

Galleria di immagini

Stefan Djokovic

Esame TIW 2019/2020

Analisi dei dati

Esercizio 1: galleria di immagini- Versione pure HTML.

Un'applicazione web consente la gestione di una galleria d'immagini. Ogni immagine è memorizzata nella base di dati mediante un titolo, una data, un testo descrittivo e il percorso del file dell'immagine nel file system del server su cui l'applicazione è rilasciata. Le immagini sono raggruppate in album e sono associate a uno o più commenti. Un commento ha un testo e il nome dell'utente che lo ha creato. Un album ha un titolo e una data di creazione. Quando l'utente accede all'HOME PAGE, questa presenta l'elenco degli album, ordinato per data di creazione decrescente. Quando l'utente clicca su un album nell'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.

Entità:

- Galleria d'immagini (album)
- Immagini
- Commenti
- Utente

Attributi:

- Titolo (Immagine e Album)
- Data (Immagine)
- Descrizione (Immagine)
- Filepath file system (Immagine)
- ...

Relazione:

- «Immagini raggruppate in album»
- «Immagini associate a uno o più commenti»

Analisi dei dati RIA

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.
- 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 gli 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.

In più rispetto all'HTML:

- L'ordine dell'album
(personalizzato per utente)

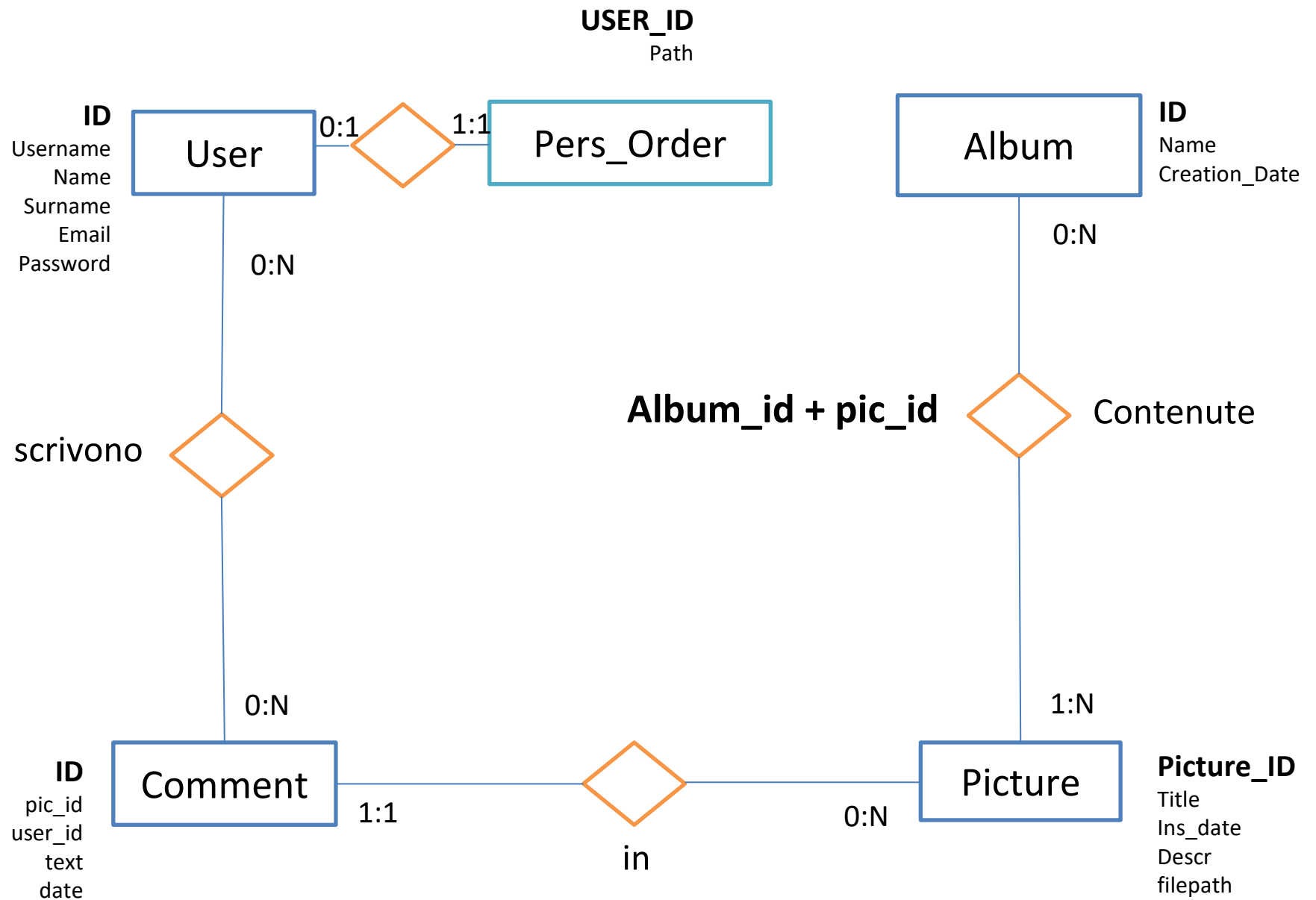


Table user

