



POLITECNICO
MILANO 1863

**Progetto del corso Tecnologie
Informatiche per il web
Anno Accademico 2023/2024**

Scianna Marco
Codice Persona: 10815900 Matricola: 984631

Indice

1	Specifiche.....	2
1.1	Versione pure HTML	2
1.2	Versione con JavaScript.....	3
2	Analisi dei requisiti.....	4
2.1	Analisi dei dati.....	4
2.2	Analisi requisiti dell'applicazione.....	5
2.3	Completamento delle specifiche	6
3	Design	6
3.1	Database design.....	6
3.2	Progettazione concettuale	7
3.3	Application design.....	8
4	Componenti	9
4.1	Componenti pure HTML	9
4.2	Componenti JavaScript	10
5	JavaScript	12
6	Eventi	13
6.1	Login.....	14
6.2	Registrazione.....	15
6.3	Accesso alla Home Page.....	16
6.4	Creazione cartella	17
6.5	Creazione documento.....	18
6.6	Accesso alla pagina dei contenuti.....	19
6.7	Accesso alla pagina del documento.....	20
6.8	Sposta documento	21
6.9	Seleziona cartella	22
6.10	Elimina elemento.....	22

1 Specifiche

1.1 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. 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.

1.2 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).

2 Analisi dei requisiti

2.1 Analisi dei dati

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", anche a lato client. 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.

Entità, **attributi**, **relazioni**

2.2 Analisi requisiti dell'applicazione

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", anche a lato client. 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.

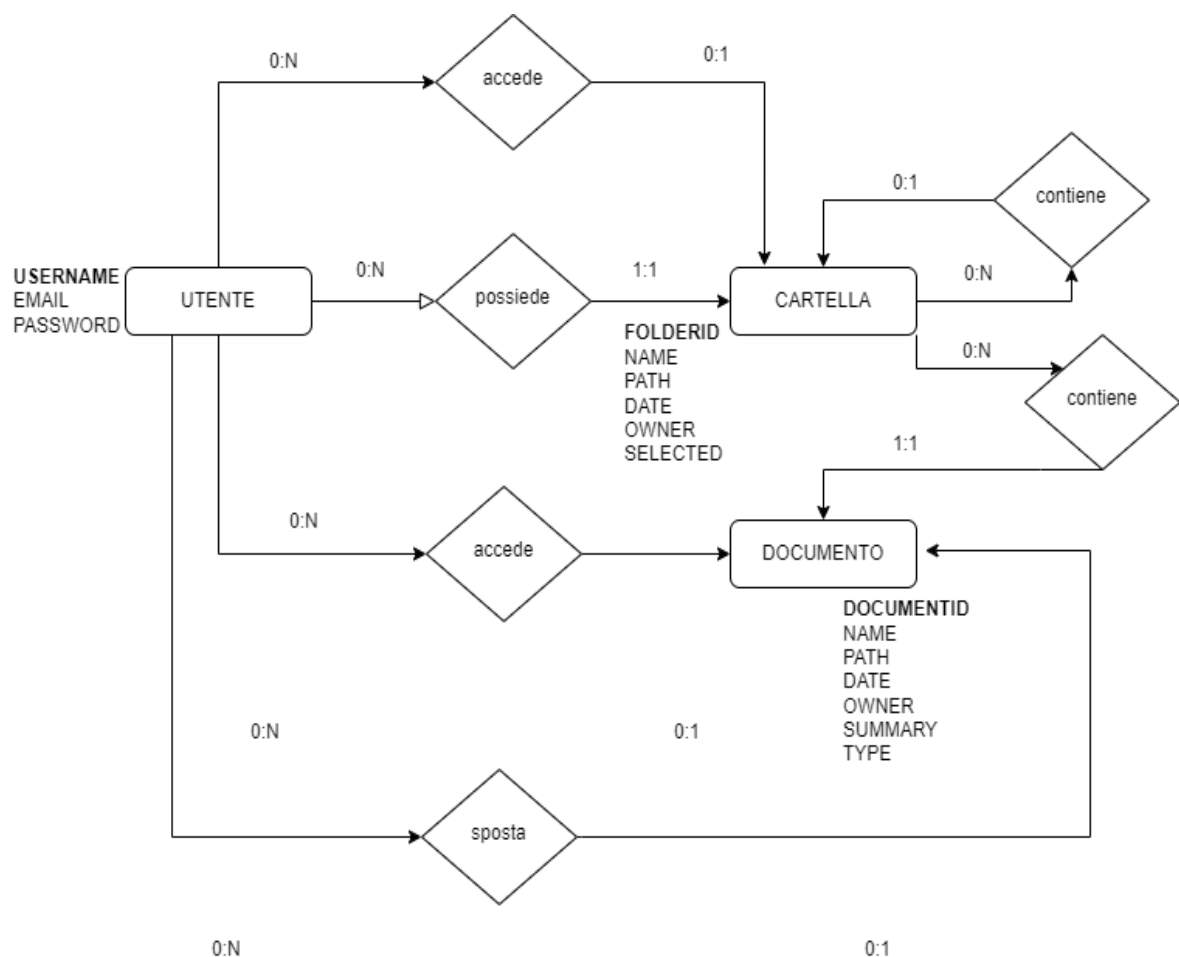
Pagine, **view components**, **eventi**, **azioni**

2.3 Completamento delle specifiche

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).

