

# Tecnologie Informatiche per il Web

## Traccia 2: Gestione Documenti

A cura di Fabio Vokrri  
(10738386)

A.A. 2023-2024

## Analisi dei dati

L'applicazione supporta **registrazione** e **login** mediante una pagina pubblica con opportune form. La registrazione di un **utente** 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.

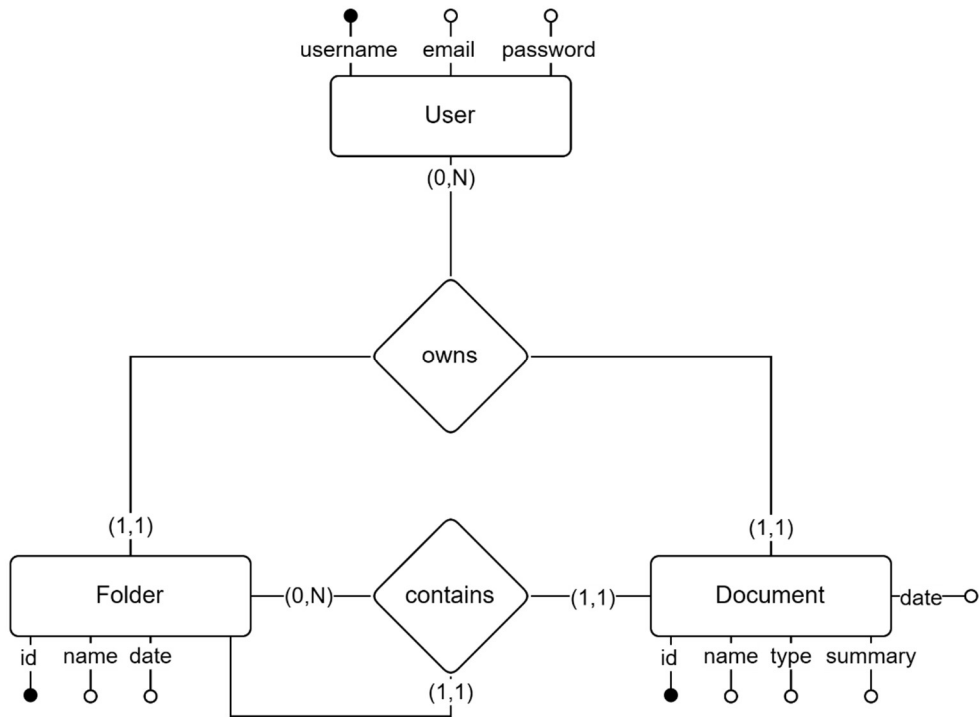
In cui:

- **Pages (Views)**
- **View Components**
- **Actions**
- **Events**

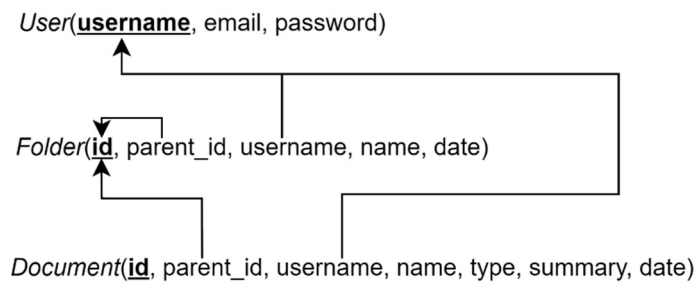
- **Entities**
- **Attributes**
- **Relationships**

# Database Design

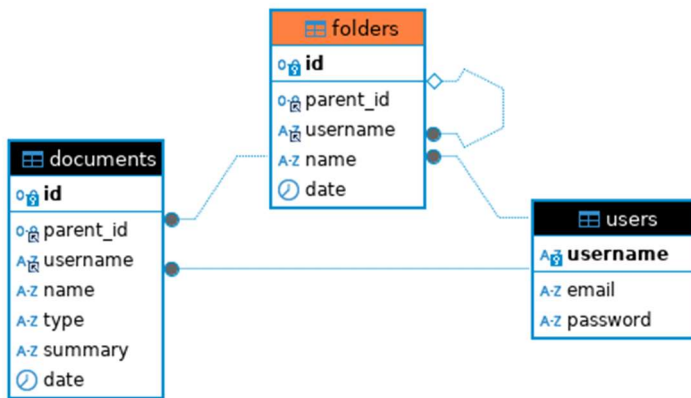
## Concettuale



## Logico



## Schema



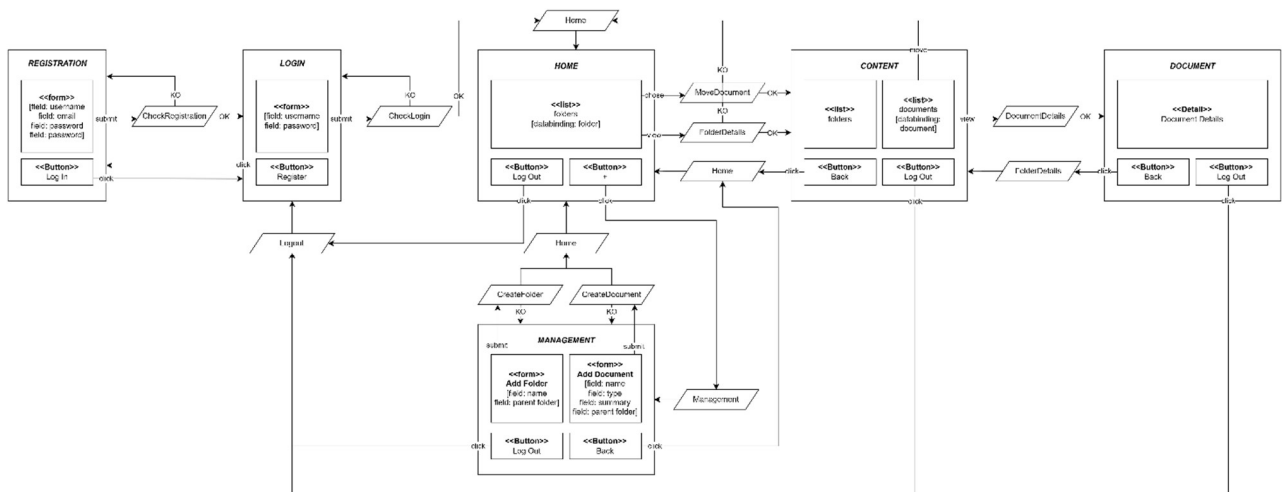
