

GALLERIA DI IMMAGINI

Tecnologie Informatiche per il Web
AA 2021-2022

PROGETTO REALIZZATO DAL GRUPPO 121
Matteo Spreafico (10669138)
Ludovica Tassini (10663137)

SPECIFICA DEL PROGETTO

Versione pure HTML

Un'applicazione web consente la gestione di una galleria d'immagini. 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”. La registrazione controlla l'unicità dello username. Ogni immagine è memorizzata come file nel file system del server su cui l'applicazione è rilasciata. Inoltre nella base di dati sono memorizzati i seguenti attributi: un titolo, una data, un testo descrittivo e il percorso del file dell'immagine nel file system del server. Le immagini sono associate all'utente che le carica. L'utente può creare album e associare a questi le proprie immagini. Un album ha un titolo, il creatore e la data di creazione. Le immagini sono associate a uno o più commenti inseriti dagli utenti (dal proprietario o da altri utenti). Un commento ha un testo e il nome dell'utente che lo ha creato. Quando l'utente accede all'HOME PAGE, questa presenta l'elenco degli album che ha creato e l'elenco degli album creati da altri utenti. Entrambi gli elenchi sono ordinati per data di creazione decrescente. Quando l'utente clicca su un album che appare negli elenchi della HOME PAGE, appare la pagina ALBUM PAGE che contiene inizialmente una tabella di una riga e cinque colonne. Ogni cella contiene una miniatura (thumbnail) e il titolo dell'immagine. Le miniature sono ordinate da sinistra a destra per data decrescente. Se l'album contiene più di cinque immagini, sono disponibili comandi per vedere il precedente e successivo insieme di cinque immagini. Se la pagina ALBUM PAGE mostra il primo blocco d'immagini e ne esistono altre successive nell'ordinamento, compare a destra della riga il bottone SUCCESSIVE, che permette di vedere le successive cinque immagini. Se la pagina ALBUM PAGE mostra l'ultimo blocco d'immagini e ne esistono altre precedenti nell'ordinamento, compare a sinistra della riga il bottone PRECEDENTI, che permette di vedere le cinque immagini precedenti. Se la pagina ALBUM PAGE mostra un blocco d'immagini e ne esistono altre precedenti e successive nell'ordinamento, compare a destra della riga il bottone SUCCESSIVE, che permette di vedere le successive cinque immagini, e a sinistra il bottone PRECEDENTI, che permette di vedere le cinque immagini precedenti. Quando l'utente seleziona una miniatura, la pagina ALBUM PAGE mostra tutti i dati dell'immagine scelta, tra cui la stessa immagine a grandezza naturale e i commenti eventualmente presenti. La pagina mostra anche una form per aggiungere un commento. L'invio del commento con un bottone INVIA ripresenta la pagina ALBUM PAGE, con tutti i dati aggiornati della stessa immagine. La pagina ALBUM PAGE contiene anche un collegamento per tornare all'HOME PAGE. L'applicazione consente il logout dell'utente.

SPECIFICA DEL PROGETTO

Versione con JavaScript

Si realizzi un'applicazione client server web che modifica le specifiche precedenti come segue:

- La registrazione 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 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.
- L'evento di visualizzazione del blocco precedente/successivo d'immagini di un album è gestito a lato client senza generare una richiesta al server.
- Quando l'utente passa con il mouse su una miniatura, l'applicazione mostra una finestra modale con tutte le informazioni dell'immagine, tra cui la stessa a grandezza naturale, i commenti eventualmente presenti e la form per inserire un commento.
- L'applicazione controlla anche a lato client che non si invii un commento vuoto.
- Errori a lato server devono essere segnalati mediante un messaggio di allerta all'interno della pagina.
- Si deve consentire all'utente di riordinare l'elenco dei propri album con un criterio diverso da quello di default (data decrescente). L'utente trascina il titolo di un album nell'elenco e lo colloca in una posizione diversa per realizzare l'ordinamento che desidera, senza invocare il server. Quando l'utente ha raggiunto l'ordinamento desiderato, usa un bottone “salva ordinamento”, per memorizzare la sequenza sul server. Ai successivi accessi, l'ordinamento personalizzato è usato al posto di quello di default.

COMPLETAMENTO DELLE SPECIFICHE

Aspetti funzionali

- È stato deciso di permettere il login all'applicazione web unicamente tramite username e password, senza ricorrere all'email.
- Un apposito filtro controlla che utenti non autorizzati non abbiano accesso a pagine diverse da quella di login.

Per quanto riguarda la versione pure HTML:

- È permesso il logout anche dalla pagina ALBUM PAGE.
- Una volta creato un album, viene data immediatamente la possibilità di aggiungervi immagini mediante un'opportuna form. Qualora non lo si voglia fare subito, è possibile, dalla pagina ALBUM PAGE, accedere nuovamente alla pagina contenente la form per effettuare l'inserimento di immagini nell'album selezionato.

Per quanto riguarda la versione con JavaScript:

- Una volta creato o selezionato un album, viene data la possibilità di aggiungervi immagini mediante un'opportuna form.

COMPLETAMENTO DELLE SPECIFICHE

Limitazioni agli input testuali dell'utente

- L'username e la password di un utente non devono superare i 20 caratteri, mentre l'email non deve superare i 100 caratteri. In ambedue i casi, non sono ammessi spazi bianchi.
- Il titolo di un album non accetta più di 30 caratteri.
- Il titolo di un'immagine non può superare i 30 caratteri, mentre la descrizione può raggiungere al massimo 280 caratteri.
- Il testo di un commento può essere lungo al più 280 caratteri.

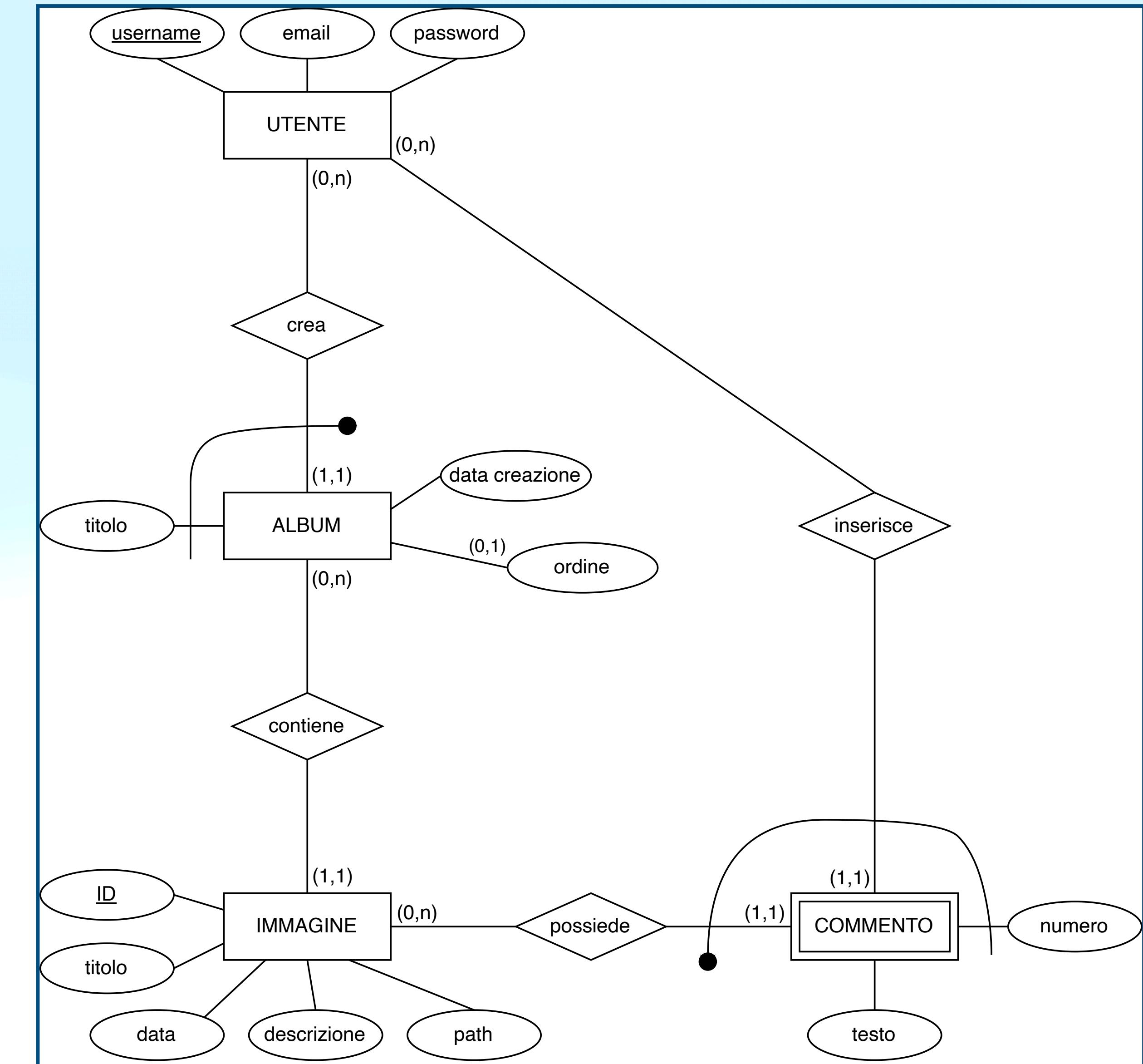
ANALISI DEI DATI

Entità, Attributi, Relazioni

Un'applicazione web consente la gestione di una galleria d'immagini. 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”. La registrazione controlla l'unicità dello **username**. Ogni **immagine** è memorizzata come file nel file system del server su cui l'applicazione è rilasciata. Inoltre nella base di dati sono memorizzati i seguenti attributi: un **titolo**, una **data**, un **testo descrittivo** e il **percorso** del file dell'immagine nel file system del server. Le **immagini** sono associate all'utente che le carica. L'**utente** può creare **album** e associare a questi le proprie **immagini**. Un **album** ha un **titolo**, il **creatore** e la **data di creazione**. Le **immagini** sono associate a uno o più **commenti** inseriti dagli **utenti** (dal proprietario o da altri utenti). Un **commento** ha un **testo** e il **nome dell'utente che lo ha creato**. Quando l'utente accede all'HOME PAGE, questa presenta l'elenco degli album che ha creato e l'elenco degli album creati da altri utenti. Entrambi gli elenchi sono ordinati per data di creazione decrescente. Quando l'utente clicca su un album che appare negli elenchi della HOME PAGE, appare la pagina ALBUM PAGE che contiene inizialmente una tabella di una riga e cinque colonne. Ogni cella contiene una miniatura (thumbnail) e il titolo dell'immagine. Le miniature sono ordinate da sinistra a destra per data decrescente. Se l'album contiene più di cinque immagini, sono disponibili comandi per vedere il precedente e successivo insieme di cinque immagini. Se la pagina ALBUM PAGE mostra il primo blocco d'immagini e ne esistono altre successive nell'ordinamento, compare a destra della riga il bottone **SUCCESSIONE**, che permette di vedere le successive cinque immagini. Se la pagina ALBUM PAGE mostra l'ultimo blocco d'immagini e ne esistono altre precedenti nell'ordinamento, compare a sinistra della riga il bottone **PRECEDENTI**, che permette di vedere le cinque immagini precedenti. Se la pagina ALBUM PAGE mostra un blocco d'immagini e ne esistono altre precedenti e successive nell'ordinamento, compare a destra della riga il bottone **SUCCESSIONE**, che permette di vedere le successive cinque immagini, e a sinistra il bottone **PRECEDENTI**, che permette di vedere le cinque immagini precedenti. Quando l'utente seleziona una miniatura, la pagina ALBUM PAGE mostra tutti i dati dell'immagine scelta, tra cui la stessa immagine a grandezza naturale e i commenti eventualmente presenti. La pagina mostra anche una form per aggiungere un commento. L'invio del commento con un bottone **INVIA** ripresenta la pagina ALBUM PAGE, con tutti i dati aggiornati della stessa immagine. La pagina ALBUM PAGE contiene anche un collegamento per tornare all'HOME PAGE. L'applicazione consente il logout dell'utente.