3 Design

3.1 Database design



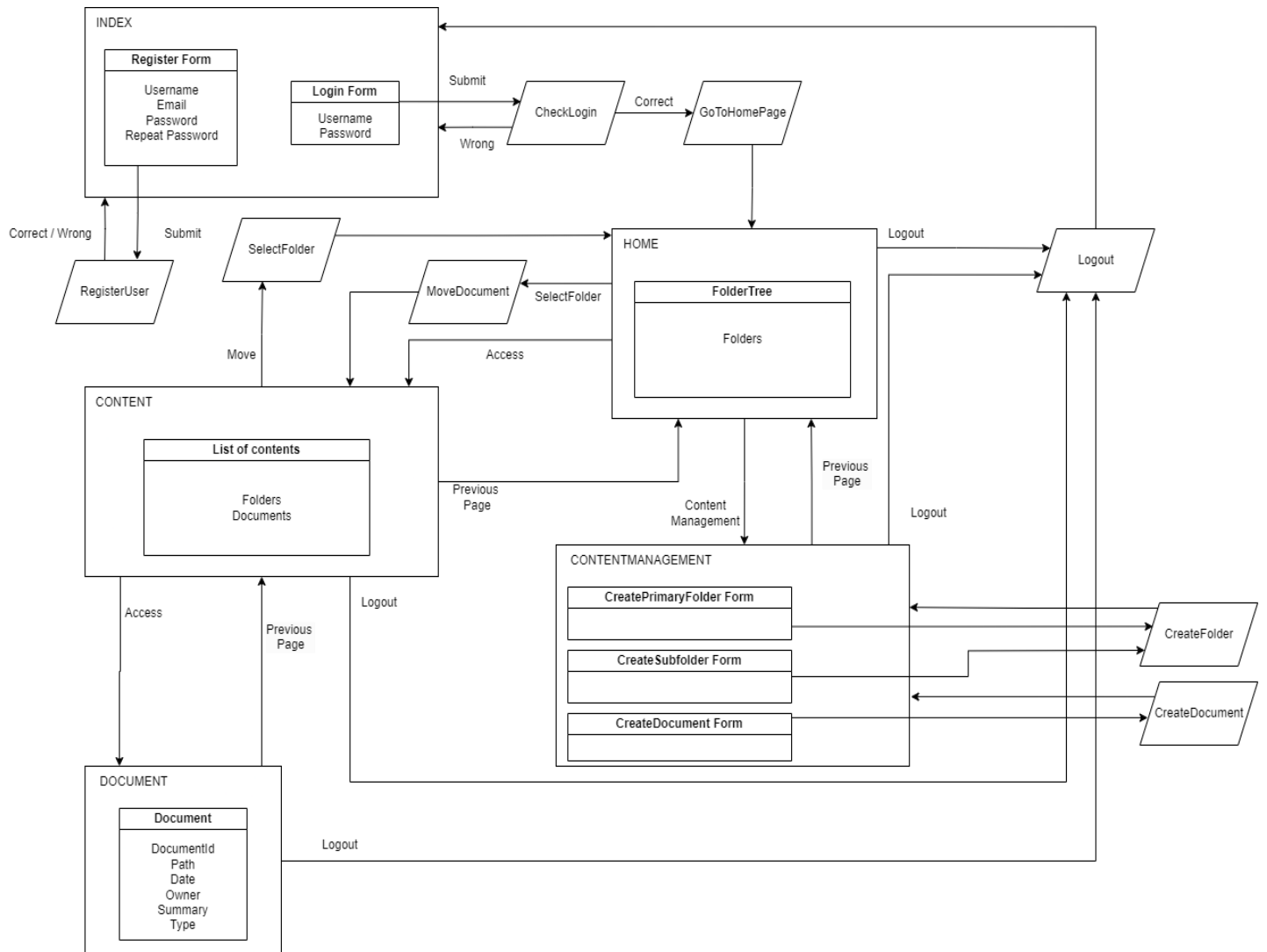
3.2 Progettazione concettuale

```
CREATE TABLE user (  
  `username` varchar(50) UNIQUE NOT NULL PRIMARY KEY,  
  `email` varchar(100) NOT NULL,  
  `password` varchar(100) NOT NULL  
);
```

```
CREATE TABLE folders (  
  `folderId` int NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,  
  `owner` varchar(50) NOT NULL,  
  `path` varchar(200) NOT NULL,  
  `name` varchar(100) NOT NULL,  
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `selected` tinyint(1) NOT NULL DEFAULT '0',  
  FOREIGN KEY (owner) REFERENCES user(username) ON UPDATE  
  CASCADE ON DELETE CASCADE  
);
```

```
CREATE TABLE documents (  
  `documentId` int NOT NULL UNIQUE AUTO_INCREMENT PRIMARY KEY,  
  `owner` varchar(50) NOT NULL,  
  `name` varchar(100) NOT NULL,  
  `path` varchar(200) NOT NULL,  
  `summary` text NOT NULL,  
  `type` varchar(50) NOT NULL,  
  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  FOREIGN KEY (owner) REFERENCES user(username) ON UPDATE  
  CASCADE ON DELETE CASCADE  
);
```


3.3 Application design



4 Componenti

4.1 Componenti pure HTML

Model Objects (Beans)

- User
- Folder
- Document

Data Access Objects (Classes)

- UserDao
 - checkUser(String username, String password)
 - checkUsernameAvailability(String username)
 - repeatPasswordField(String password, String repeatPassword)
 - registration(user userBean)
- FolderDAO
 - createFolder(String owner, String name, String path)
 - findFolderById(int folderId)
 - findFolderByPath(String username, String path)
 - findFolderByUser(String username)
 - sortFolders(List<folder> folders)
- DocumentDAO
 - createDocument(String owner, String name, String path, String summary, String type)
 - moveDocument(String newPath, int documentId)
 - findDocumentByPath(String username, String path)
 - findDocumentById(Integer documentId)

Controllers (Servlets)

- registerUser
- checkLogin
- createDocument
- createFolder
- goToHomePage
- goToContentPage
- goToDocumentPage
- moveDocument
- goToContentManagementPage
- logout
- selectFolder

Views (Templates)

- index
- home
- content
- contentManagement
- document

Database Connection

- Stabilita nel metodo init() dei controllers e passata ai DAOs.

