

Documentazione gruppo 164

Buccheri Giuseppe, Canepari Michele

Analisi dei dati

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.

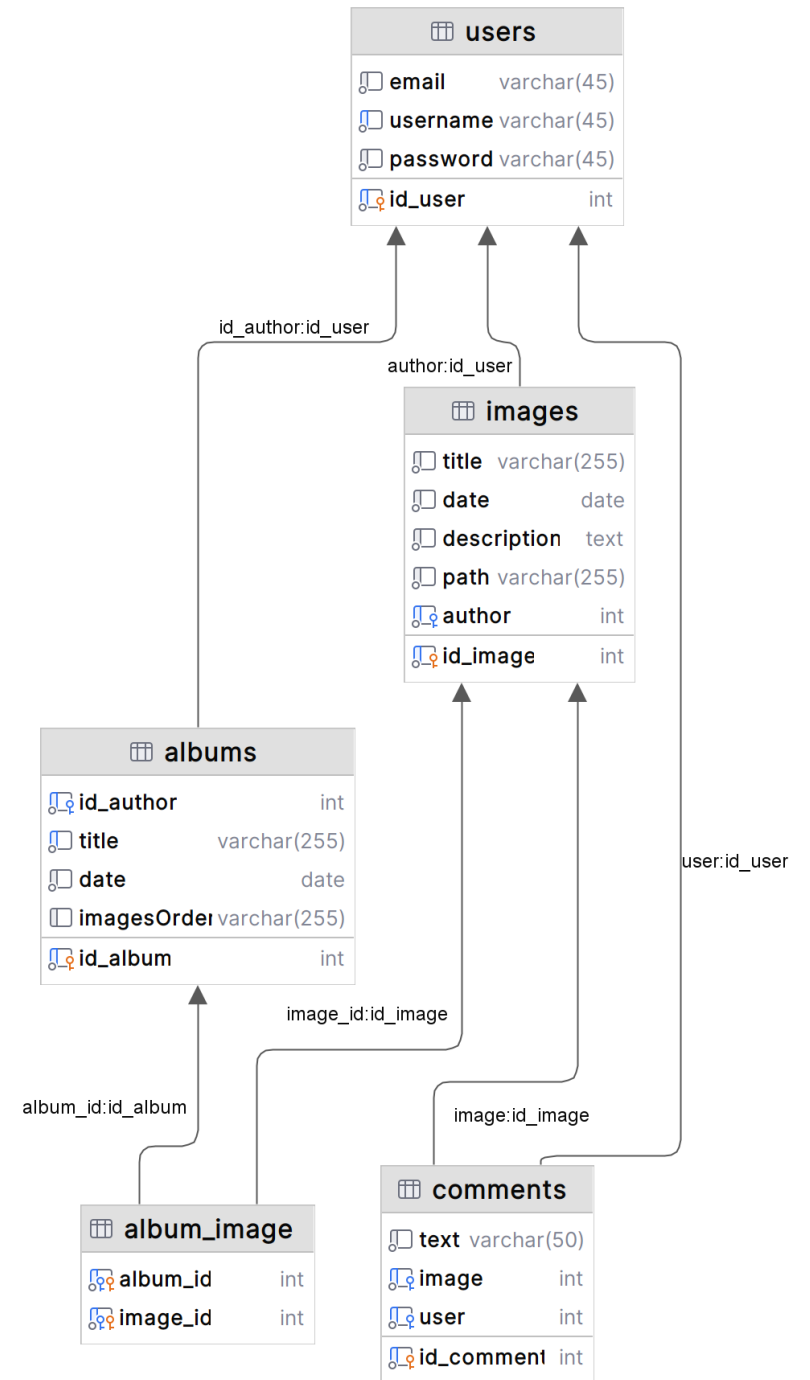
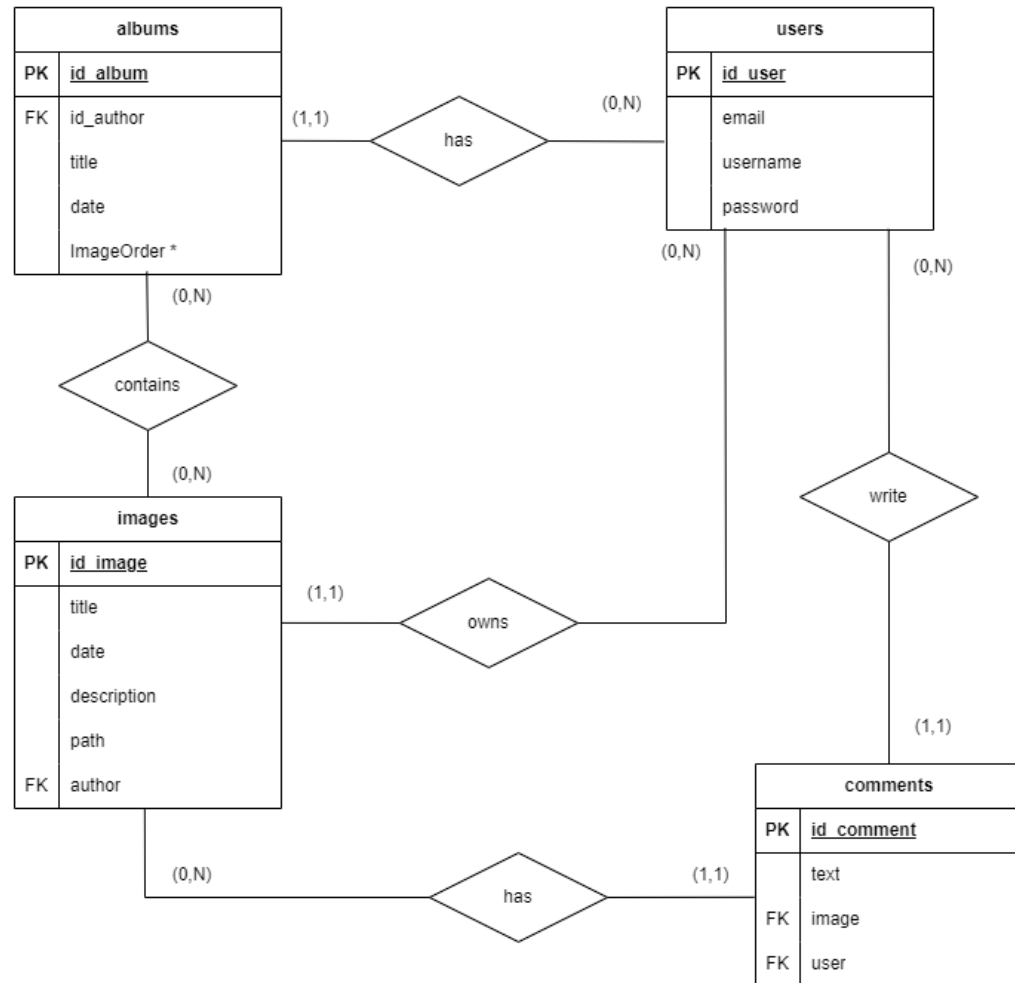
L'utente può creare album dalla HOME PAGE e associare a questi le proprie immagini. Un album ha un titolo, il creatore e la data di creazione. La stessa immagine può appartenere a più di un album. Le immagini sono associate a uno o più commenti inseriti dagli utenti (dal proprietario o da altri utenti). 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.

