



# **PROGETTO TECNOLOGIE INFORMATICHE PER IL WEB**

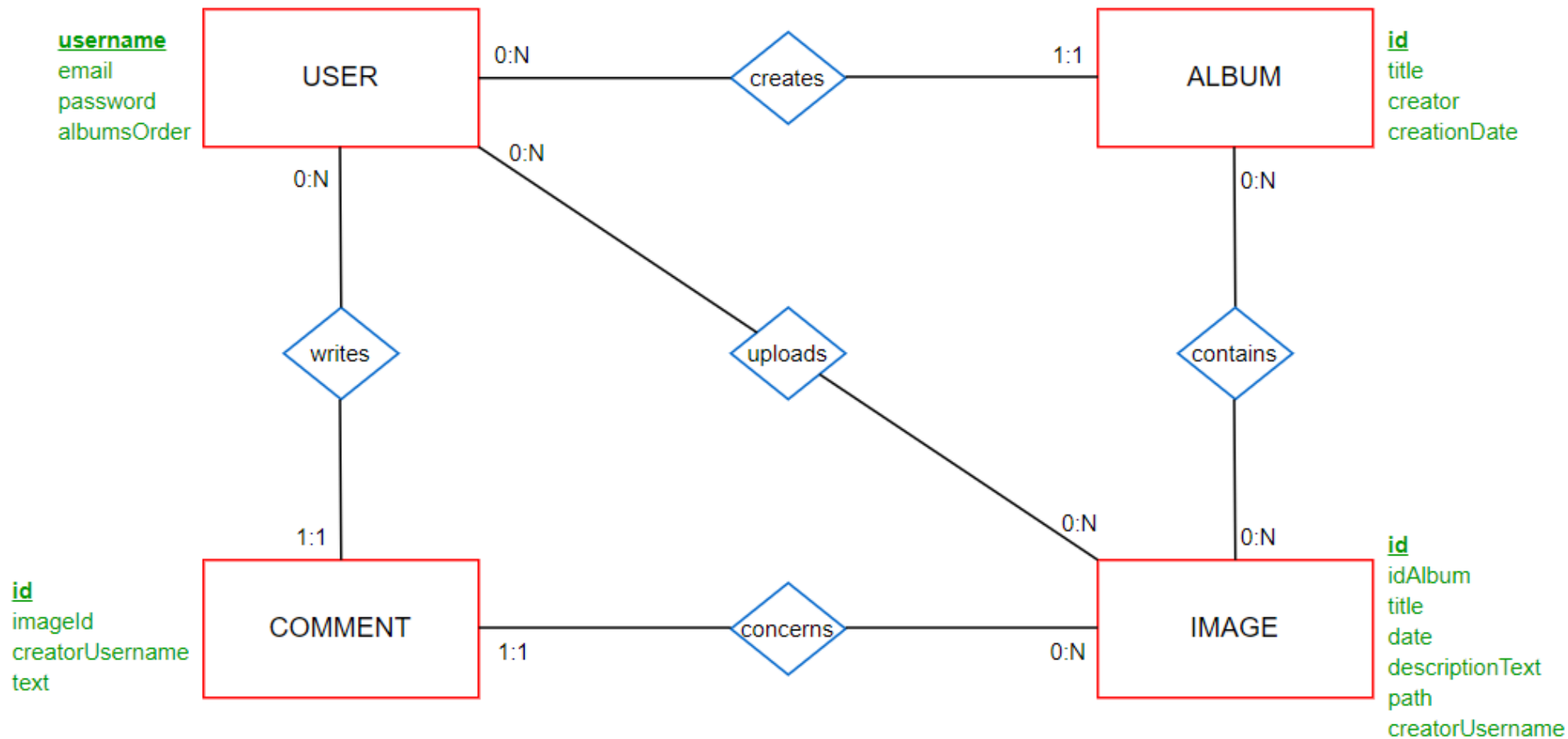
Galleria di Immagini – Pure HTML  
Federica Di Filippo - 10666561

# DATA ANALYSIS

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.