4.2 Componenti JavaScript

SERVER SIDE

Model Objects (Beans)

- User
- Folder
- Document

Data Access Objects (Classes)

- UserDao
 - checkUser(String username, String password)
 - checkUsernameAvailability(String username)
 - repeatPasswordField(String password, String repeatPassword)
 - registration(user userBean)
- FolderDAO
 - createFolder(String owner, String name, String path)
 - findFolderById(int folderId)
 - findFolderByPath(String username, String path)
 - findFolderByUser(String username)
 - sortFolders(List<folder> folders)
 - deleteFolder(int id, String owner)
- DocumentDAO
 - createDocument(String owner, String name, String path, String summary, String type)
 - moveDocument(String newPath, int documentId)
 - findDocumentByPath(String username, String path)
 - findDocumentById(Integer documentId)

- deleteDocument(int id, String owner)

Controllers (Servlets)

- registerUser
- checkLogin
- createDocument
- createFolder
- moveDocument
- delete
- getContent
- getDocument
- getFolders
- logout

Views (Templates)

- home
- index

Database Connection

- Stabilita nel metodo init() dei controllers e passata ai DAOs.

CLIENT SIDE

- Index
 - LoginWizard
 - registerEvents(): associa al componente le funzioni per gestirne gli eventi
 - makeCall(): con i dati del login
 - RegistrationWizard
 - registerEvents(): associa al componente le funzioni per gestirne gli eventi
 - makeCall(): con i dati della registrazione
- Home
 - listFolder: : richiede al server l'elenco di cartelle dell'utente e lo mostra
 - showSubfolder: richiede al server l'elenco di sottocartelle e documenti di una cartella e lo mostra
 - showDocument: richiede al server i dati di sottocartelle del documento passato e lo mostra

- addSubfolder: aggiunge una sottocartella nella cartella affianco
- addFolder: aggiunge una cartella primaria
- addDocument: aggiunge un documento nella cartella affianco
- deleteItem: chiama il server per eliminare l'elemento passato per id
- dropFolder: prende l'id della cartella su cui viene rilasciato l'elemento e chiama moveDocument
- moveDocument : chiama il server e sposta un documento da una cartella ad una altra
- drag: prende l'id dell'elemento su cui viene fatto drag
- dropBin: prende i parametri dell'oggetto che viene rilasciato e chiama showDeleteModal
- showDeleteModal: aggiorna la vista mostrando un modal che permette di annullare o confermare chiamando poi deleteItem

5 JavaScript

Client Side			Server Side		
Event	Action	Controller	Event	Action	Controller
Index > login form > submit	Controllo dati	makeCall	POST username, password	Controlla validità	checkLogin(servlet)
Index > signup form > submit	Controllo dati	makeCall	POST username, email, password, repeat password	Controlla validità	registerUser(servlet)
Home > load	Vista aggiornata con le cartelle	PageOrchestrator()	POST	Cerca tutte le cartelle	getFolders(servlet)
Home > document > drop	Drop in una cartella	dropFolder()	POST folderId, documentId	Sposta documento	moveDocument (servlet)
Home > item > drop	Drop sul cestino	dropBin()	x	x	x

home > Delete Modal > confirmation> click	Conferma eliminazione oggetto	deleteItem ()	DELETE id	Controlla se è il proprietario ed elimina	Delete(servlet)
home > folder > click	Vista aggiornata con contenuto cartella	showSubf older()	POST folderId	Cerca documenti e sottocartelle	getContent(servlet)
home > folder > click > access button	Vista aggiornata con dati del documento	showDocu ment()	POST documentId	Cerca dati documento	getDocument(servlet)
Home > logout > click	logout	Logout()	GET	logout	Logout(servlet)
home > subfolder add button	Controlla presenza dei dati	addSubfol der()	POST name, folderId	Controlla validità e crea	createFolder(servlet)
home > document add button	Controlla presenza dei dati	addDocum ent()	POST name, summary, type, folderId	Controlla validità e crea	createDocument(servlet)
home > folder add button	Controlla presenza dei dati	addFolder ()	POST name, folderId= -1	Controlla validità e crea	createFolder(servlet)

6 Eventi

Nei seguenti eventi non sono presenti tutte le casistiche, sono mostrati i casi di eventi con richieste ben formate. Gli eventi reali, invece, dispongono di tutti i controlli specifici e restituiscono al cliente il messaggio di errore appropriato.

Legenda:



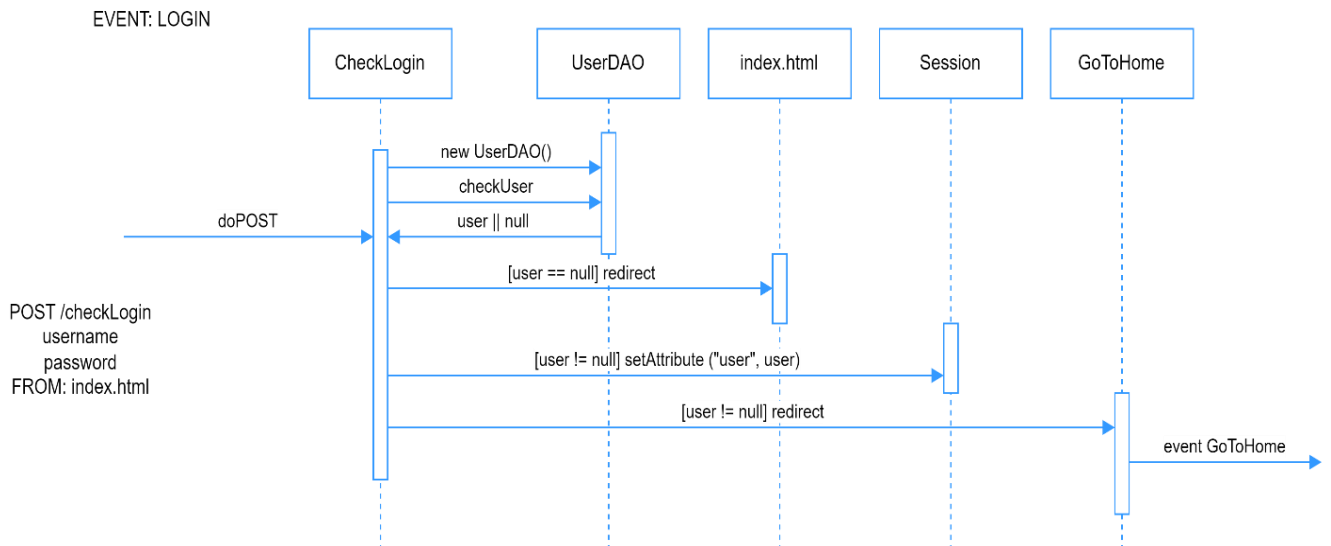
Client side



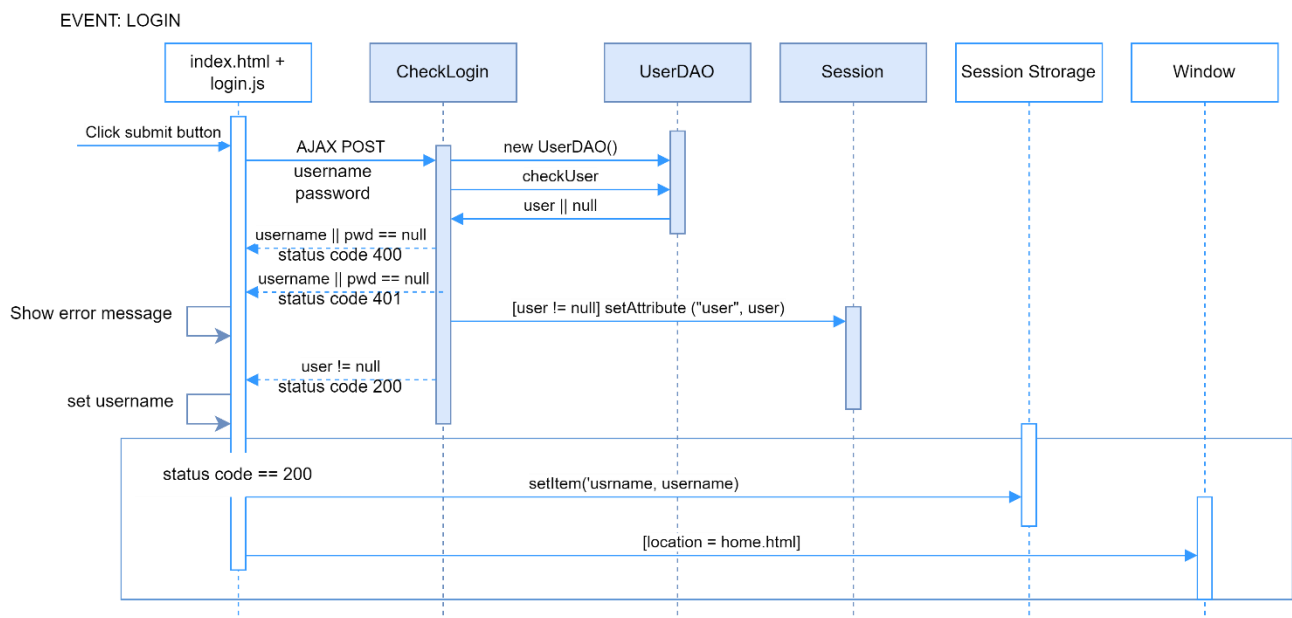
Server side

6.1 Login

Versione pure HTML



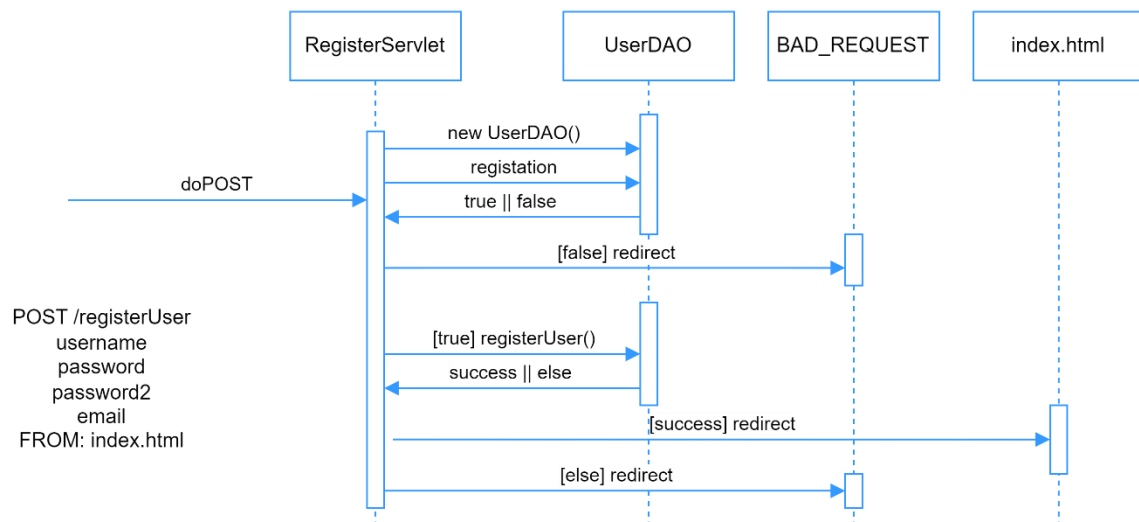
Versione con JavaScript



6.2 Registrazione

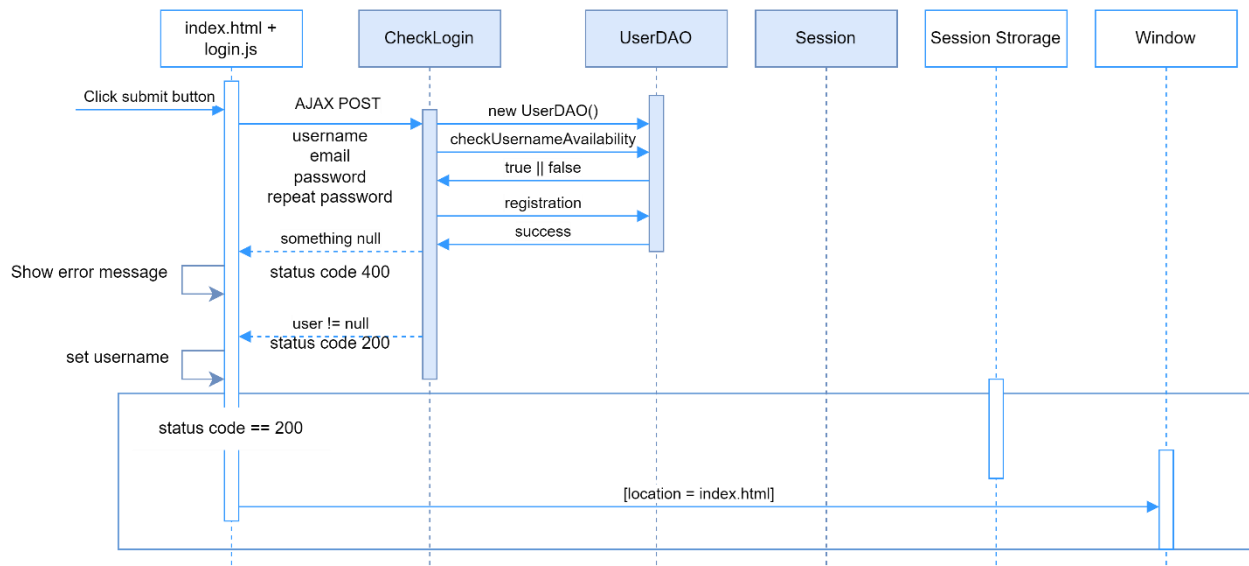
Versione pure HTML

EVENT: REGISTER



Versione con JavaScript

