Progetto di Tecnologie informatiche per il web 2023/2024

Prof. Fraternali Piero

Simone Rodari

Cod. persona: [redacted]

Matricola: [redacted]

POLITECNICO
MILANO 1863

Sommario

Specifica

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 di creazione, 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 dalla HOME PAGE e associare a questi le proprie immagini. Un album ha un titolo, il creatore e la data di creazione. La stessa immagine può appartenere a più di un album. 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. Se il numero di immagini non è un multiplo di 5 la tabella deve avere sempre 5 celle, lasciando quelle più a destra vuote. 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, una pagina IMAGE 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 e un bottone per cancellare l'immagine e tutti i commenti ad essa associati. Il bottone di cancellazione appare solo se l'immagine appartiene all'utente. L'invio del commento con un bottone INVIA ripresenta la pagina IMAGE PAGE, con tutti i dati aggiornati della stessa immagine.

La pagina IMAGE PAGE contiene collegamenti per tornare all'HOME PAGE e alla pagina ALBUM PAGE. La pagina ALBUM PAGE contiene un collegamento per tornare all'HOME PAGE. L'applicazione consente il logout dell'utente.

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 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.
- 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. L'applicazione carica le informazioni necessarie per la visualizzazione di tutte le immagini di un album e dei relativi commenti mediante un'unica chiamata.
- 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 delle immagini all'interno di un album con un criterio personalizzato diverso da quello di default (data decrescente). L'utente accede a un elenco dei titoli delle immagini ordinato secondo il criterio correntemente in uso: default o personalizzato. Trascina il titolo di un'immagine 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, per quell'utente, è usato al posto di quello di default.

Analisi della specifica

Analisi dei dati

Legenda:

- Entità
- Relazioni
- Attributi

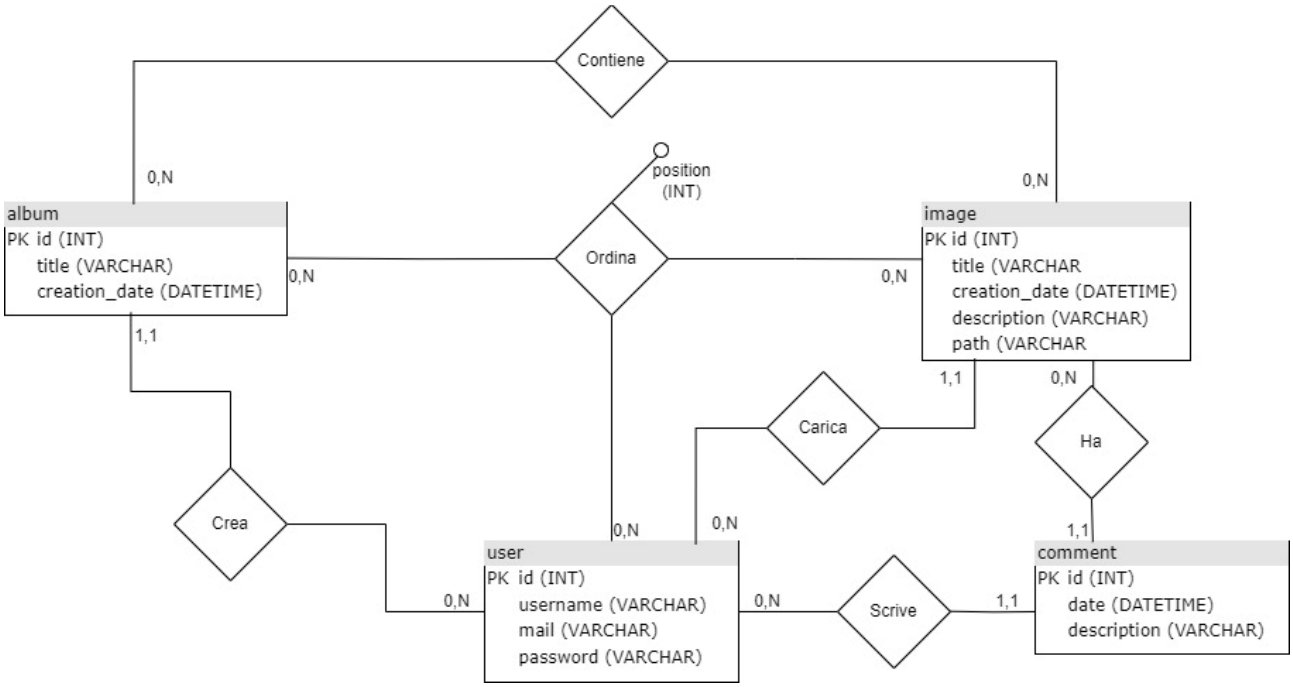
Un'applicazione web consente la gestione di una galleria d'immagini. L'applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password". La 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 di creazione, 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 dalla HOME PAGE e associare a questi le proprie immagini. Un album ha un titolo, il creatore e la data di creazione. La stessa immagine può appartenere a più di un album. 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.

Si deve consentire all'utente di riordinare l'elenco delle immagini all'interno di un album con un criterio personalizzato diverso da quello di default (data decrescente). L'utente accede a un elenco dei titoli delle immagini ordinato secondo il criterio correntemente in uso: default o personalizzato. Trascina il titolo di un'immagine 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, per quell'utente, è usato al posto di quello di default.

Database design



Codice SQL

```
CREATE TABLE `comment` (
 `id` int NOT NULL AUTO_INCREMENT,
 `id_image` int NOT NULL,
 `id_user` int NOT NULL,
 `date` datetime NOT NULL,
 `text` varchar(400) NOT NULL,
 PRIMARY KEY (`id`),
 KEY `id_image_idx` (`id_image`),
 KEY `id_user_idx` (`id_user`),
 CONSTRAINT `fk_image` FOREIGN KEY (`id_image`) REFERENCES `image` (`id`) ON DELETE
CASCADE,
 CONSTRAINT `fk_user` FOREIGN KEY (`id_user`) REFERENCES `user` (`id`)
)

CREATE TABLE `album` (
 `id` int NOT NULL AUTO_INCREMENT,
 `title` varchar(45) NOT NULL,
 `owner` int NOT NULL,
 `creation_date` datetime NOT NULL,
 PRIMARY KEY (`id`),
 KEY `fk_user_album_idx` (`owner`),
 CONSTRAINT `fk_user_album` FOREIGN KEY (`owner`) REFERENCES `user` (`id`)
);

CREATE TABLE `album_image` (
 `id_album` int NOT NULL,
 `id_image` int NOT NULL,
```

```
 PRIMARY KEY (`id_album`,`id_image`),
 KEY `fk_image_album_idx` (`id_image`),
 CONSTRAINT `fk_album_image` FOREIGN KEY (`id_album`) REFERENCES `album` (`id`) ON
DELETE CASCADE,
 CONSTRAINT `fk_image_album` FOREIGN KEY (`id_image`) REFERENCES `image` (`id`) ON
DELETE CASCADE
);

CREATE TABLE `image` (
 `id` int NOT NULL AUTO_INCREMENT,
 `id_user` int NOT NULL,
 `title` varchar(45) NOT NULL,
 `creation_date` datetime NOT NULL,
 `description` varchar(100) DEFAULT NULL,
 `path` varchar(100) NOT NULL,
 PRIMARY KEY (`id`),
 KEY `fk_user_idx` (`id_user`),
 CONSTRAINT `fk_user_image` FOREIGN KEY (`id_user`) REFERENCES `user` (`id`)
);

CREATE TABLE `user` (
 `id` int NOT NULL AUTO_INCREMENT,
 `username` varchar(45) NOT NULL,
 `mail` varchar(45) NOT NULL,
 `password` varchar(45) NOT NULL,
 PRIMARY KEY (`id`),
 UNIQUE KEY `username_UNIQUE` (`username`),
 UNIQUE KEY `mail_UNIQUE` (`mail`)
);

CREATE TABLE `user_image_order` (
 `id_user` int NOT NULL,
 `id_album` int NOT NULL,
 `id_image` int NOT NULL,
 `position` int NOT NULL,
 PRIMARY KEY (`id_user`,`id_album`,`id_image`),
 KEY `fk_order_album_idx` (`id_album`,`id_image`),
 CONSTRAINT `fk_order_album_image` FOREIGN KEY (`id_album`, `id_image`) REFERENCES
`album_image` (`id_album`, `id_image`) ON DELETE CASCADE ON UPDATE RESTRICT,
 CONSTRAINT `fk_order_user` FOREIGN KEY (`id_user`) REFERENCES `user` (`id`) ON DELETE
CASCADE ON UPDATE RESTRICT
);
```

Analisi funzionale (pure HTML)

Legenda:

- Pagine
- View components
- Azioni
- Eventi

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 di creazione, 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 dalla HOME PAGE e associare a questi le proprie immagini. Un album ha un titolo, il creatore e la data di creazione. La stessa immagine può appartenere a più di un album. 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. Se il numero di immagini non è un multiplo di 5 la tabella deve avere sempre 5 celle, lasciando quelle più a destra vuote. 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, una pagina IMAGE 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 e un bottone per cancellare l'immagine e tutti i commenti ad essa associati. Il bottone di cancellazione appare solo se l'immagine appartiene all'utente. L'invio del commento con un bottone INVIA ripresenta la pagina IMAGE PAGE, con tutti i dati aggiornati della stessa immagine.

La pagina IMAGE PAGE contiene collegamenti per tornare all'HOME PAGE e alla pagina ALBUM PAGE. La pagina ALBUM PAGE contiene un collegamento per tornare all'HOME PAGE. L'applicazione consente il logout dell'utente.

Completamento specifiche:

- Quando si torna indietro da IMAGE PAGE ad ALBUM PAGE tramite l'apposito pulsante, si ritorna alle ultime 5 immagini visualizzate precedentemente
- È possibile il caricamento delle immagini dell'utente da fileSystem mediante un apposito form
- È possibile creare un album vuoto
- I commenti hanno anche una data di creazione