SCHEMA ER



Database table

```
CREATE TABLE `albums` (  
  `id_album` int NOT NULL AUTO_INCREMENT,  
  `id_author` int NOT NULL,  
  `title` varchar(255) NOT NULL,  
  `date` date NOT NULL,  
  PRIMARY KEY (`id_album`),  
  UNIQUE KEY `albums_pk` (`title`),  
  KEY `id_author` (`id_author`),  
  CONSTRAINT `albums_ibfk_1` FOREIGN KEY (`id_author`) REFERENCES `users` (`id_user`))  
  
CREATE TABLE `comments` (  
  `id_comment` int NOT NULL AUTO_INCREMENT,  
  `text` varchar(50) NOT NULL,  
  `image` int NOT NULL,  
  `user` int NOT NULL,  
  PRIMARY KEY (`id_comment`),  
  KEY `image` (`image`),  
  KEY `comments_users_id_user_fk` (`user`),  
  CONSTRAINT `comments_ibfk_1` FOREIGN KEY (`image`) REFERENCES `images` (`id_image`),  
  CONSTRAINT `comments_users_id_user_fk` FOREIGN KEY (`user`) REFERENCES `users` (`id_user`) ON UPDATE CASCADE)  
  
CREATE TABLE `images` (  
  `id_image` int NOT NULL AUTO_INCREMENT,  
  `title` varchar(255) NOT NULL,  
  `date` date NOT NULL,  
  `description` text NOT NULL,  
  `path` varchar(255) NOT NULL,  
  `author` int NOT NULL,  
  PRIMARY KEY (`id_image`),  
  KEY `id_author` (`author`),  
  CONSTRAINT `images_ibfk_1` FOREIGN KEY (`author`) REFERENCES `users` (`id_user`) ON DELETE CASCADE ON UPDATE CASCADE)
```

Albums(id_album, id_author, title, date, imageOrder*)

Album_image(id_album, id_image)

Image(id_image, title, description, path, author)

Users(id_user, email, username, password)

Comments(id_comment, text, image, user)

```
CREATE TABLE `album_image` (  
  `album_id` int NOT NULL,  
  `image_id` int NOT NULL,  
  PRIMARY KEY (`album_id`, `image_id`),  
  KEY `image_id` (`image_id`),  
  CONSTRAINT `album_image_ibfk_1` FOREIGN KEY (`album_id`) REFERENCES `albums` (`id_album`) ON DELETE CASCADE,  
  CONSTRAINT `album_image_ibfk_2` FOREIGN KEY (`image_id`) REFERENCES `images` (`id_image`) ON DELETE CASCADE)
```

```
CREATE TABLE `users` (  
  `id_user` int NOT NULL AUTO_INCREMENT,  
  `email` varchar(45) NOT NULL,  
  `username` varchar(45) NOT NULL,  
  `password` varchar(45) NOT NULL,  
  PRIMARY KEY (`id_user`),  
  UNIQUE KEY `username_UNIQUE` (`username`))
```

Analisi dei dati

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**.

