Documentazione progetto TIW AA 2021/2022

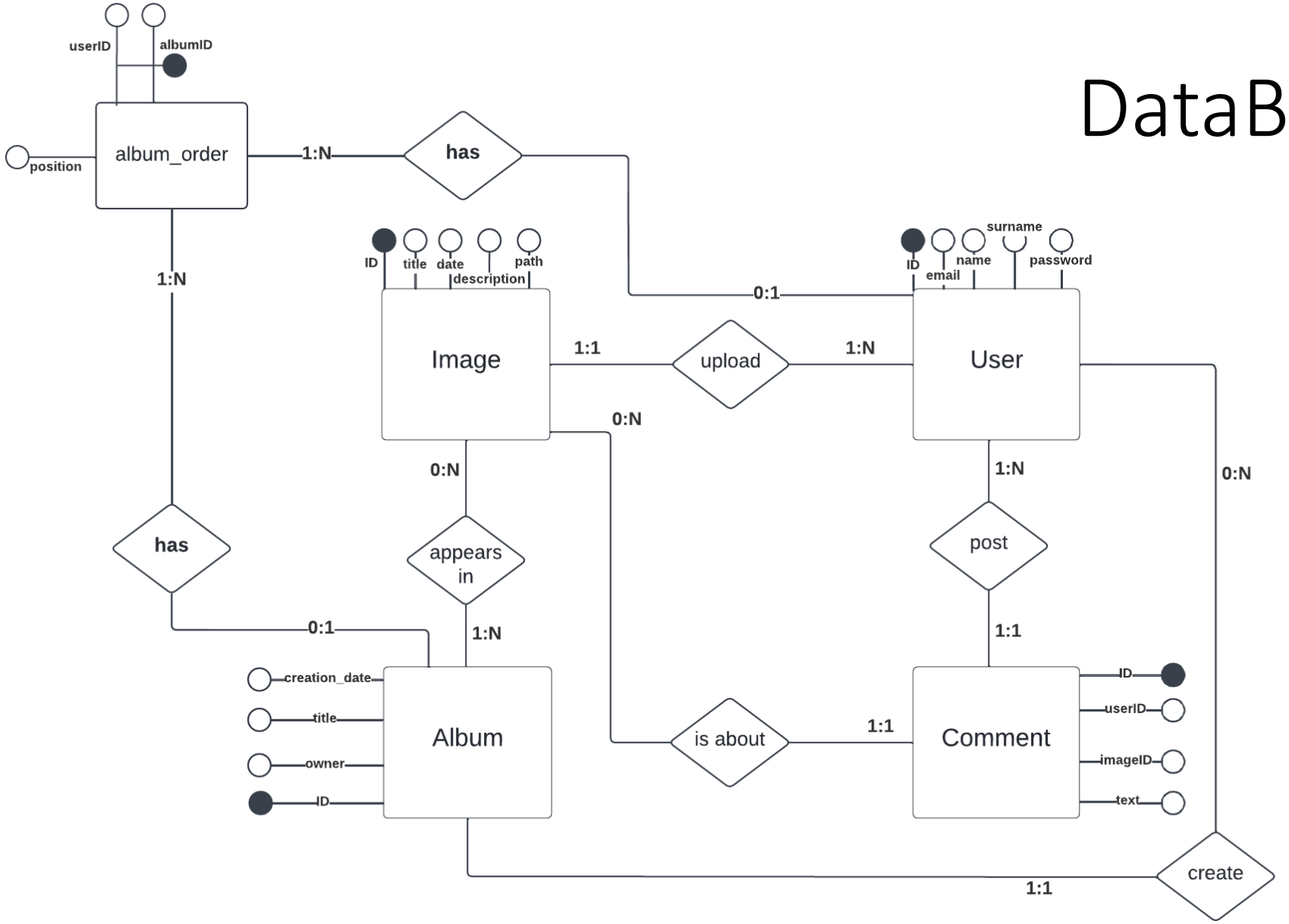
Componenti gruppo: Brontesi Milo, Colombo Andrea