DIAGRAMMA ENTITÀ-RELAZIONE

Progetto della base di dati



STRUTTURA LOCALE DEL DATABASE

Tabella “utente”

```
CREATE TABLE `utente` (
    `username` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    `email` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    `password` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    PRIMARY KEY (`username`),
    UNIQUE KEY `email_UNIQUE` (`email`)
)
```

STRUTTURA LOCALE DEL DATABASE

Tabella “album”

```
CREATE TABLE `album` (
    `proprietario` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    `titolo` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    `data_creazione` timestamp NOT NULL,
    `ordine` int DEFAULT NULL,
    PRIMARY KEY (`proprietario`,`titolo`),
    KEY `titolo_idx` (`titolo`),
    CONSTRAINT `proprietario` FOREIGN KEY (`proprietario`) REFERENCES `utente` (`username`) ON DELETE CASCADE ON UPDATE CASCADE
)
```

STRUTTURA LOCALE DEL DATABASE

Tabella “immagine”

```
CREATE TABLE `immagine` (
    `id` int NOT NULL AUTO_INCREMENT,
    `proprietario_album` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    `titolo_album` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    `titolo_immagine` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    `data` timestamp NOT NULL,
    `descrizione` varchar(280) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    `path` varchar(260) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
    PRIMARY KEY (`id`),
    KEY `titolo_album_idx` (`titolo_album`),
    KEY `proprietario_album_idx` (`proprietario_album`),
    CONSTRAINT `proprietario_album` FOREIGN KEY (`proprietario_album`) REFERENCES `album` (`proprietario`) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT `titolo_album` FOREIGN KEY (`titolo_album`) REFERENCES `album` (`titolo`) ON DELETE CASCADE ON UPDATE CASCADE
)
```

STRUTTURA LOCALE DEL DATABASE

Tabella “commento”

```
CREATE TABLE `commento` (
    `autore` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    `id_immagine` int NOT NULL,
    `numero_commento` int NOT NULL,
    `testo` varchar(280) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_cs NOT NULL,
    PRIMARY KEY (`autore`,`id_immagine`,`numero_commento`),
    KEY `id_immagine_idx` (`id_immagine`),
    CONSTRAINT `autore` FOREIGN KEY (`autore`) REFERENCES `utente` (`username`) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT `id_immagine` FOREIGN KEY (`id_immagine`) REFERENCES `immagine` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
)
```

VERSIIONE PURE HTML

Progetto dell'applicazione

ANALISI DEI REQUISITI DELL'APPLICAZIONE

Pagine, Componenti della vista, Eventi, Azioni

Un'applicazione web consente la gestione di una galleria d'immagini. 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". La registrazione controlla l'unicità dello username. Ogni immagine è memorizzata come file nel file system del server su cui l'applicazione è rilasciata. Inoltre nella base di dati sono memorizzati i seguenti attributi: un titolo, una data, un testo descrittivo e il percorso del file dell'immagine nel file system del server. Le immagini sono associate all'utente che le carica. L'utente può **creare album** e **associare a questi le proprie immagini**. Un album ha un titolo, il creatore e la data di creazione. Le immagini sono associate a uno o più commenti inseriti dagli utenti (dal proprietario o da altri utenti). Un commento ha un testo e il nome dell'utente che lo ha creato. Quando l'utente accede all'**HOME PAGE**, questa presenta l'**elenco degli album** che ha creato e l'**elenco degli album creati da altri utenti**. Entrambi gli elenchi sono ordinati per data di creazione decrescente. Quando l'utente **clicca su un album** che appare negli elenchi della HOME PAGE, appare la pagina **ALBUM PAGE** che contiene inizialmente una **tabella di una riga e cinque colonne**. Ogni cella contiene una **miniatura (thumbnail)** e il **titolo dell'immagine**. Le miniature sono ordinate da sinistra a destra per data decrescente. Se l'album contiene più di cinque immagini, sono disponibili **comandi per vedere il precedente e successivo insieme di cinque immagini**. Se la pagina ALBUM PAGE mostra il primo blocco d'immagini e ne esistono altre successive nell'ordinamento, compare a destra della riga il bottone **SUCCESSIVE**, che permette di vedere le successive cinque immagini. Se la pagina ALBUM PAGE mostra l'ultimo blocco d'immagini e ne esistono altre precedenti nell'ordinamento, compare a sinistra della riga il bottone **PRECEDENTI**, che permette di vedere le cinque immagini precedenti. Se la pagina ALBUM PAGE mostra un blocco d'immagini e ne esistono altre precedenti e successive nell'ordinamento, compare a destra della riga il bottone **SUCCESSIVE**, che permette di vedere le successive cinque immagini, e a sinistra il bottone **PRECEDENTI**, che permette di vedere le cinque immagini precedenti. Quando l'utente **seleziona una miniatura**, la pagina ALBUM PAGE mostra tutti i dati dell'immagine scelta, tra cui la stessa immagine a grandezza naturale e i commenti eventualmente presenti. La pagina mostra anche una **form per aggiungere un commento**. L'**invio del commento** con un bottone **INVIA** ripresenta la pagina ALBUM PAGE, con tutti i dati aggiornati della stessa immagine. La pagina ALBUM PAGE contiene anche un **collegamento per tornare all'HOME PAGE**. L'applicazione consente il **logout** dell'utente.

COMPONENTI DELL'APPLICAZIONE

Viste, Beans, DAOs e Servlets

- **VISTE:** login, HomePage, AlbumPage, AddImage
- **BEANS:** Album, Comment, Image, User
- **DATA ACCESS OBJECTS:** AlbumDAO, CommentDAO, ImageDAO, UserDAO
- **SERVLETS:** AddComment, AddImage, AlbumPage, CheckLogin, CheckRegister, CreateAlbum, GetImage, GoToAlbumPage, GoToAddImage, HomePage, Logout

L'applicazione prevede anche l'uso dei **filtri** “AccessControl” e “AlbumFilter” .

Nota: i componenti sottolineati si riferiscono a funzionalità non richieste che sono state implementate facoltativamente.

COMPONENTI DELL'APPLICAZIONE

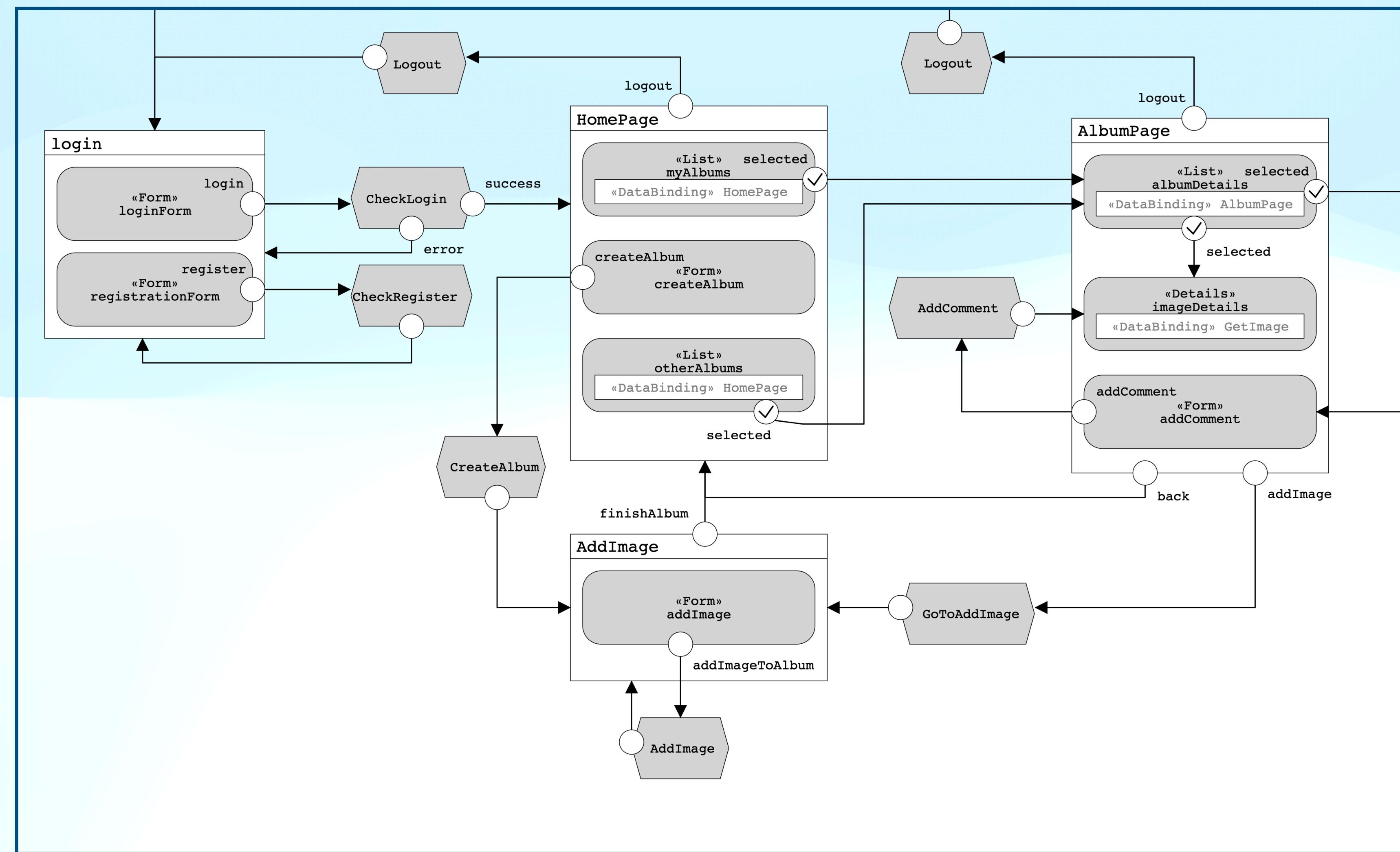
Approfondimento sui Data Access Objects (DAOs)

La tabella a fianco elenca i metodi dei DAOs che permettono l'interrogazione e l'aggiornamento del database.