Ricapitolazione e completamento dell'analisi funzionale

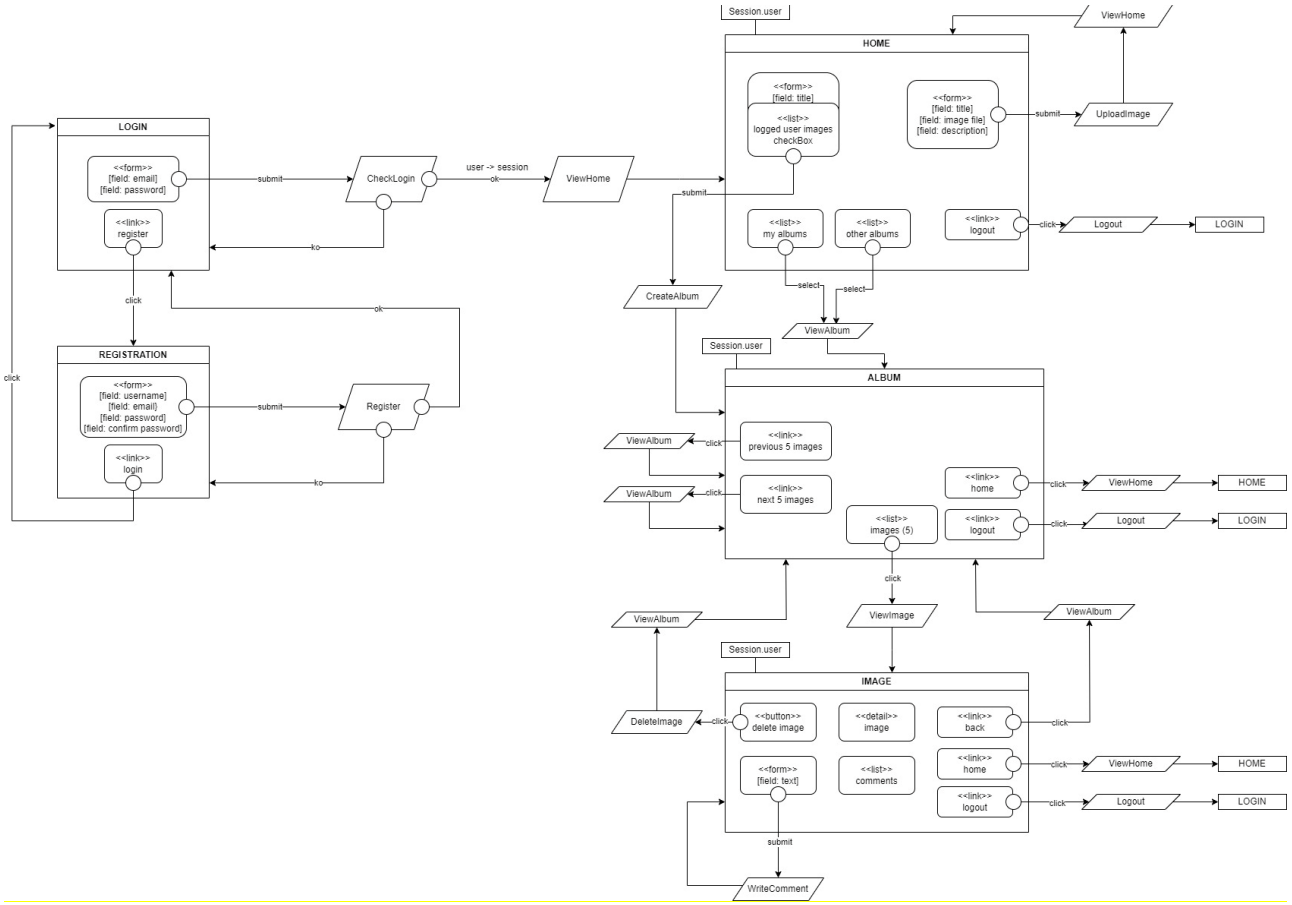
Pagine e view components:

- Login
  - Form di login
  - Pulsante di login
  - Link per Registration
- Registration
  - Form di registrazione
  - Pulsante di registrazione
  - Link per login
- Home
  - Form per l'inserimento del titolo dell'album
  - Lista delle immagini dell'utente con relative checkbox per contrassegnare quelle da aggiungere
  - Pulsante per creare l'album
  - Form per caricare sul server un'immagine da file system e inserire i suoi dati
  - Pulsante per caricare l'immagine sul server
  - Lista degli album dell'utente corrente
  - Lista degli album degli altri utenti
  - Link per il logout dell'utente
- Album
  - Lista delle immagini (5 per volta)
  - Pulsante per visualizzare le 5 immagini precedenti (se presenti)
  - Pulsante per visualizzare le 5 immagini successive (se presenti)
  - Link per tornare a Home
  - Link per il logout dell'utente
- Image
  - Dati dell'immagine
  - Immagine a grandezza naturale
  - Lista dei commenti associati all'immagine
  - Form per l'inserimento di un nuovo commento
  - Pulsante per eliminare l'immagine e tutti i commenti ad essa associata (solo se l'utente corrente è colui che ha caricato l'immagine)
  - Link per tornare ad Album
  - Link per tornare a Home
  - Link per il logout dell'utente

Eventi e azioni:

- Click del pulsante di login
  - Verifica credenziali
- Click sul link di registrazione
  - Reindirizzamento alla pagina di registrazione
- Click del pulsante di registrazione
  - Creazione dell'utente
  - Reindirizzamento alla pagina di login
- Click sul link di login
  - Reindirizzamento alla pagina di login
- Click su un album
  - Reindirizzamento alla pagina ALBUM
- Click sul pulsante "Create album"
  - Creazione dell'album avente titolo specificato nell'opportuno form
  - Associazione delle immagini selezionate tramite checkbox all'album
- Click sul pulsante "Upload image"
  - Caricamento di un'immagine da file system alla cartella di upload
  - Inserimento dei dati dell'immagine nella base dati
- Click su una miniatura
  - Reindirizzamento alla pagina Image
- Click sul bottone "Prev"
  - Ricaricamento pagina IMAGE
  - Visualizzazione delle 5 immagini precedenti
- Click sul bottone "Next"
  - Ricaricamento pagina IMAGE
  - Visualizzazione delle 5 immagini successive
- Click sull'icona "Trash"
  - Eliminazione dell'immagine visualizzata e dei commenti ad essa associati
  - Reindirizzamento alla pagina Album
- Click sul pulsante "Send comment"
  - Inserimento del commento associato all'immagine e all'utente corrente
  - Ricaricamento pagina
- Click sull'icona "Back"
  - Reindirizzamento alla pagina Album
- Click sull'icona "Home"
  - Reindirizzamento alla pagina Home
- Click sul link di logout
  - Logout dell'utente
  - Reindirizzamento alla pagina di login

Design dell’applicazione web



Componenti

Model Objects (beans):

- Album
- Comment
- Image
- User

Data Access Objects (DAO):

- AlbumDAO:
  - getUserAlbums(int owner)
  - getOtherUsersAlbums(int user)
  - createAlbum(String title, int owner, Timestamp creation_date)
  - isExistingAlbum(albumId)
- AlbumImageDAO:
  - saveAlbumImages(int albumId, List<Integer> imageIds)
- CommentDAO:
  - getImageComments(int imageId)
  - writeComment(int imageId, int userId, Timestamp date, String text)
- ImageDAO
  - getImageById(int imageId)
  - getAlbumImages(int albumId, int limit, int offset)
  - countImagesByAlbumId(int albumId)
  - getUserImages(int userId)
  - deleteImage(int imageId)
  - isMyImage(int imageId, int userId)
  - areImagesOwnedByUser(List<Integer> imageIds, int userId)
  - uploadImage(int userId, String title, Timestamp date, String description, String path)
  - isNewImage(string path)
- UserDAO
  - getUserById(int id)
  - checkCredentials(String mail, String password)
  - registerUser(String username, String mail, String password)
  - isNewUsername(String username)
  - isNewEmail(String mail)

Controllori (servlets):

- CheckLogin
- CreateAlbum
- DeleteImage
- ImageGetter
- Logout
- Register
- ViewAlbum
- ViewHome
- ViewImage
- UploadImage

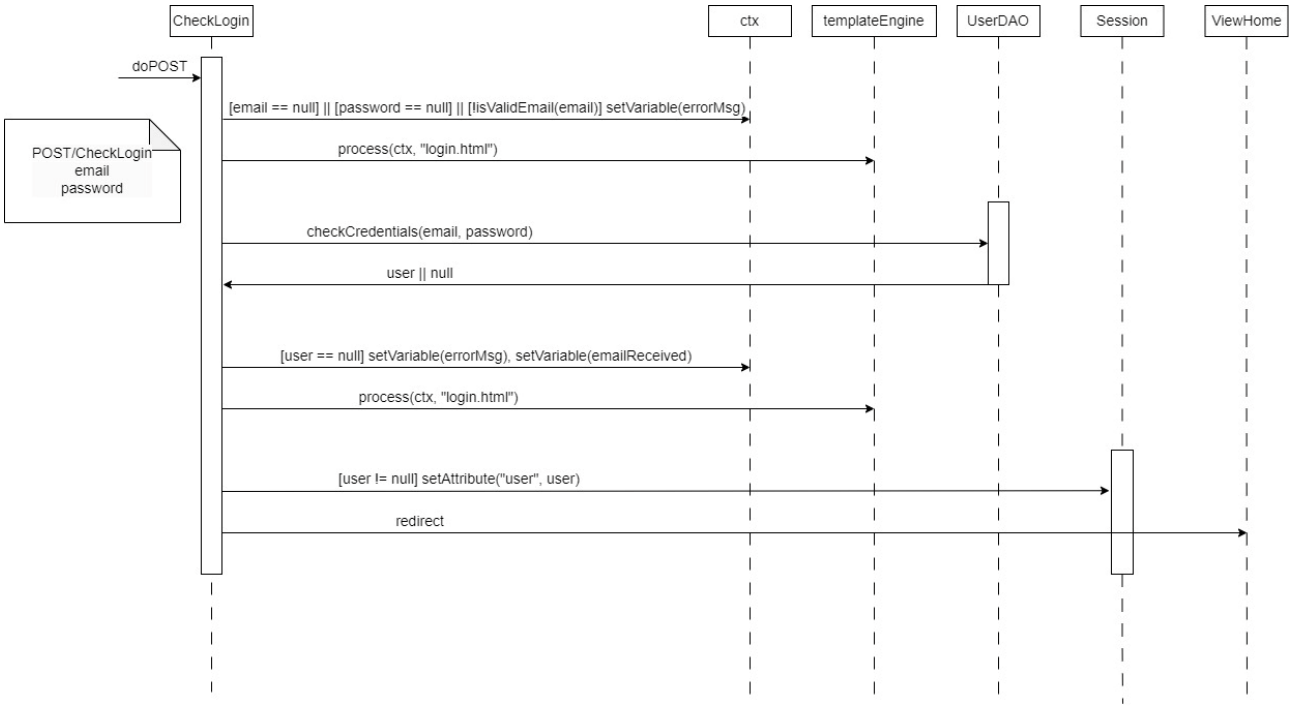
- WriteComment

Pagine (templates):

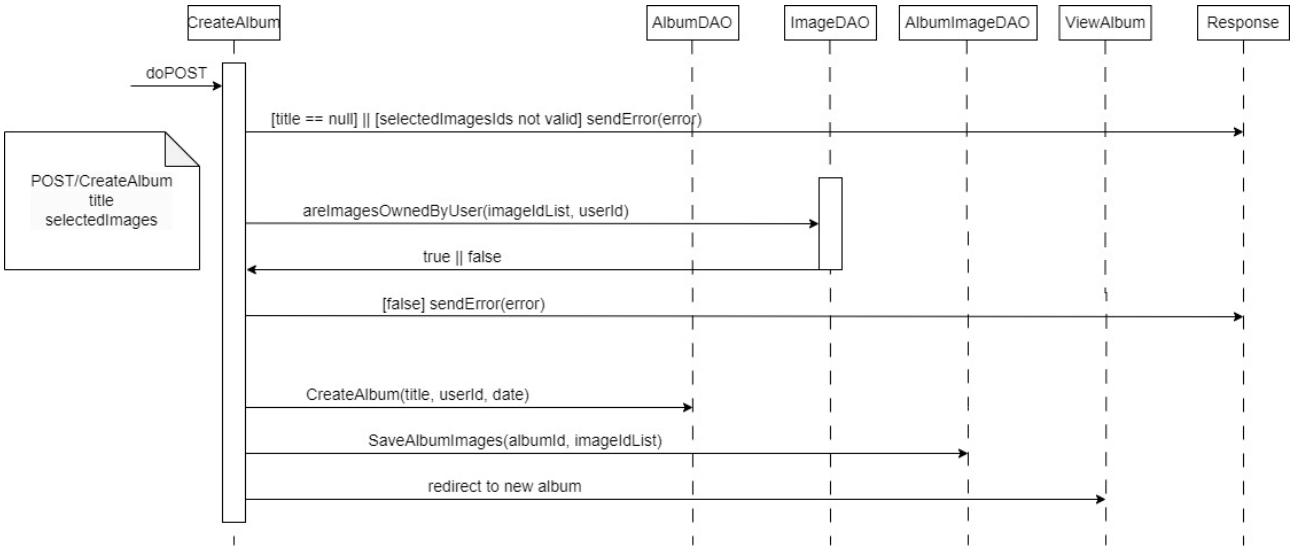
- LOGIN
- REGISTRATION
- ALBUM
- HOME
- IMAGE