```
-- tiw_project.users definition
CREATE TABLE `users` (
  `username` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(64) NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

-- tiw_project.folders definition
CREATE TABLE `folders` (
  `id` int NOT NULL AUTO_INCREMENT,
  `parent_id` int DEFAULT NULL,
  `username` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  `date` timestamp NOT NULL,
  PRIMARY KEY (`id`),
  KEY `folders_folders_FK` (`parent_id`),
  KEY `folders_users_FK` (`username`),
  CONSTRAINT `folders_folders_FK` FOREIGN KEY (`parent_id`) REFERENCES `folders` (`id`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `folders_users_FK` FOREIGN KEY (`username`) REFERENCES `users` (`username`)
    ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

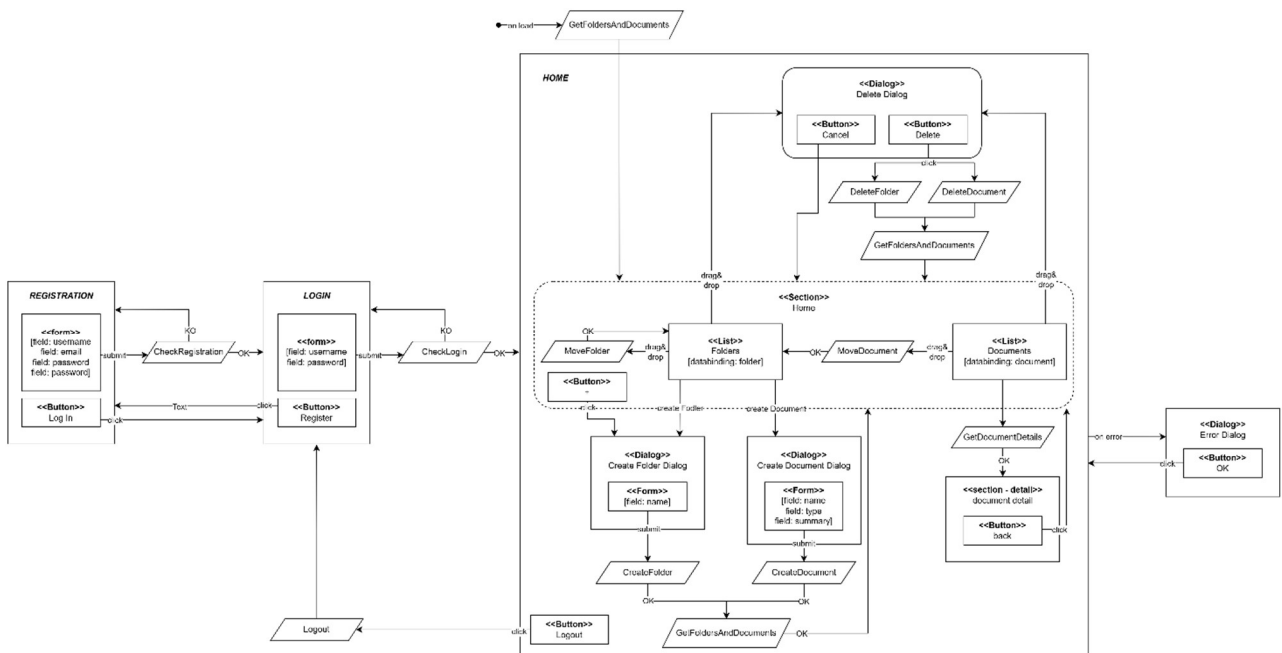
-- tiw_project.documents definition
CREATE TABLE `documents` (
  `id` int NOT NULL AUTO_INCREMENT,
  `parent_id` int NOT NULL,
  `username` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  `type` varchar(255) NOT NULL,
  `summary` text NOT NULL,
  `date` timestamp NOT NULL,
  PRIMARY KEY (`id`),
  KEY `documents_users_FK` (`username`),
  KEY `documents_folders_FK` (`parent_id`),
  CONSTRAINT `documents_folders_FK` FOREIGN KEY (`parent_id`) REFERENCES `folders` (`id`)
    ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `documents_users_FK` FOREIGN KEY (`username`) REFERENCES `users` (`username`)
    ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

# Design Applicativo

## Pure HTML



## RIA



# Components

## Pure HTML - Server Side Components

### Controllers – Back End

- CheckLogin
- CheckRegistration
- CreateDocument
- CreateFolder
- MoveDocument
- Logout

### Controllers – Front End

- Home
- Management
- FolderDetails
- DocumentDetails

### Model Objects – Beans

- User
- Folder
- Document

### Data Access Objects – DAOs

- UserDAO
  - login()
  - checkRegistration()
  - registerUser()
  - getByUserName()
- FolderDAO
  - getAllFoldersOf()
  - getById()
- DocumentDAO
  - getFirstLevelFoldersOf()
  - getChildrenFoldersOf()
  - getChildrenDocumentsOf()
  - addNewFolder()
- DocumentDAO
  - getChildrenDocumentsOf()
  - getByld()
  - moveDocumentToFolder()
  - addNewDocument()

## RIA – Server Side Components

### Controllers

- CheckLogin
- CheckRegistration
- Logout
- CreateDocument
- CreateFolder
- DeleteDocument
- DeleteFolder
- MoveDocument
- MoveFolder
- GetDocumentDetails
- GetFoldersAndDocuments

### Model Objects – Beans

- User
- Folder
- Document

## Data Access Objects – DAOs

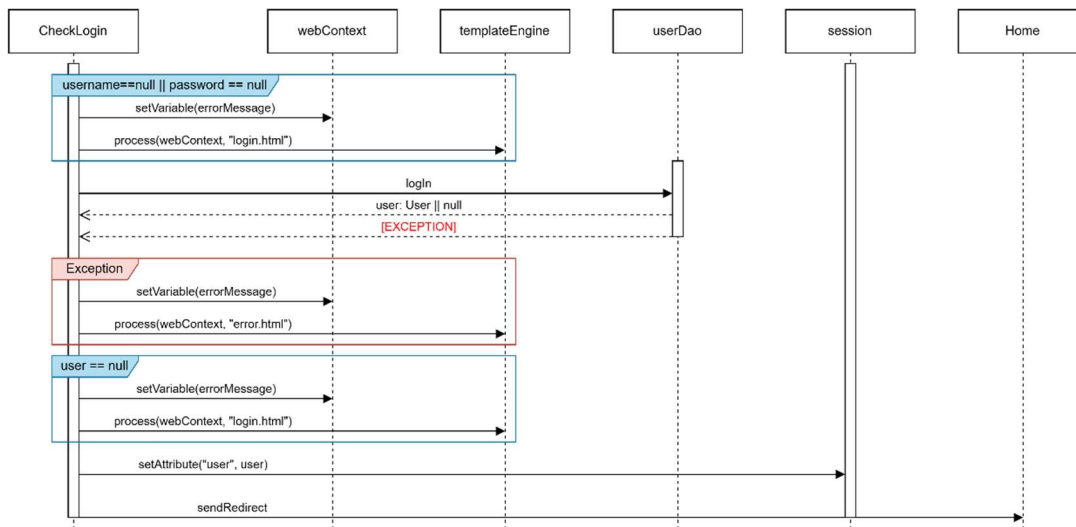
- UserDao
  - login()
  - checkRegistration()
  - registerUser()
  - getByUserName()