**Entities**, **attributes**, **relationships**

# DATABASE DESIGN



# LOCAL DATABASE SCHEMA

```
CREATE TABLE `album` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `title` varchar(45) NOT NULL,  
  `creator` varchar(45) NOT NULL,  
  `creationDate` date NOT NULL, PRIMARY KEY (`id`))
```

```
CREATE TABLE `image` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `idAlbum` int NOT NULL,  
  `title` varchar(45) NOT NULL,  
  `date` date NOT NULL,  
  `descriptionText` varchar(100) NOT NULL,  
  `path` varchar(200) NOT NULL,  
  `creatorUsername` varchar(45) NOT NULL, PRIMARY KEY (`id`))
```

```
CREATE TABLE `comment` (  
  `id` int NOT NULL AUTO_INCREMENT, `imageId` int  
  NOT NULL,  
  `creatorUsername` varchar(45) NOT NULL, `text`  
  varchar(400) NOT NULL, PRIMARY KEY (`id`))
```

```
CREATE TABLE `user` (  
  `username` varchar(45) NOT NULL,  
  `email` varchar(45) NOT NULL,  
  `password` varchar(45) NOT NULL,  
  `albumsOrder` varchar(45) DEFAULT NULL, PRIMARY KEY  
  (`username`))
```

# APPLICATION REQUIREMENTS ANALYSIS

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.

**Pages (views), view components, events, actions**



# APPLICATION REQUIREMENTS ANALYSIS

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** rappresenta 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.

# SPECIFICATIONS INTEGRATION

- L'utente è identificato dal suo username (unico);
- La **pagina di default** contiene la **form di login**;
- Se l'utente è già registrato ed **effettua il login**, accede alla HomePage, altrimenti viene reindirizzato alla pagina di login;
- Dalla pagina di login è possibile **accedere** ad una **pagina di registrazione**;
- Tutti gli errori di compilazione dei form di login/registrazione vengono mostrati all'interno del form;
- Se si verificano errori/comportamenti anomali, l'utente è reindirizzato ad una **pagina di errore** in cui è possibile visualizzare il problema.

# COMPONENTS

- Model objects (Beans)

- User
- Album
- Image
- Comment

- Data Access Objects (Classes)

- UserDAO
- AlbumDAO
- ImageDAO
- CommentDAO

- Controllers

- AddComment
- AddImageToAlbum
- CheckLogin
- CreateAlbum
- GetAlbumPage
- GetHomePage
- GetLoginPage
- GetSignUpPage
- Logout
- SignUp

- Views

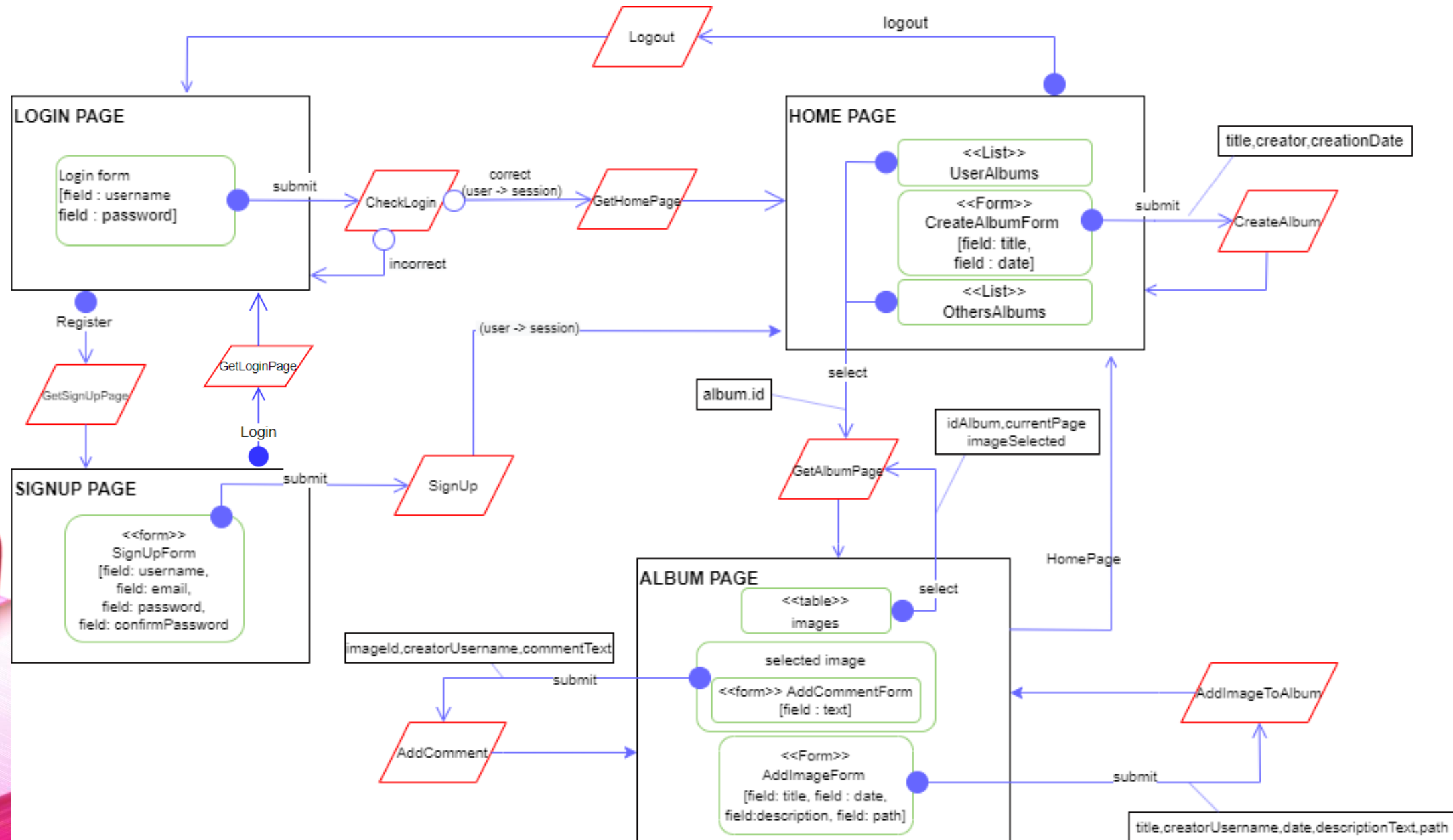
- index.html
- HomePage.html
- SignUpPage.html
- AlbumPage.html