Docenti: Fraternali Piero, Milani Federico, Pinciroli Nicolò

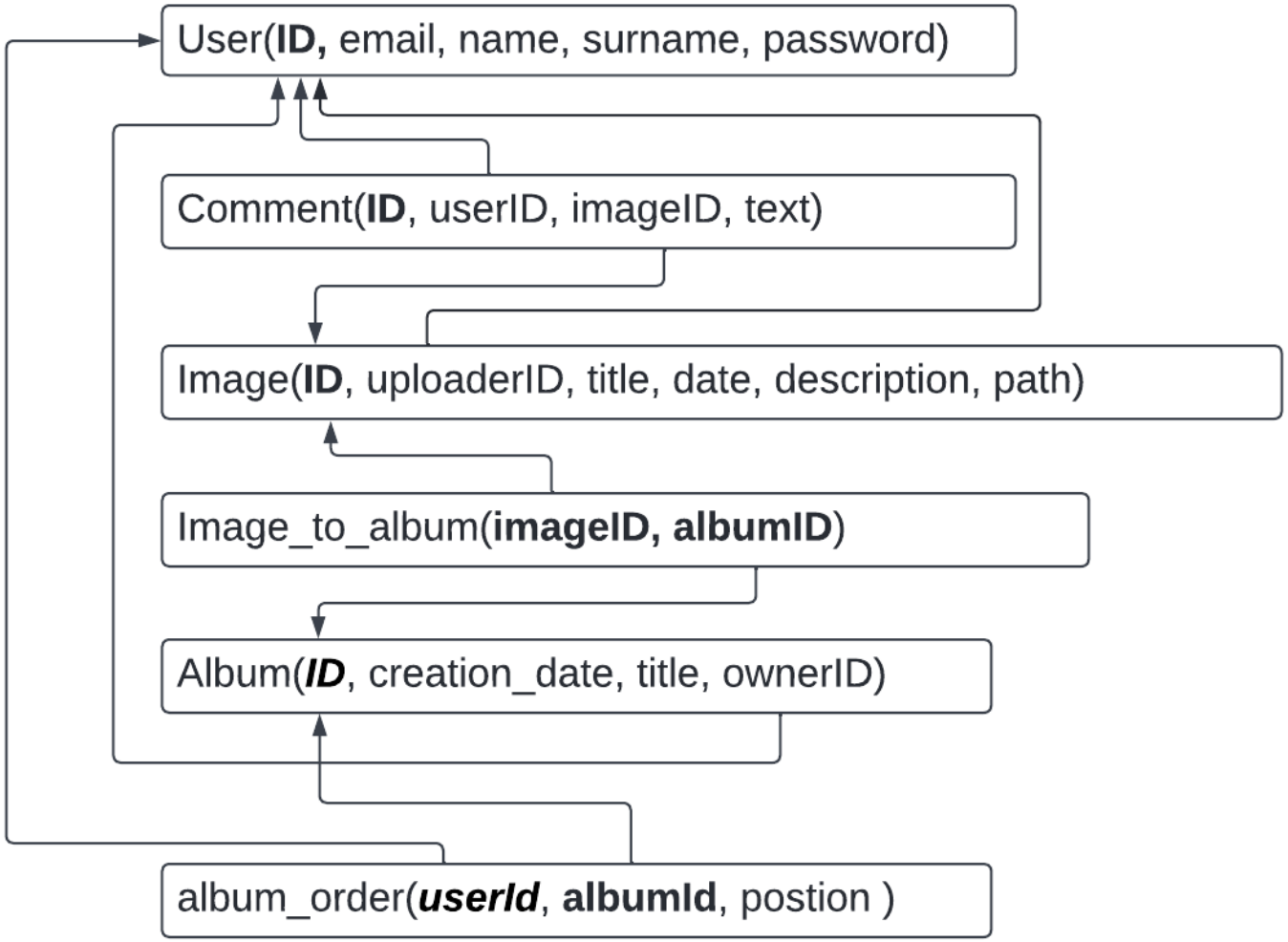
DataBase requirements

- 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.
- Entities, attributes, relationships

DataBase Design



DataBase Design



Local DataBase schema

- **CREATE TABLE** album (
  ID int NOT NULL AUTO_INCREMENT,
  creation_date date NOT NULL,
  title varchar(32) NOT NULL,
  ownerID int NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (ownerID) REFERENCES user (ID) ON DELETE CASCADE ON UPDATE CASCADE
)

- **CREATE TABLE** album_order (
  userID int NOT NULL,
  albumID int NOT NULL,
  position int NOT NULL,
  PRIMARY KEY (userID,albumID),
  KEY albumID (albumID),
  FOREIGN KEY (albumID) REFERENCES album (ID) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (userID) REFERENCES user (ID) ON DELETE CASCADE ON UPDATE CASCADE
)

Local DataBase schema

- **CREATE TABLE** image (
  ID int NOT NULL AUTO INCREMENT,
  uploaderID int NOT NULL,
  title varchar(32) NOT NULL,
  path varchar(260) NOT NULL,
  date date NOT NULL,
  description varchar(280) NOT NULL,
  PRIMARY KEY (ID),
  KEY uploaderID (uploaderID),
  FOREIGN KEY (uploaderID) REFERENCES user (ID)
  ON DELETE CASCADE ON UPDATE CASCADE
- **CREATE TABLE** comment (
  ID int NOT NULL AUTO_INCREMENT,*
  userID int NOT NULL,
  imageID int NOT NULL,
  text varchar(280) NOT NULL,
  PRIMARY KEY (ID),
  FOREIGN KEY (userID) REFERENCES user (ID) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (imageID) REFERENCES image (ID) ON DELETE CASCADE ON UPDATE CASCADE,
)

Local DataBase schema

- **CREATE TABLE** image_to_album (
  imageID int NOT NULL,
  albumID int NOT NULL,
  PRIMARY KEY (imageID,albumID),
  KEY userID(albumID),
  KEY userID (albumID),
  FOREIGN KEY (albumID) REFERENCES album (ID),
  FOREIGN KEY (imageID) REFERENCES image (ID)
)
- **CREATE TABLE** user (
  ID int NOT NULL AUTO_INCREMENT,
  email varchar(320) NOT NULL,
  name varchar(45) NOT NULL,
  surname varchar(45) NOT NULL,
  password varchar(45) NOT NULL,
  PRIMARY KEY (ID),
  UNIQUE KEY email_UNIQUE (email)
)