```
CREATE TABLE `user` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `username` varchar(32) NOT NULL,  
  `name` varchar(32) NOT NULL,  
  `surname` varchar(32) NOT NULL,  
  `email` varchar(64) NOT NULL,  
  `password` varchar(32) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `username_UNIQUE` (`username`)  
);
```

Table: pers_order

```
CREATE TABLE `pers_order` (  
  `user_id` int NOT NULL AUTO_INCREMENT,  
  `path` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`user_id`),  
  CONSTRAINT `user_id` FOREIGN KEY (`user_id`)  
REFERENCES `user` (`id`) ON DELETE CASCADE ON UPDATE  
CASCADE  
);
```

Table: comment

```
CREATE TABLE `tiwexam`.`comment` (  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `pic_id` INT NOT NULL,  `user_id` INT NOT NULL,  
  `text` VARCHAR(255) NOT NULL,  
  `date` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  CONSTRAINT `pic_construser` FOREIGN KEY (`pic_id`)   
  REFERENCES `tiwexam`.`picture` (`picture_id`) ON   
  DELETE CASCADE ON UPDATE NO ACTION,  
  CONSTRAINT `user` FOREIGN KEY (`user_id`) REFERENCES   
  `tiwexam`.`user` (`id`) ON DELETE CASCADE ON UPDATE   
  NO ACTION  
);
```

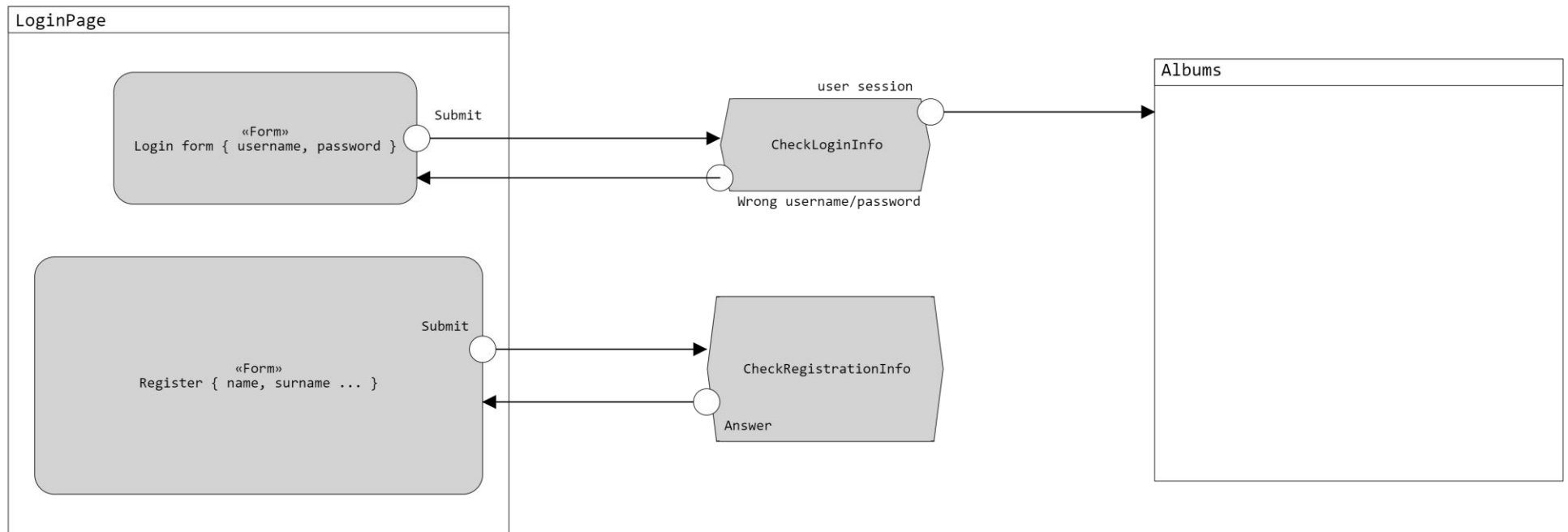
Table: album

```
CREATE TABLE `album` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(32) NOT NULL,  
  `creation_date` datetime NOT NULL DEFAULT  
CURRENT_TIMESTAMP,  
PRIMARY KEY (`id`)  
);
```