- Utils

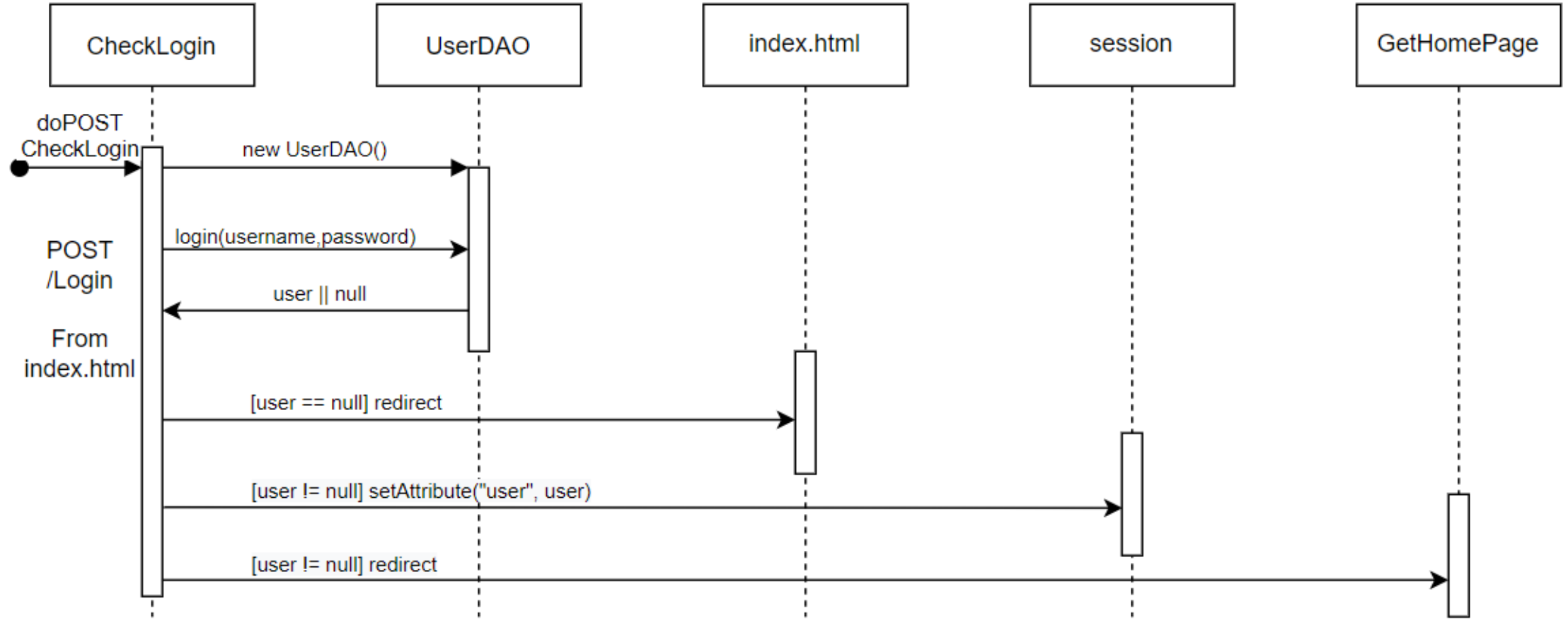
- ConnectionHandler