COMPONENTI

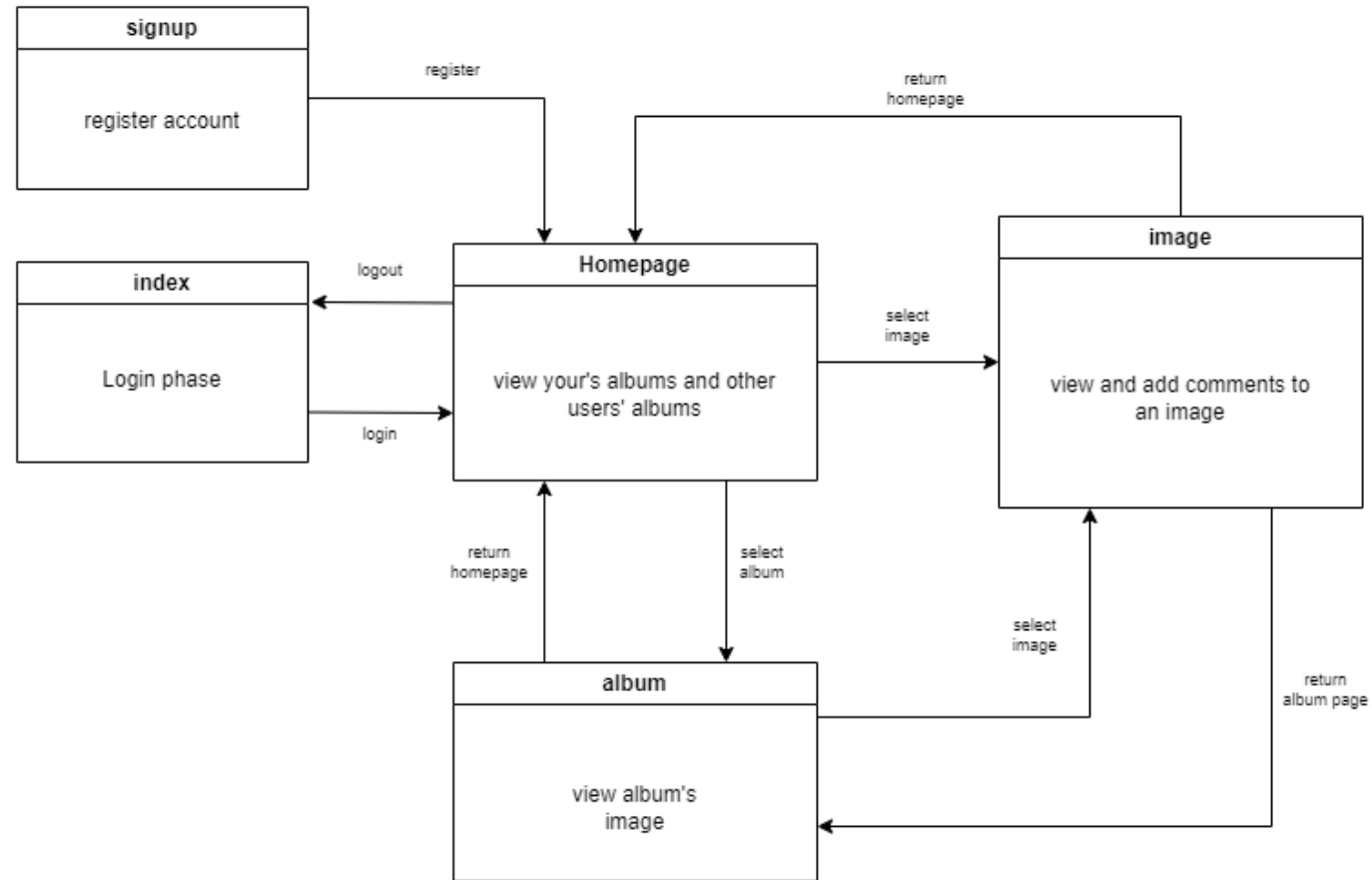
- Controllers
 - AddComment
 - DeleteImage
 - CreateAlbum
 - GetAlbum
 - GetImage
 - GoToHome
 - GetAllAlbums
 - CheckLogins
 - Logout
 - Signup
 - SaveAlbumsOrder
- Beans
 - Album
 - Comment
 - Image
 - User
- Filters
 - LoggedUser