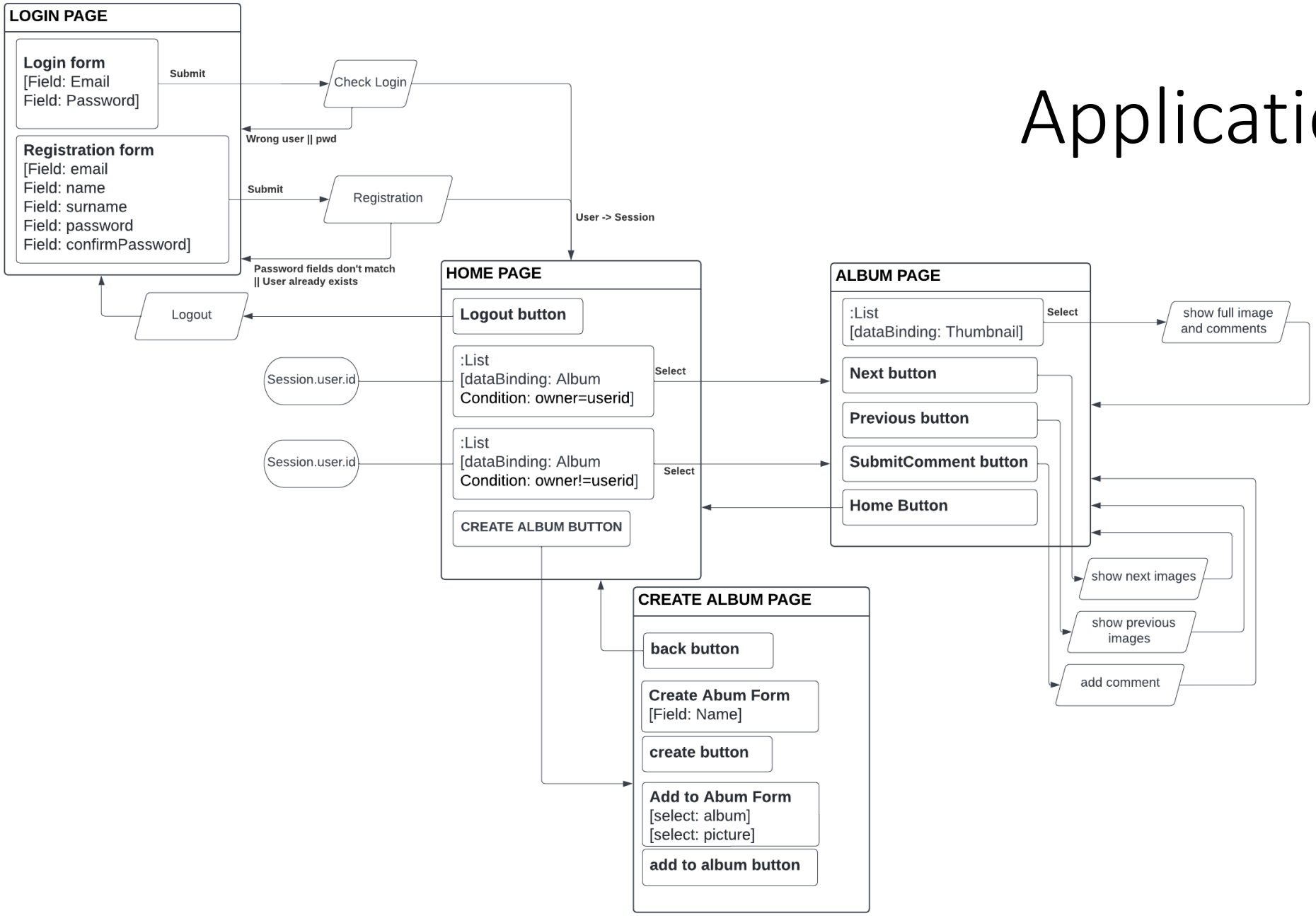
Application requirements PureHTML

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.

Page(views), view components, actions, events

Application Design



COMPONENTS (PureHTML)

- **MODEL OBJECTS** (beans):
  - Album
  - Comment
  - Image
  - User
- **DATA ACCESS OBJECTS** (classes):
  - AlbumDAO
  - UserDAO
  - ImageDAO
  - CommentDAO
  - TestDAO
- **CONTROLLERS** (servlets):
  - CheckLogin
  - CheckRegistration
  - CreateComment
  - GoToAlbumPage
  - GoToHomePage
  - Logout
  - ShowImage
  - AddToAlbum
  - CreateAlbum
  - GoToCreateAlbum
- **VIEWS** (Templates):
  - Home
  - Album

Richieste per versione RIA

- 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

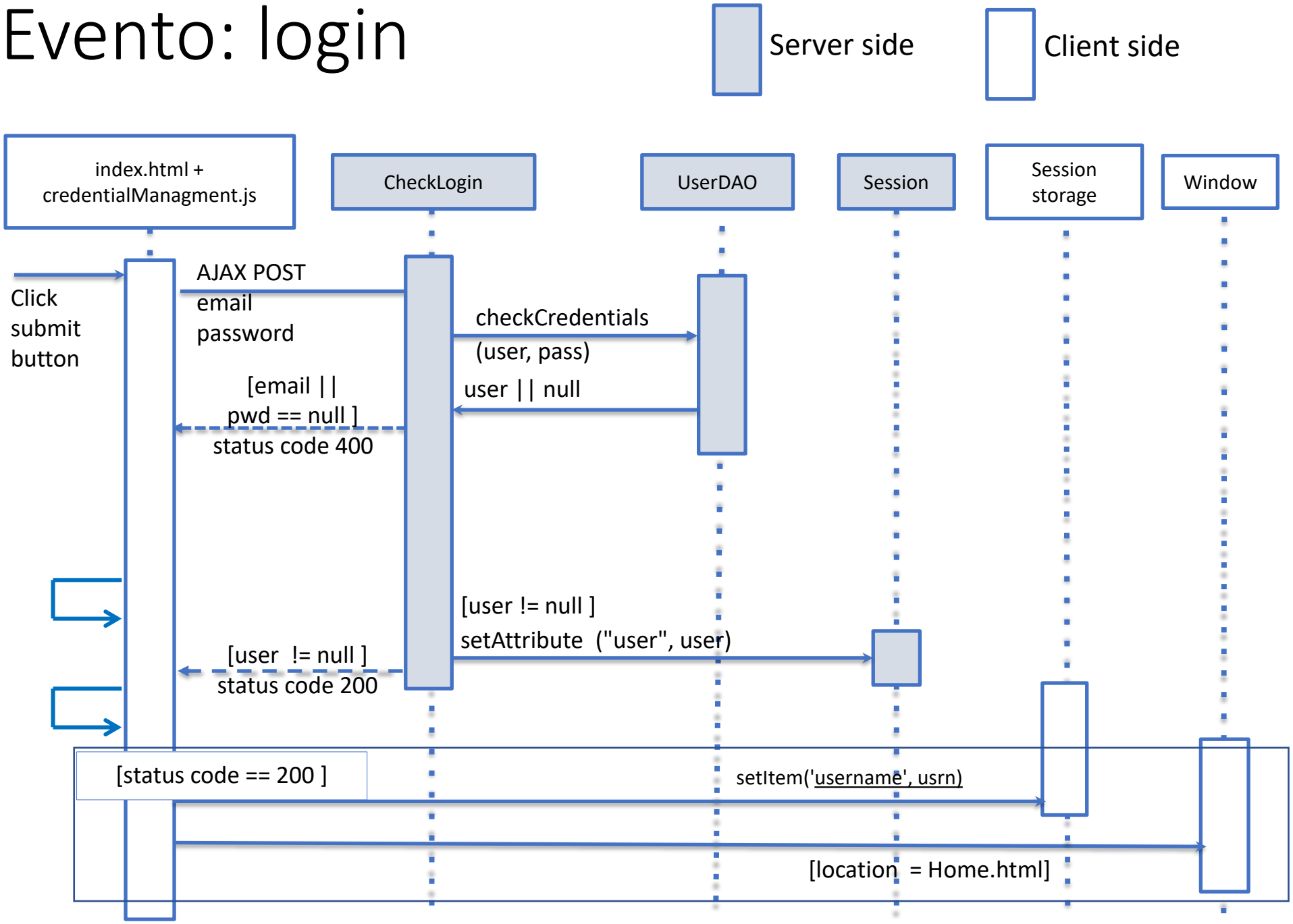
Components (RIA)

- **MODEL OBJECTS** (beans):
  - Album
  - Comment
  - Image
  - User
- **DATA ACCESS OBJECTS** (classes):
  - AlbumDAO
  - UserDAO
  - ImageDAO
  - CommentDAO
  - TestDAO
- **CONTROLLERS** (servlets):
  - AddToAlbum
  - CheckLogin
  - CheckRegistration
  - CreateAlbum
  - CreateComment
  - GetAlbumData
  - GetCreateAlbumData
  - GetHomeData
  - GetImageData
  - Logout
  - SaveAlbumOrder