Sequence diagrams

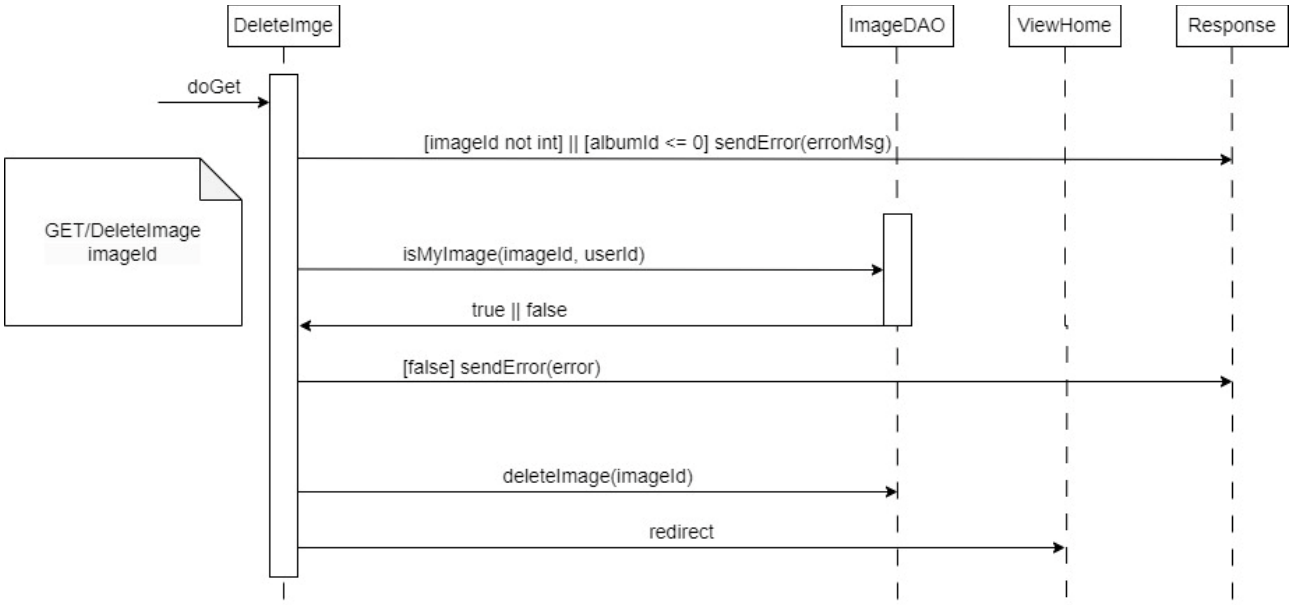
Login



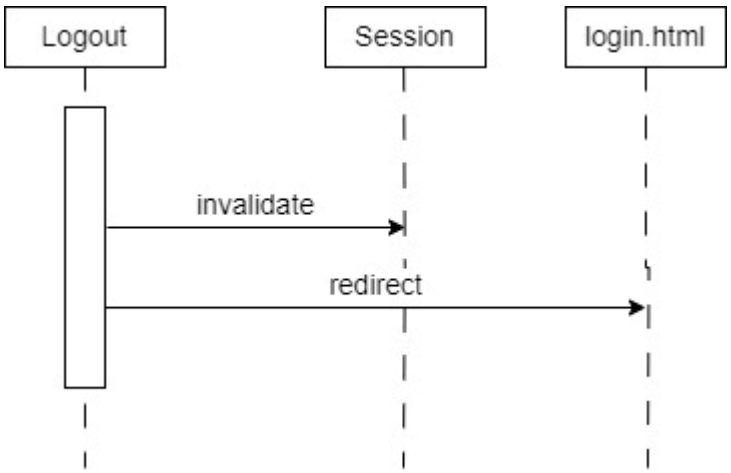
Create album



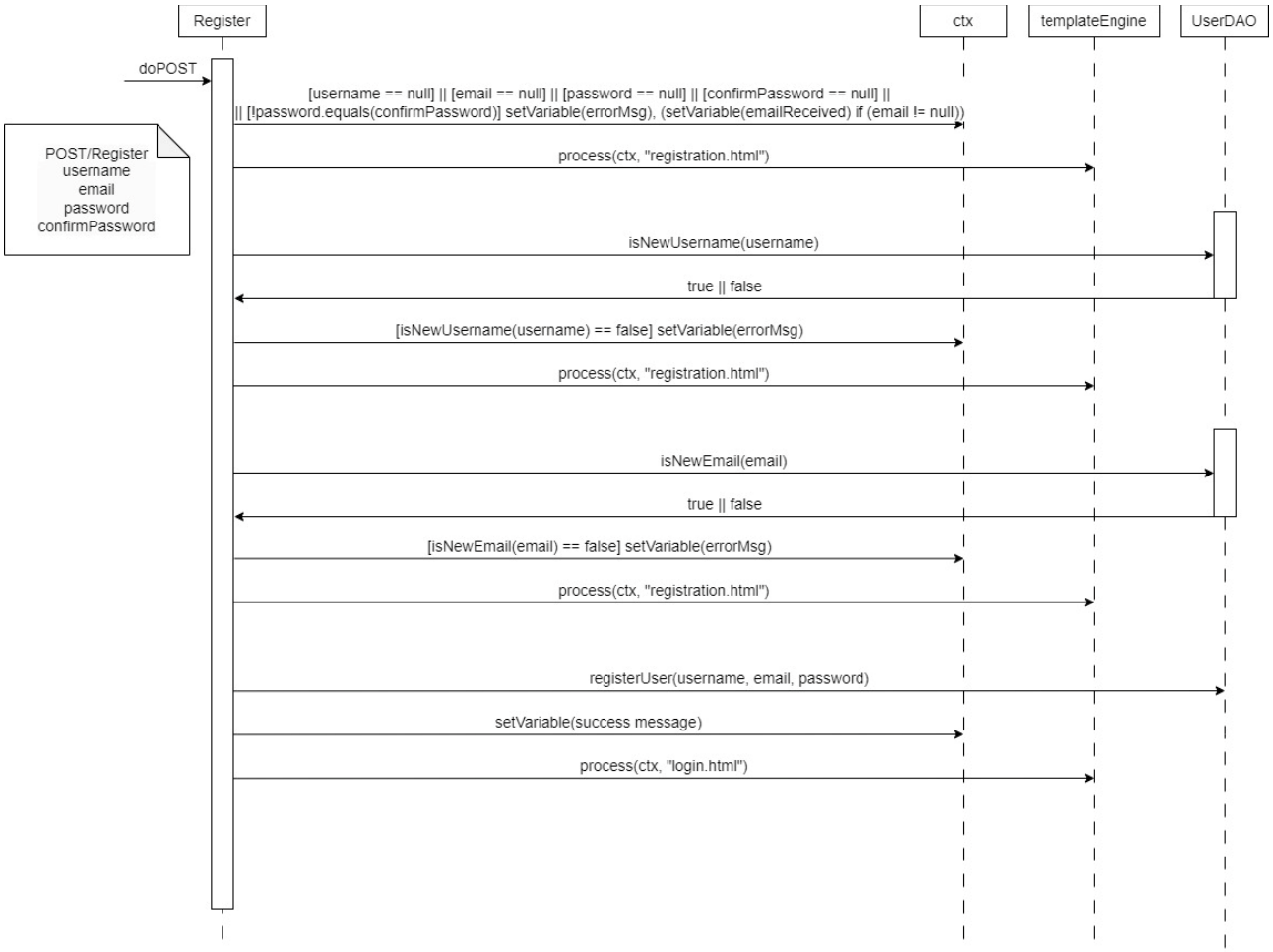
Delete image



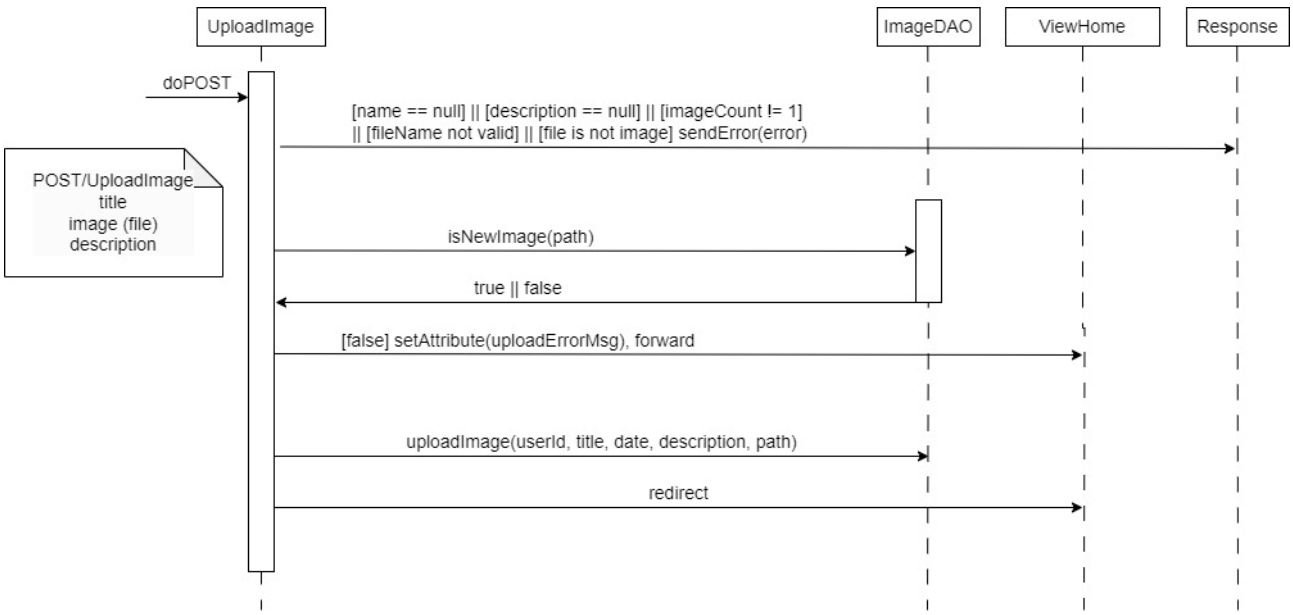
Logout



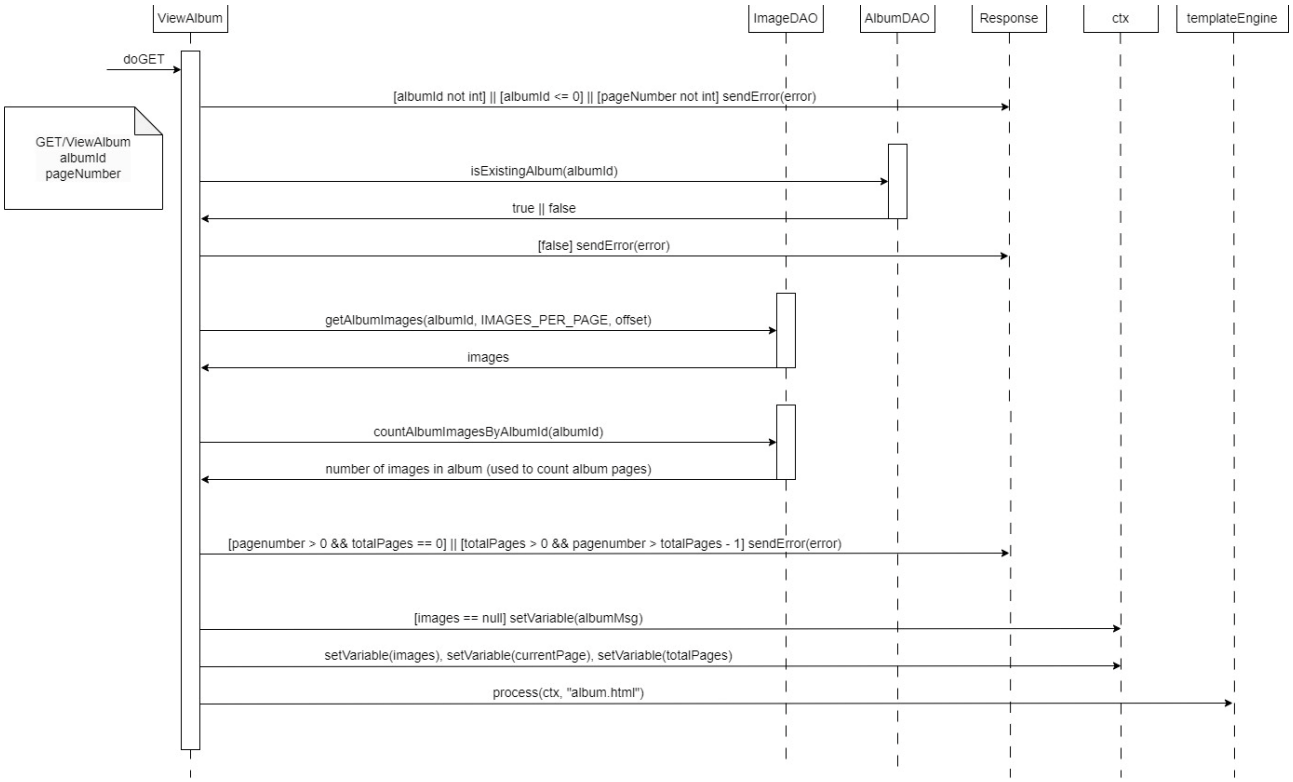
Register



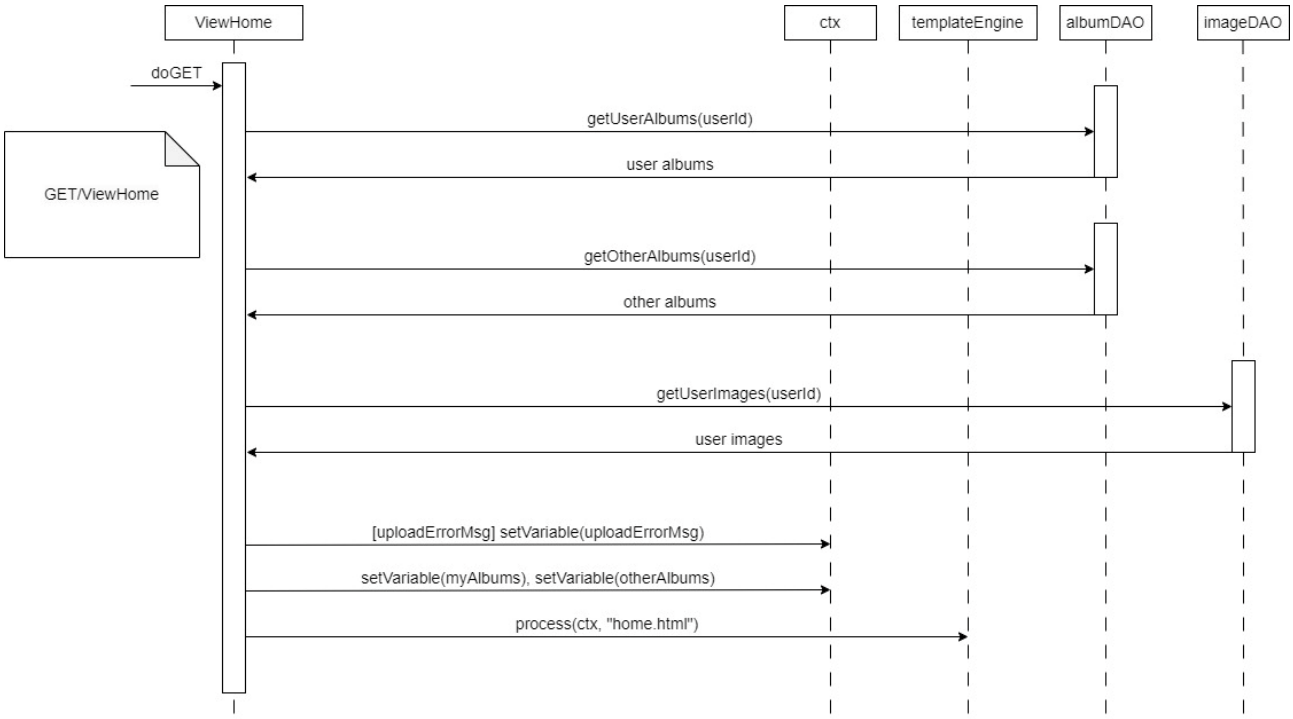
Upload image



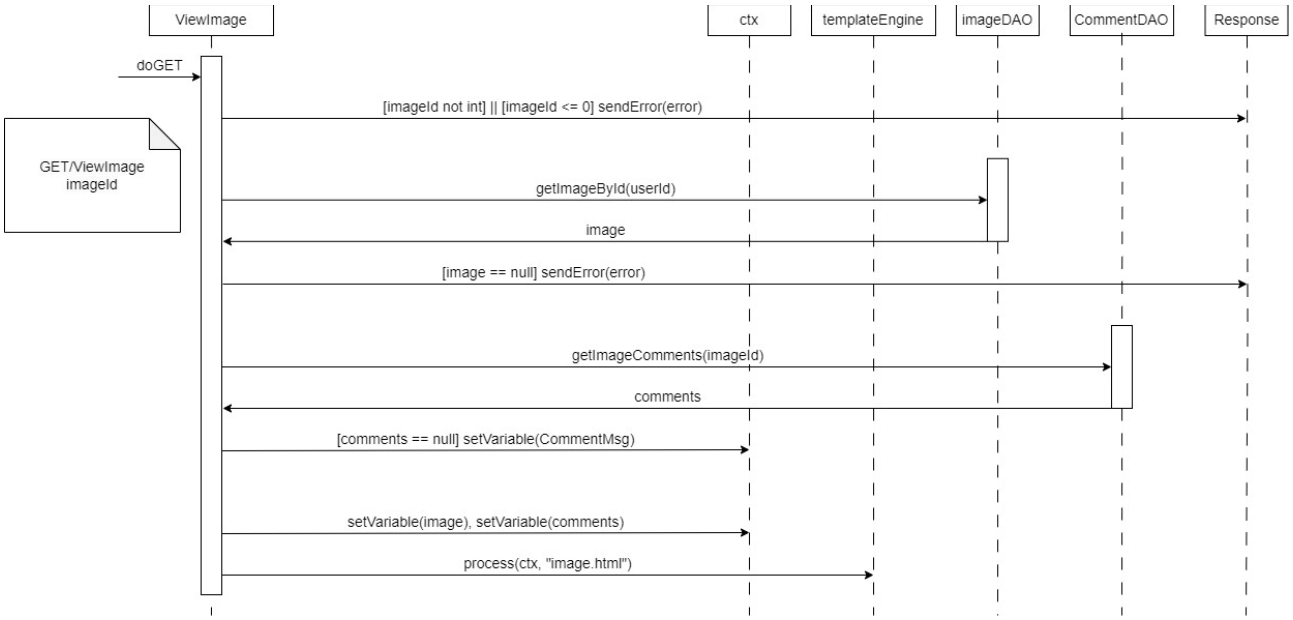
View album



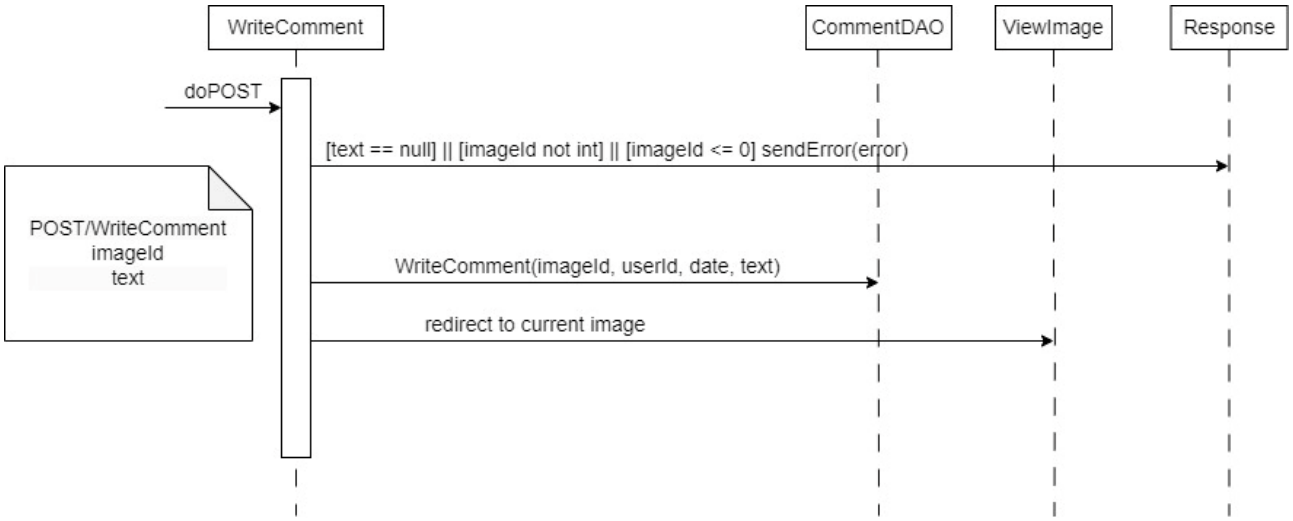
View home



View image



Write comment



NOTE:

- **ImageGetter** semplicemente recupera le immagini dal file system
- Per ogni variabile String viene verificato che essa non sia vuota o contenga solo spazi bianchi
- Un filtro controlla che ad ogni chiamata di una servlet venga controllato se l'utente è loggato o no, in caso negativo viene reindirizzato alla schermata di login

Analisi funzionale (JavaScript)

Legenda:

- Pagine
- View components
- Azioni
- Eventi

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 di creazione, 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 dalla HOME PAGE e associare a questi le proprie immagini. Un album ha un titolo, il creatore e la data di creazione. La stessa immagine può appartenere a più di un album. 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. Se il numero di immagini non è un multiplo di 5 la tabella deve avere sempre 5 celle, lasciando quelle più a destra vuote. 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, una pagina IMAGE 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 e un bottone per cancellare l'immagine e tutti i commenti ad essa associati. Il bottone di cancellazione appare solo se l'immagine appartiene all'utente. L'invio del commento con un bottone INVIA ripresenta la pagina IMAGE PAGE, con tutti i dati aggiornati della stessa immagine.

