di login.

LoggedOutChecker: il filtro controlla che l’utente non abbia effettuato il login, altrimenti invia un errore di tipo BAD_REQUEST(400) in caso la richiesta sia in metodo POST.

LoggedOutCheckerRedirect: il filtro controlla che l’utente non abbia effettuato il login altrimenti reindirizza l’utente alla pagina principale.

Client side : JavaScript

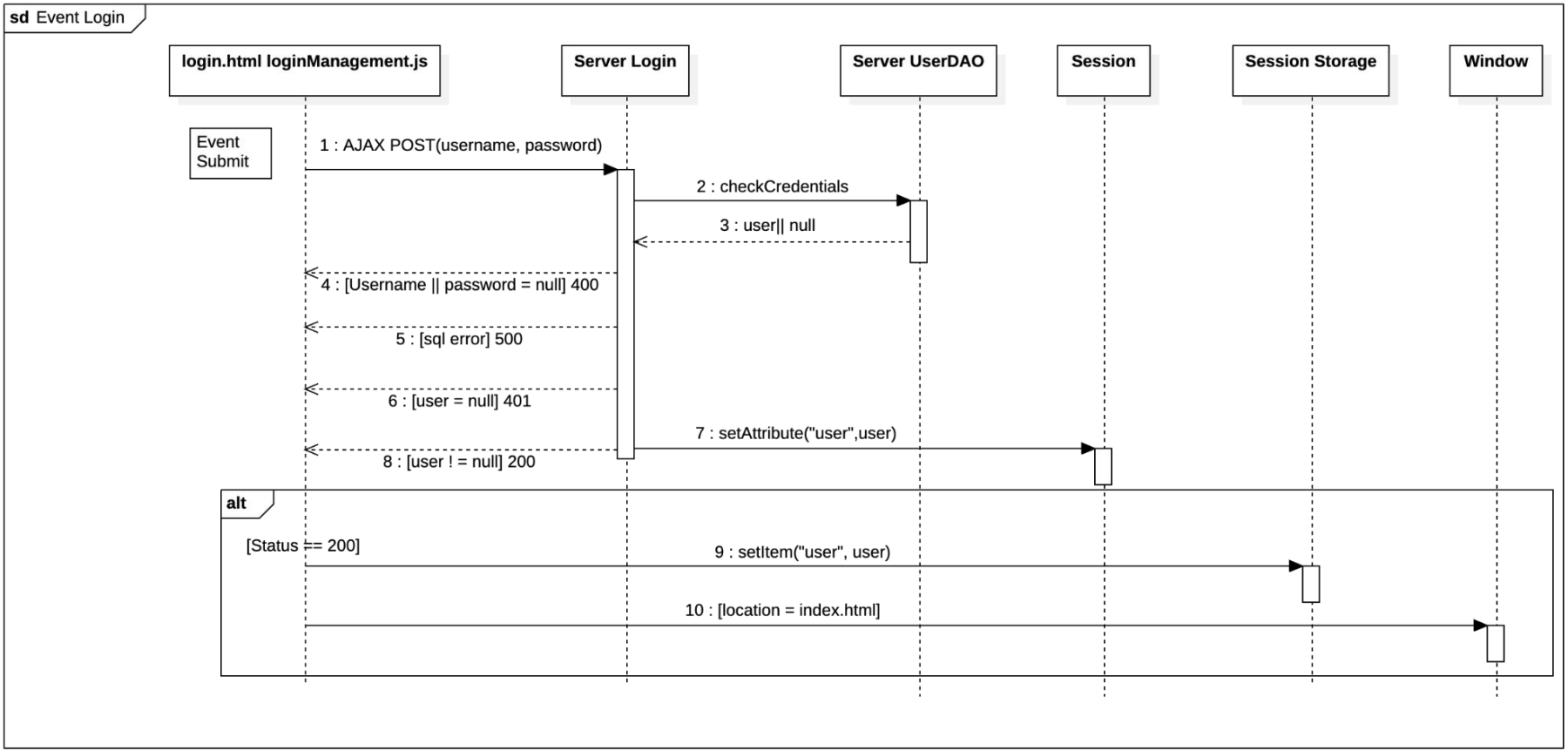
utils.js:
makeCall()
sendFormData()
checkEmail()

loginManagment.js:
function()