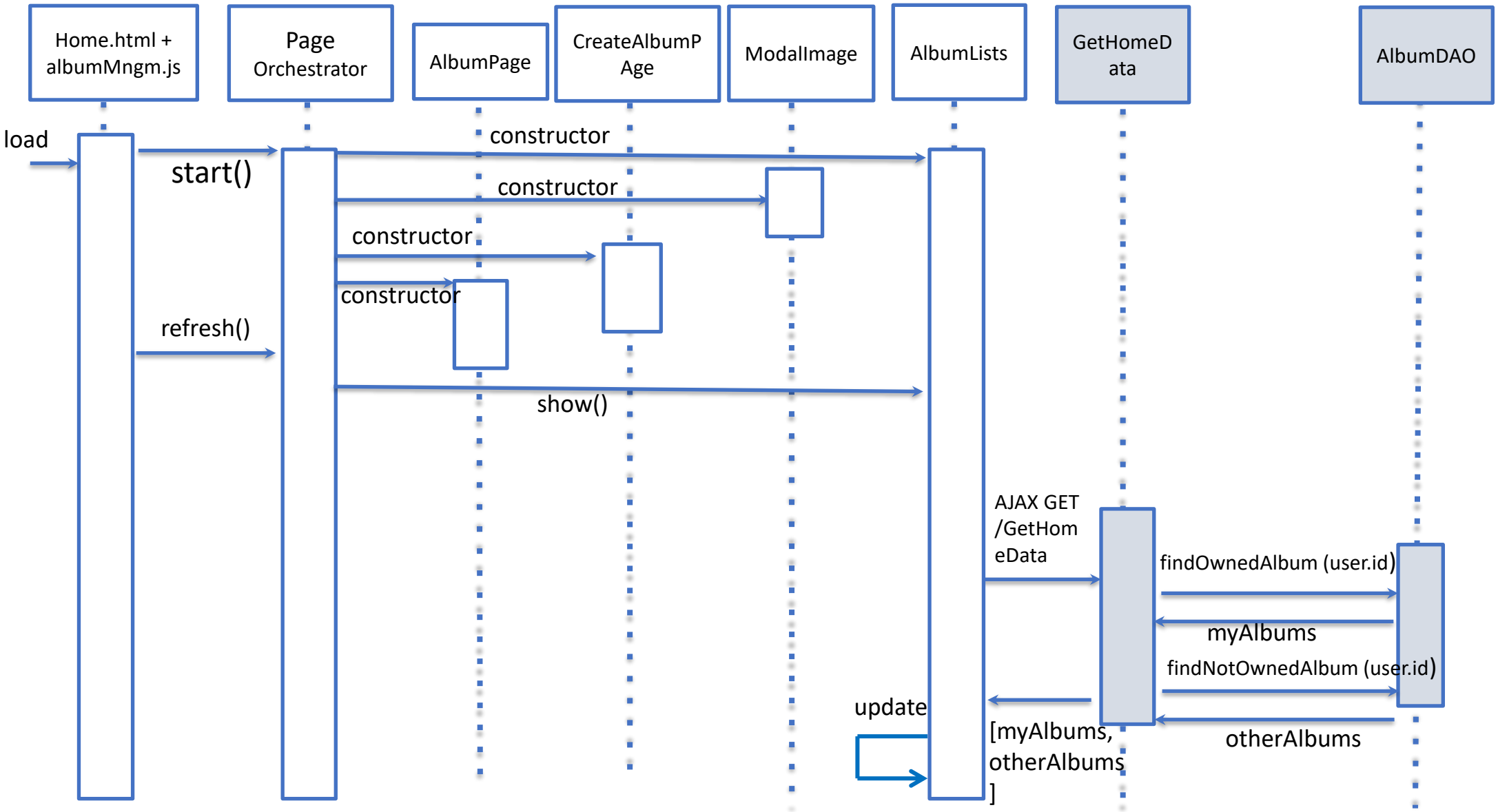
Descrizione eventi/interfaccia

- **Login**: Dopo aver compilato il form vengono effettuati dei controlli lato client sulla validità dei campi (formato email) e, se questi controlli vanno a buon fine, vengono effettuati i controlli a lato server. In caso di esito positivo anche su essi si viene reindirizzati alla Home page.
- **Modifica ordine album**: l'utente ha la possibilità di ordinare i propri album a sua discrezione attraverso un sistema di drag and drop, dopo l'avvenuta modifica e il click sul bottone di save order, l'ordine scelto dal nostro utente verrà salvato sul DB e diverrà l'ordine predefinito.
- **Apertura di un album**: dopo il click su «open» verranno mostrate le immagini presenti all'interno di quell'album (a gruppi di 5), la versione RIA differisce da quella pure HTML in quanto il server verrà interrogato solo una volta per il recupero delle immagini (e in seguito vengono gestite a lato client) mentre la versione Pure HTML interroga il server per ogni gruppo di immagini (attraverso l'utilizzo di LIMIT+offset nella query).
- **Click su create album**: dopo aver cliccato sul bottone viene mostrata l'interfaccia che permette all'utente di creare un nuovo album da aggiungere ai suoi già presenti.
  Attraverso questa interfaccia è inoltre possibile aggiungere delle immagini ai propri album (verranno mostrate come opzione le immagini non presenti in alcun album).
- **Click su back**: dopo aver cliccato su back «back» si viene riportati all'interfaccia che mostra le due liste di album, la modifica della pagina è gestita lato client.
- **Hover su immagine**: dopo aver portato il puntatore su un'immagine verrà mostrata l'interfaccia che mostra l'immagine, i commenti già presenti e un form per l'aggiunta di un proprio commento.
- **Creazione commento**: dopo la compilazione del form apposito il commento verrà aggiunto al database e sarà visibile nell'interfaccia citata sopra.
- **Logout**: distruzione sessione e reindirizzamento a Index.
- **Caricamento Home page**: al caricamento della Home page saranno visibili due liste di album (albums dell'utente che ha fatto il login e albums degli altri utenti).