Table: album_contains_pic

```
CREATE TABLE `tiwexam`.`album_contains_pic` (  
  `album_id` INT NOT NULL,  
  `pic_id` INT NOT NULL,  
  CONSTRAINT `album` FOREIGN KEY (`album_id`)  
    REFERENCES `tiwexam`.`album` (`id`) ON DELETE  
    CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `pic` FOREIGN KEY (`pic_id`) REFERENCES  
    `tiwexam`.`picture` (`picture_id`) ON DELETE  
    CASCADE ON UPDATE CASCADE,  
  PRIMARY KEY(`album_id`, `pic_id`)  
);
```

Login design



Login Page

Welcome!

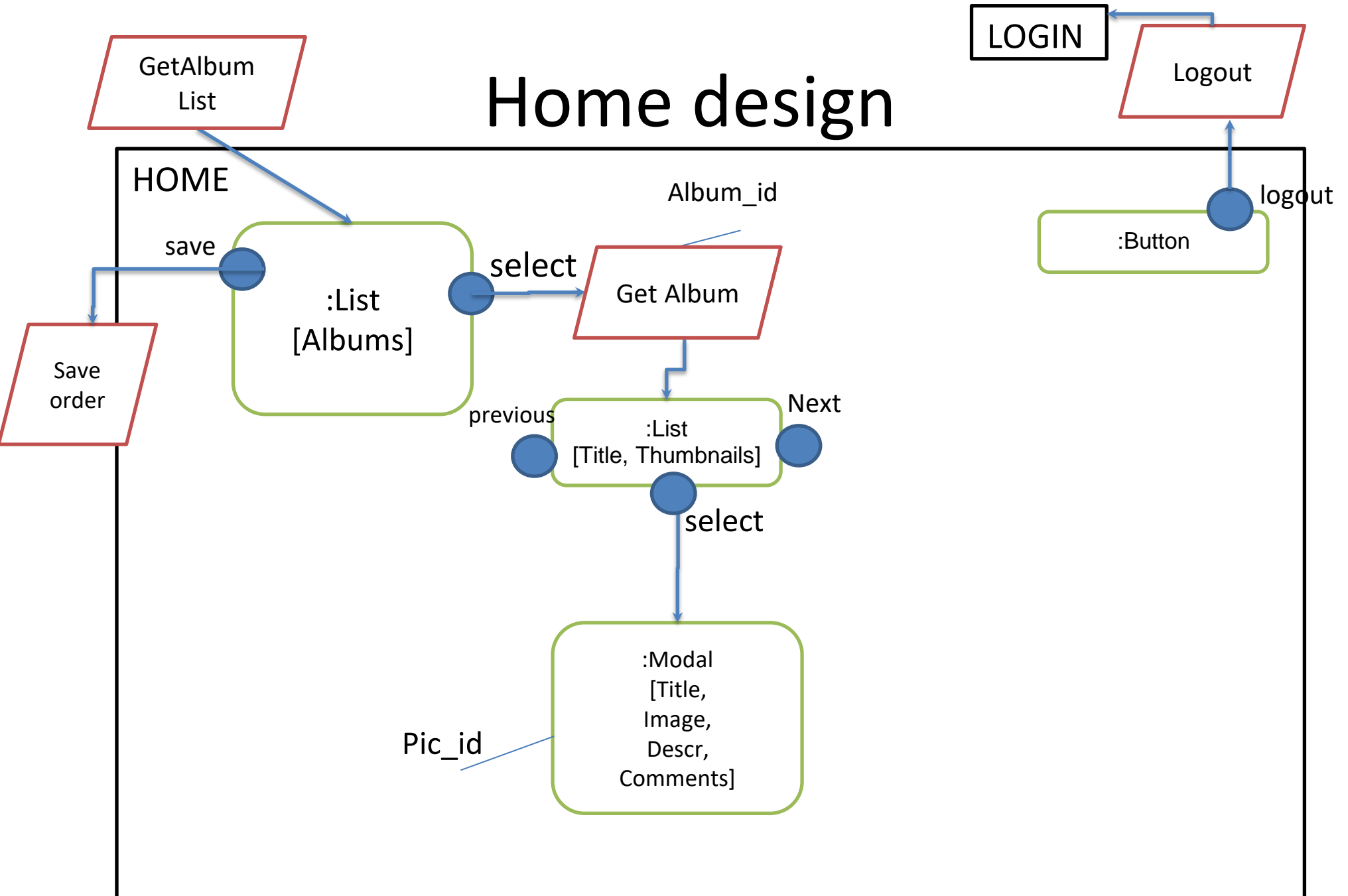
Username
Password

login

Name
Surname
Email
Username
Password
Repeat Password

Register

Home design




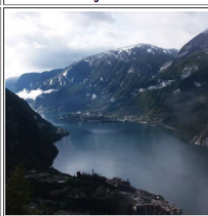
Welcome stefan, enjoy your time!

Albums Available:



Save Changes

[Logout](#)

Danis
album2
album3
album5
album4
album1

JoJo	Detective Conan	No Title	Memes!	Fjord!
				



JoJo	Detective Conan	No Title	Memes!	Fjord!
				

Fjord!

x



I miss them

Comment

Comment

Submit

jonas

Det var veldig deilig Stefan! :)