AlbumDAO	CommentDAO	ImageDAO	UserDAO
getUserAlbums (username)	createComment (comment)	getImageID() getAlbumImages (albumOwner, albumTitle)	checkCredentials (username, password)
getOtherAlbums (username)	getCommentNumber (imageID)	getImageFromID (imageID)	checkCredentials Registration(email, username)
buildAlbumList (query, username)	getImageComments (imageID)	getImageFromIDs FromTitle(imageID)	
createAlbum(album)		createImage(image)	
checkAlbumTitle (album)		hasNextSetOfFive (albumOwner, albumTitle, pageOffset)	registerUser(user)

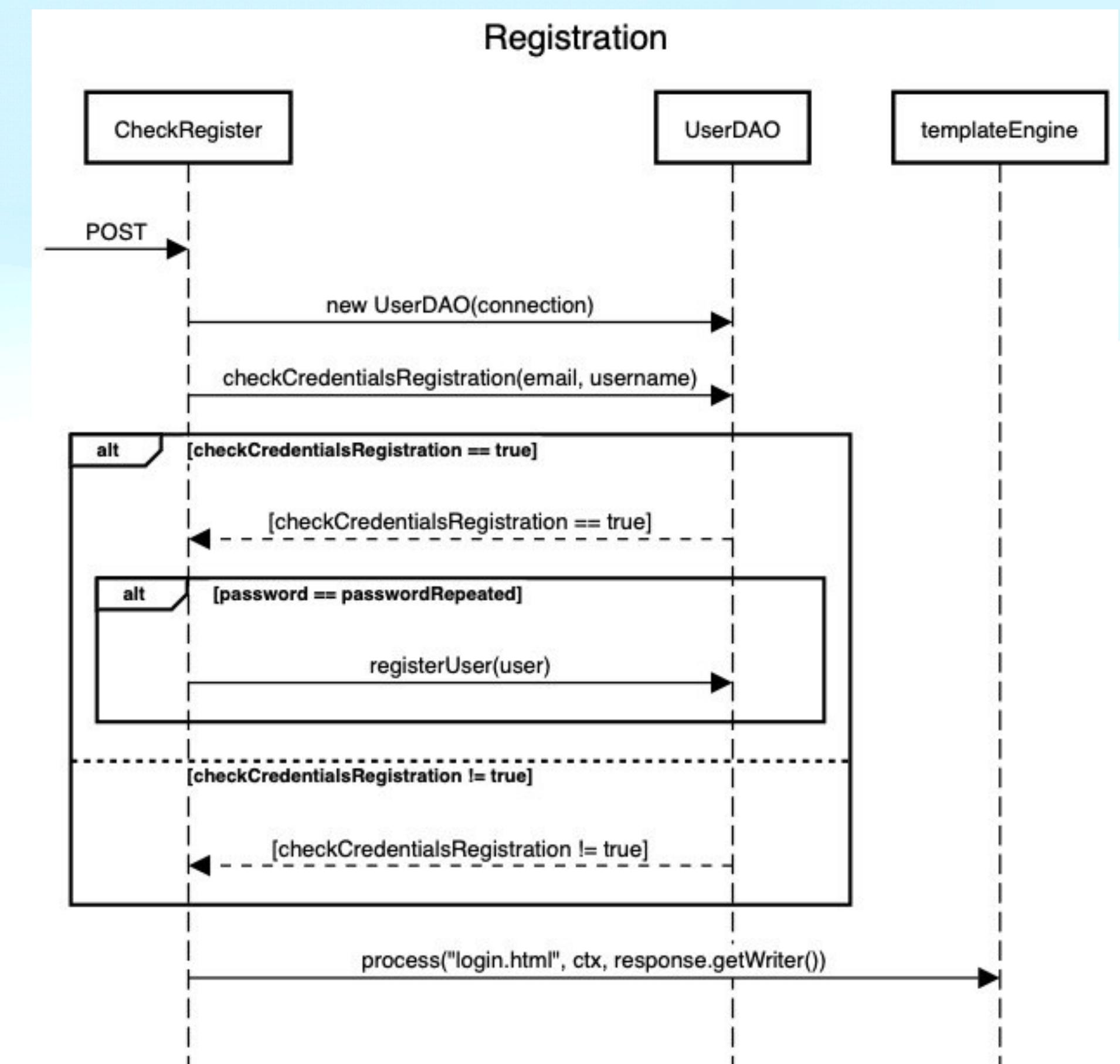
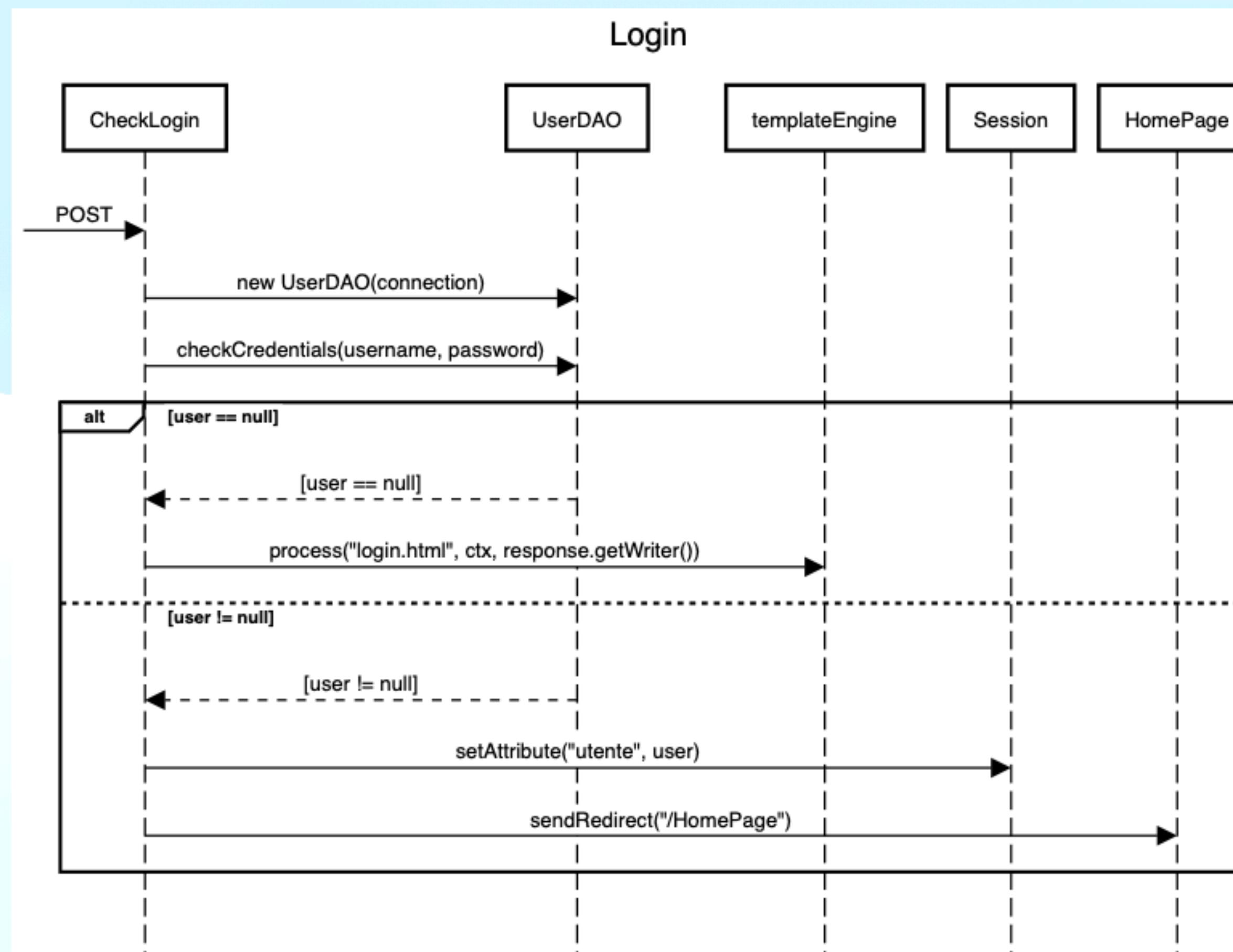
DIAGRAMMA IFML

Design dell'applicazione



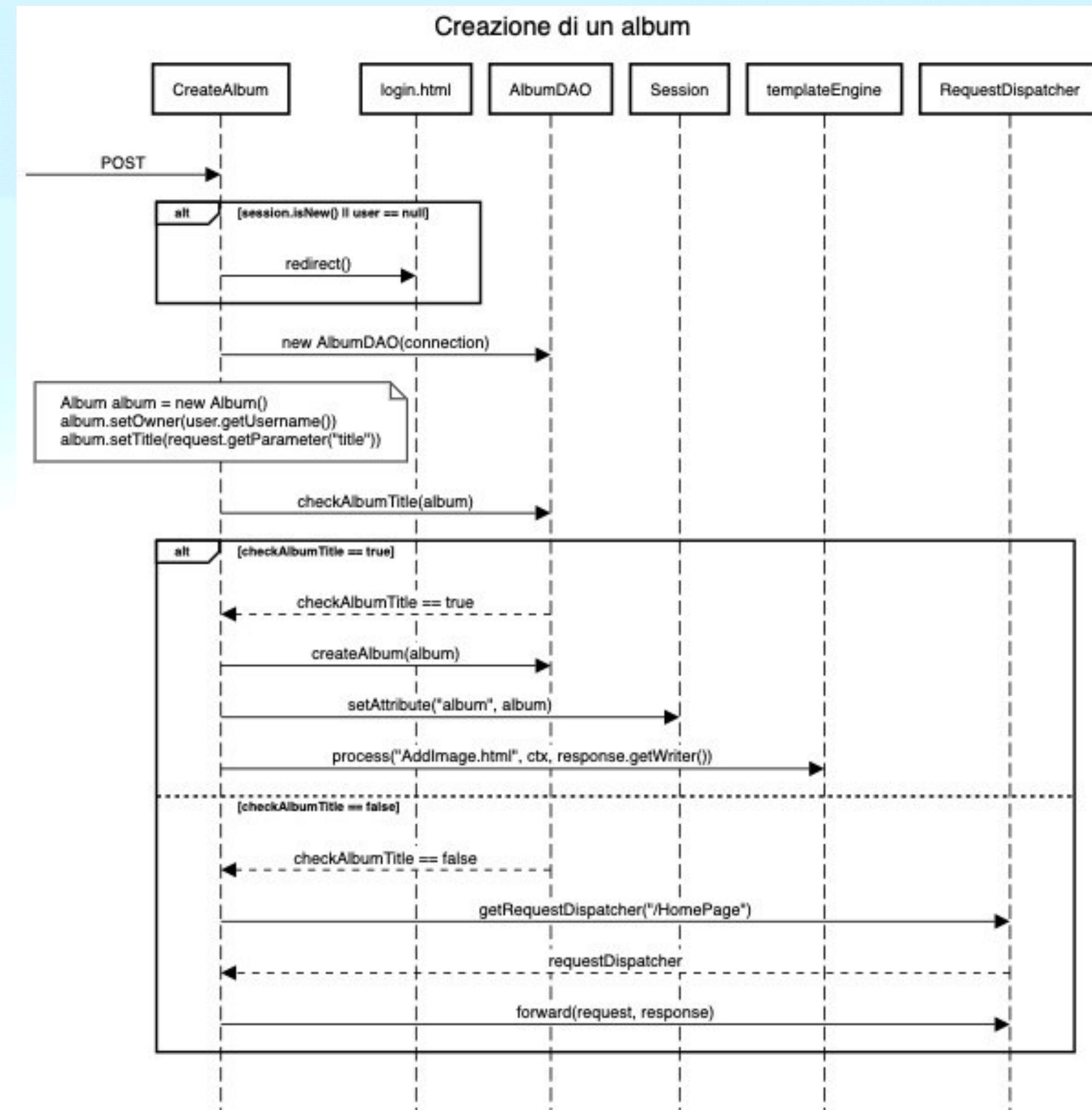
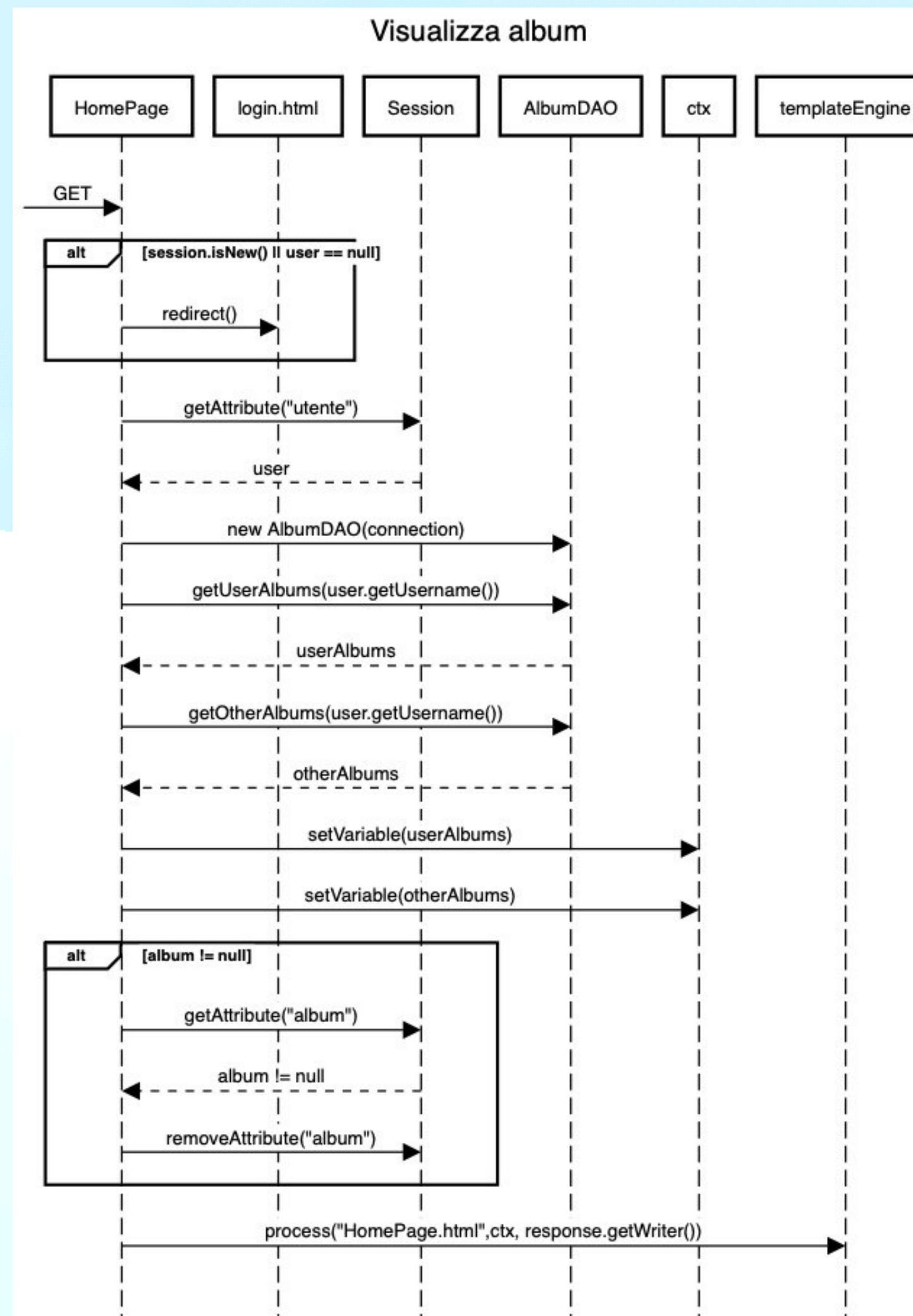
SEQUENCE DIAGRAMS