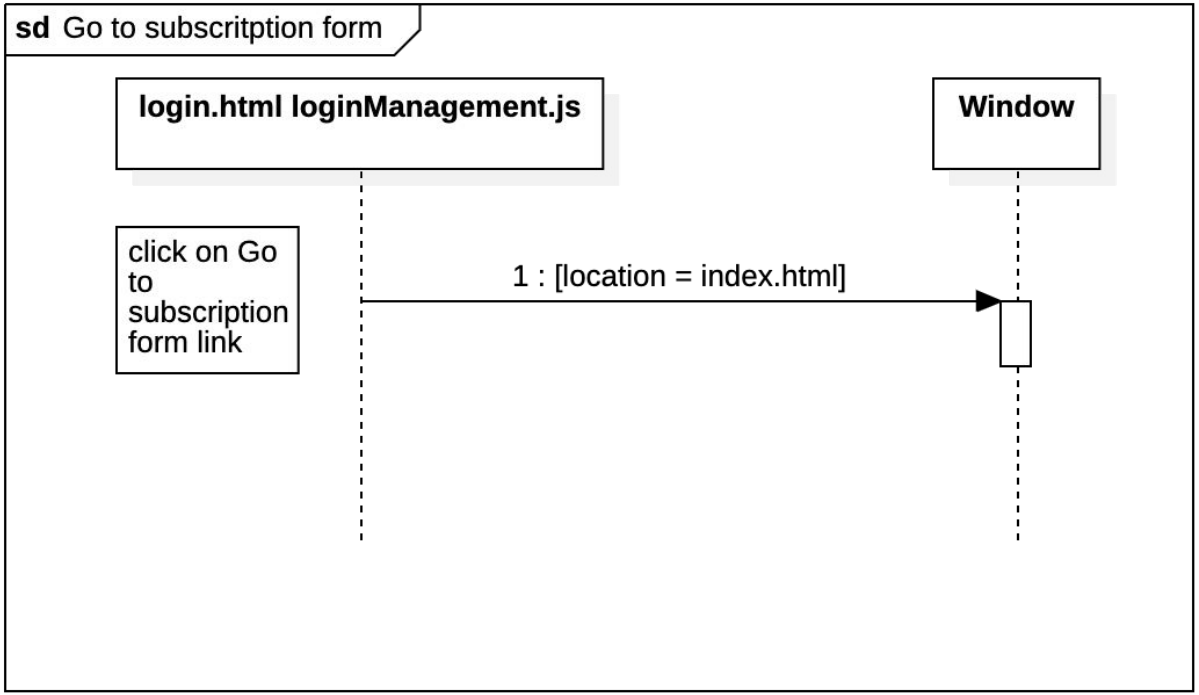
registerManagment.js:
function()

homeManagment.js:
-PageOrchestrator
    start()
    refresh()
    hideContent()
-FolderList
    show()
    edit()
    undo()
    showContent()
-CheckResponse()
-DragAndDropManager
    setupDragAndDrop()
    setMove()
    setDelete()
    setTrashCan()
    findNotDroppable()
    setDocumentDrop()
    resetDroppable()
-ShowDocument
    hide()
    showDocument()
    setDocumentDetails()
-CreateFolder
    hide()
    enableForm()
-CreateSubFolder
    hide()
    enableForm()
-CreateDocument
    hide()
    enableForm()
-start()
-logout()