stefan

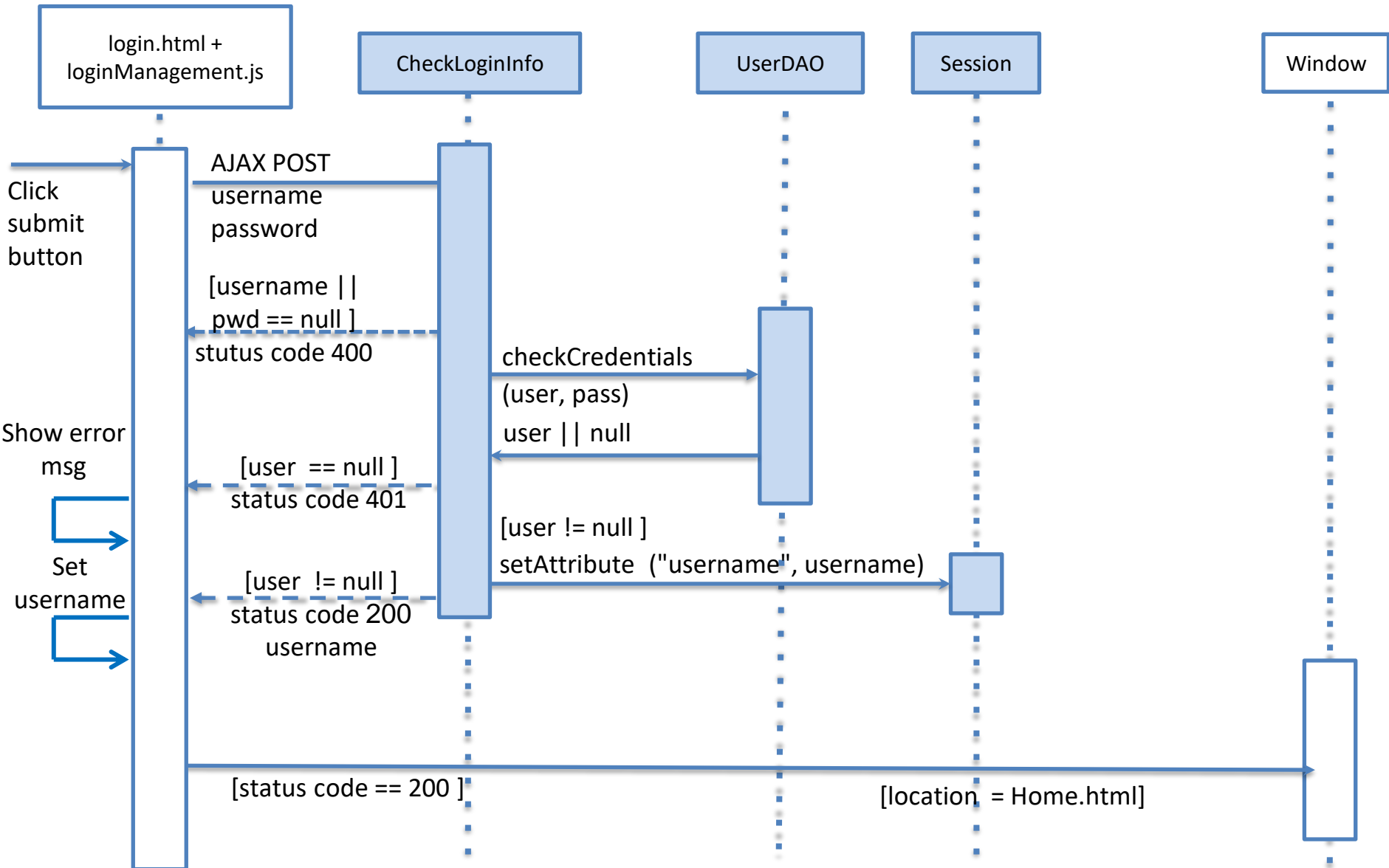
Ja, det var det! ^^

Presentazione Eventi

Client side

Server side