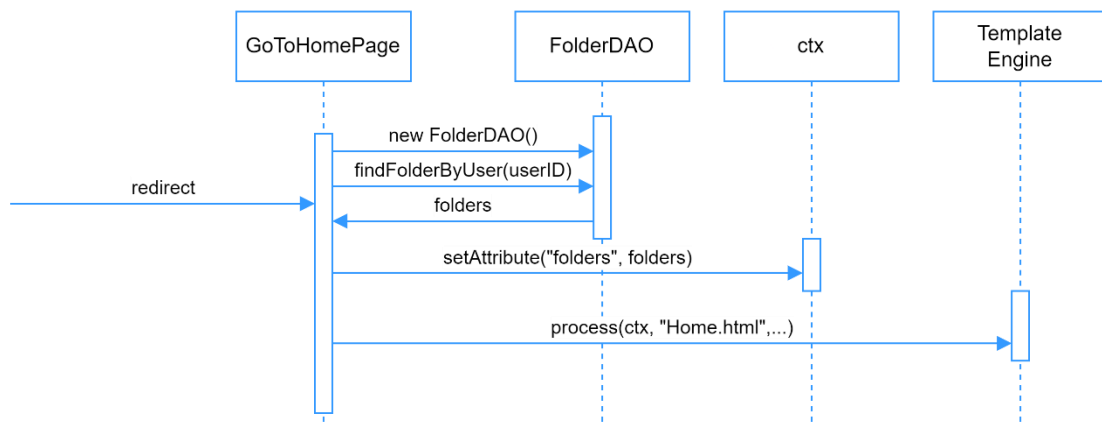
EVENT: REGISTRATION



6.3 Accesso alla Home Page

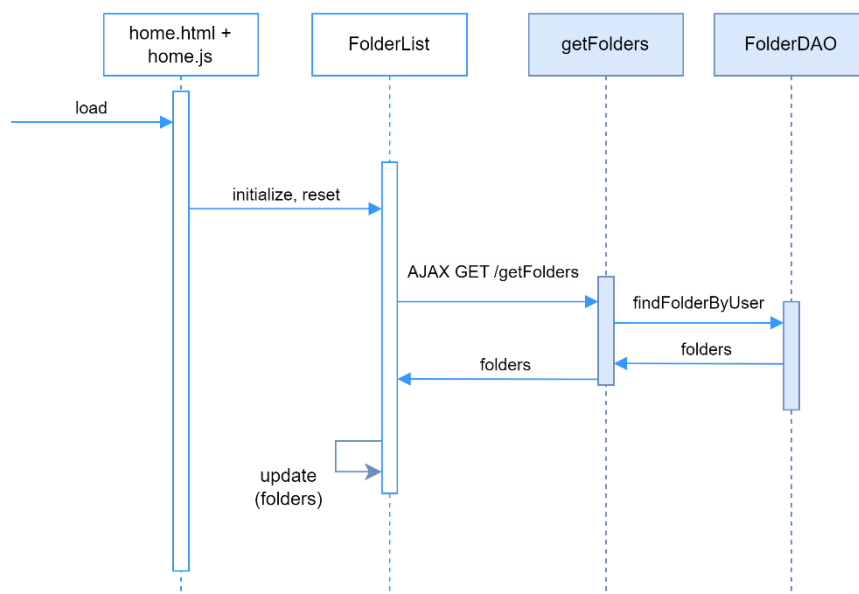
Versione pure HTML

EVENT: GO TO HOME PAGE



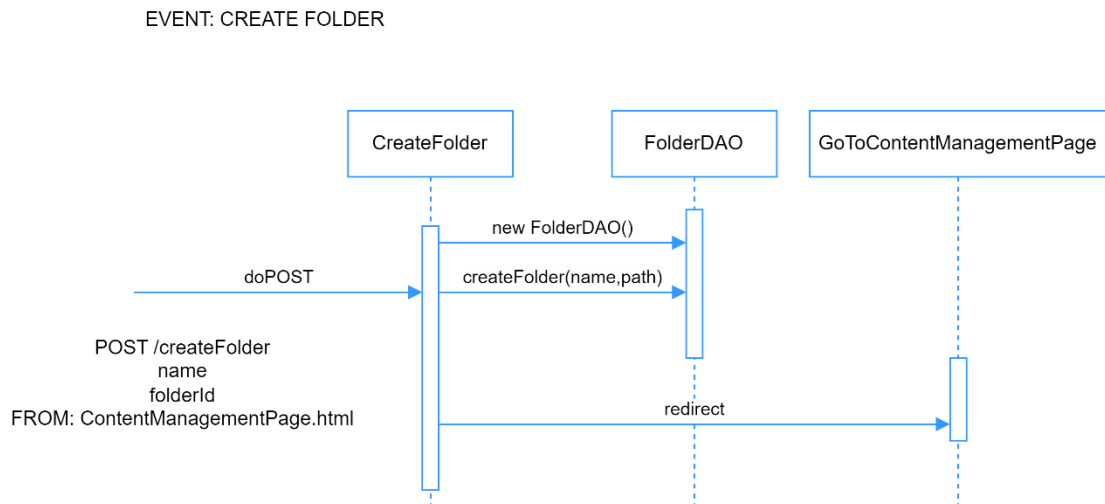
Versione con JavaScript

EVENT: Go to home page

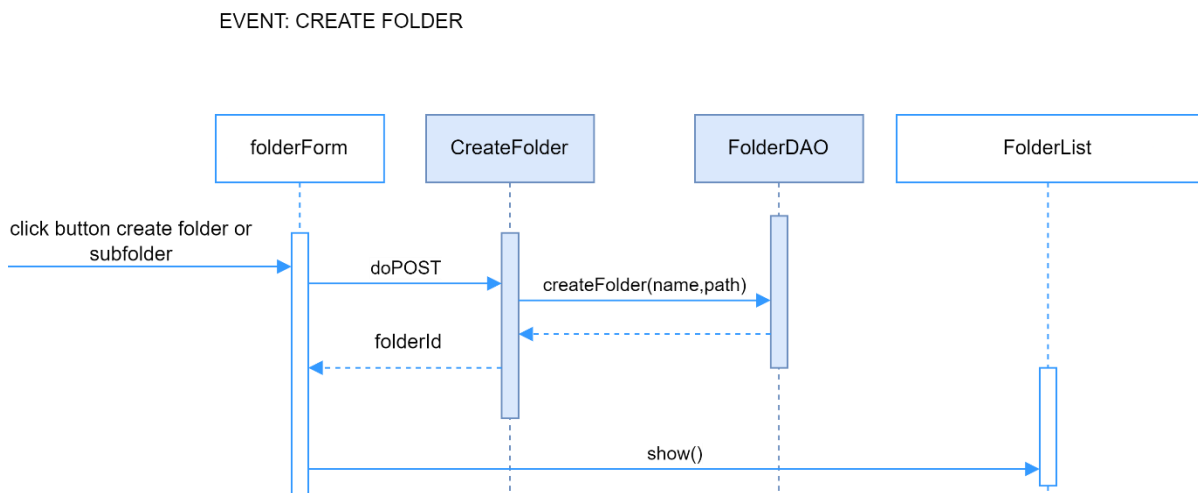


6.4 Creazione cartella

Versione pure HTML