La pagina IMAGE PAGE contiene collegamenti per tornare all'HOME PAGE e alla pagina ALBUM PAGE. La pagina ALBUM PAGE contiene un collegamento per tornare all'HOME PAGE. L'applicazione consente il logout dell'utente.

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 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.
- 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. L'applicazione carica le informazioni necessarie per la visualizzazione di tutte le immagini di un album e dei relativi commenti mediante un'unica chiamata.
- 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 delle immagini all'interno di un album con un criterio personalizzato diverso da quello di default (data decrescente). L'utente accede a un elenco dei titoli delle immagini ordinato secondo il criterio correntemente in uso: default o personalizzato. Trascina il titolo di un'immagine 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, per quell'utente, è usato al posto di quello di default.

Completamento specifiche:

- È possibile il caricamento delle immagini dell'utente da fileSystem mediante un apposito form
- È possibile creare un album vuoto
- I commenti hanno anche una data di creazione

Ricapitolazione e completamento dell'analisi funzionale

Pagine e view components:

- Login
  - Form di login
  - Pulsante di login
  - Link per mostrare il form di registrazione
  - Form di registrazione
  - Pulsante di registrazione
  - Link per mostrare il form di login
- Home
  - Form per l'inserimento del titolo dell'album
  - Lista delle immagini dell'utente con relative checkbox per contrassegnare quelle da aggiungere

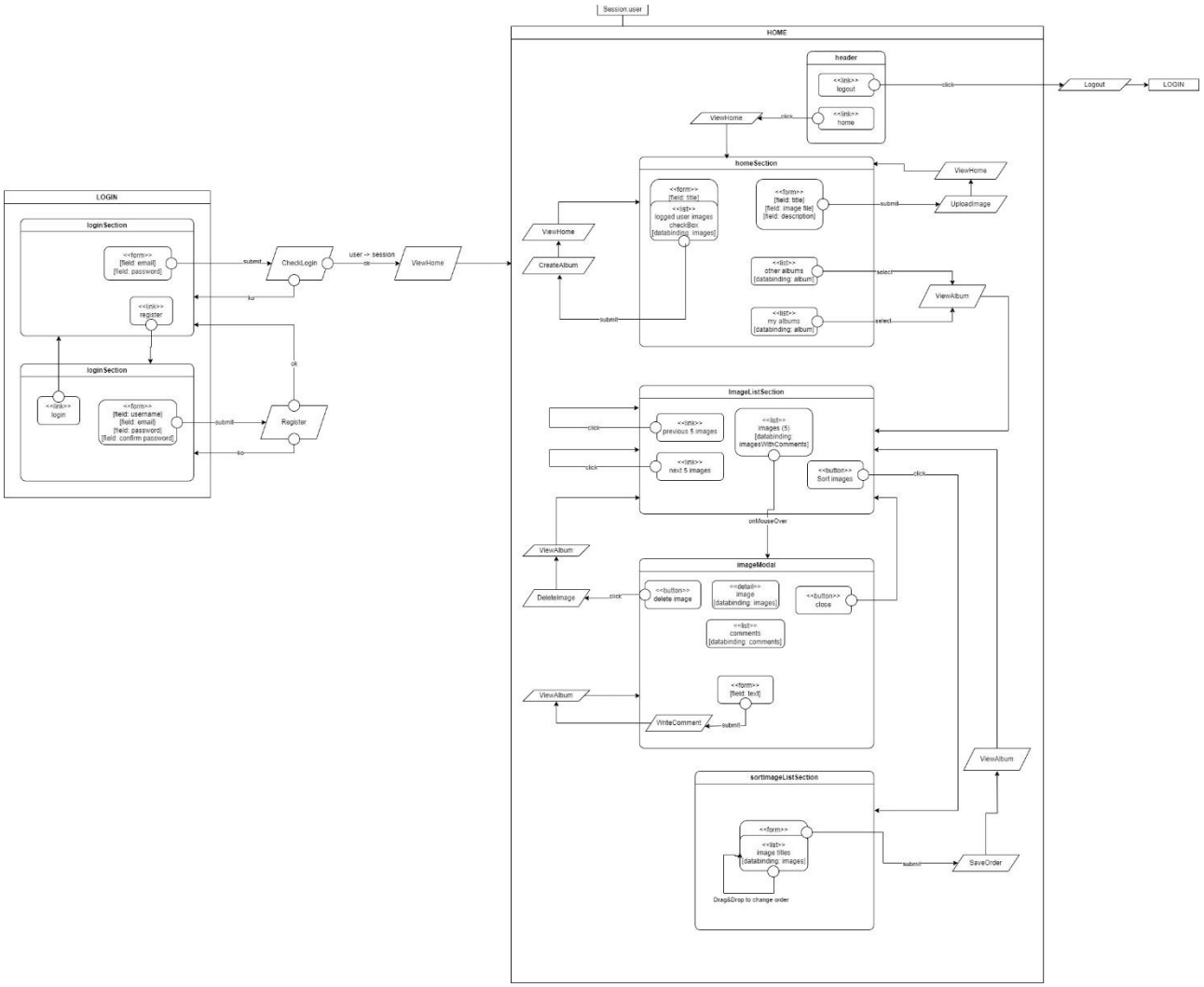
    - Pulsante per creare l'album
    - Form per caricare sul server un'immagine da file system e inserire i suoi dati
    - Pulsante per caricare l'immagine sul server
    - Lista degli album dell'utente corrente
    - Lista degli album degli altri utenti
    - Lista delle immagini (5 per volta)
    - Pulsante per visualizzare le 5 immagini precedenti (se presenti)
    - Pulsante per visualizzare le 5 immagini successive (se presenti)
    - Finestra modale per visualizzazione delle immagini e dei loro dati
    - Dati dell'immagine
    - Immagine a grandezza naturale
    - Lista dei commenti associati all'immagine
    - Form per l'inserimento di un nuovo commento
    - Pulsante per eliminare l'immagine e tutti i commenti ad essa associata (solo se l'utente corrente è colui che ha caricato l'immagine)
    - Link per tornare a Home
    - Link per il logout dell'utente
    - Pulsante per visualizzare l'elenco dei titoli delle immagini nell'album
    - Lista dei titoli delle immagini nell'album
    - Pulsante per salvare l'ordinamento delle immagini

Eventi e azioni:

- Click del pulsante di login
    - Verifica credenziali
- Click sul link di registrazione
    - Mostra form di registrazione
- Click del pulsante di registrazione
    - Creazione dell'utente
    - Mostra form di login
- Click sul link di login
    - Mostra form di login
- Click su un album
    - Visualizzazione dell'album (caricamento dei dati delle immagini e i relativi commenti in un'unica richiesta)
- Click sul pulsante "Create album"
    - Creazione dell'album avente titolo specificato nell'opportuno form
    - Associazione delle immagini selezionate tramite checkbox all'album
- Click sul pulsante "Upload image"
    - Caricamento di un'immagine da file system alla cartella di upload
    - Inserimento dei dati dell'immagine nella base dati
- Mouse su una miniatura
    - Mostra pagina modale con le informazioni dell'immagine
- Click sul bottone "Prev"
    - Visualizzazione delle 5 immagini precedenti
- Click sul bottone "Next"
    - Visualizzazione delle 5 immagini successive
- Click pulsante "Delete"
    - Eliminazione dell'immagine visualizzata e dei commenti ad essa associati

    - Mostra album corrente
- Click sul pulsante “Send comment”
    - Inserimento del commento associato all’immagine e all’utente corrente
    - Aggiornamento pagina
- Click sull’icona “Home”
    - Caricamento home
- Click sul link di logout
    - Logout dell’utente
    - Reindirizzamento alla pagina di login
- Click sul pulsante “Order”
    - Visualizzazione dei titoli delle immagini dell’album
- Trascinamento titolo immagine
    - Collocazione nella posizione in cui il titolo viene rilasciato
- Click sul pulsante “Save order”
    - Salvataggio dell’ordine delle immagini nell’album per quell’utente

Design dell’applicazione web



Componenti

Model Objects (beans):

- Album
- Comment
- ImageWithComments
- User

Data Access Objects (DAO):

- AlbumDAO:
  - getUserAlbums(int owner)
  - getOtherUsersAlbums(int user)
  - createAlbum(String title, int owner, Timestamp creation_date)
  - isExistingAlbum(albumId)
- AlbumImageDAO:
  - saveAlbumImages(int albumId, List<Integer> imageIds)
  - getImageIdsByAlbum(int albumId)
- CommentDAO:
  - getImageComments(int imageId)
  - writeComment(int imageId, int userId, Timestamp date, String text)
- ImageDAO
  - getImageById(int imageId)
  - getAlbumImages(int albumId)
  - getAlbumImagesOrdered(int albumId, userId)
  - countImagesByAlbumId(int albumId)
  - getUserImages(int userId)
  - deleteImage(int imageId)
  - isMyImage(int imageId, int userId)
  - areImagesOwnedByUser(List<Integer> imageIds, int userId)
  - uploadImage(int userId, String title, Timestamp date, String description, String path)
  - isNewImage(string path)
- UserDAO
  - getUserById(int id)
  - checkCredentials(String mail, String password)
  - registerUser(String username, String mail, String password)
  - isNewUsername(String username)
  - isNewEmail(String mail)
- UserImageOrderDAO
  - isExistingOrder(int userId, int albumId)
  - updateImageOrder(int userId, intalbumId, List<Integer> imageIds, List<Integer> positions)
  - insertimageOrder(int userId, intalbumId, List<Integer> imageIds, List<Integer> positions)

Controllori (servlets):

- CheckLogin
- CreateAlbum

- DeleteImage
- ImageGetter
- Logout
- Register
- SaveOrder
- ViewAlbum
- ViewHome
- UploadImage
- WriteComment

Pagine (templates):

- LOGIN
- HOME

Eventi e azioni

N	Client side		Server side	
	Evento	Azione	Evento	Azione
1	Login -> login form -> submit	Login	POST (mail, password)	Controllo credenziali
2	Login -> link Registration	Visualizza form registrazione	-	-
3	Login -> registration form -> submit	Registrazione	POST (username, mail, password, confirmPassword)	Registrazione utente
4	Login -> link Login	Visualizza form login	-	-
5	Home -> load	Visualizzazione immagini utente. Visualizzazione album dell'utente e degli altri	GET()	Estrazione delle immagini dell'utente e degli album dell'utente e degli altri
6	Home -> pulsante Home	Visualizzazione le immagini dell'utente. Visualizzazione album dell'utente e degli altri	GET()	Estrazione delle immagini dell'utente e degli album dell'utente e degli altri
7	Home -> lista album -> click	Visualizzazione immagini album secondo l'ordinamento corrente	GET(albumId)	Estrazione delle immagini contenute nell'album e tutti i commenti relativi ad esse.
8	Home -> pulsante Create album	Creazione di un album con il titolo inserito e le immagini selezionate. Evento 6	POST(title, selectedImages)	Creazione di un album con il titolo inserito e le immagini selezionate