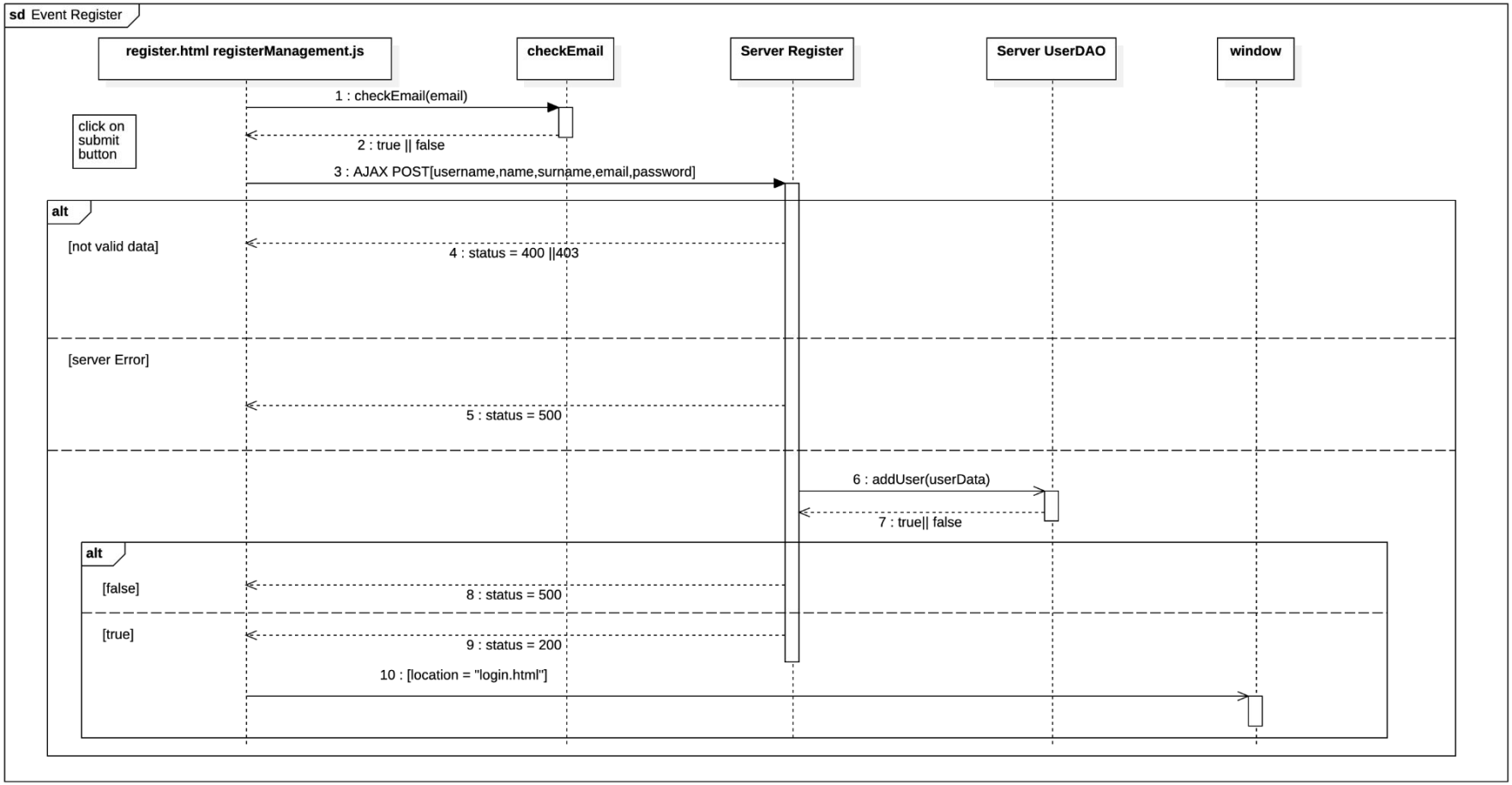
Evento Login



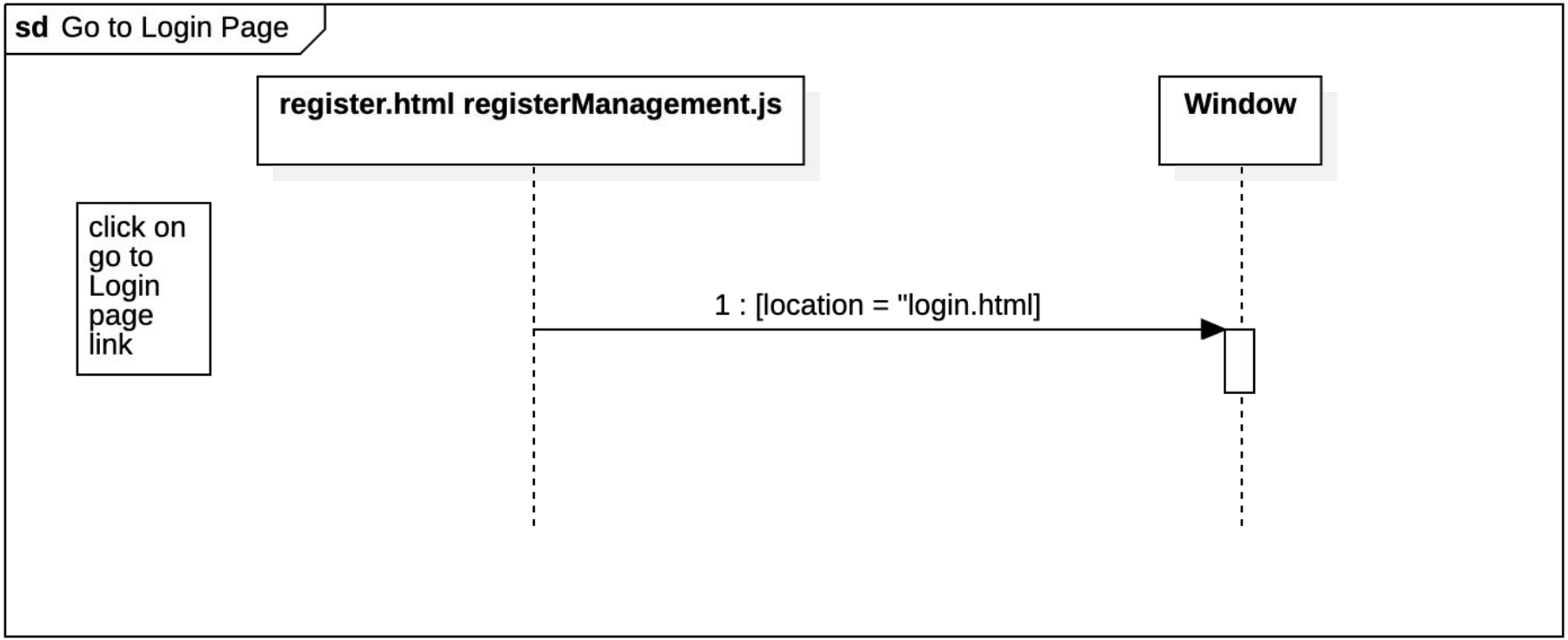