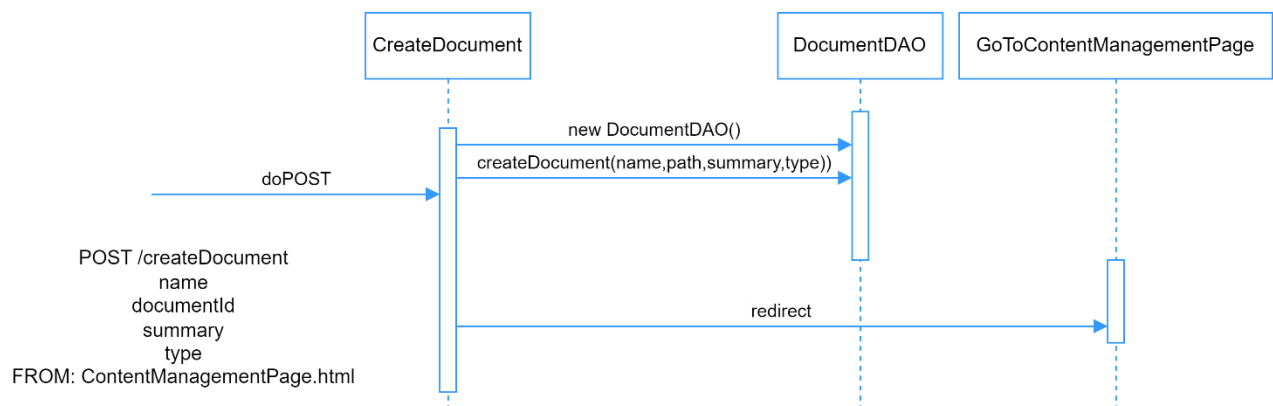
Versione con JavaScript



6.5 Creazione documento

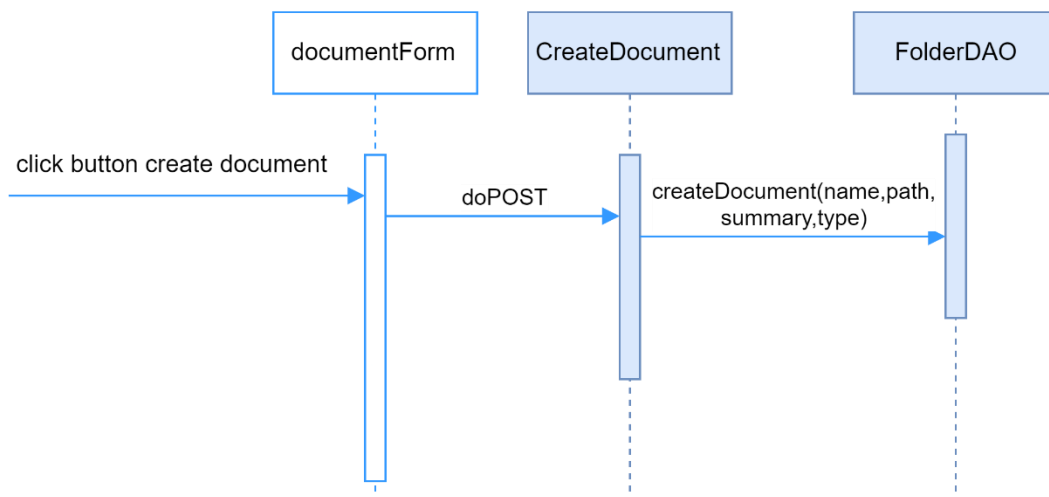
Versione pure HTML

EVENT: CREATE DOCUMENT



Versione con JavaScript

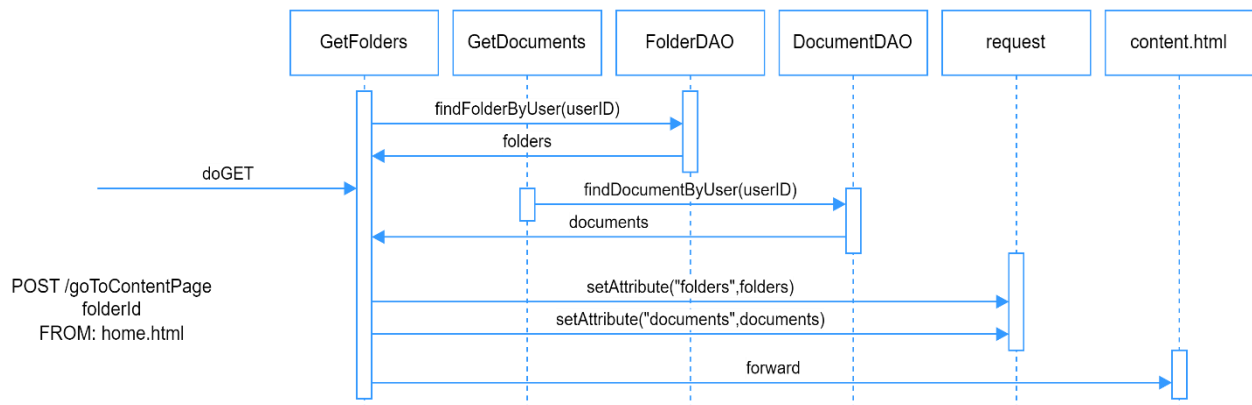
EVENT: CREATE DOCUMENT



6.6 Accesso alla pagina dei contenuti

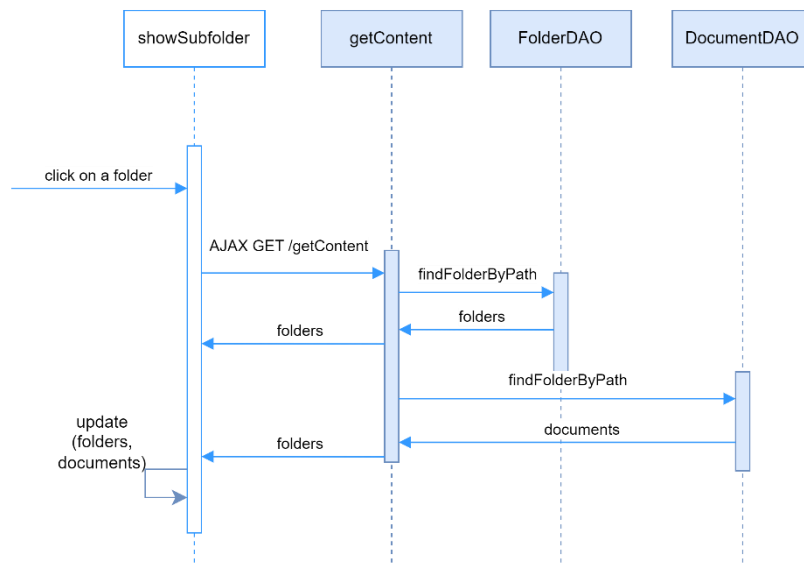
Versione pure HTML