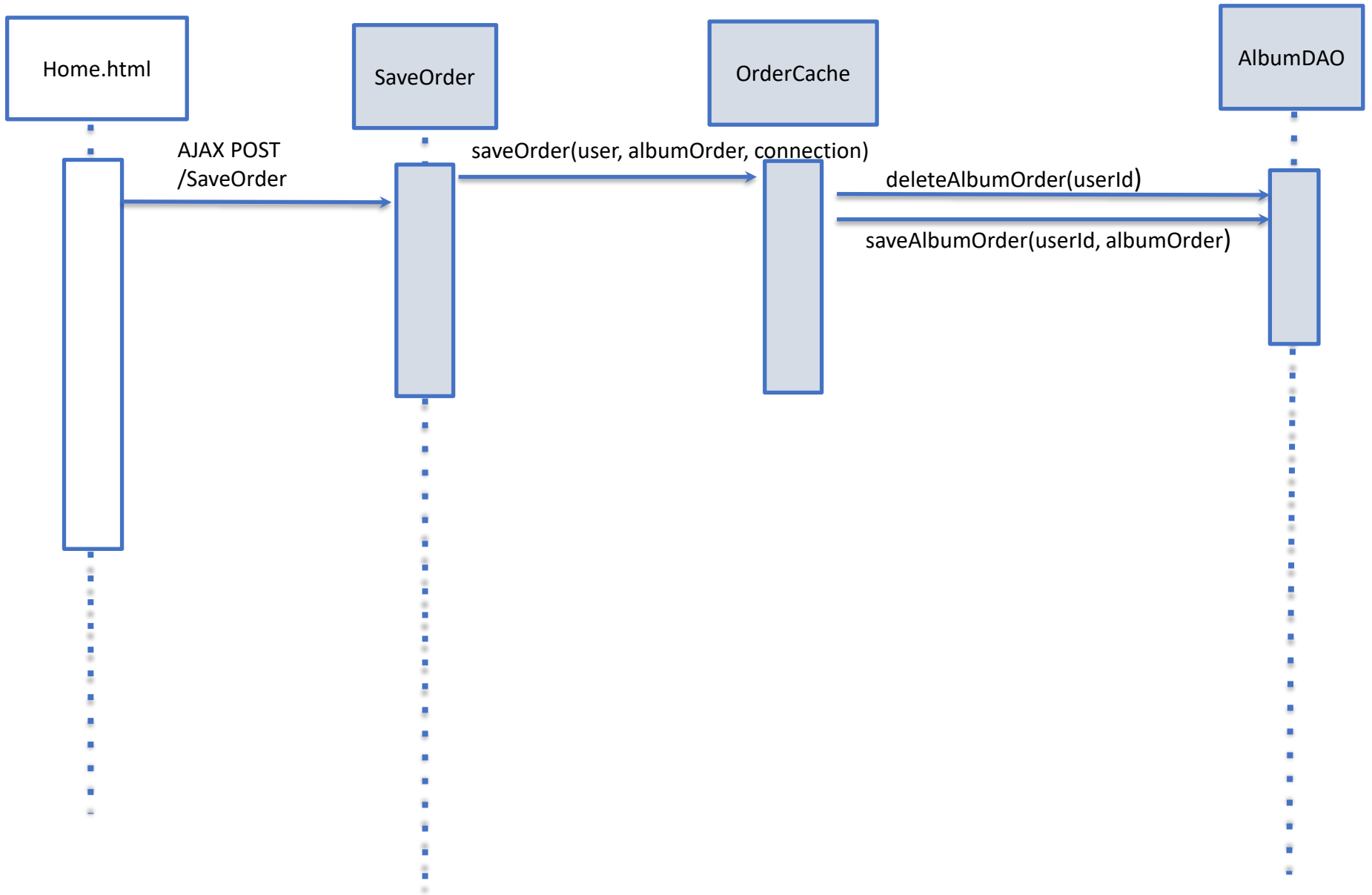
Evento: login



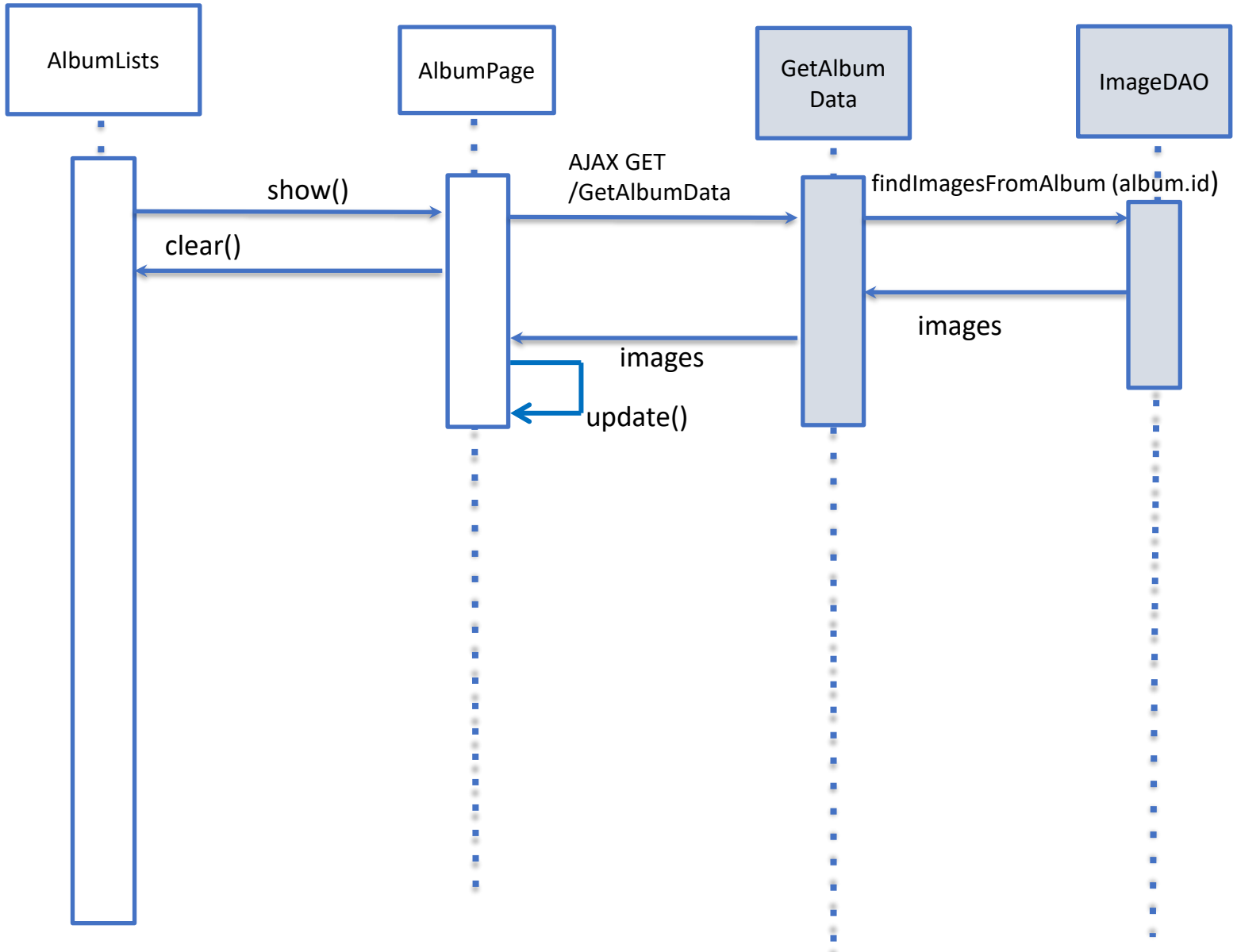
Evento: caricamento Home page