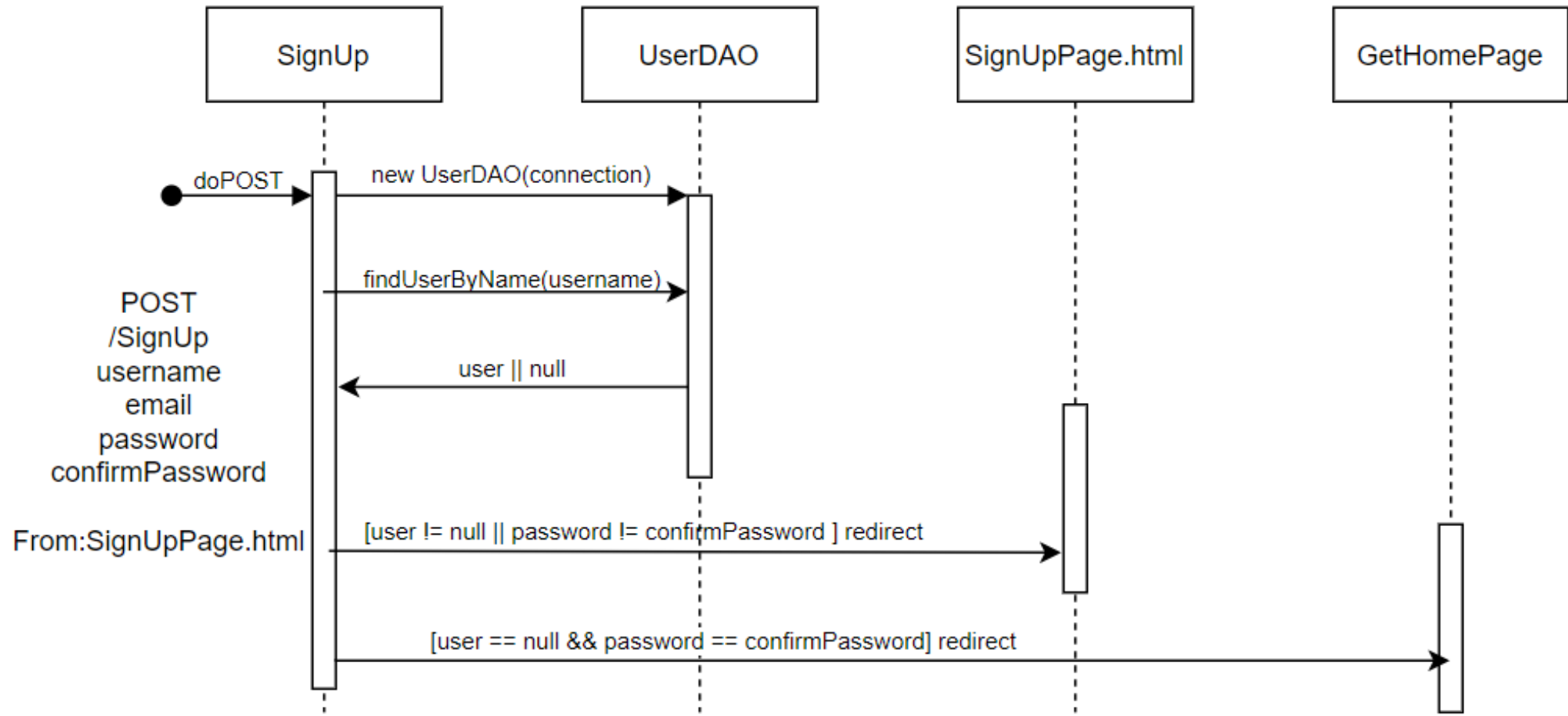
# APPLICATION DESIGN



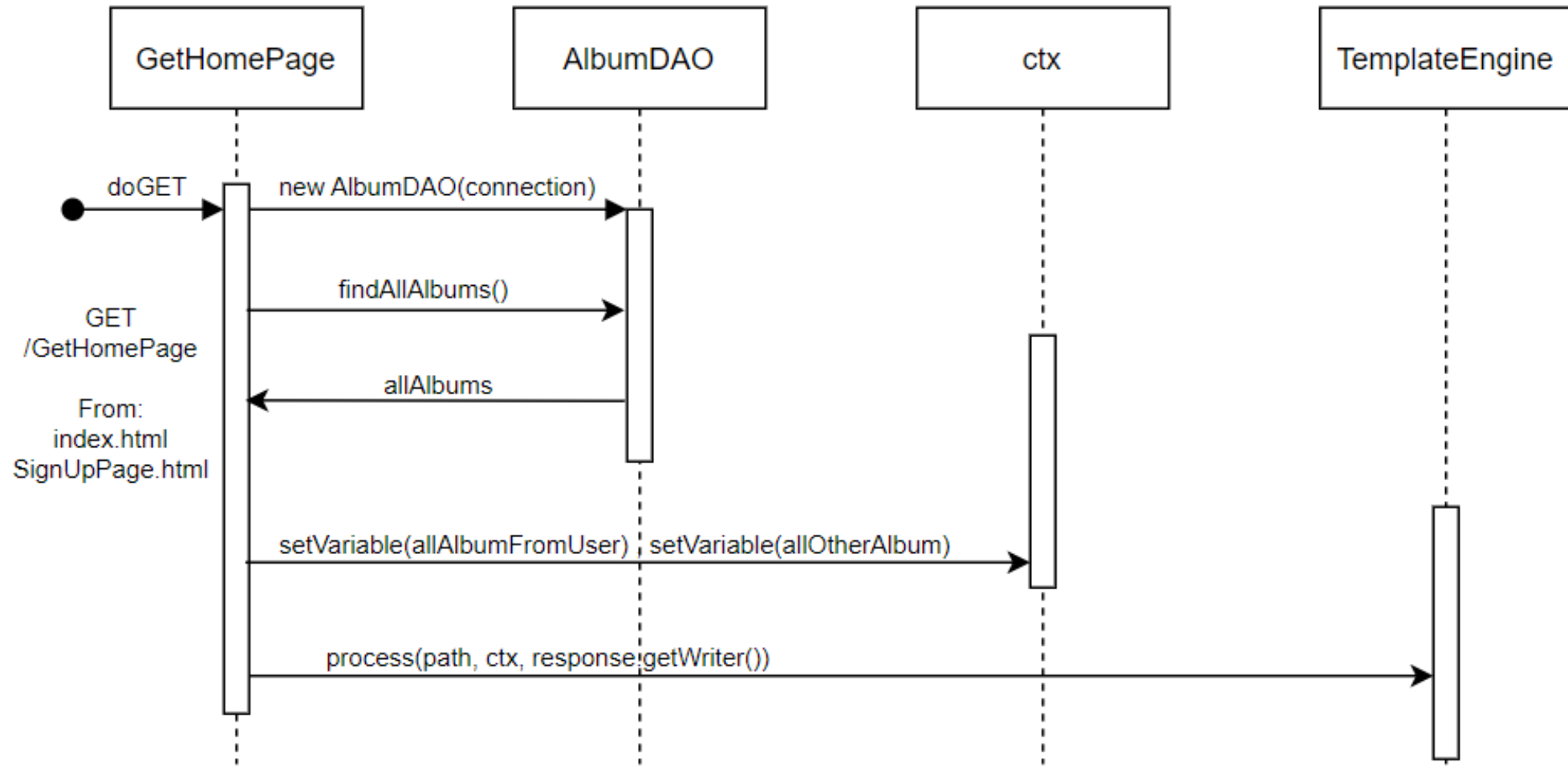