Login e registrazione



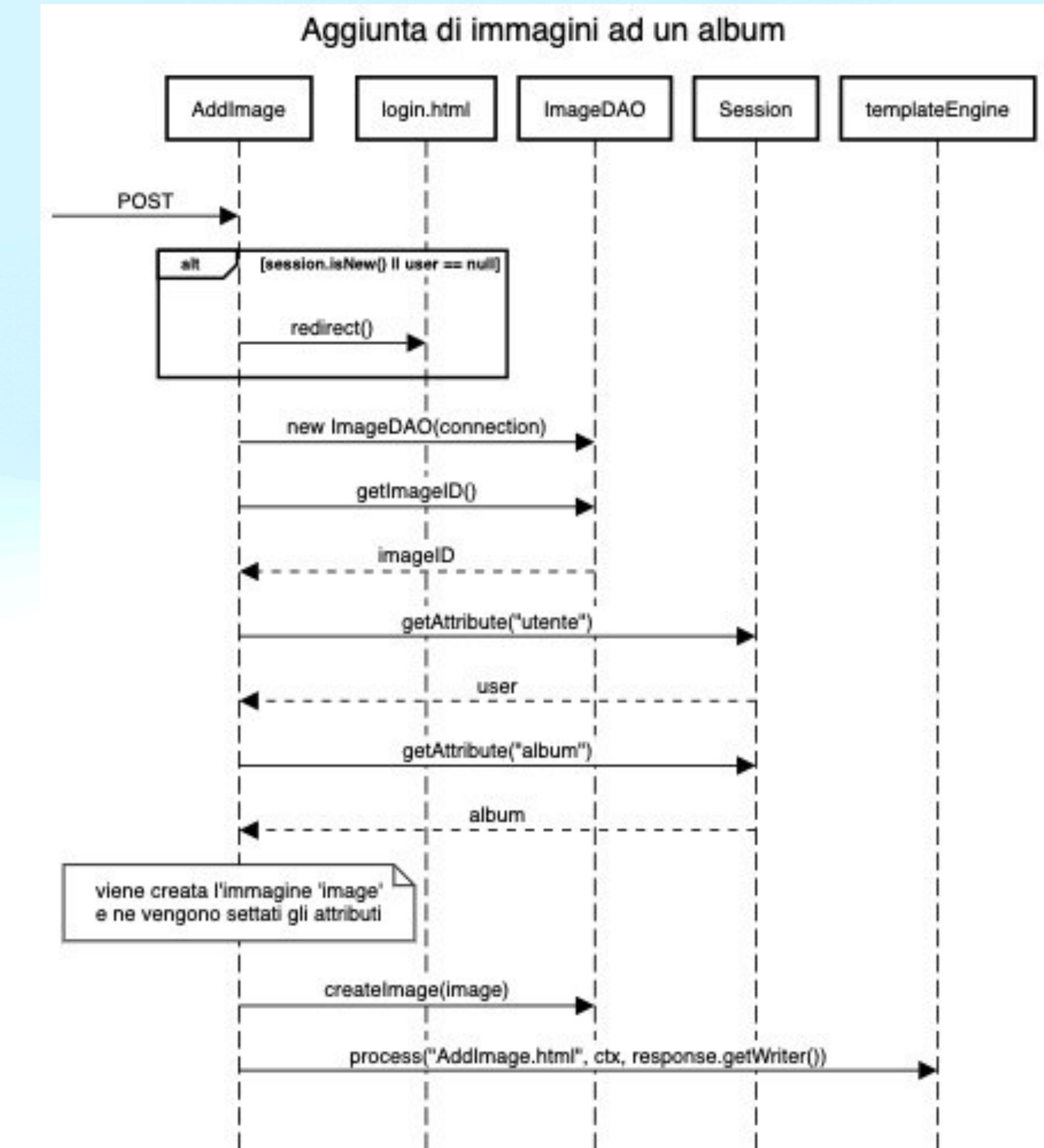
SEQUENCE DIAGRAMS

Visualizzazione e creazione di un album



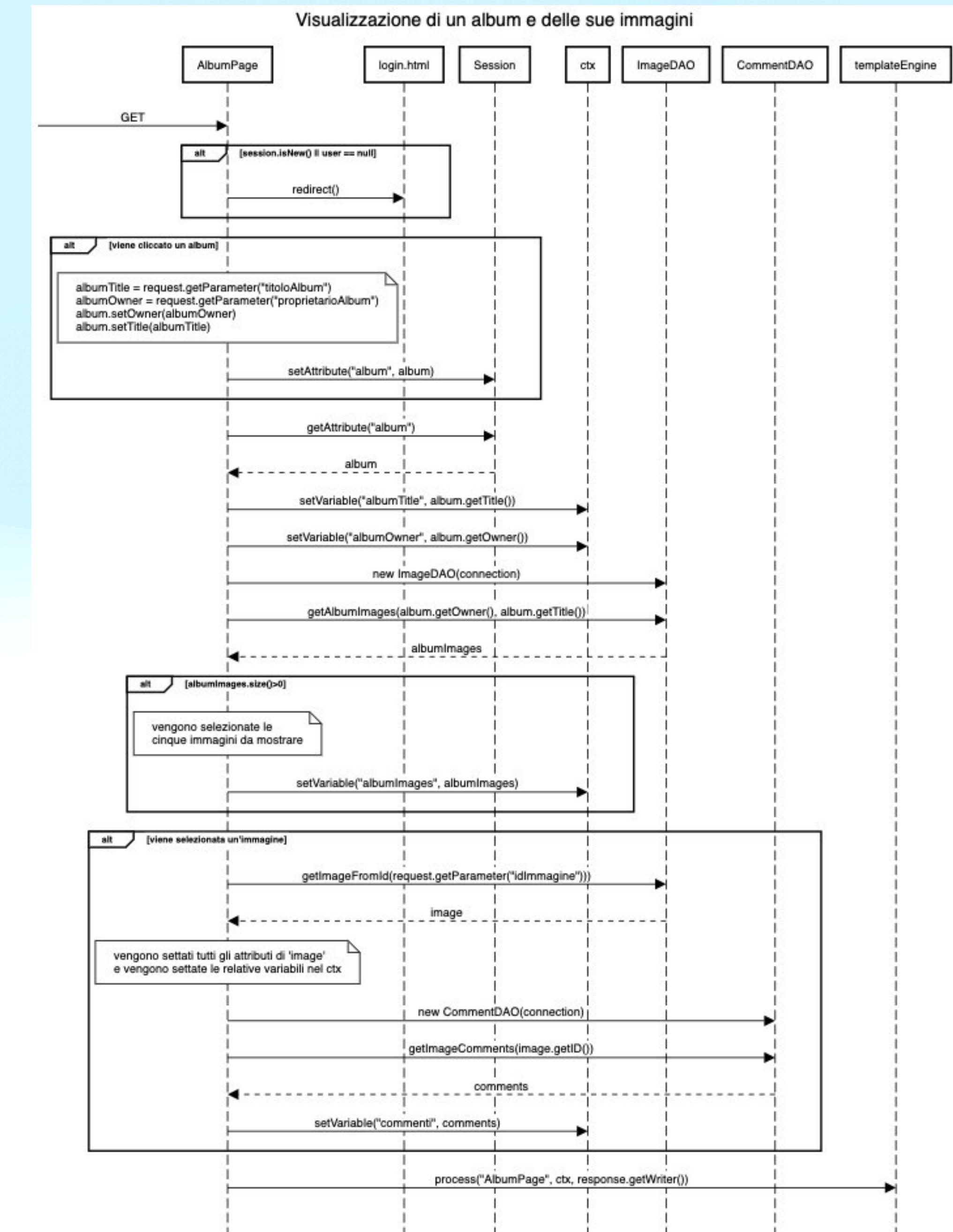
SEQUENCE DIAGRAMS

Aggiunta di un'immagine a un album (facoltativo)