EVENT: ACCESS TO CONTENT PAGE



Versione con JavaScript

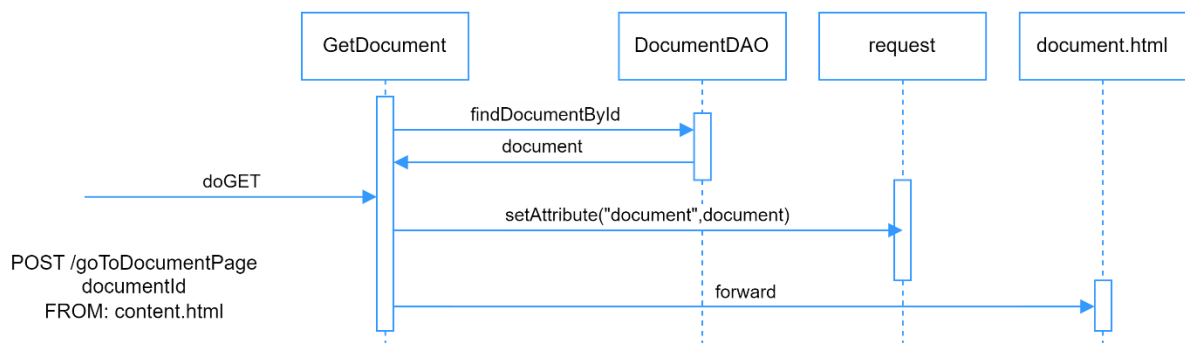
EVENT: ACCESS TO CONTENT PAGE



6.7 Accesso alla pagina del documento

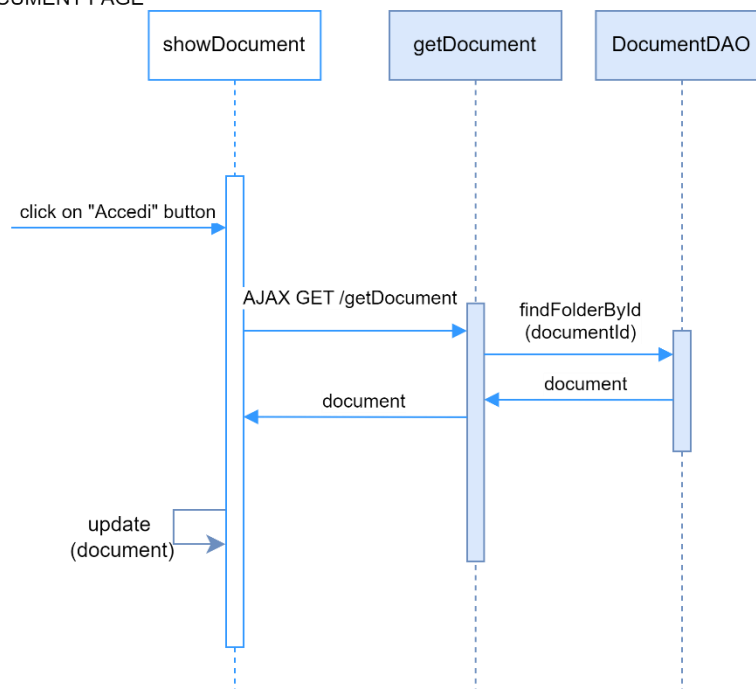
Versione pure HTML

EVENT: ACCESS TO DOCUMENT PAGE



Versione con JavaScript

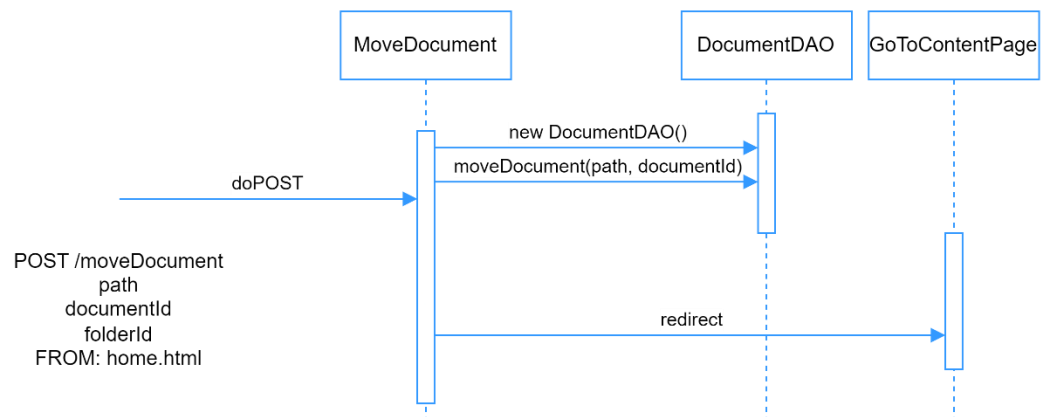