RIA only

HTML pure only

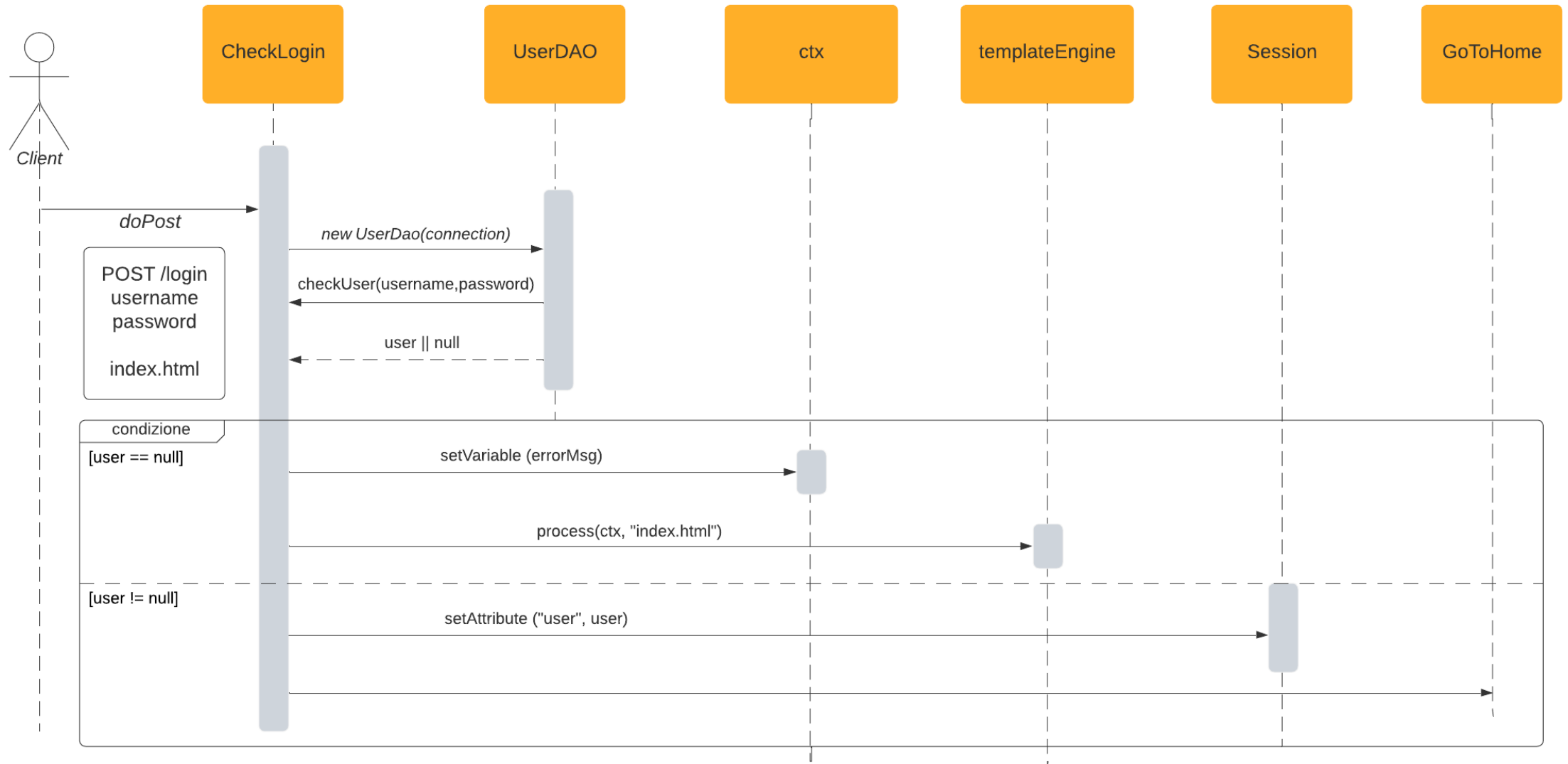
- DAO
 - AlbumDAO
 - findAlbumImages(int id,int offset)
 - getAlbumById(int id)
 - getAuthorUsernameById(int id)
 - createAlbum(int id_author, String title, Date date)
 - addImageToAlbum(int image_id, int album_id)
 - addAlbumOrder(String order, int id)
 - countAlbumImages(int albumId)
 - CommentDAO
 - findCommentsByImage(int id)
 - addComment(int imageId, int userId, String text)
 - deleteImagesComments(int imageId)
 - getCommentById(int id)
 - ImageDAO
 - getImageById(int id)
 - getAuthorUsernameById(int id)
 - deleteImage(int imageId)
 - UserDao
 - checkUser(String username, String password)
 - findUserImages(int id)
 - findOthersAlbums(int id)
 - findUserAlbums(int id)
 - insertUser(String username, String password, String email)
 - isNew(String username)
 - getUsername(int id)