# EVENT: LOGIN



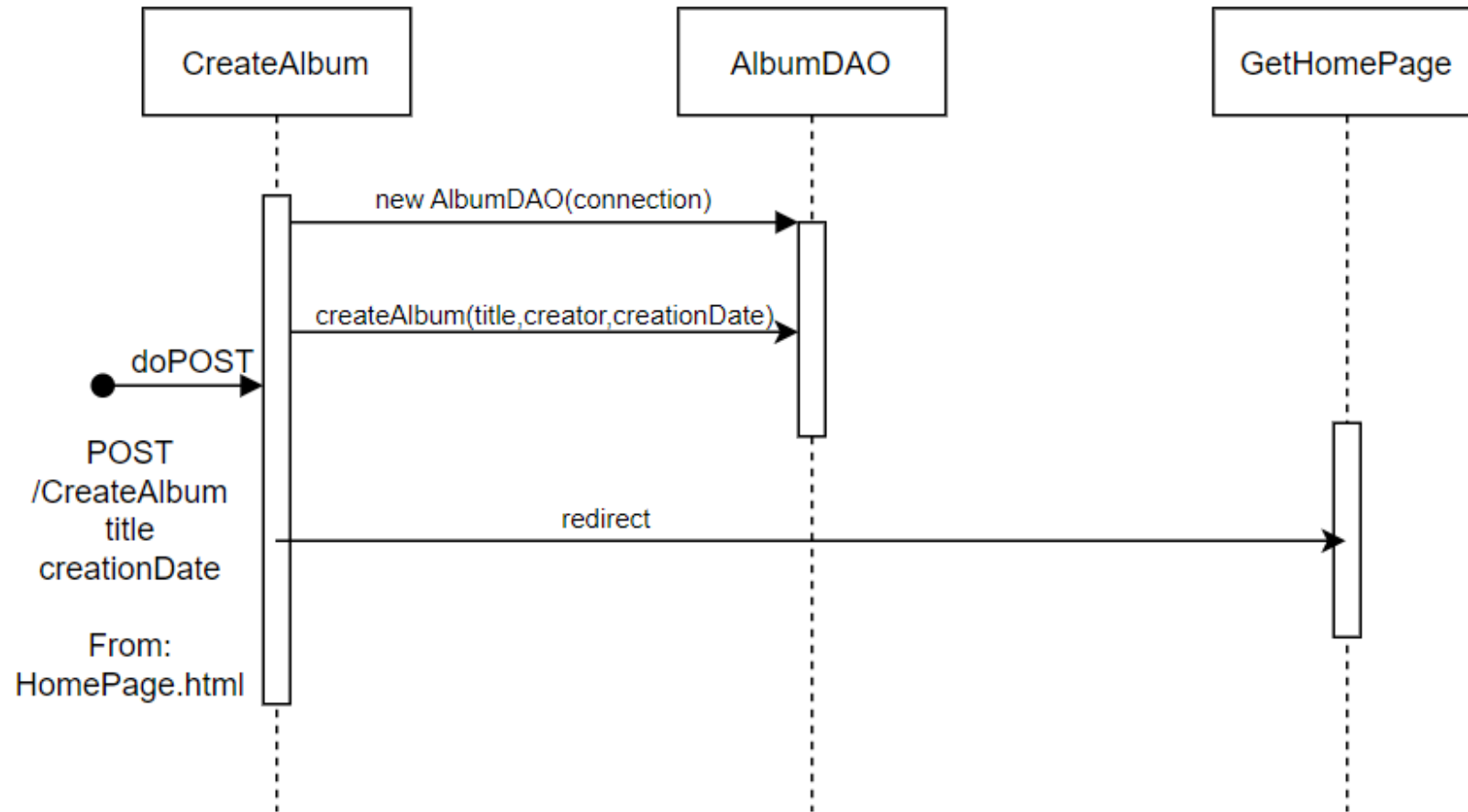
# EVENT: SIGNUP



# EVENT: GETHOMEPAGE

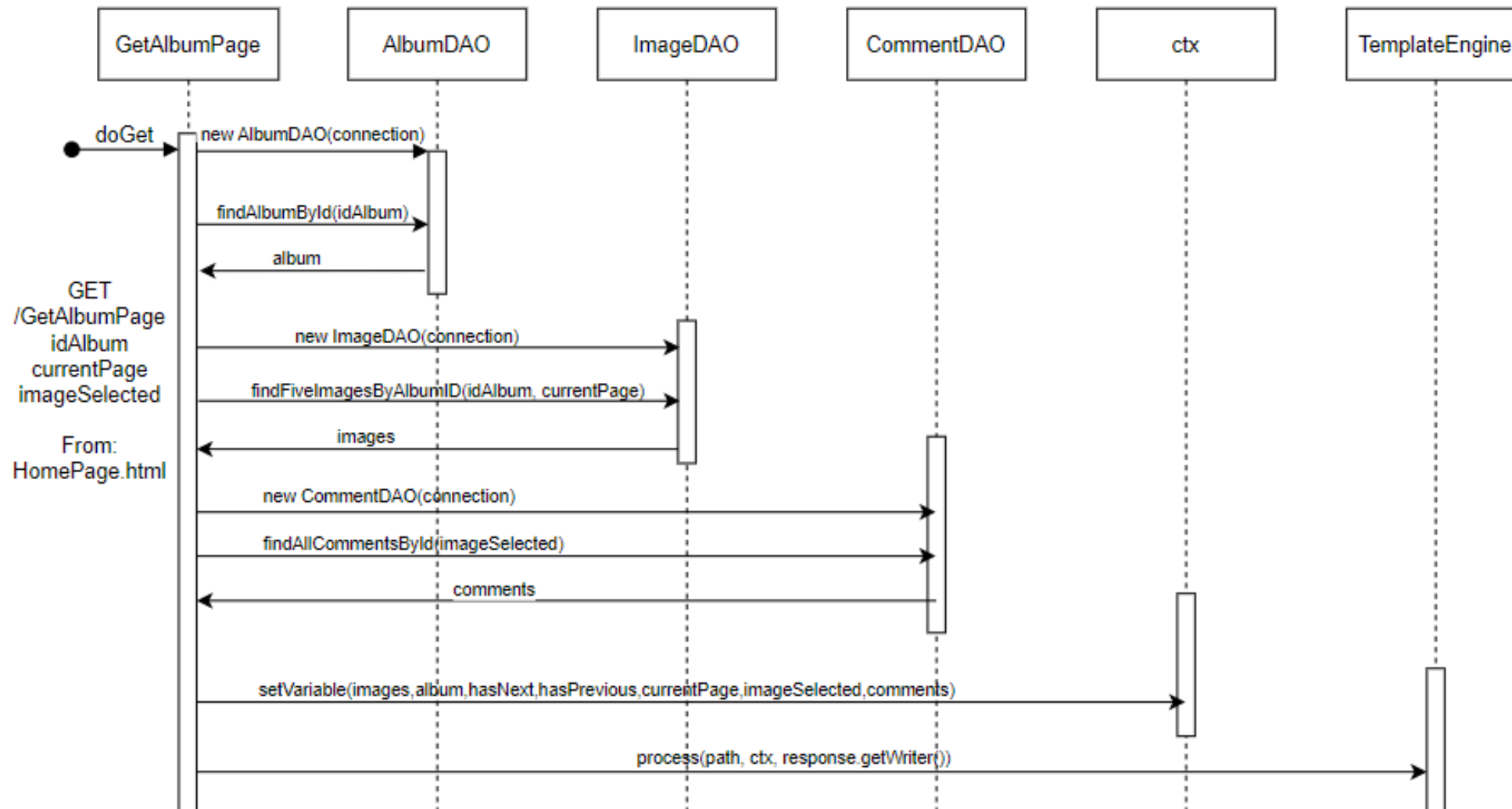


# EVENT: CREATE ALBUM