Evento Go to Subscription Form



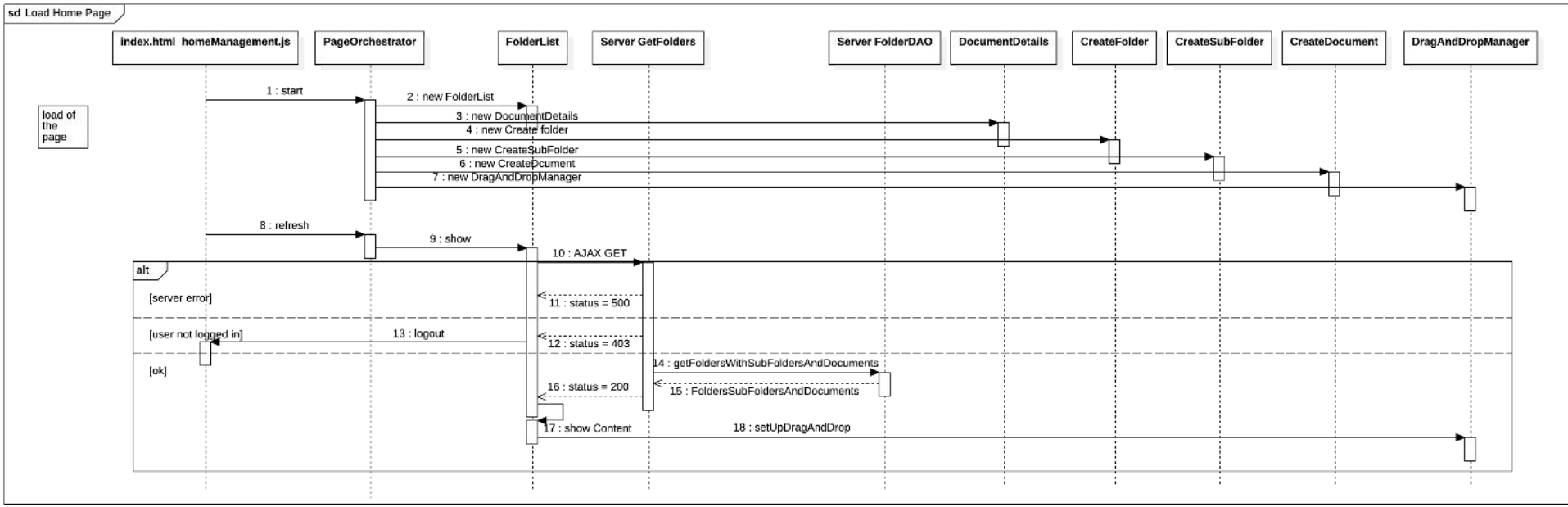
Event Registration



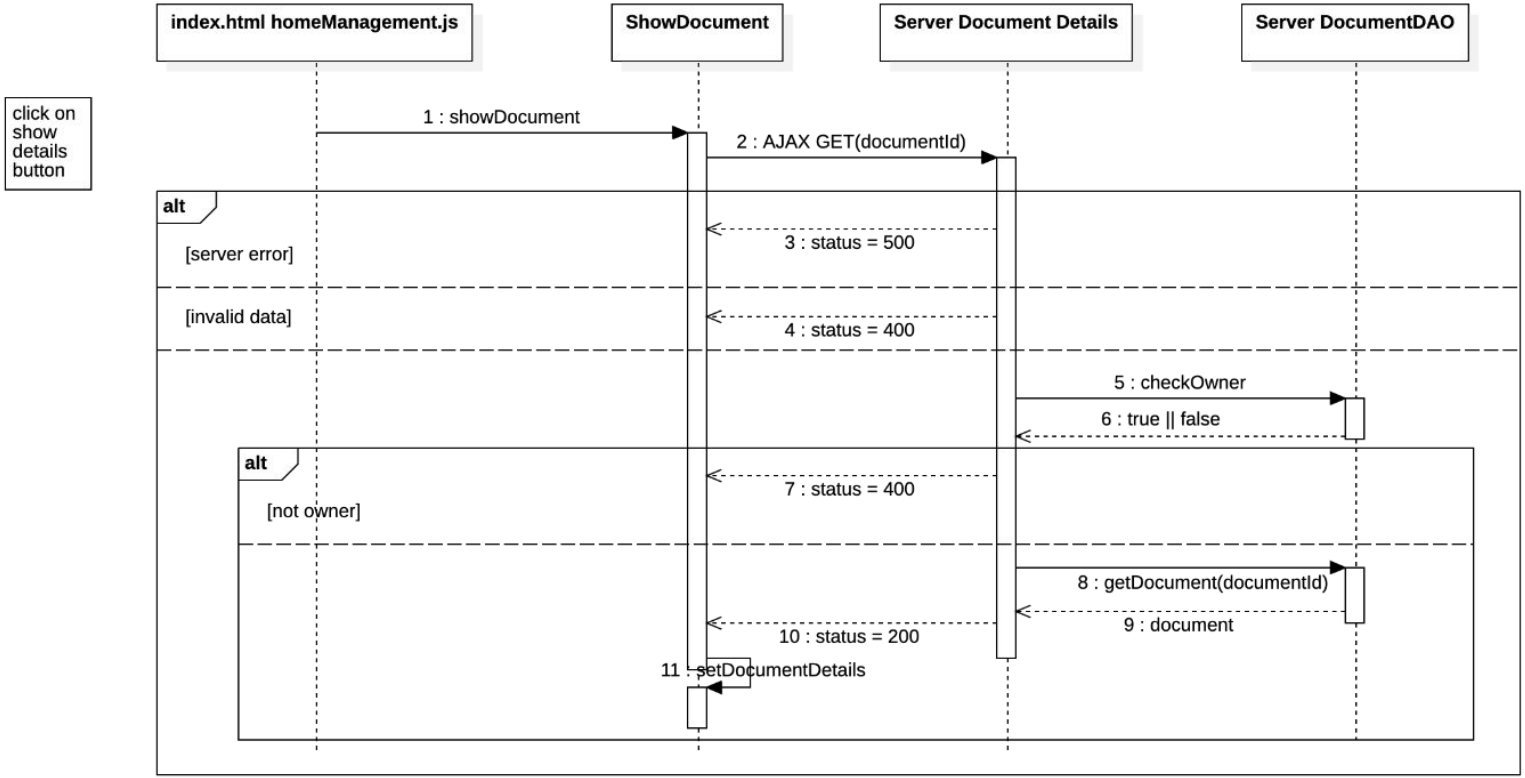