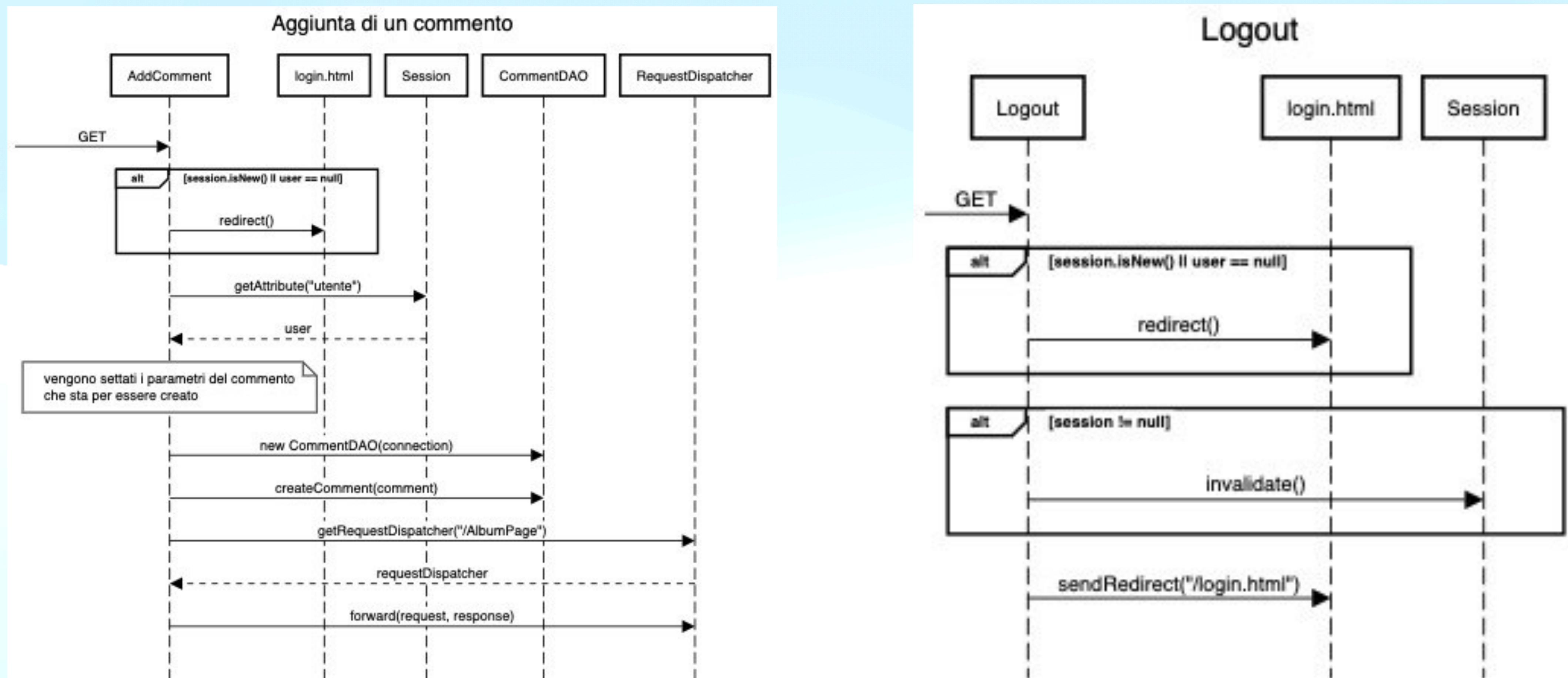
SEQUENCE DIAGRAMS

Visualizzazione dei dettagli di un album e delle sue immagini



SEQUENCE DIAGRAMS

Aggiunta di un commento e logout



VERSIONE CON JAVASCRIPT

Progetto dell'applicazione

ANALISI DEI REQUISITI DELL'APPLICAZIONE

Pagine, Componenti della vista, Eventi, Azioni

Si realizzi un'applicazione client server web che modifica le specifiche precedenti come segue:

- La **registrazione** 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** 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.
- L'evento di **visualizzazione del blocco precedente/successivo d'immagini di un album** è gestito a lato client senza generare una richiesta al server.
- Quando l'utente passa con il mouse su una miniatura, l'applicazione mostra una finestra modale con tutte le **informazioni dell'immagine**, tra cui la stessa a grandezza naturale, i commenti eventualmente presenti e la **form per inserire un commento**.
- L'applicazione controlla anche a lato client che non si invii un commento vuoto.
- Errori a lato server devono essere segnalati mediante un **messaggio di allerta** all'interno della pagina.
- Si deve consentire all'utente di **riordinare l'elenco dei propri album** con un criterio diverso da quello di default (data decrescente). L'utente trascina il titolo di un album nell'elenco e lo colloca in una posizione diversa per realizzare l'ordinamento che desidera, senza invocare il server. Quando l'utente ha raggiunto l'ordinamento desiderato, usa un bottone “salva ordinamento”, per memorizzare la sequenza sul server. Ai successivi accessi, l'ordinamento personalizzato è usato al posto di quello di default.

COMPONENTI DELL'APPLICAZIONE

Viste, Beans, DAOs, Servlets e Scripts

- **VISTE:** login, ImageGallery
- **BEANS:** Album, Comment, Image, User
- **DATA ACCESS OBJECTS:** AlbumDAO, CommentDAO, ImageDAO, UserDAO
- **SERVLETS:** AddComment, AddImage, CheckLogin, CheckRegister, CreateAlbum, GetComments, GetImages, GetOtherAlbums, GetUserAlbums, GetUserAlbumsByDate, Logout, SaveOrder
- **SCRIPTS:** ImageGalleryHandler, LoginHandler, RegistrationHandler

L'applicazione prevede anche l'uso del **filtro “AccessControl”**.

Nota: i componenti sottolineati si riferiscono a funzionalità non richieste che sono state implementate facoltativamente.

COMPONENTI DELL'APPLICAZIONE

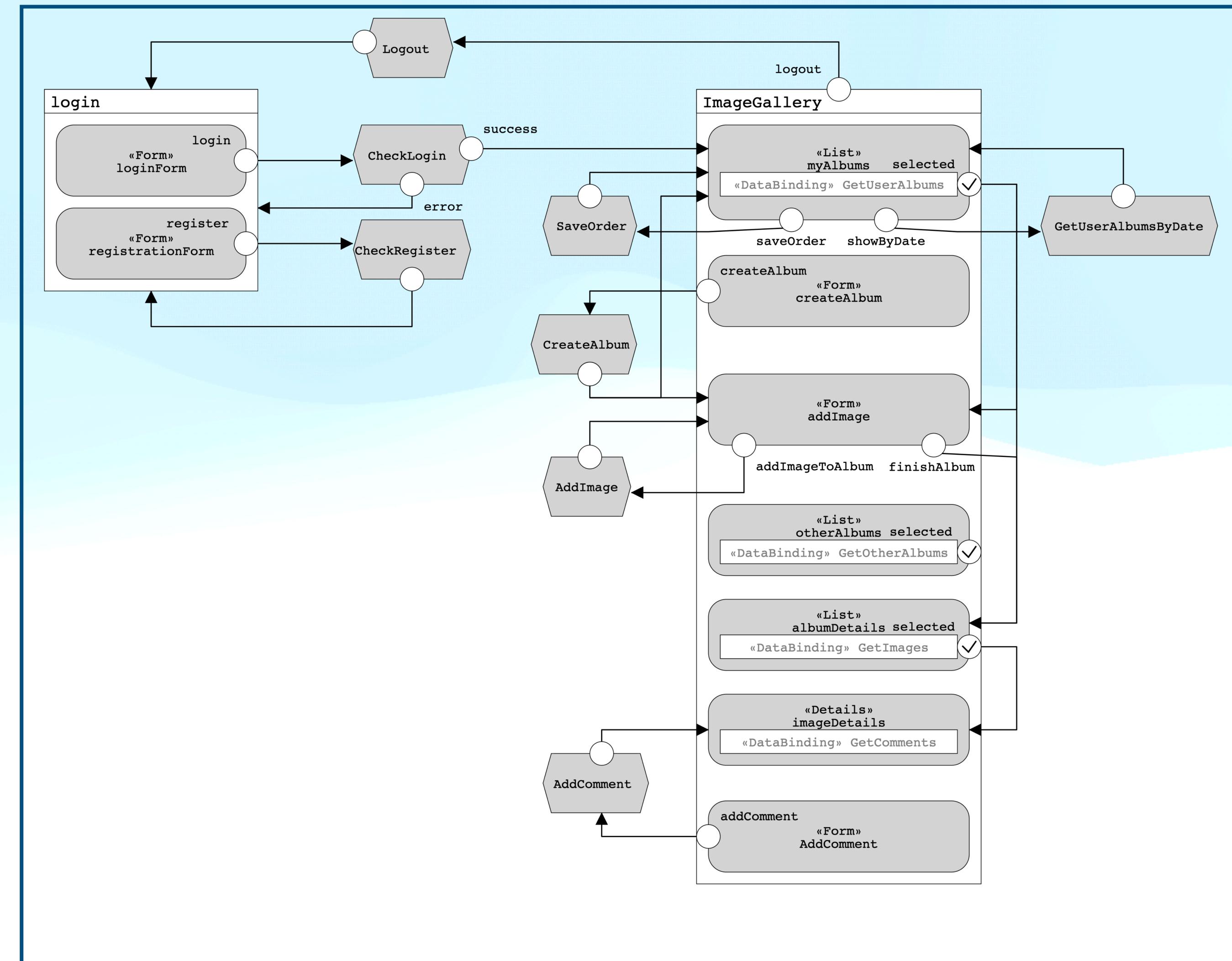
Approfondimento sui Data Access Objects (DAOs)

La tabella a fianco elenca i metodi dei DAOs che permettono l'interrogazione e l'aggiornamento del database.

AlbumDAO	CommentDAO	ImageDAO	UserDAO
getUserAlbums (username)	createComment (comment)	getImageID()	checkCredentials (username, password)
getUserAlbumsByDate (username)			
getOtherAlbums (username)	getComment Number (imageID)	getAlbumImages (albumOwner, albumTitle)	checkCredentials Registration(email, username)
buildAlbumList (query, username)			
createAlbum(album)			
checkAlbumTitle (album)	getImage Comments (imageID)	createImage(image)	
modifyAlbumOrder (newIndex, albumOwner, albumTitle)			registerUser(user)