Evento: login



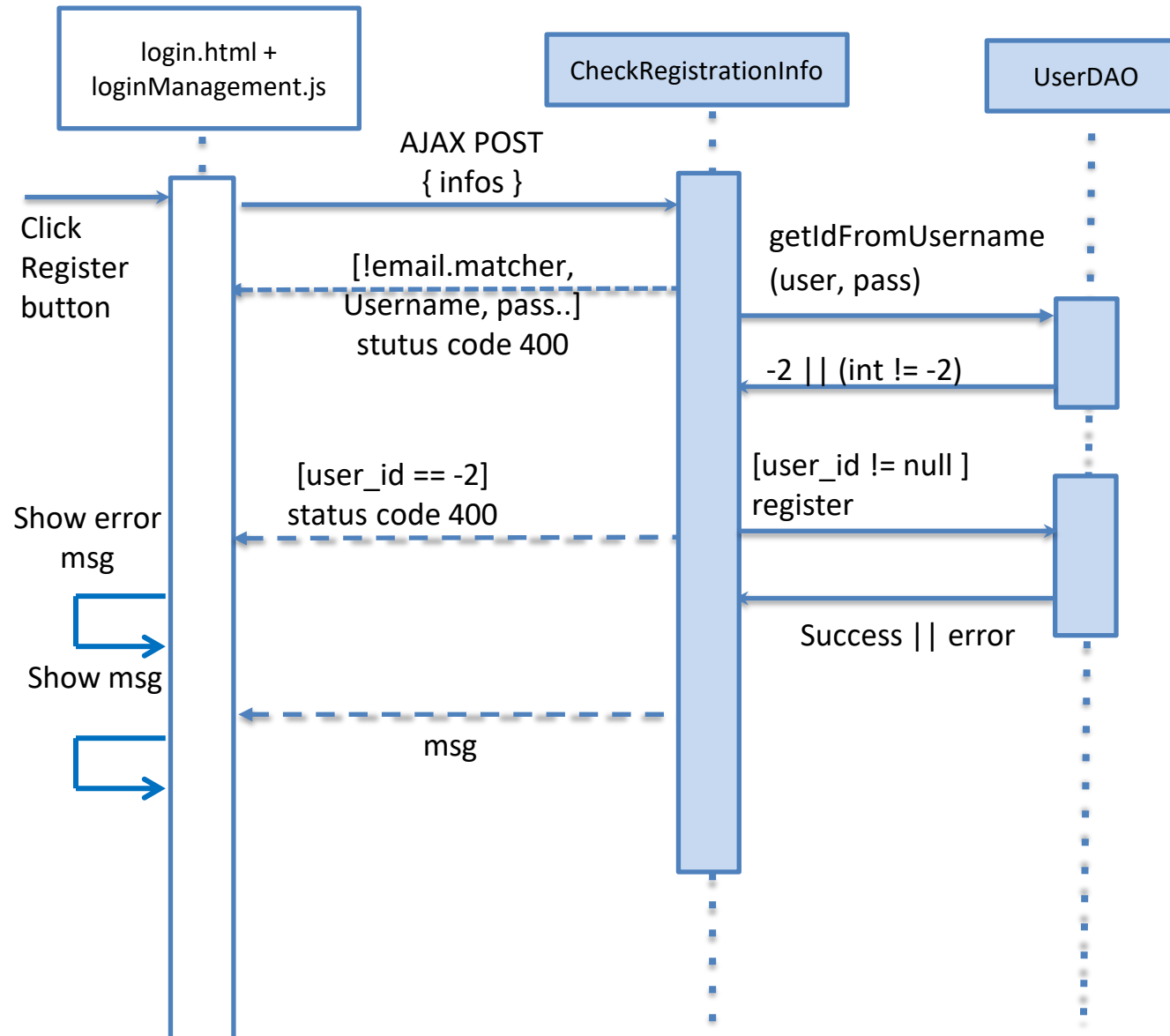


Client side

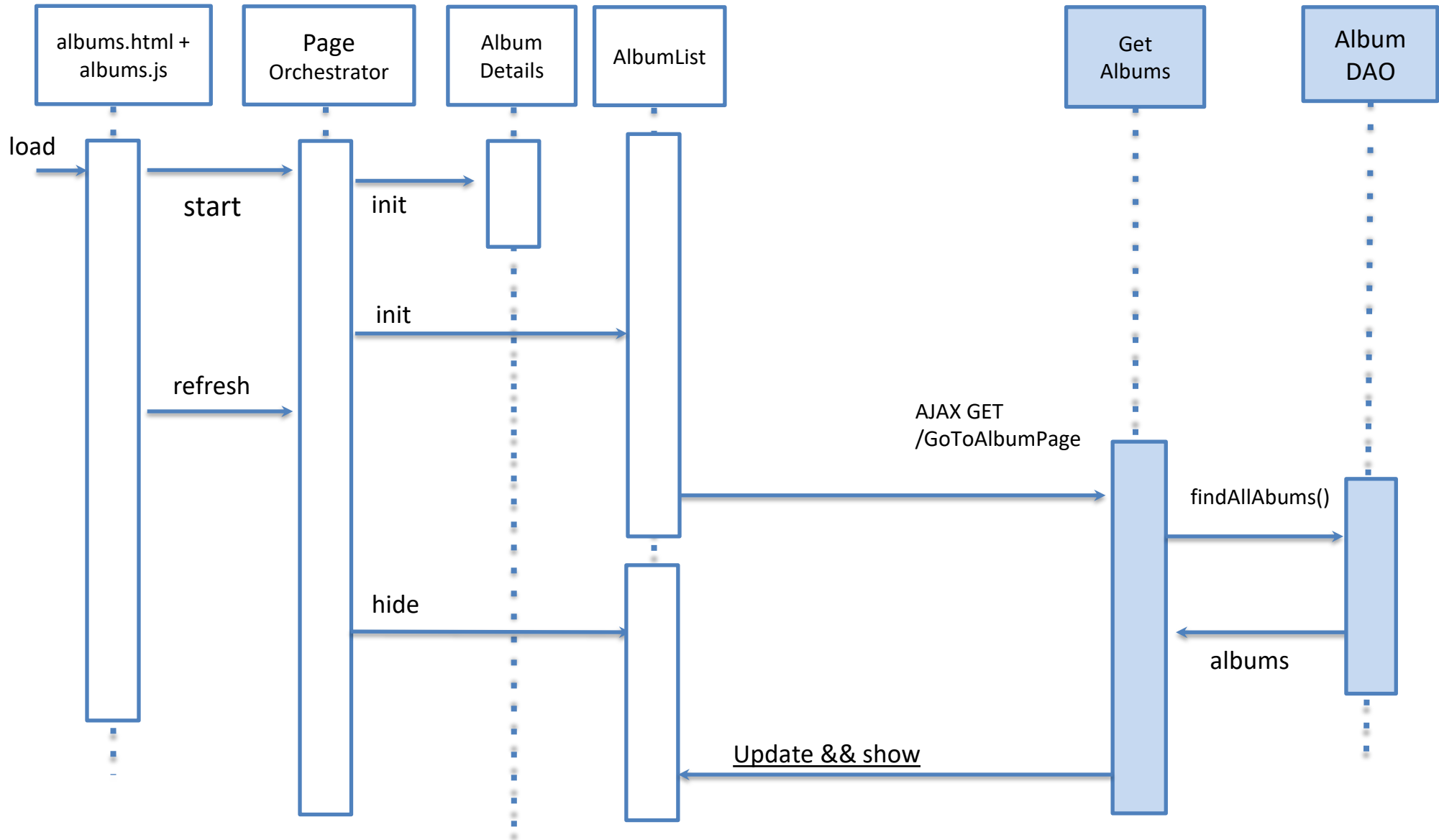