Event Go to Login Page



Event Load Page

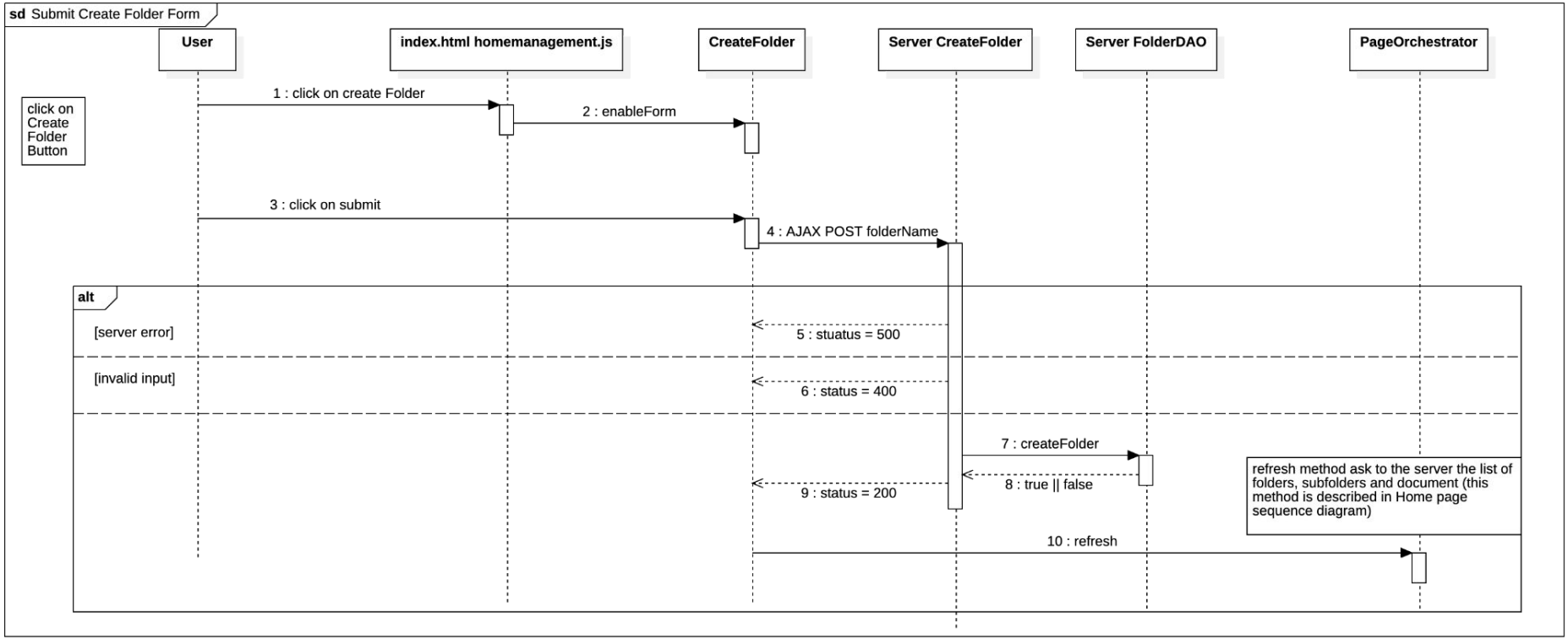