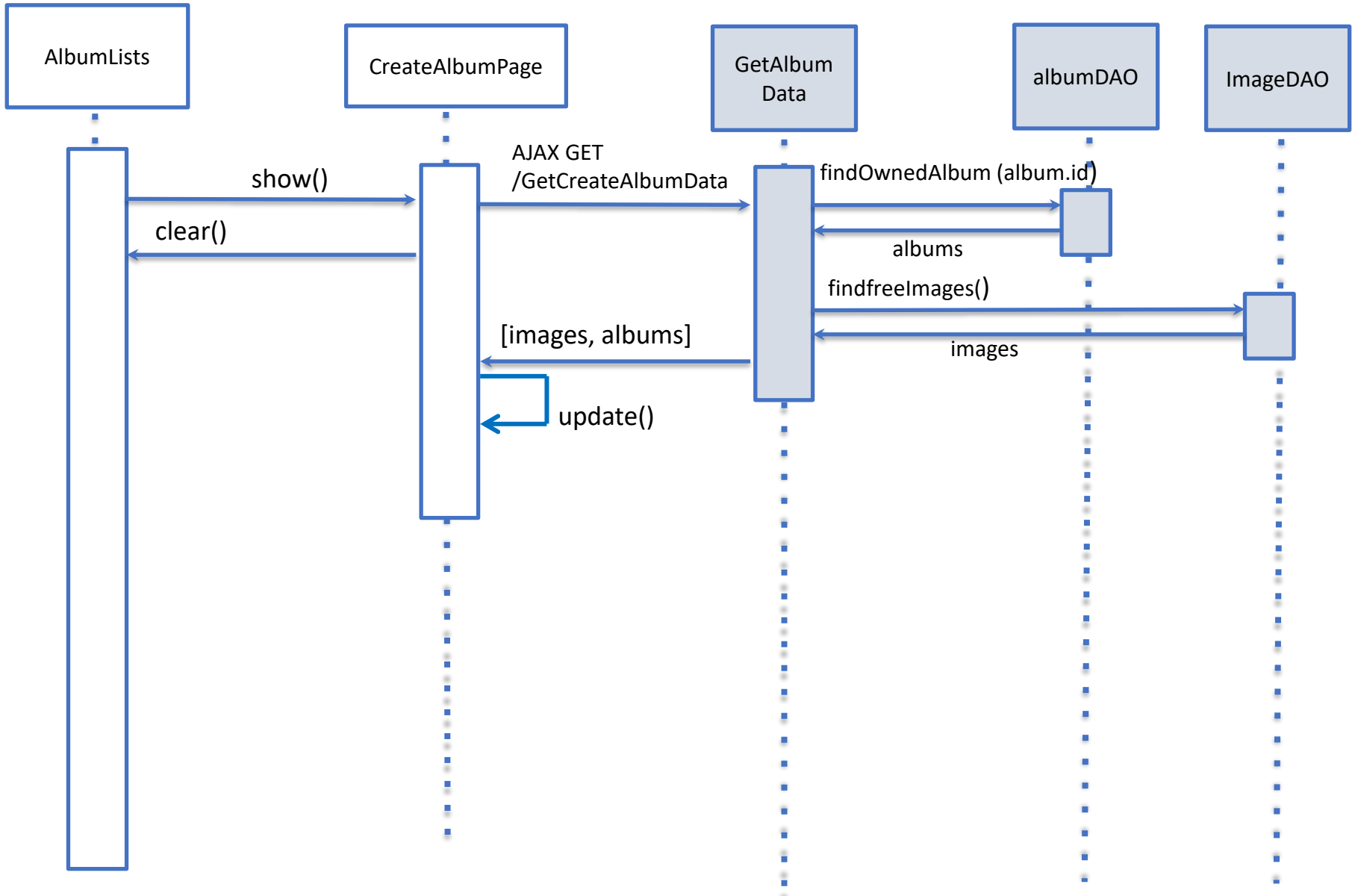
Evento: click su save order



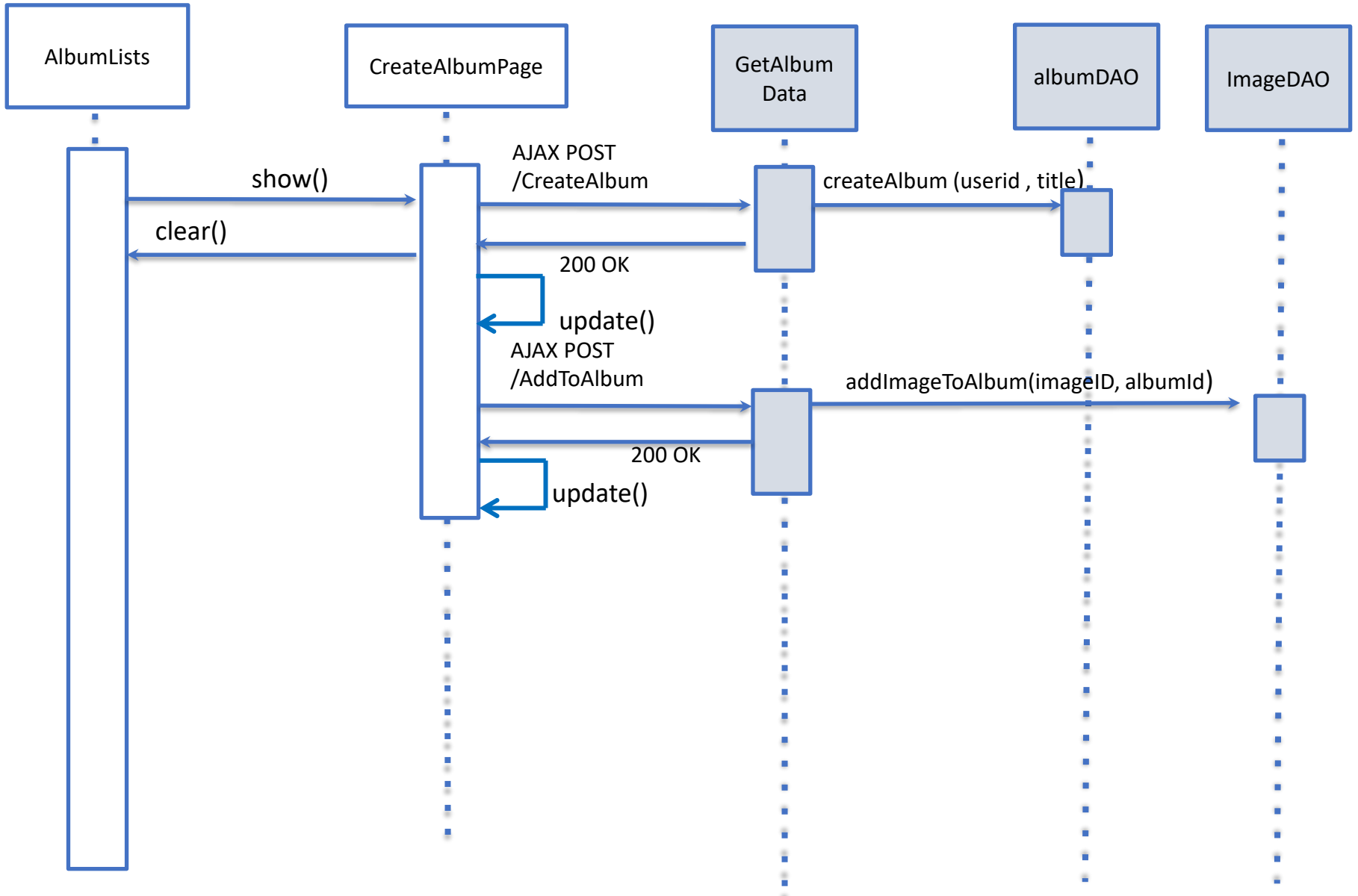
Evento: click su open Album e next page