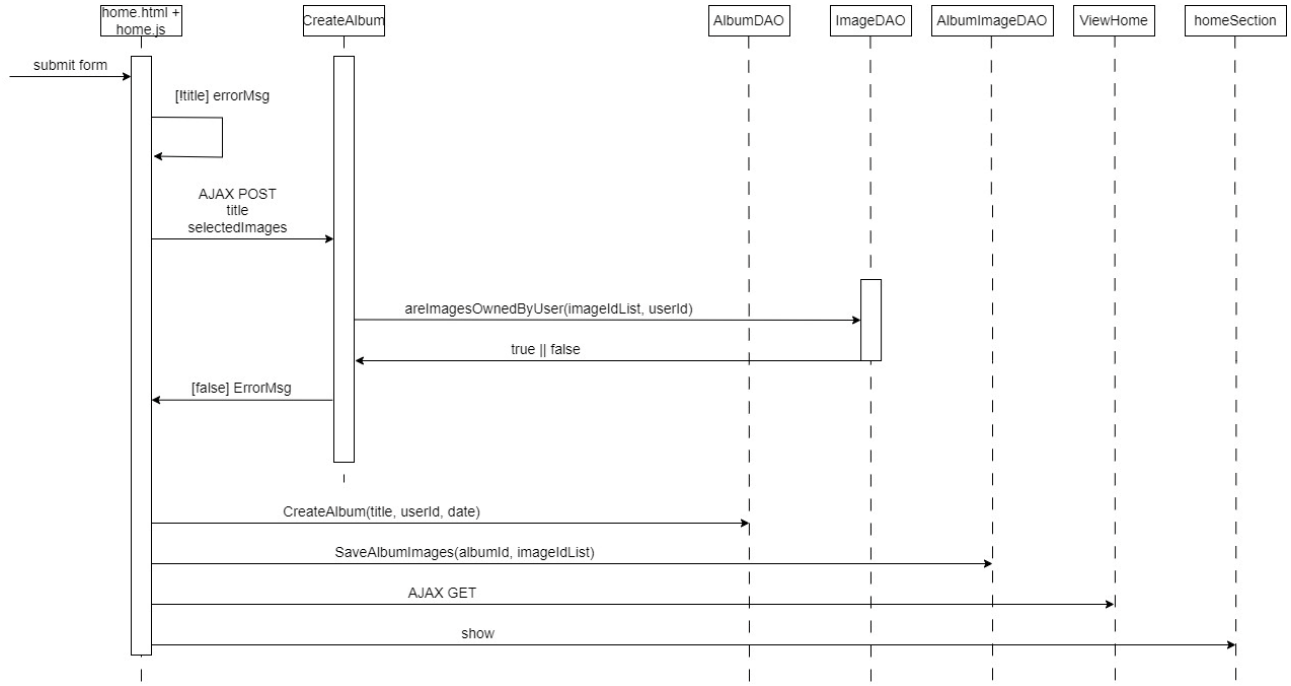
9	Home -> pulsante Upload image	Caricamento sul server di un'immagine da file system con titolo e descrizione inseriti. Evento 6	POST(title, image, description)	Caricamento sul server di un'immagine da file system con titolo e descrizione inseriti
10	Home -> lista immagini -> mouseOver	Visualizzazione finestra modale con dati dell immagine, l'immagine a grandezza naturale e un form per inserire i commenti	-	-
11	Home -> pulsante Prev	Caricamento delle 5 immagini precedenti	-	-
12	Home -> pulsante Next	Caricamento delle 5 immagini successive	-	-
13	Home -> ModalWindow -> pulsante Delete	Eliminazione immagine. Evento 7.	GET(currentImageId)	Eliminazione immagine. Aggiornamento immagini in album.
14	Home -> ModalWindow -> pulsante Send comment	Inserimento commento dell'immagine. Aggiornamento dati	POST(imageId, text)	Inserimento commento
15	Home -> pulsante Logout	Logout	GET()	Logout
16	Home -> pulsante Order	Visualizzazione titoli immagini nell'album	-	-
17	Home -> trascinamento elemento	Modifica ordine	-	-
18	Home -> pulsante Save order	Salva ordine Evento 7	POST(albumId, order{imageId,position}	Inserimento/aggiornamento dati ordine nel database

Controllori ed event handlers

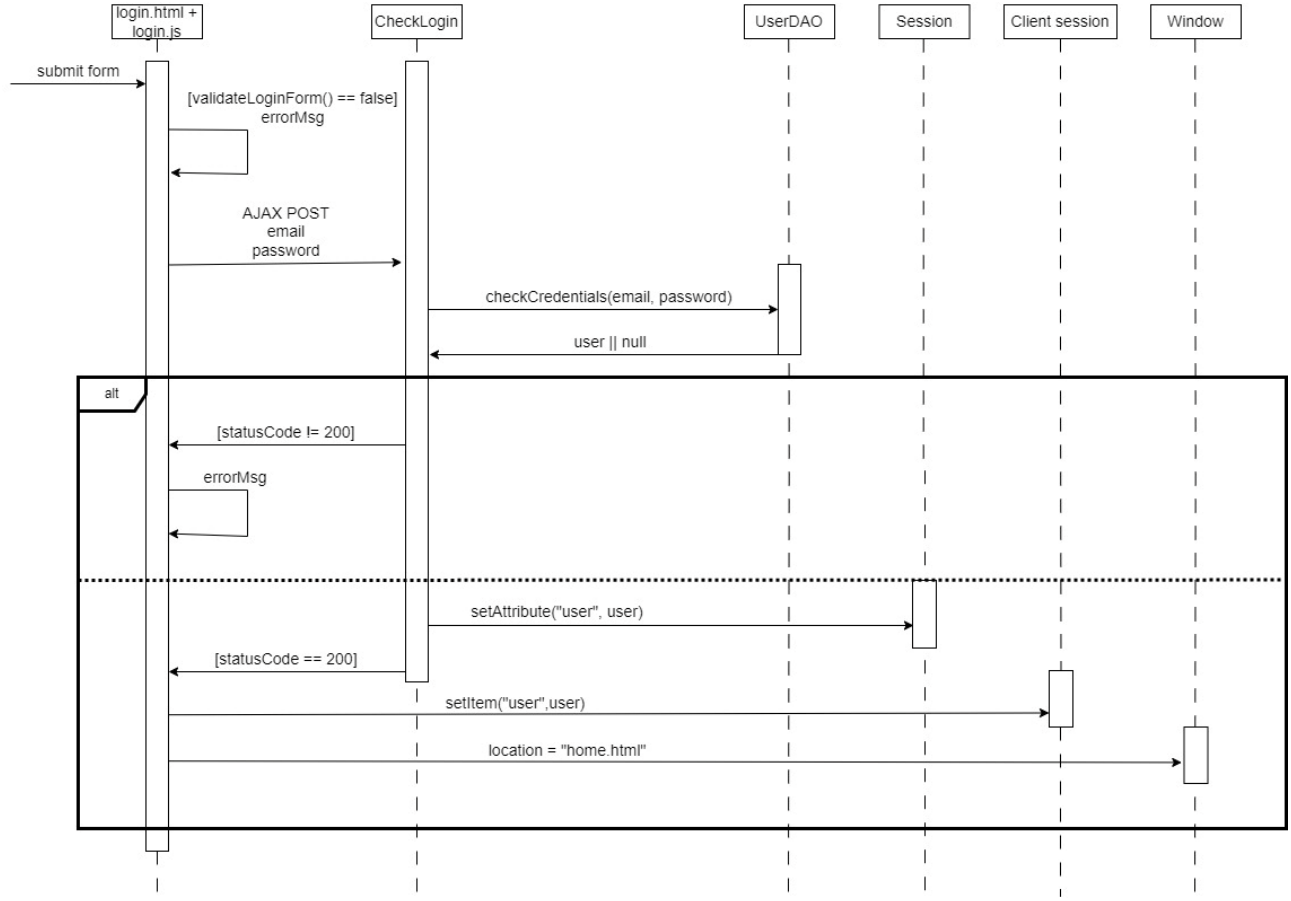
N	Client side		Server side	
	Evento	Controllore(funzione)	Evento	Controllore(servlet)
1	Login -> login form -> submit	MakeCall()	POST (mail, password)	CheckLogin
2	Login -> link Registration	showRegistrationForm()	-	-
3	Login -> registration form -> submit	MakeCall()	POST (username, mail, password, confirmPassword)	Register
4	Login -> link Login	showLoginForm()	-	-
5	Home -> load	showHomeSection() (includes makeCall())	GET()	ViewHome
6	Home -> pulsante Home	showHomeSection() (includes makeCall())	GET()	ViewHome
7	Home -> lista album -> click	showImageListSection(albumId,0) (includes makeCall())	GET(albumId)	ViewAlbum
8	Home -> pulsante Create album	MakeCall() Evento 6	POST(title, selectedImages)	CreateAlbum
9	Home -> pulsante Upload image	MakeCall() Evento 6	POST(title, image, description)	UploadImage
10	Home -> lista immagini -> mouseOver	ShowImageDetailInModal	-	-
11	Home -> pulsante Prev	currentPage— displayImages()	-	-
12	Home -> pulsante Next	currentPage++ displayImages()	-	-
13	Home -> ModalWindow -> pulsante Delete	MakeCall() Evento 7	GET(currentImageId)	DeleteImage
14	Home -> ModalWindow -> pulsante Send comment	MakeCall() loadImagesForAlbum(currentAlbumId) (includes makeCall())	POST(imageId, text)	WriteComment
15	Home -> pulsante Logout	MakeCall()	GET()	Logout
16	Home -> pulsante Order	showSortImageListSection()	-	-
17	Home -> trascinamento elemento	HandleDragStart(e) handleDragOver(e) handleDrop(e) handleDragEnd(e)	-	-
18	Home -> pulsante Save order	MakeJsonCall() Evento 7	POST(albumId, order{imageId,position}	SaveOrder