Event Show Document Details

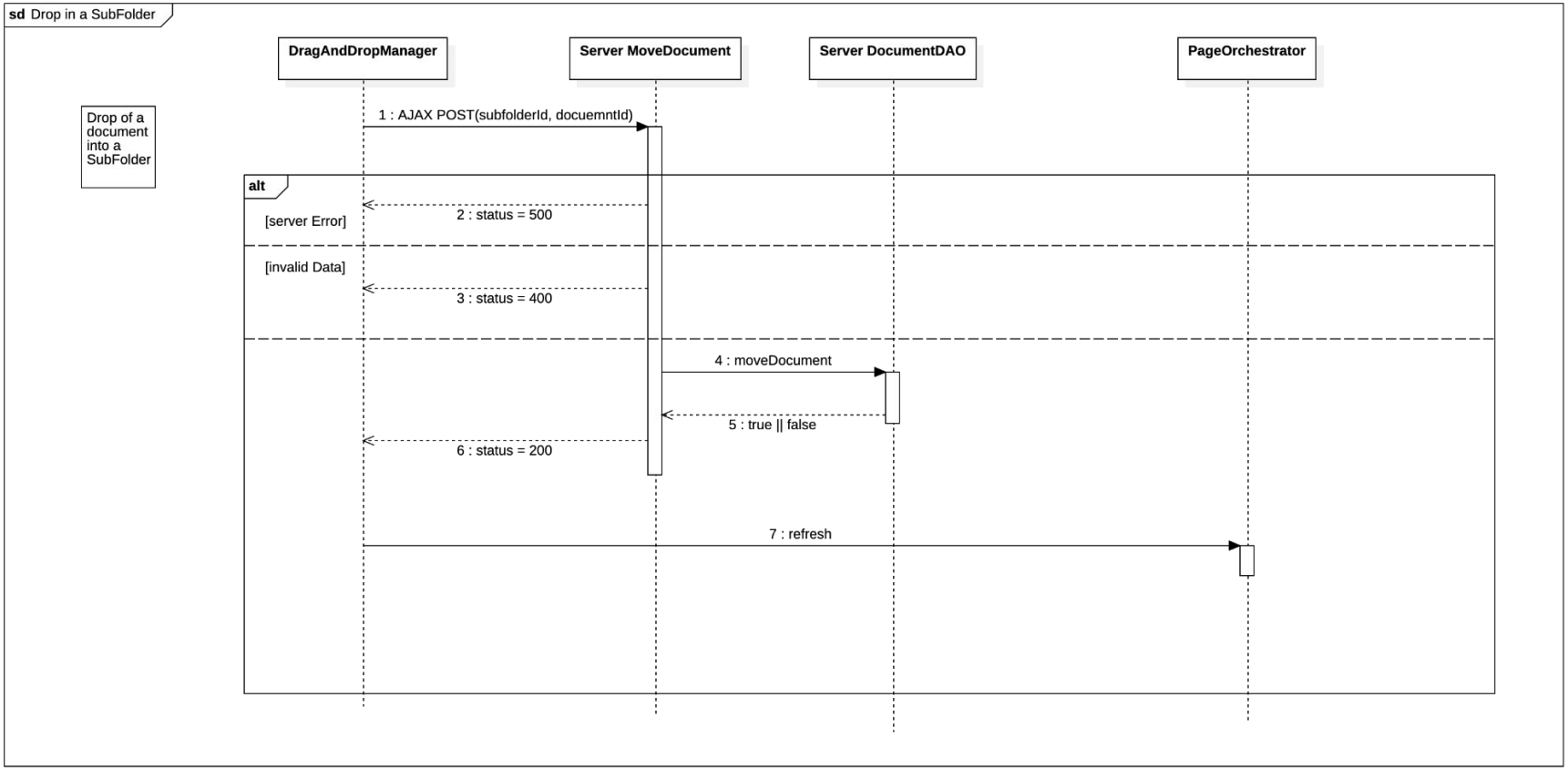


Event Submit of a Creation Form

Viene presentato un solo esempio di creazione, nel caso di creazione SubFolder o Folder le sequenze di chiamate sono analoghe