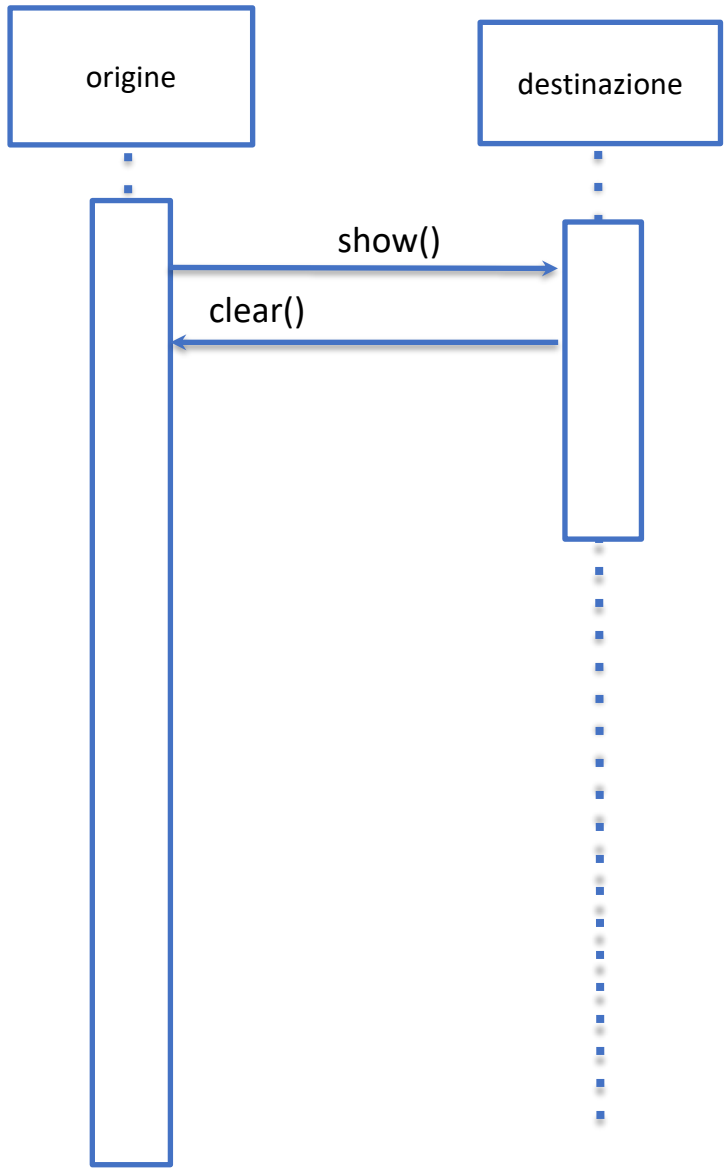
Eventi: click su create album



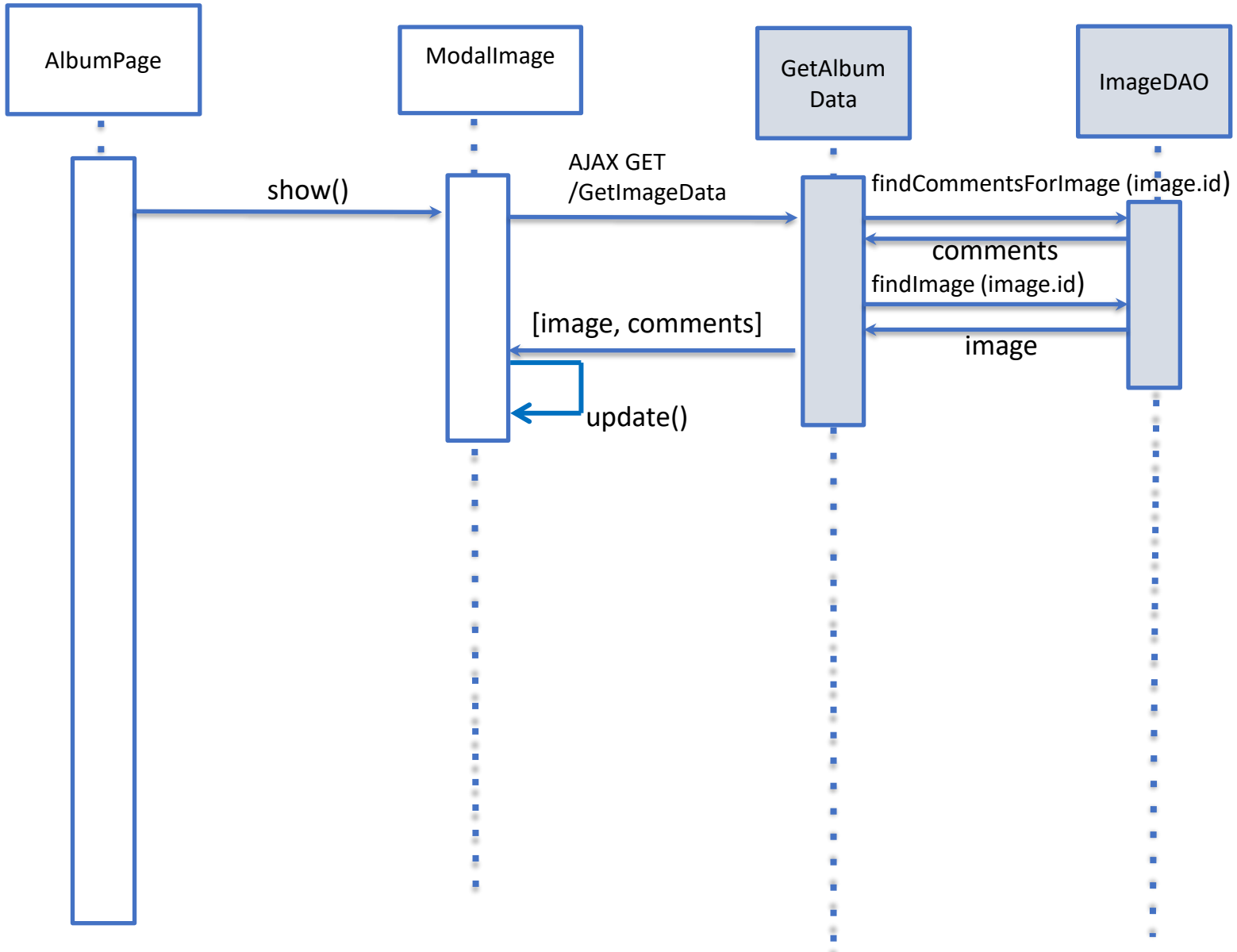
Eventi: click su create e add to album