Sequence diagrams

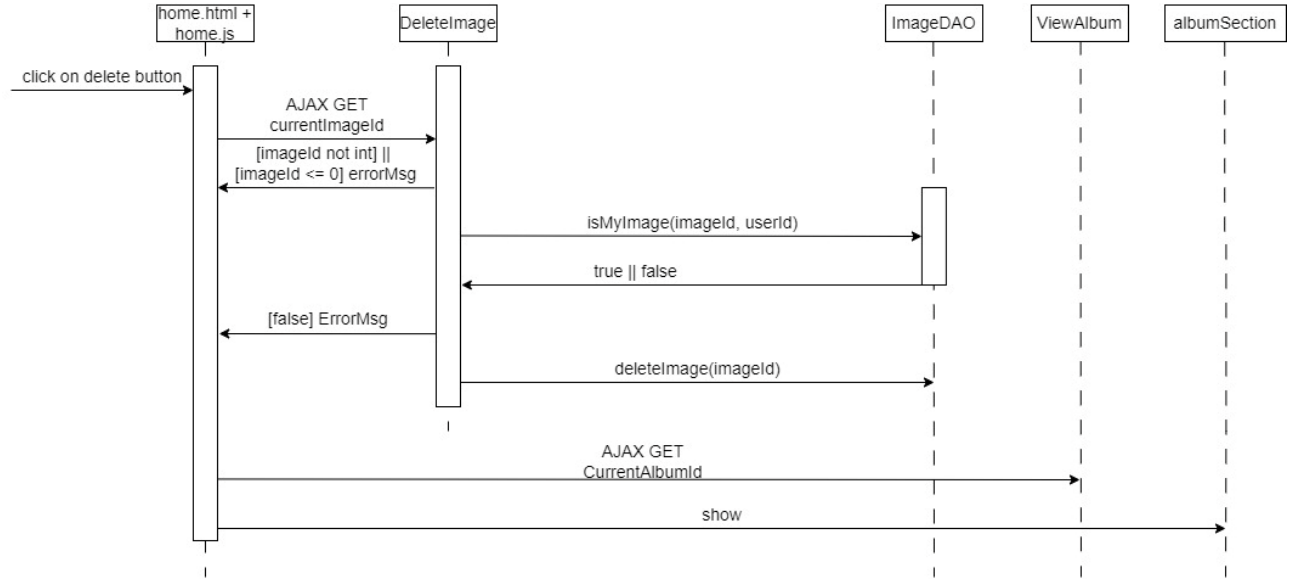
CheckLogin



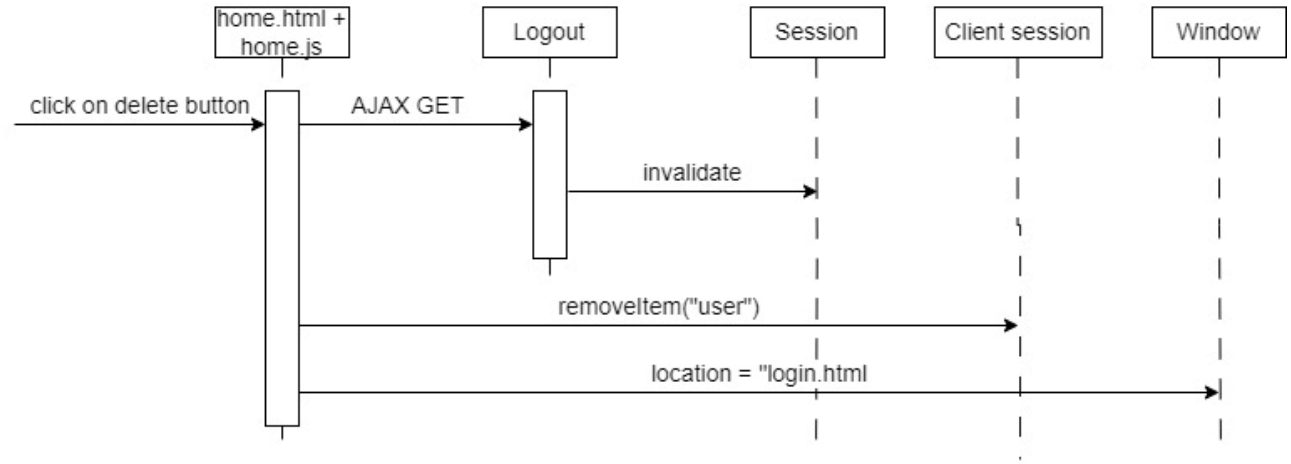
CreateAlbum



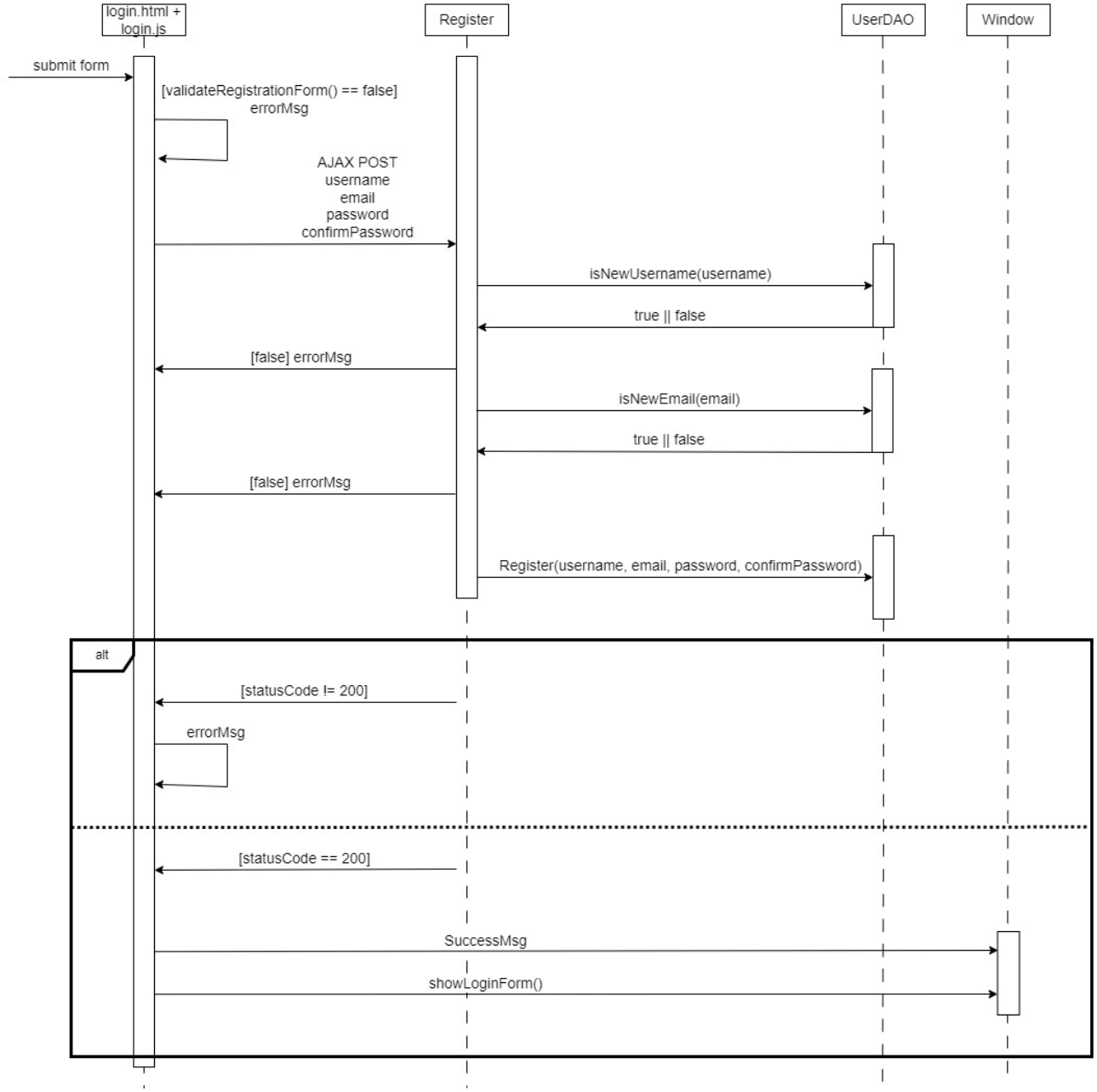
DeleteImage



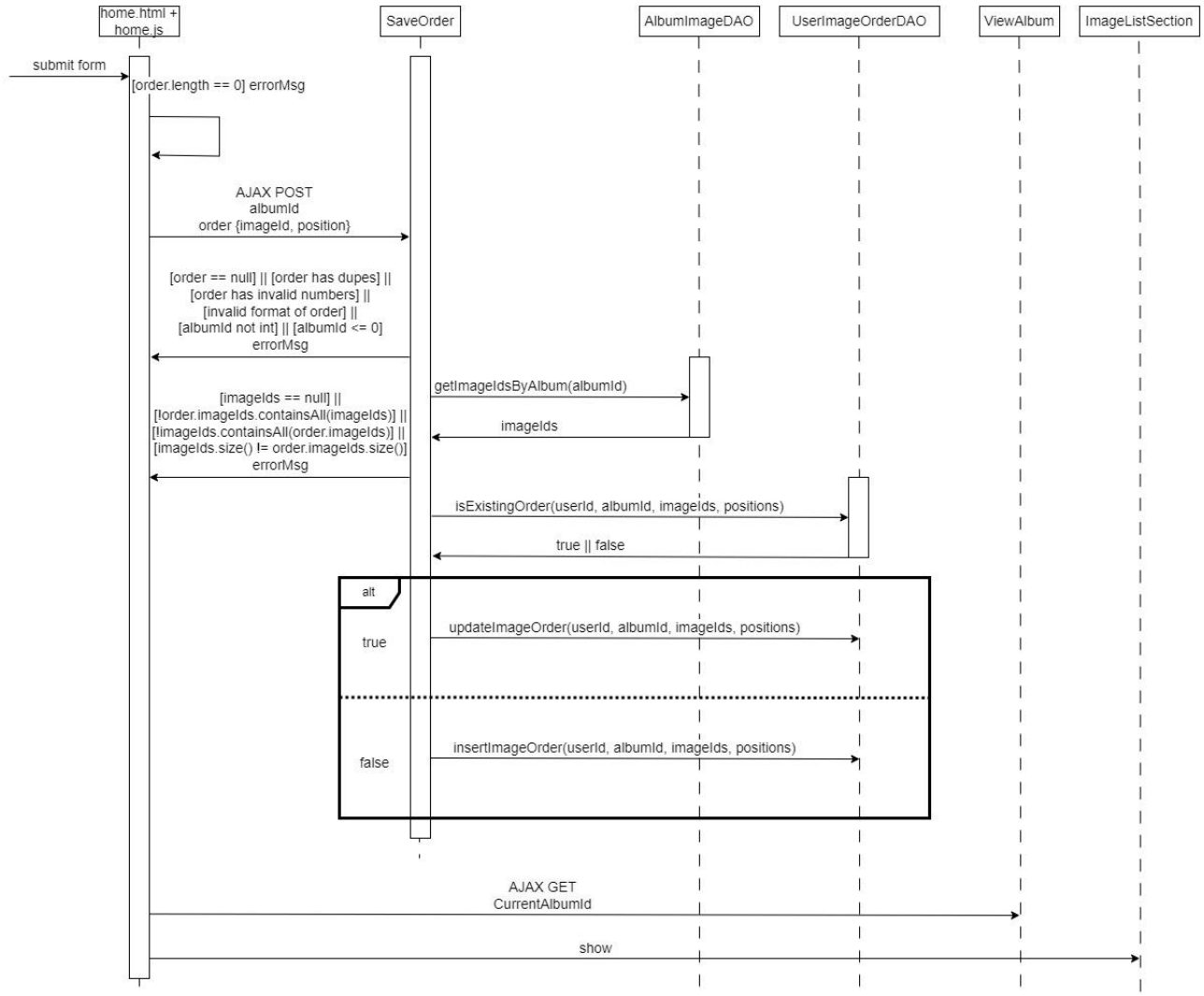
Logout



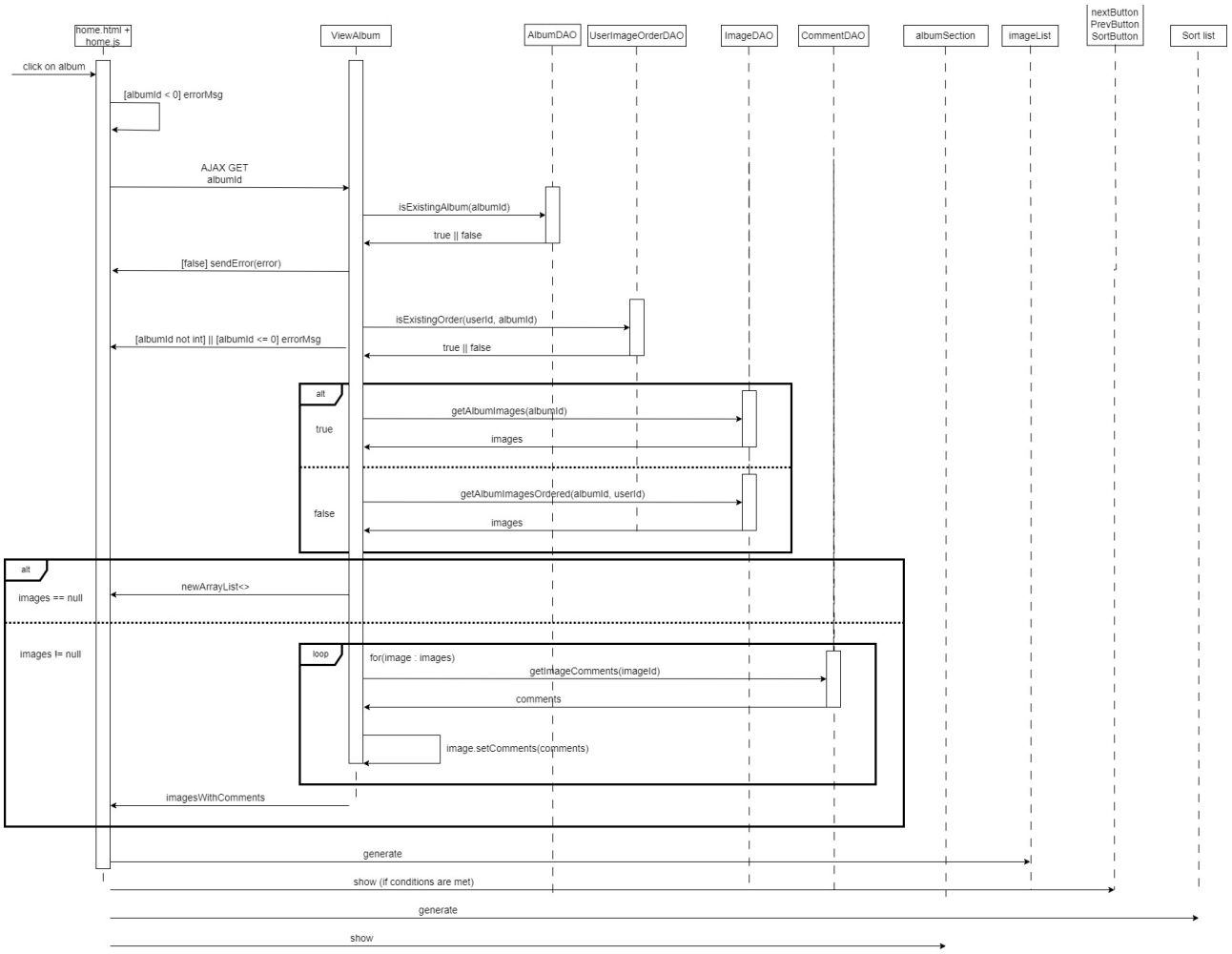
Register


login.html + login.js
Register
UserDAO
Window
submit form
[validateRegistrationForm() == false]
errorMsg
AJAX POST
username
email
password
confirmPassword
isNewUsername(username)
true || false
[false] errorMsg
isNewEmail(email)
true || false
[false] errorMsg
Register(username, email, password, confirmPassword)
alt
[statusCode != 200]
errorMsg
[statusCode == 200]
SuccessMsg
showLoginForm()

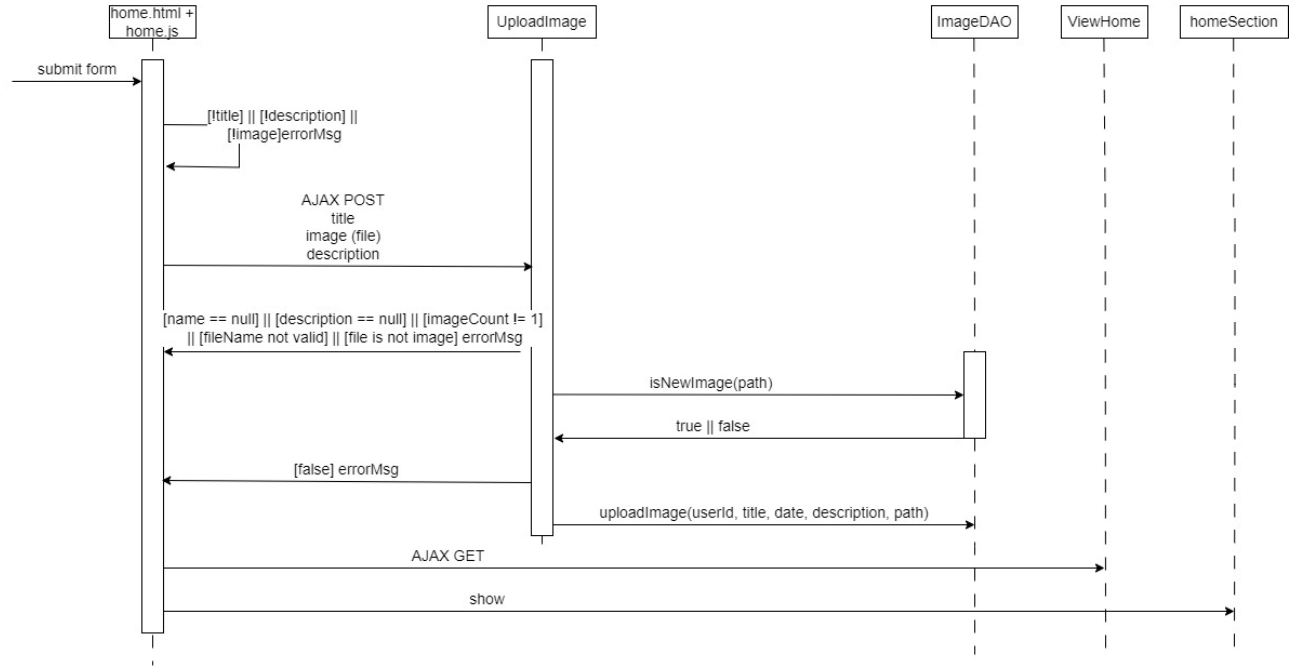
SaveOrder


home.html + home.js
SaveOrder
AlbumImageDAO
UserImageOrderDAO
ViewAlbum
ImageListSection
submit form
[order.length == 0] errorMsg
AJAX POST
albumId
order {imageId, position}
[order == null] || [order has dupes] ||
[order has invalid numbers] ||