Server side

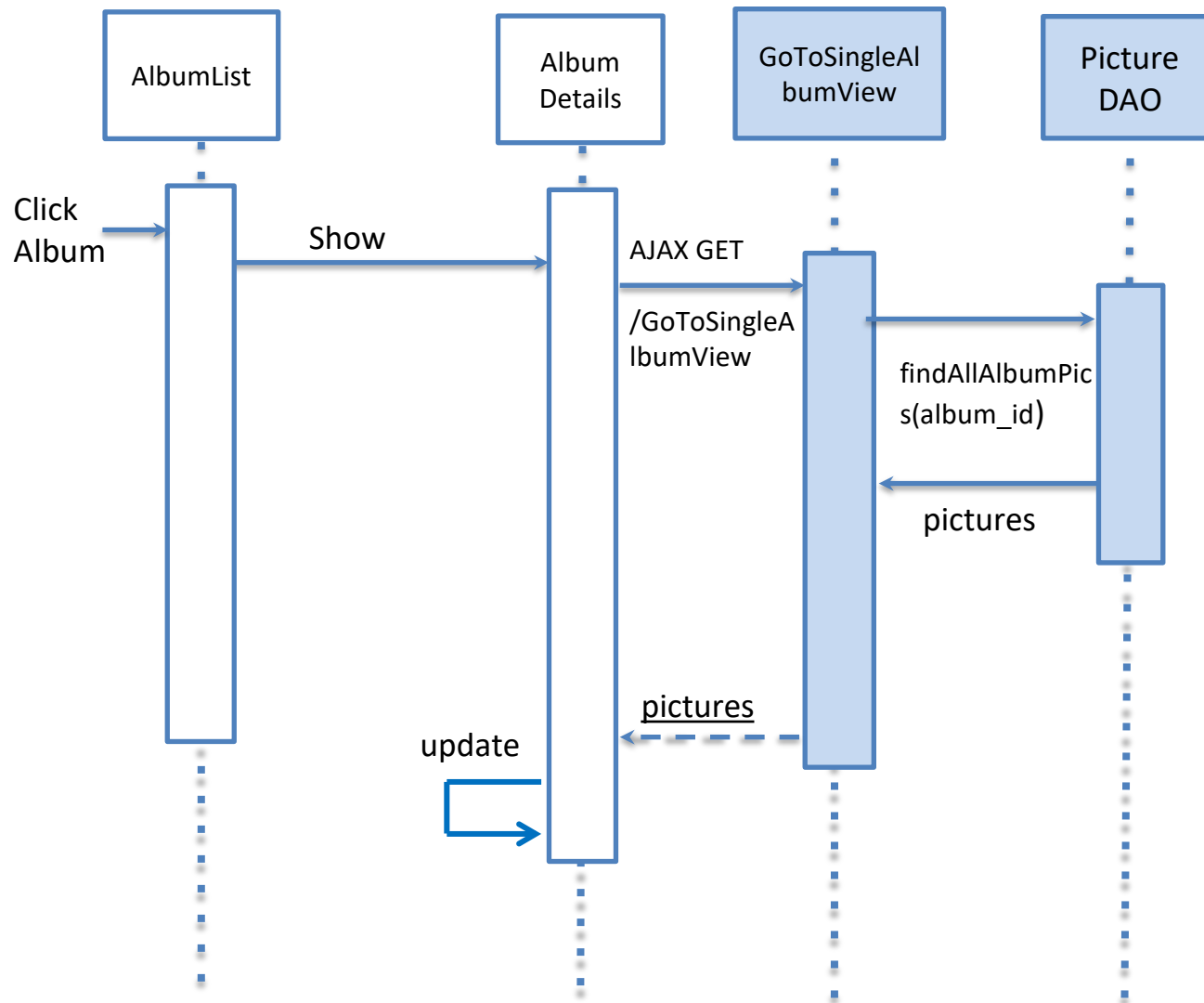
Register



Evento: caricamento Home page



Evento: selezione di un Album



Nota: da qui ho tutte le informazioni sull'immagine, mi mancano solo i commenti

Evento: selezione immagine