EVENT: ACCESS TO DOCUMENT PAGE



6.8 Sposta documento

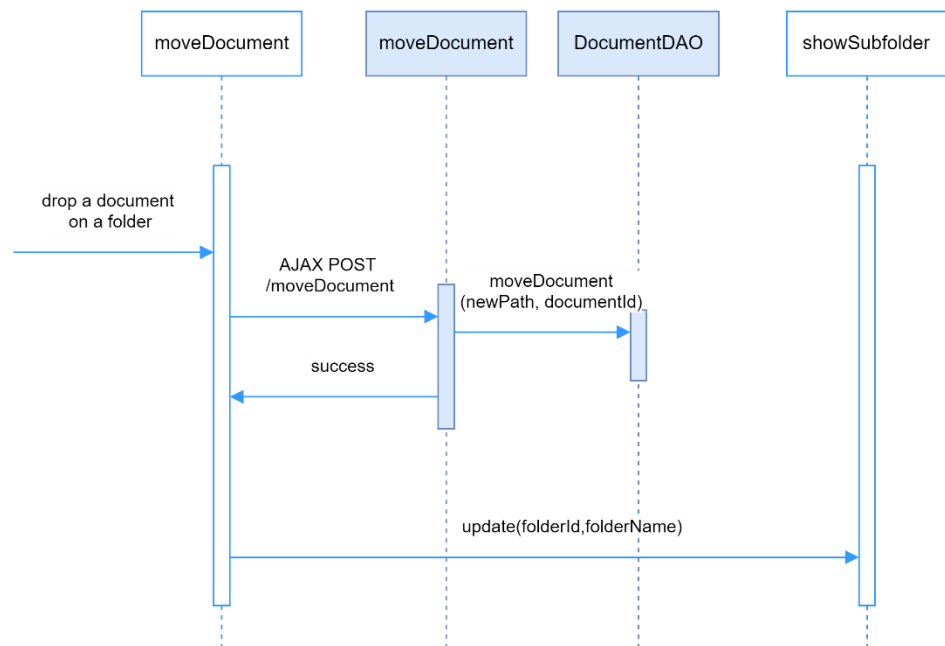
Versione pure HTML

EVENT: MOVE DOCUMENT



Versione con JavaScript

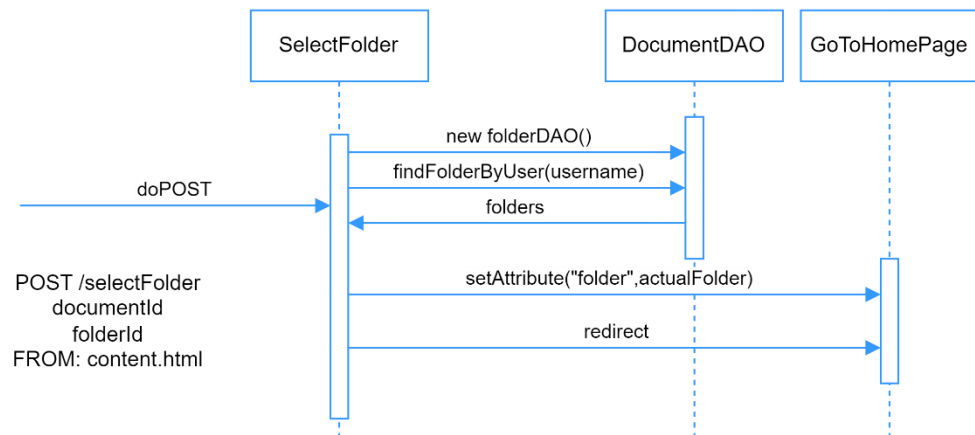
EVENT: MOVE DOCUMENT



6.9 Seleziona cartella

Versione pure HTML

EVENT: SELECT FOLDER



6.10 Elimina elemento

Versione con JavaScript

EVENT: DELETE ITEM