- FolderDAO
  - getById()
  - getFirstLevelFoldersOf()
  - getChildrenFoldersOf()
- DocumentDAO
  - getChildrenDocumentsOf()
  - addNewFolder()
  - deleteFolder()
  - moveFolderToFolder()
- DocumentDAO
  - getChildrenDocumentsOf()
  - getById()
  - moveDocumentToFolder()
  - addNewDocument()
  - deleteDocument()

## RIA – Client Side Components

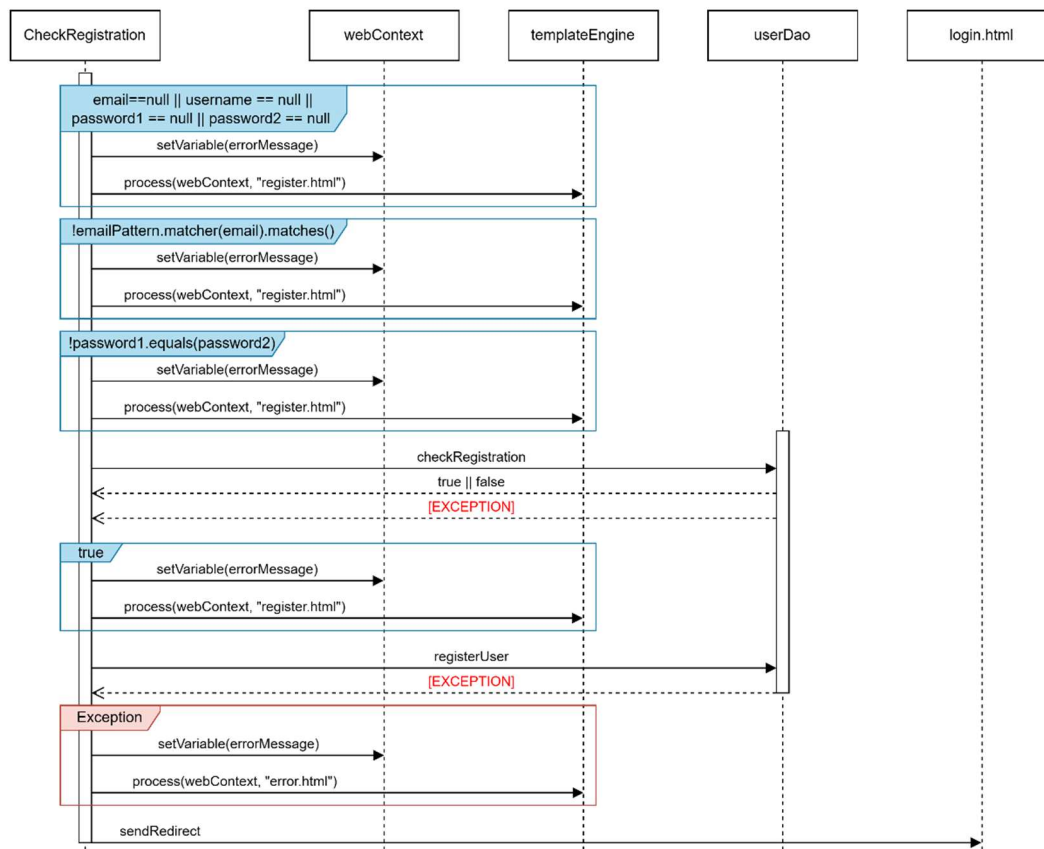
- Login
  - Login Form
- Register
  - Register Form
- Home
  - HomePage
    - show()
    - update()
    - renderFolder()
    - reset()
  - DocumentPage
    - show()
    - update()
    - reset()
  - AddFolderDialog
    - init()
    - show()
  - AddDocumentDialog
    - init()
    - show()