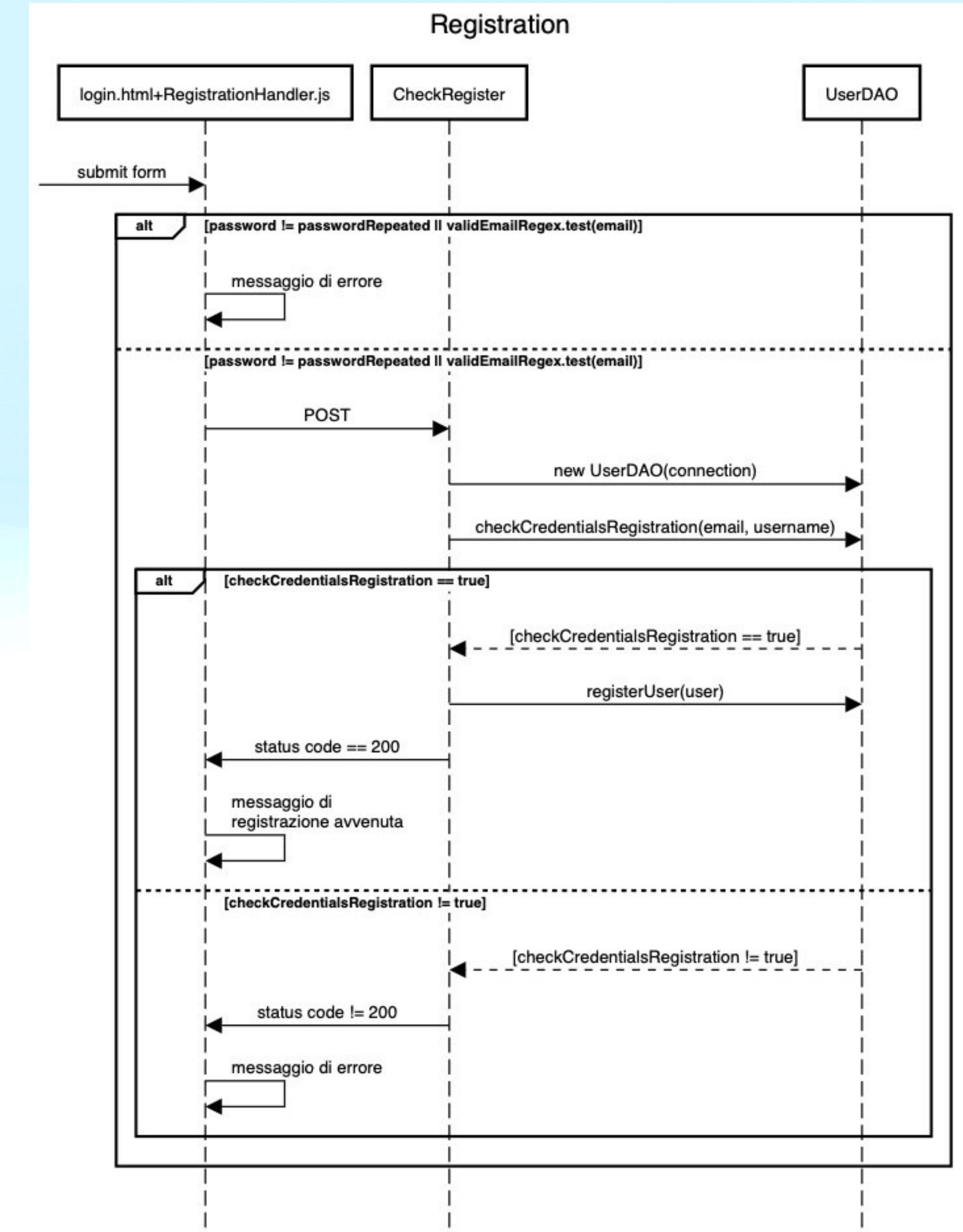
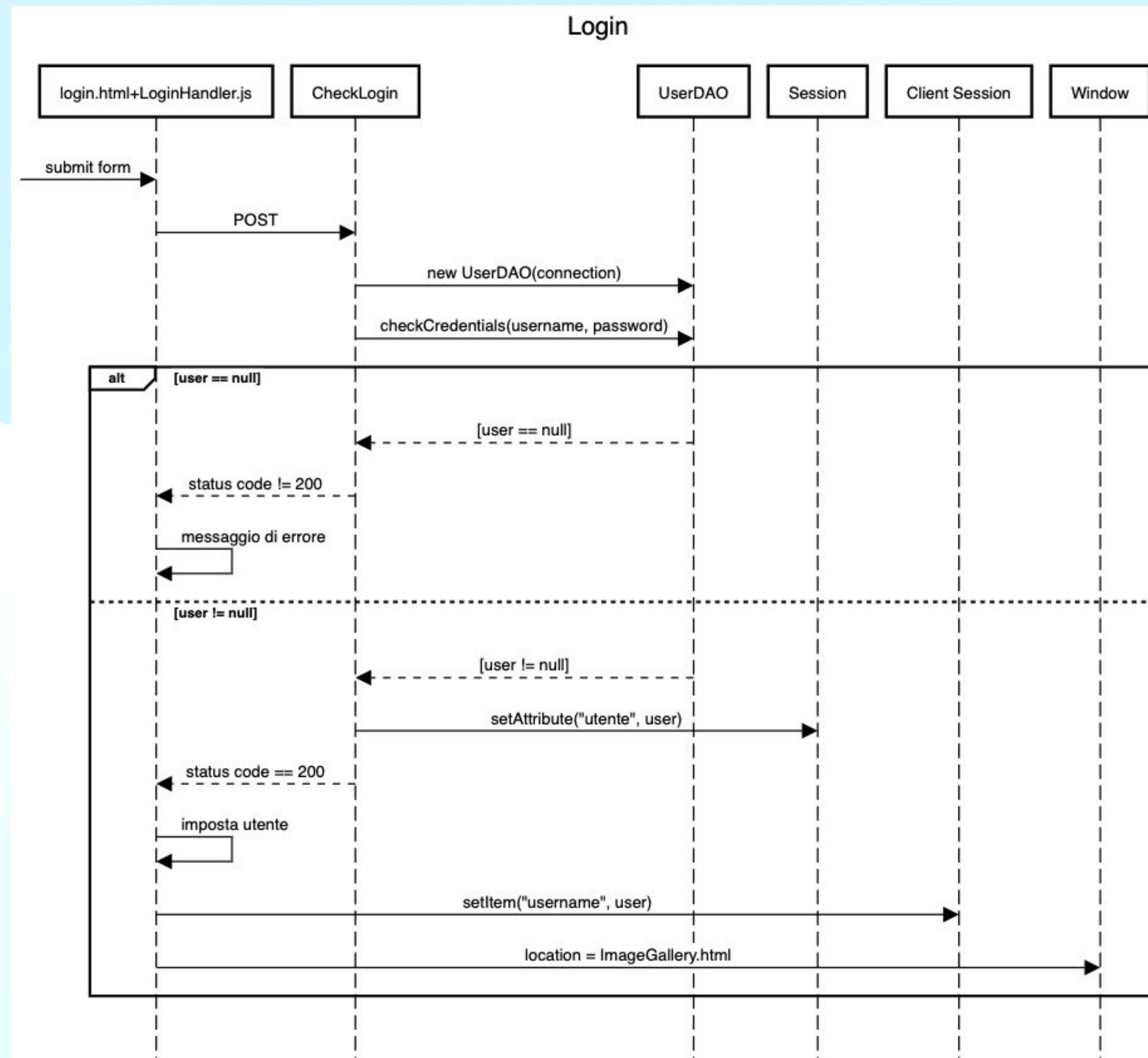
DIAGRAMMA IFML

Design dell'applicazione



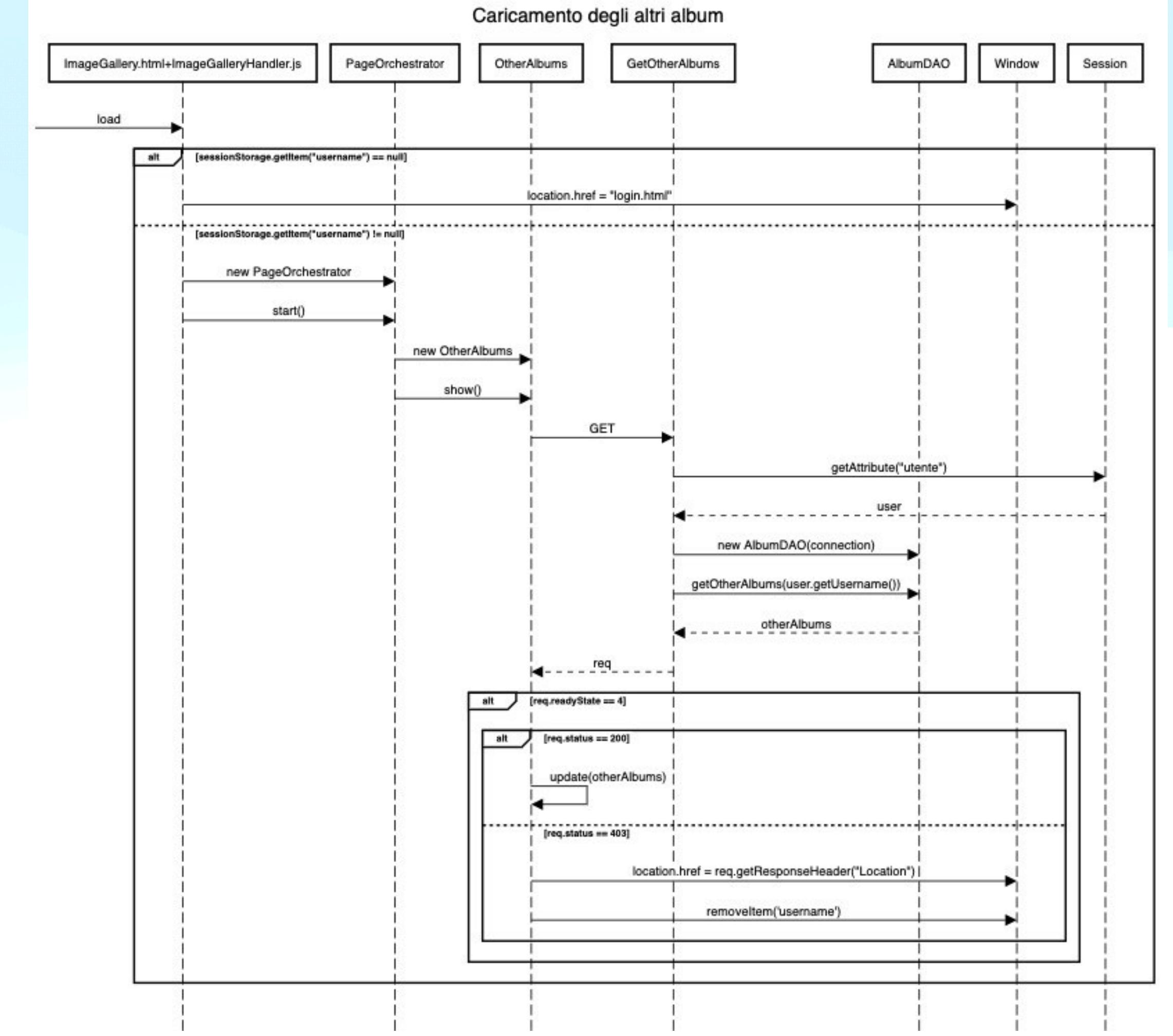
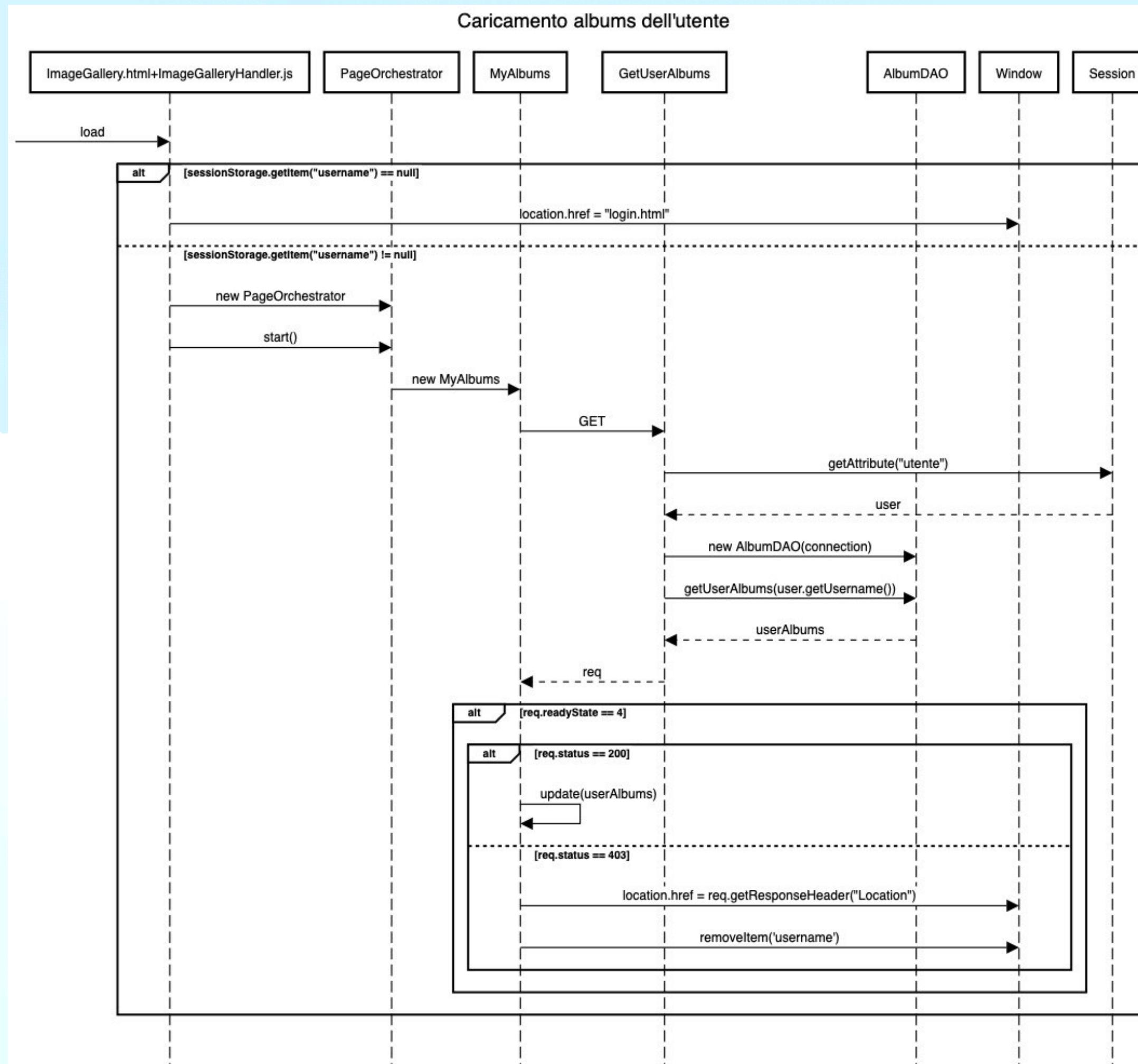
SEQUENCE DIAGRAMS