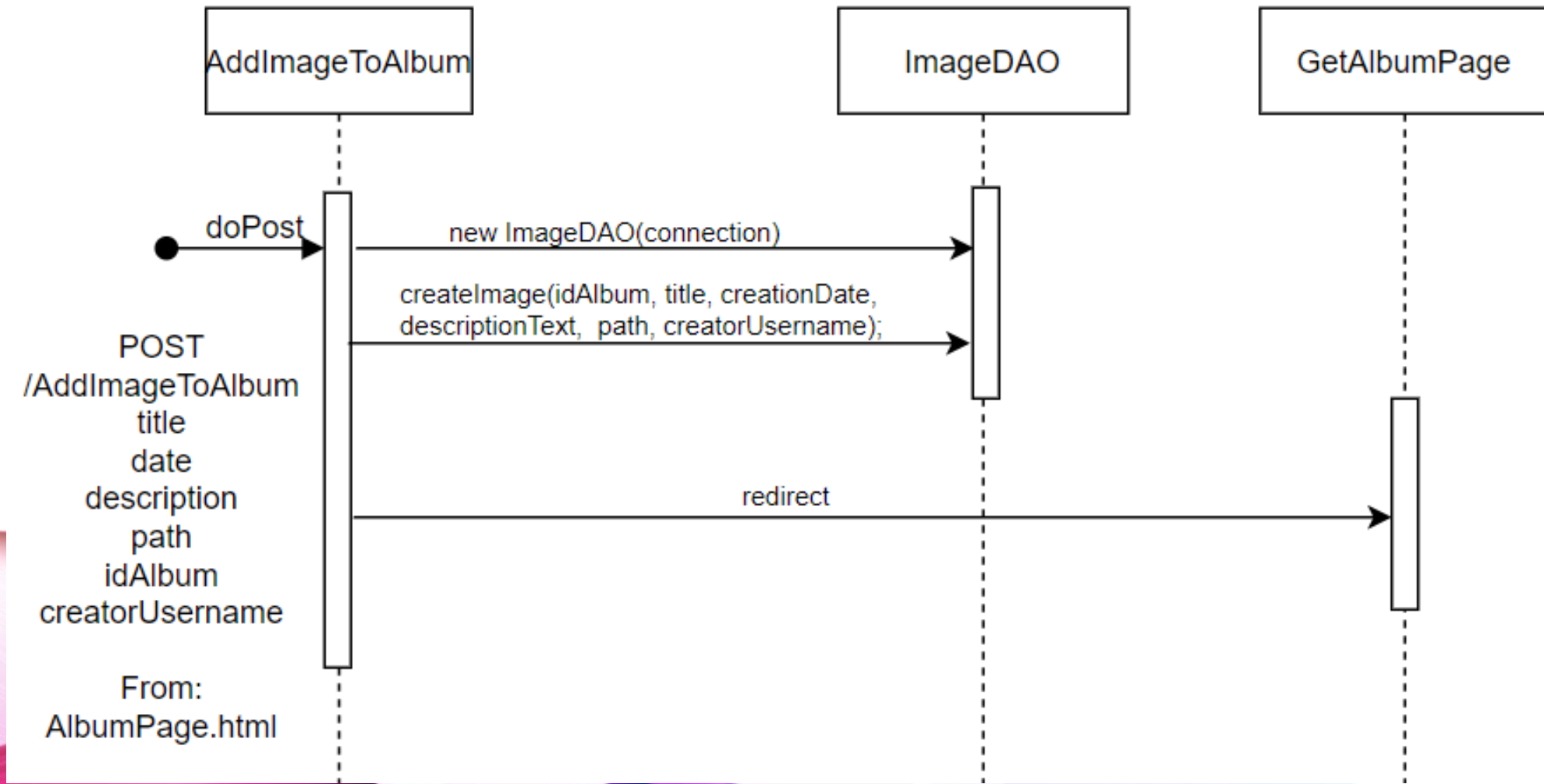




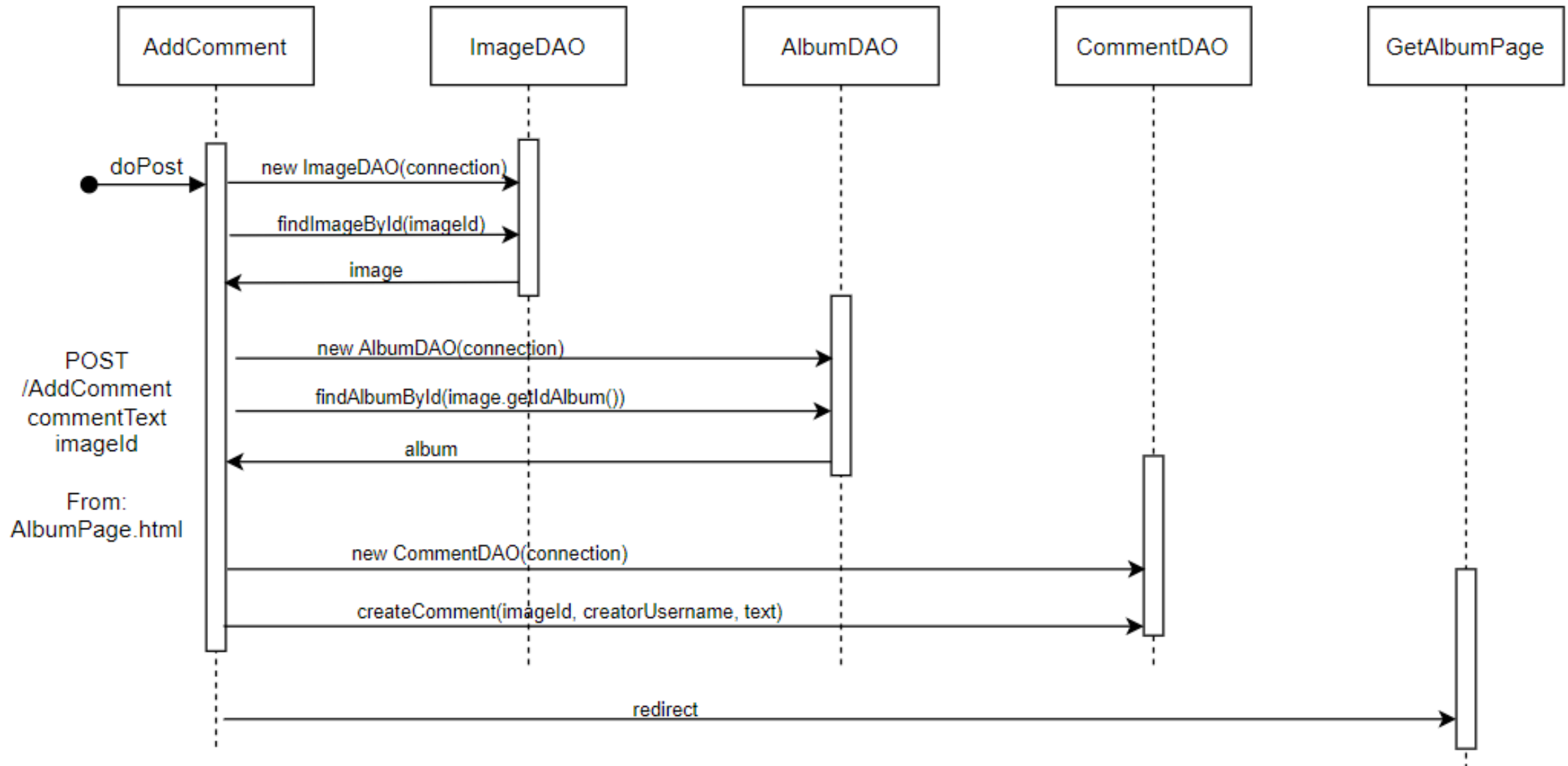
# EVENT: GETALBUMPAGE



# EVENT: ADD IMAGE TO ALBUM



# EVENT: ADD COMMENT



# EVENT: LOGOUT