  - DeleteDialog
    - show()
- Support Functions
  - logOut()
  - dragStartHandler()
  - dragOverhandler()
  - dropHandler()
  - dropDeleteHandler()
  - getIdFrom
  - showErrorDialog

# Pure HTML – Sequence Diagrams

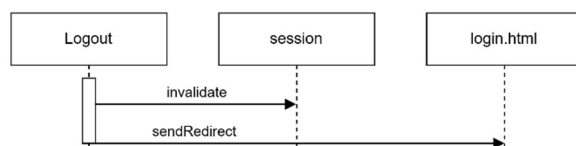
**POST /CheckLogin?username="x"&password="y"**



**POST /CheckRegistration?username="x"&email="y"&password1="z"&password2="w"**

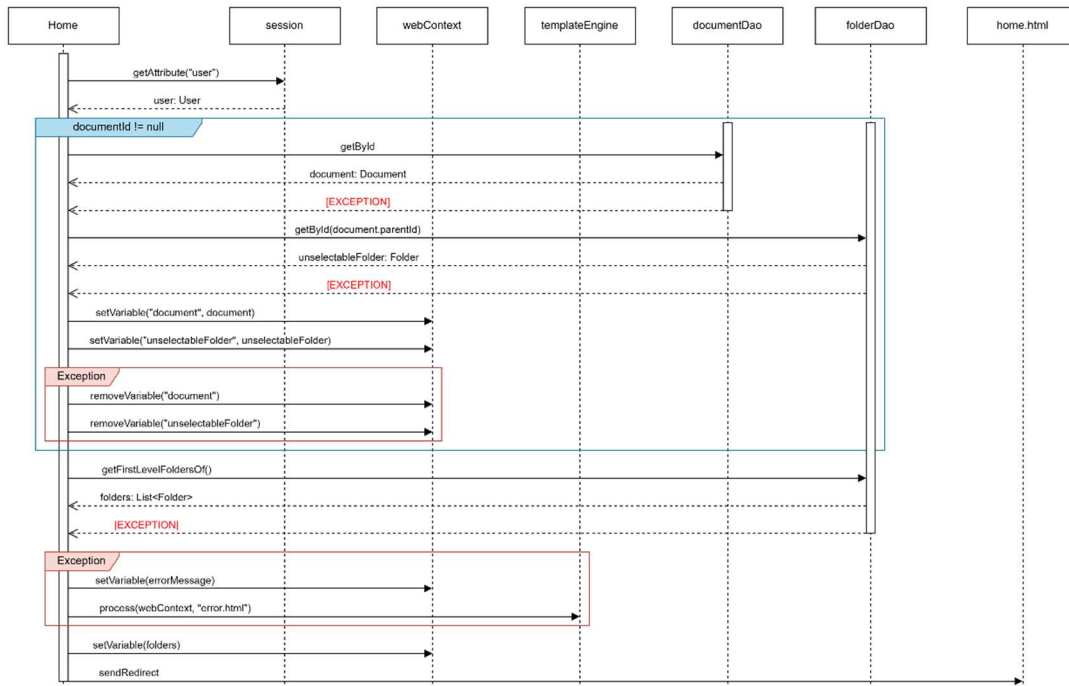


**POST /Logout**

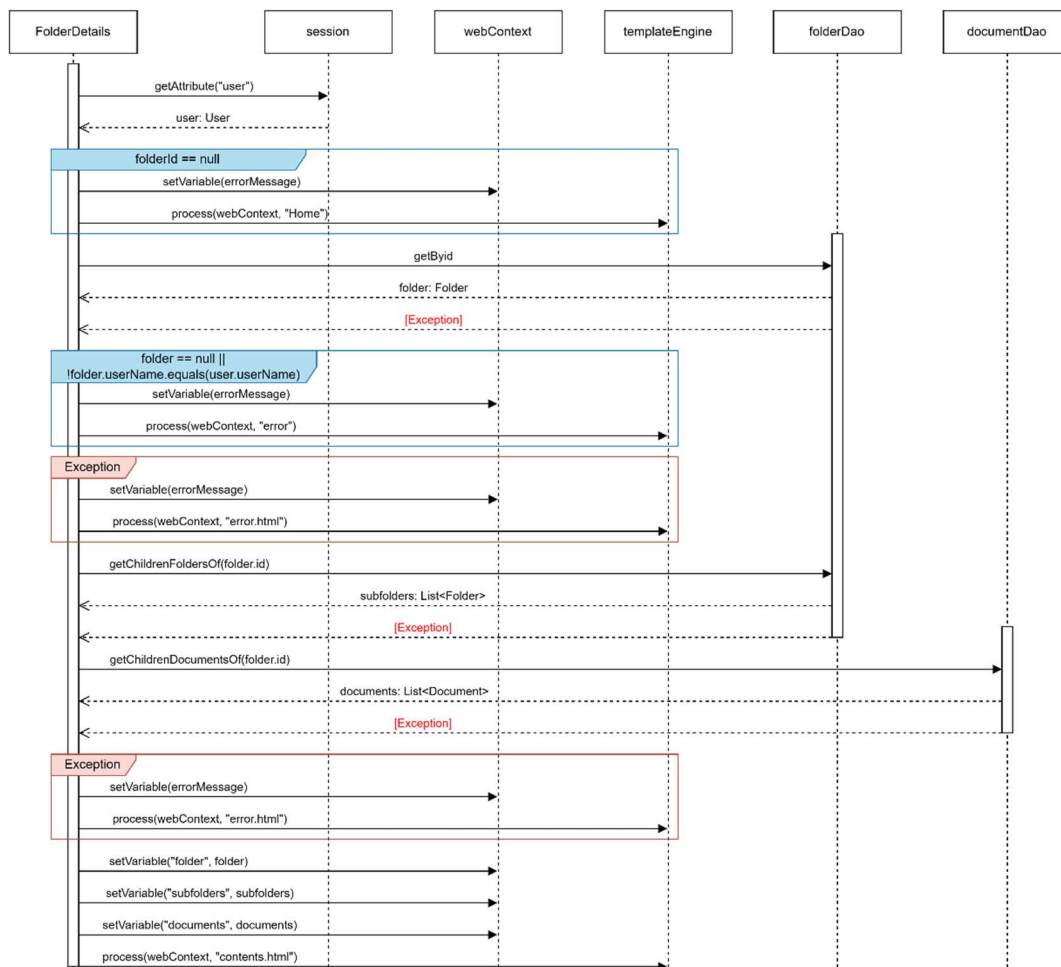




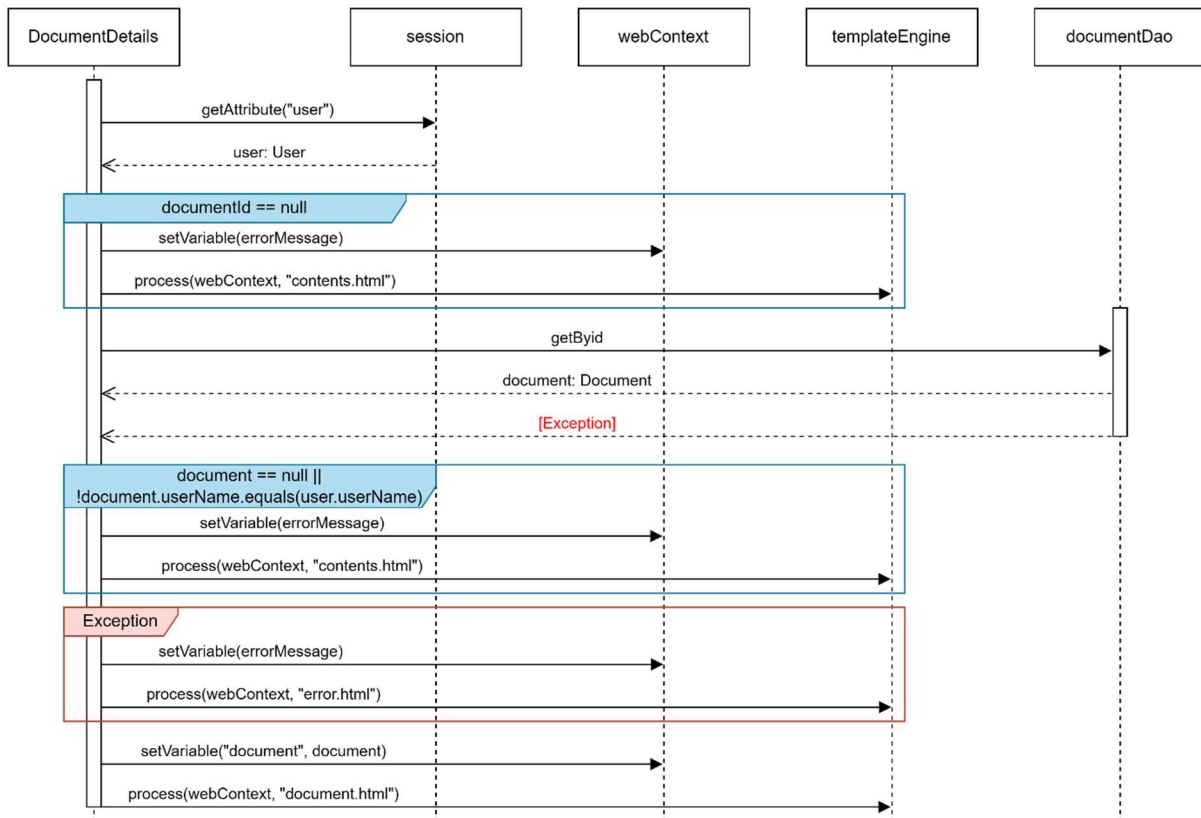
## GET /Home{?documentId="x"}



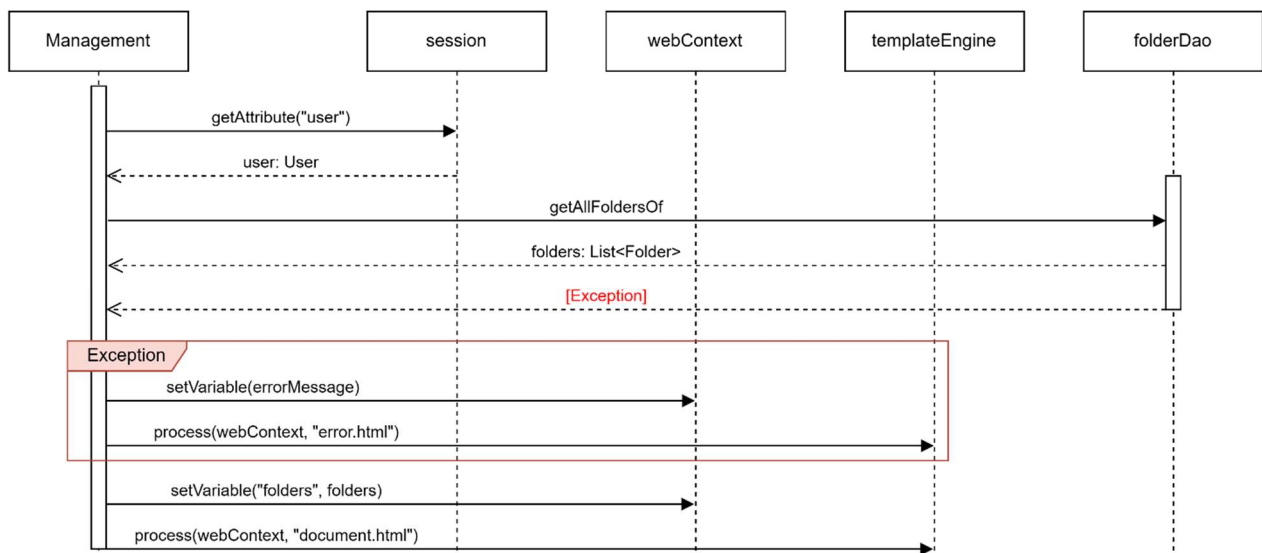
## GET /FolderDetails?folderId="x"



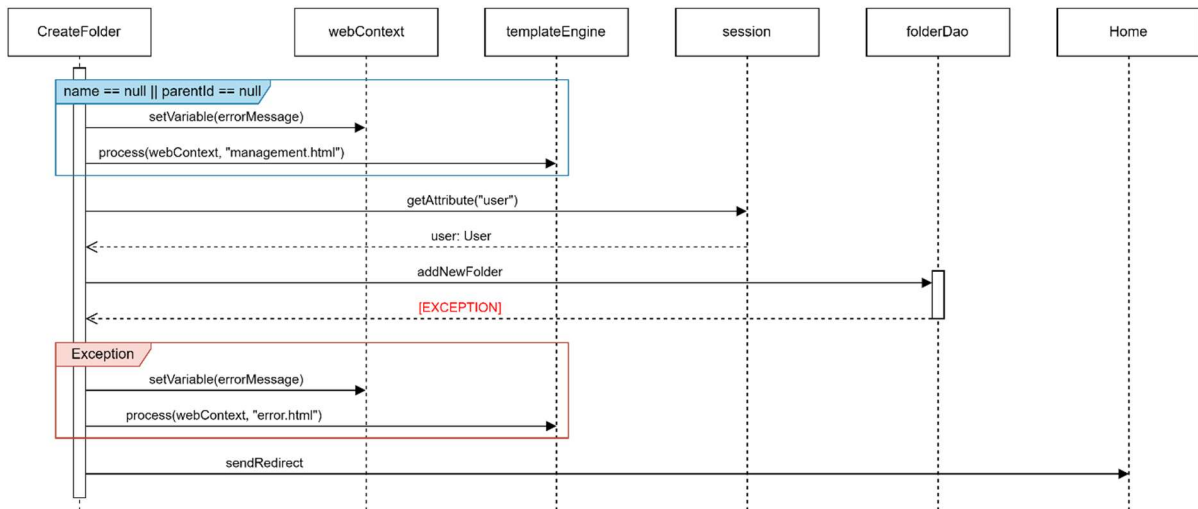