Login e registrazione



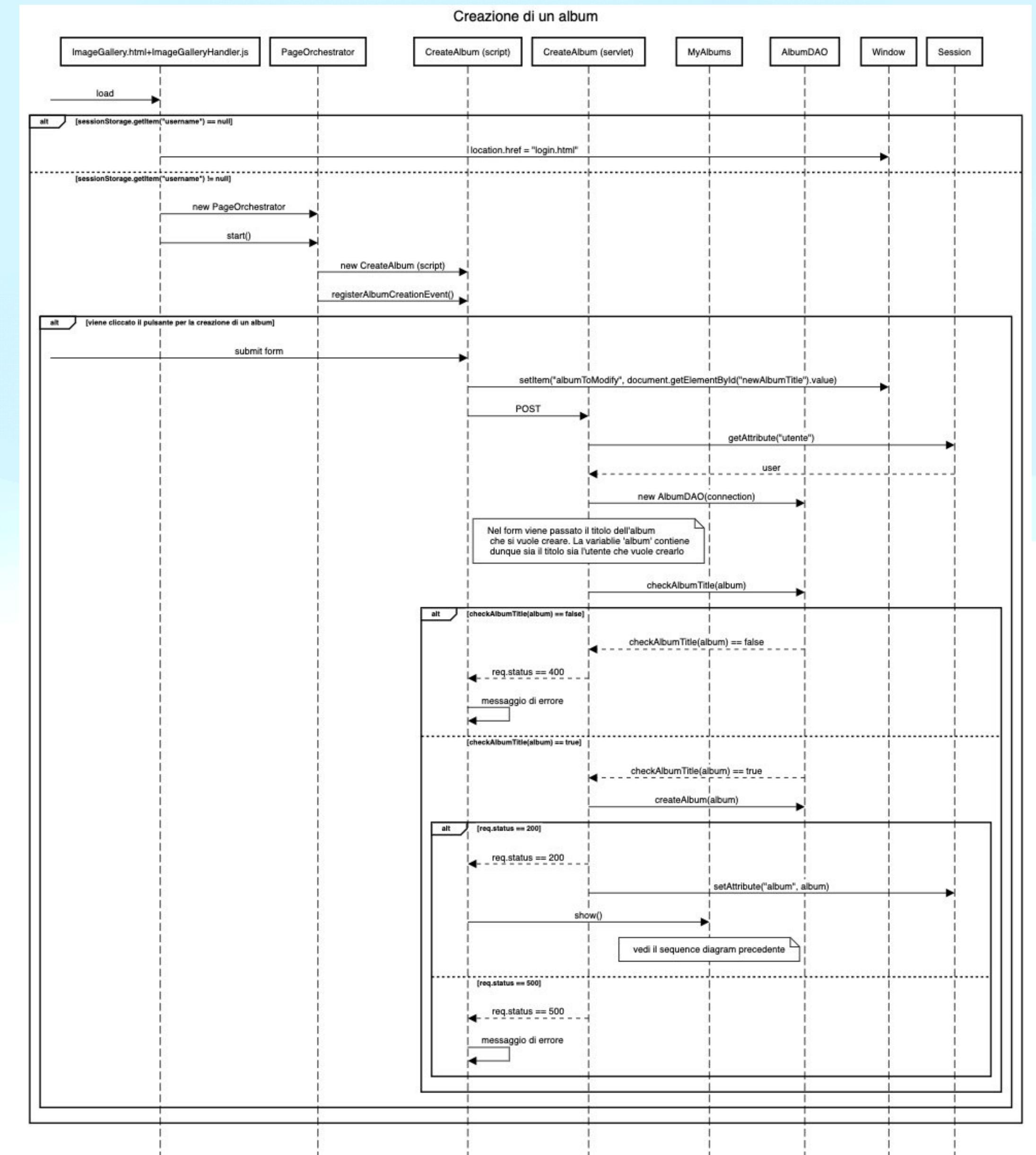
SEQUENCE DIAGRAMS

Visualizzazione degli album



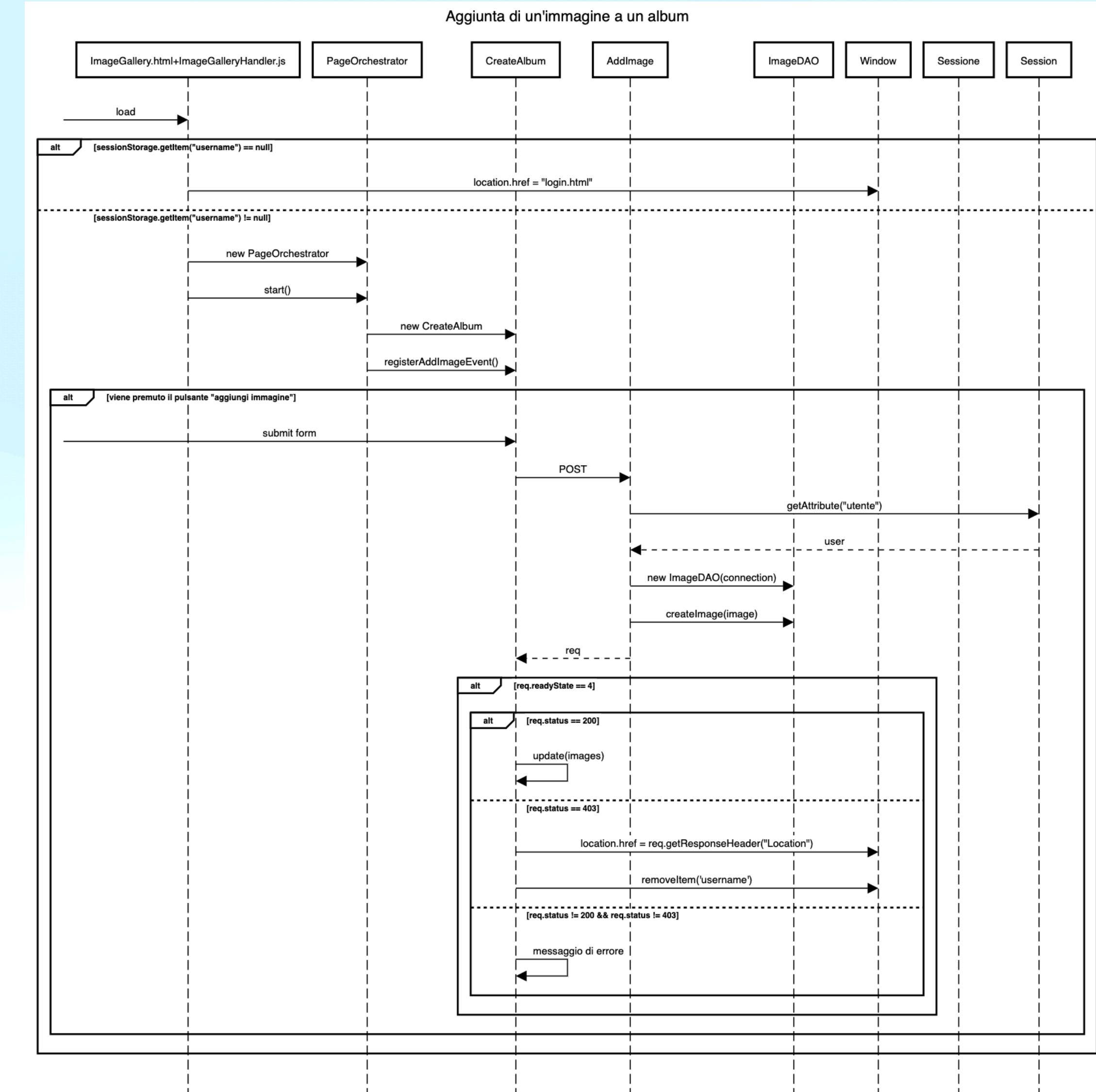
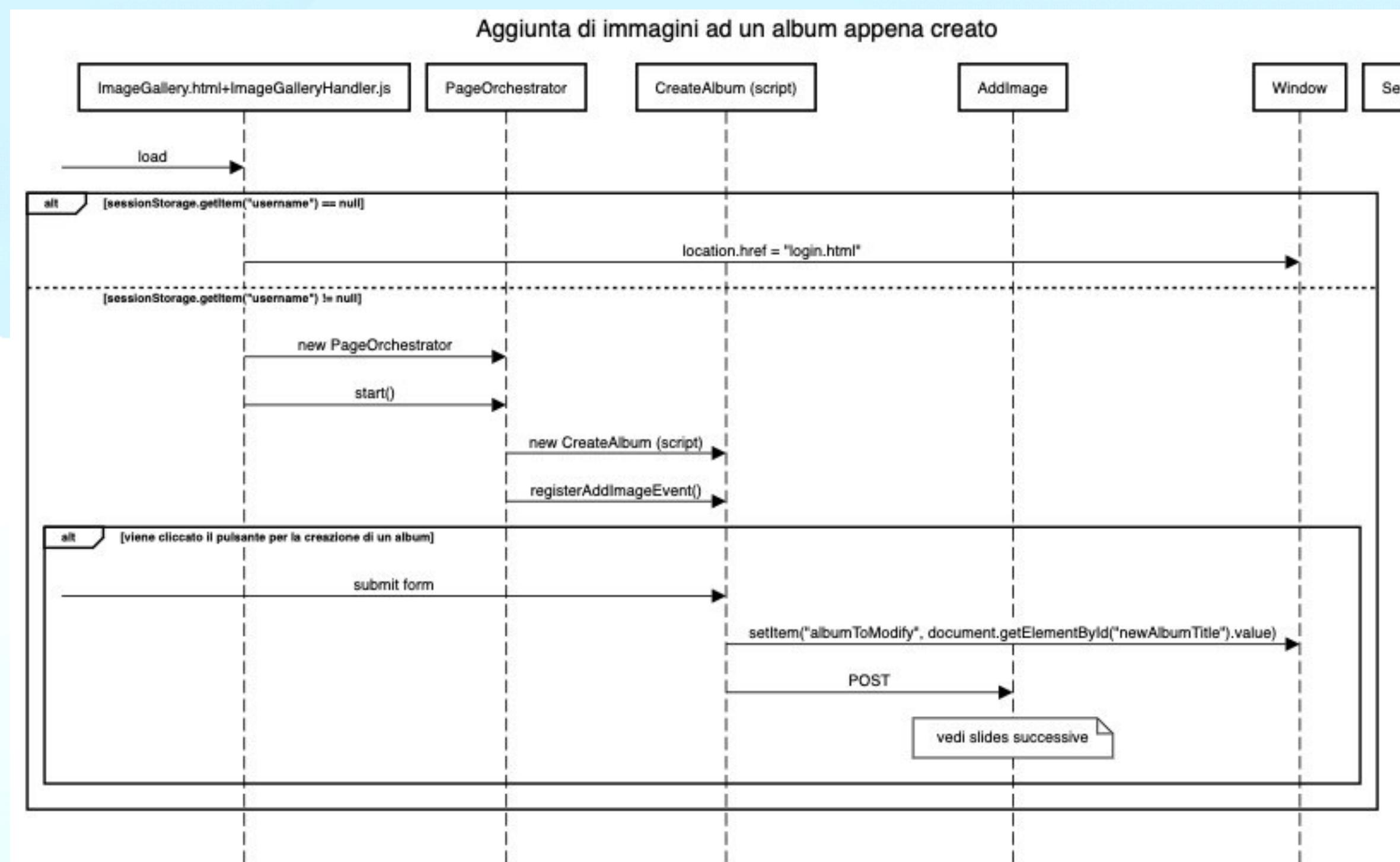
SEQUENCE DIAGRAMS