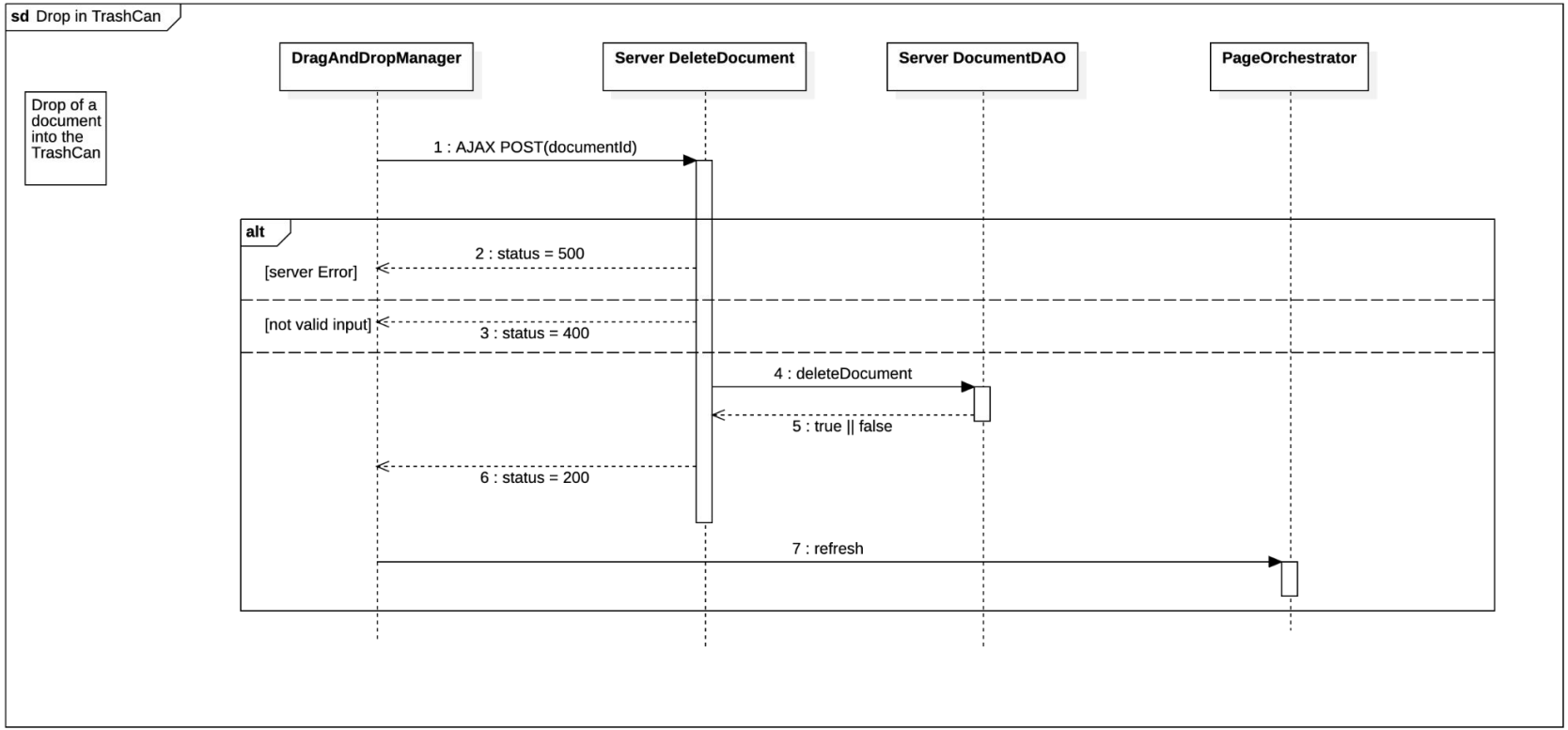


Event Drop in a SubFolder



Event drop in the TrashCan

Sono stati omessi gli eventi di eliminazione di Folder e SubFolder in quanto analoghi all' eliminazione di Document



Event click on Logout button