## GET /DocumentDetails?documentId="x"



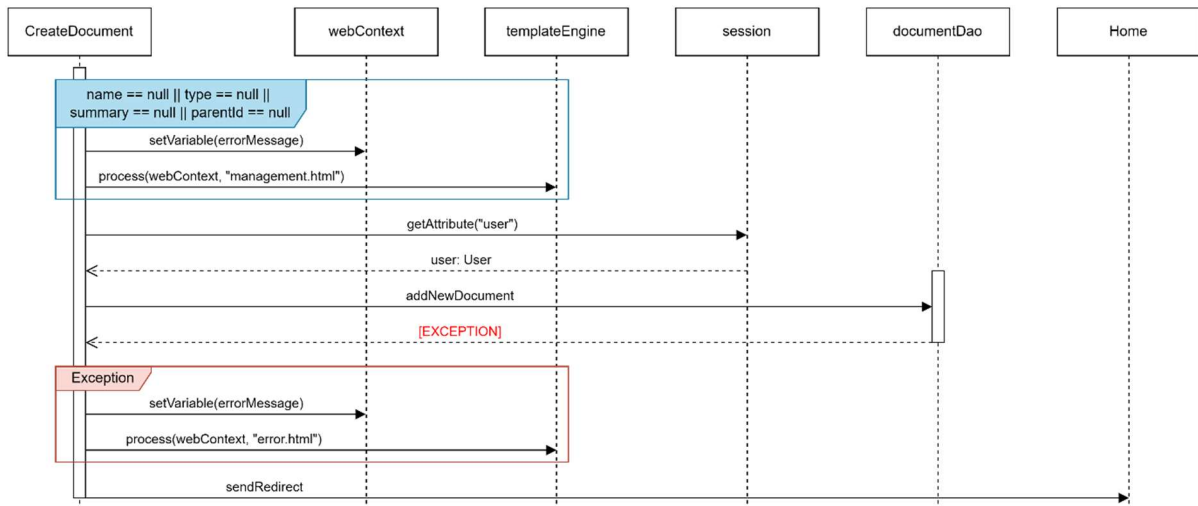
## GET /Management



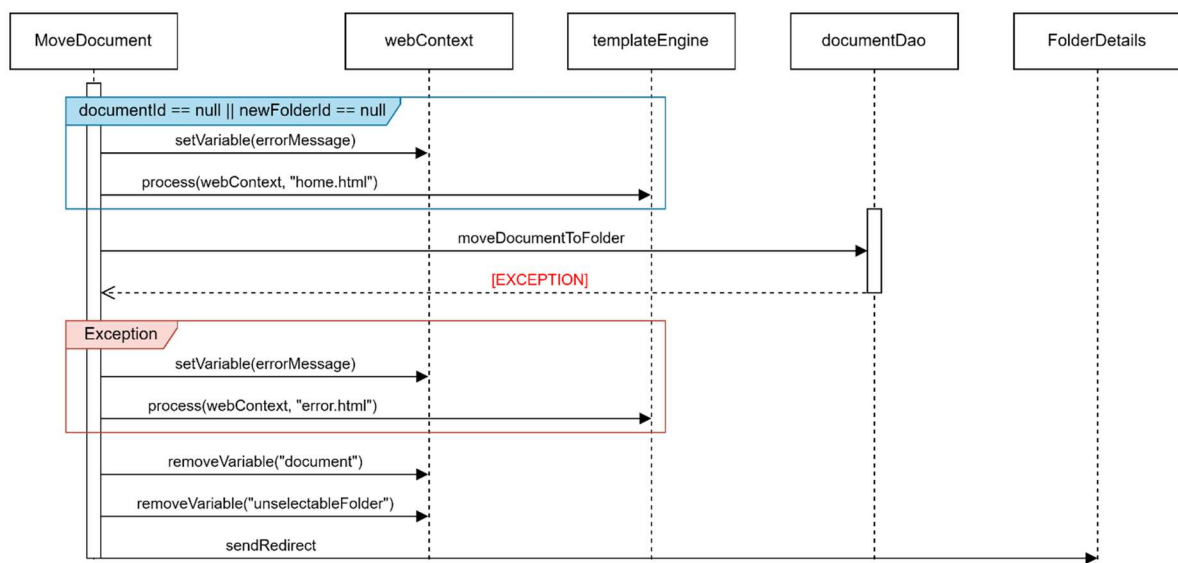
## POST /CreateFolder?name="x"&parentId="y"



## POST /CreateDocument?name="x"&type="y"&summary="z"&parentId="w"

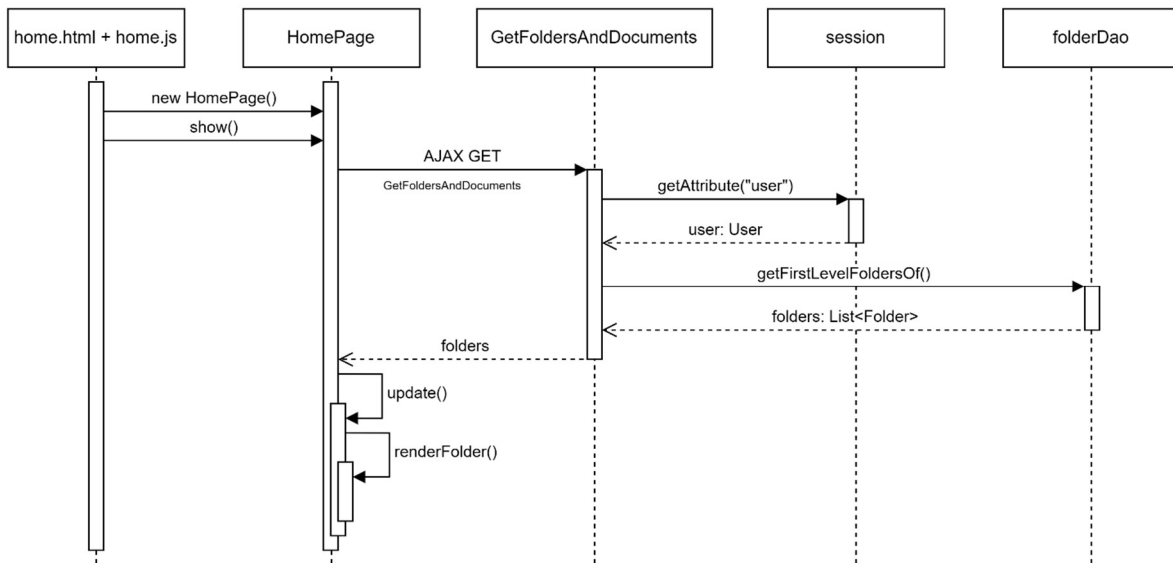


## POST /MoveDocument?documentId="x"&newFolderId="y"

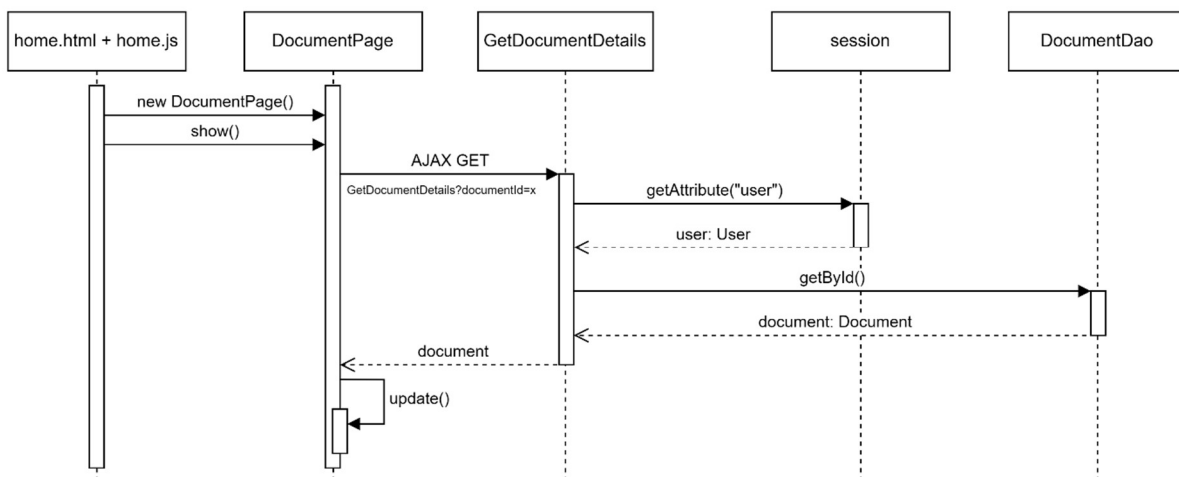


# RIA – Sequence Diagrams

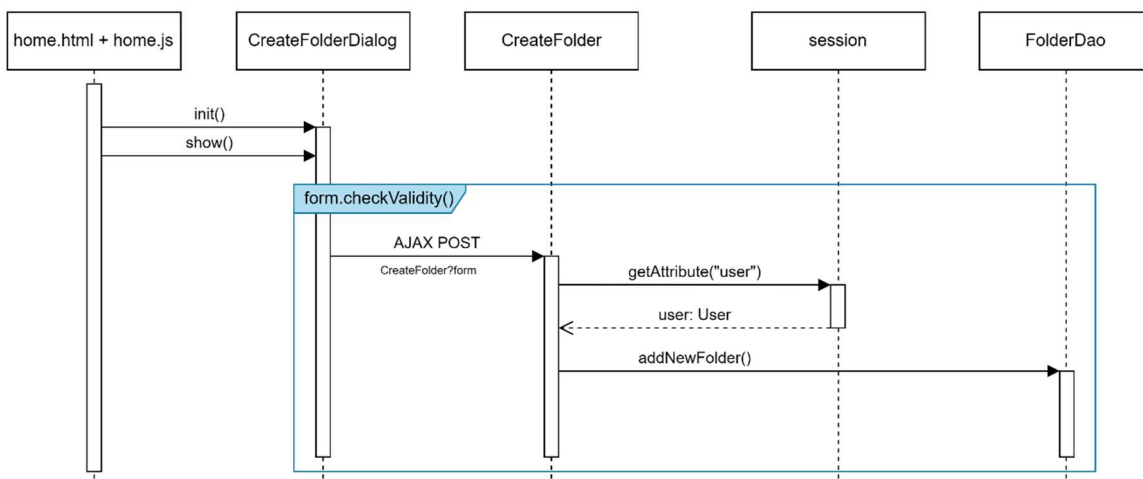
## Caricamento Home



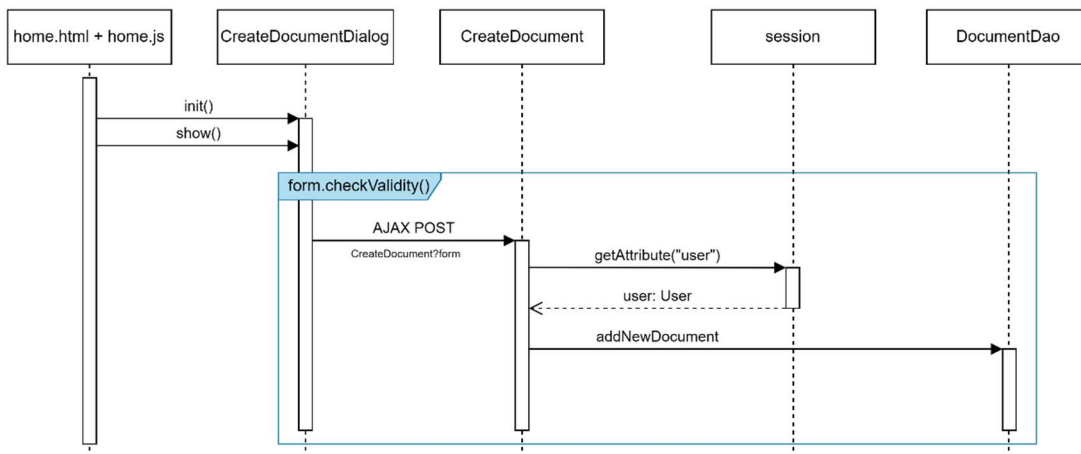
## Dettagli Documento



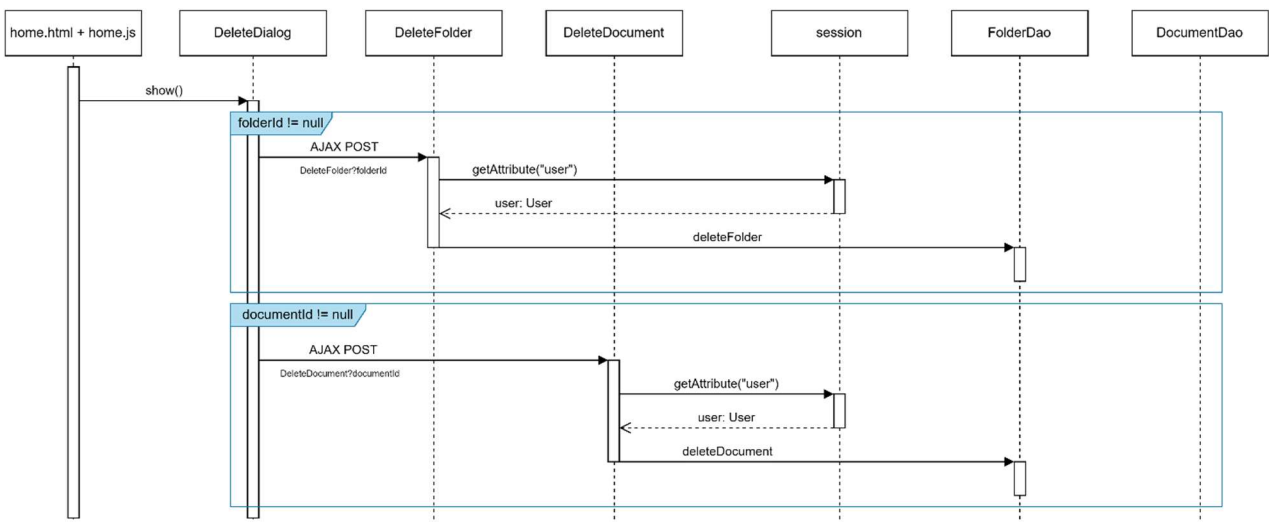
## Create Folder



## Create Document



## Delete



## Move