Creazione di un album



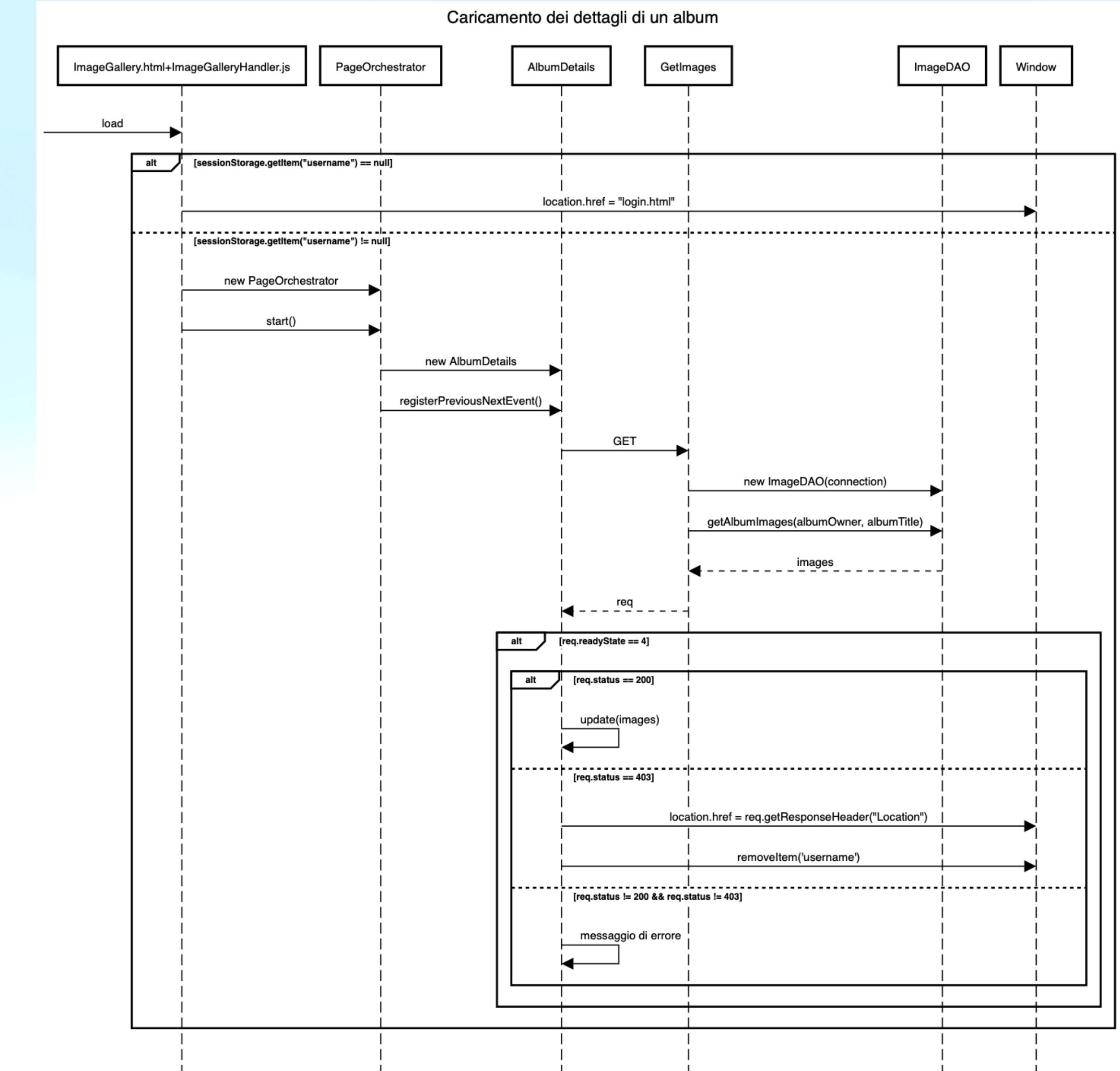
SEQUENCE DIAGRAMS

Aggiunta di un'immagine a un album (facoltativo)



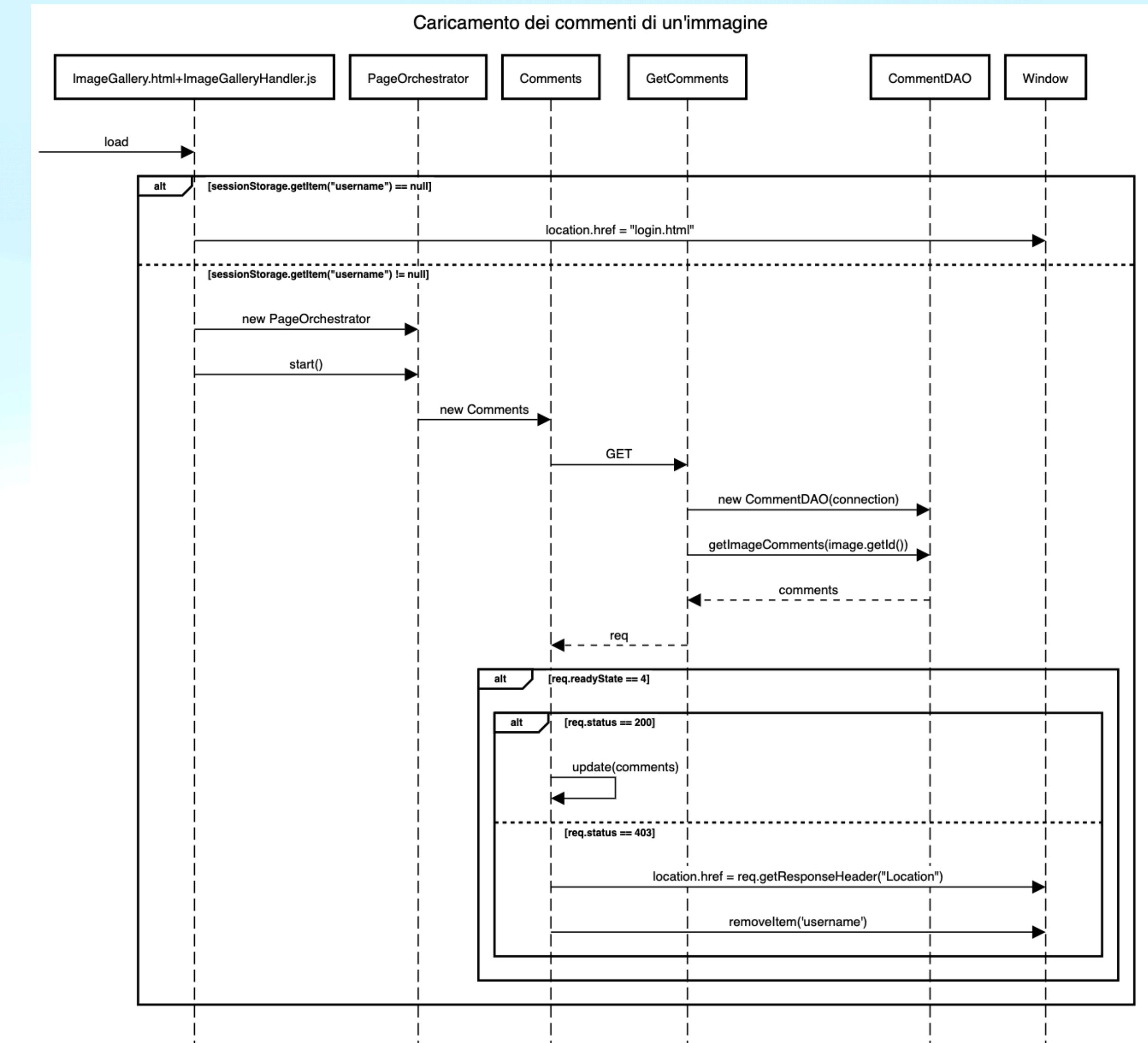
SEQUENCE DIAGRAMS

Visualizzazione dei dettagli di un album e delle sue immagini



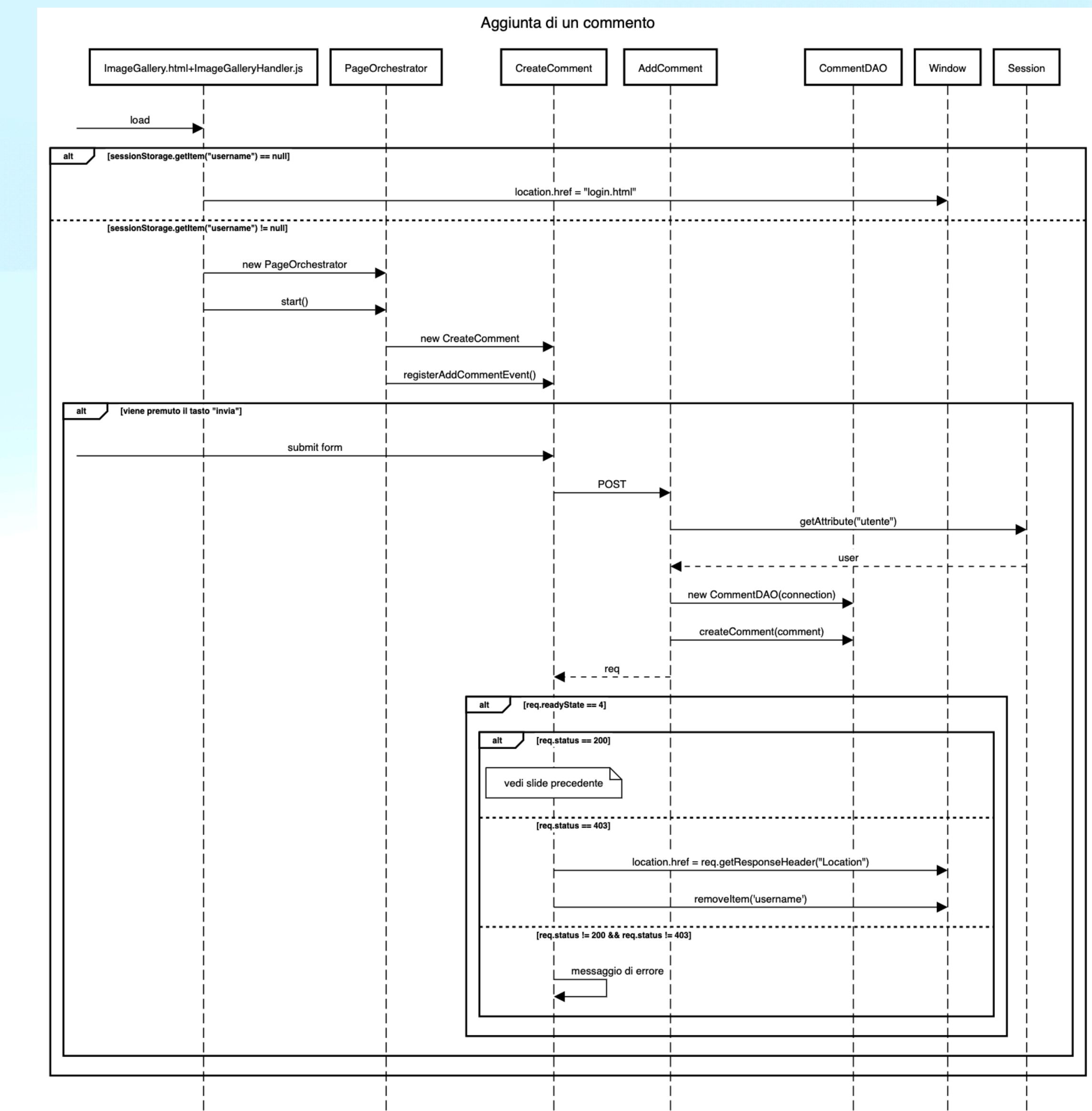
SEQUENCE DIAGRAMS

Visualizzazione dei commenti



SEQUENCE DIAGRAMS

Aggiunta di un commento



SEQUENCE DIAGRAMS

Logout