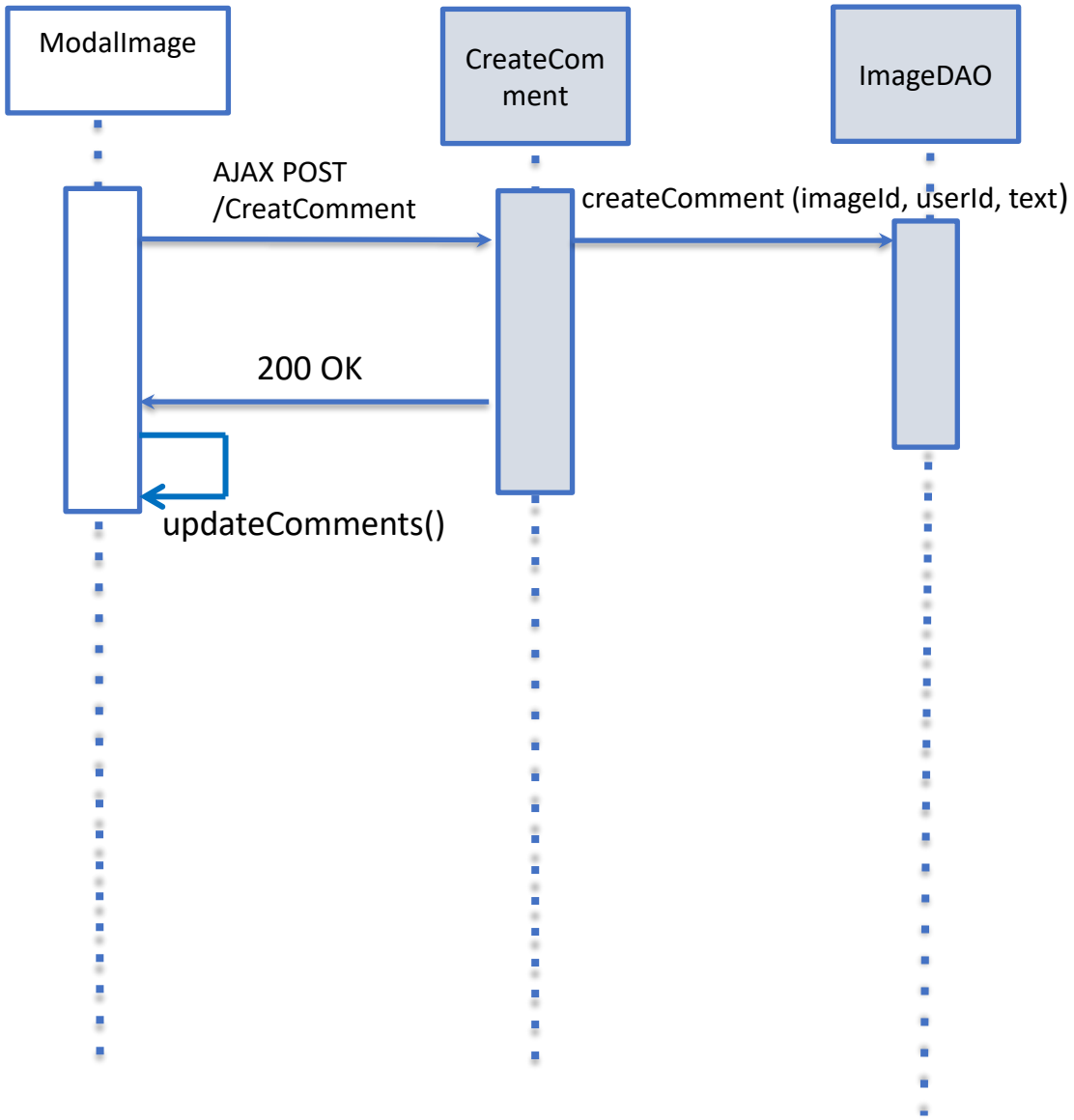
Evento: click su back



Evento: hover su immagine



Evento: creazione commento



Evento: Logout