[invalid format of order] ||
[albumId not int] || [albumId <= 0]
errorMsg
getImageIdsByAlbum(albumId)
imageIds
[imageIds == null] ||
[!order.imageIds.containsAll(imageIds)] ||
[!imageIds.containsAll(order.imageIds)] ||
[imageIds.size() != order.imageIds.size()]
errorMsg
isExistingOrder(userId, albumId, imageIds, positions)
true || false
alt
true
updateImageOrder(userId, albumId, imageIds, positions)
false
insertImageOrder(userId, albumId, imageIds, positions)
AJAX GET
CurrentAlbumId
show

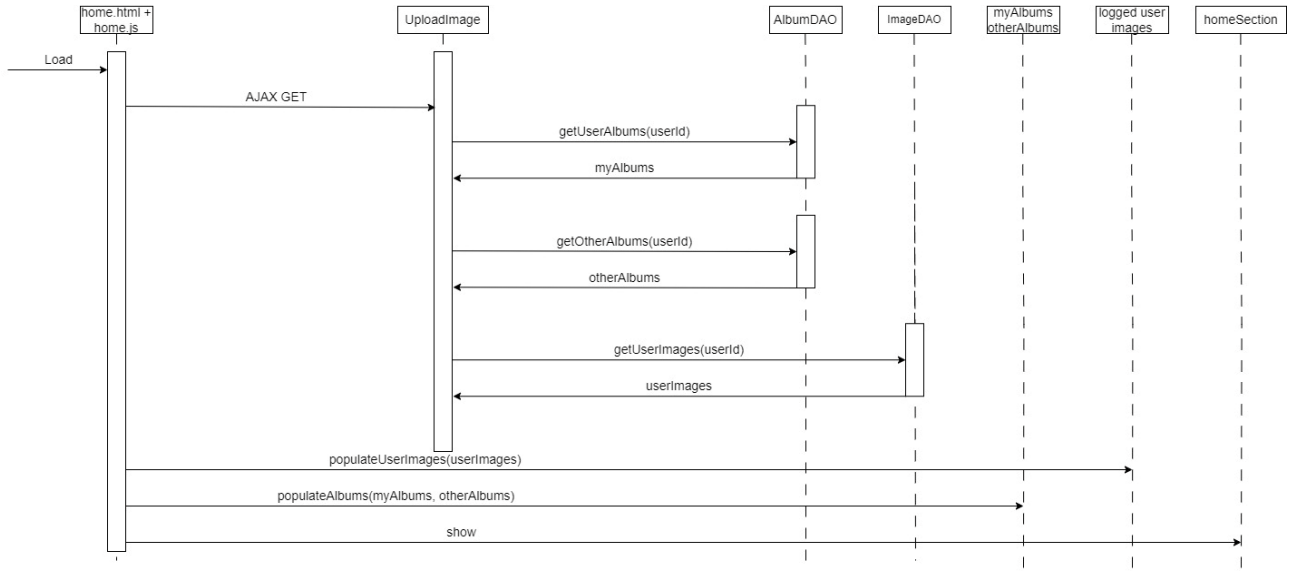
ViewAlbum



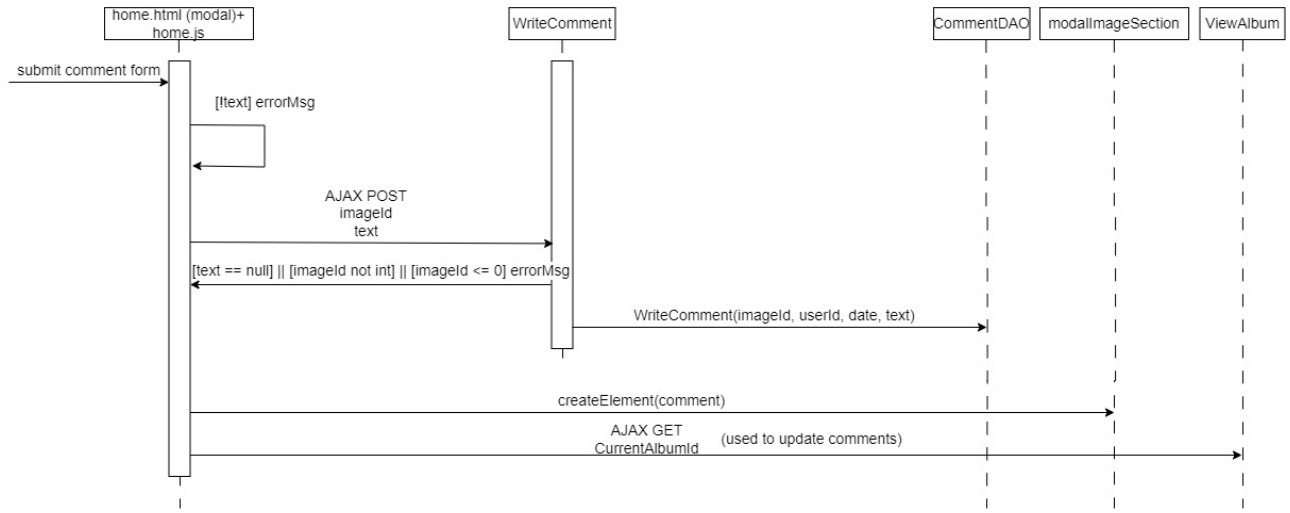
ViewHome



UploadImage



WriteComment



NOTE:

- **ImageGetter** semplicemente recupera le immagini dal file system
- Per ogni variabile String viene verificato che essa non via vuota o contenga solo spazi bianchi
- Un filtro controlla che ad ogni chiamata di una servlet venga controllato se l'utente è loggato o no, in caso negativo viene reindirizzato alla schermata di login