Application Design - HTML

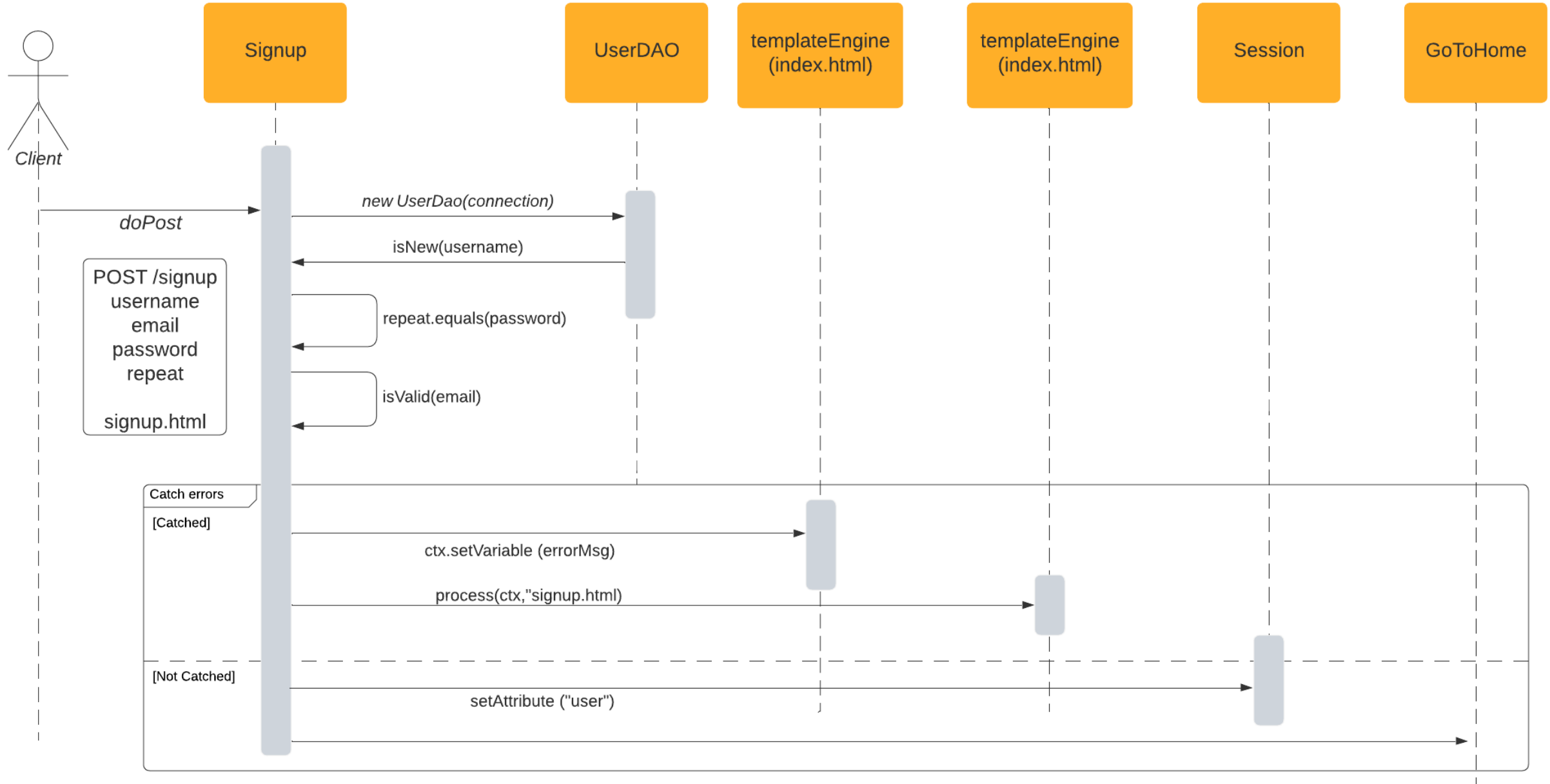


SEQUENCE DIAGRAM

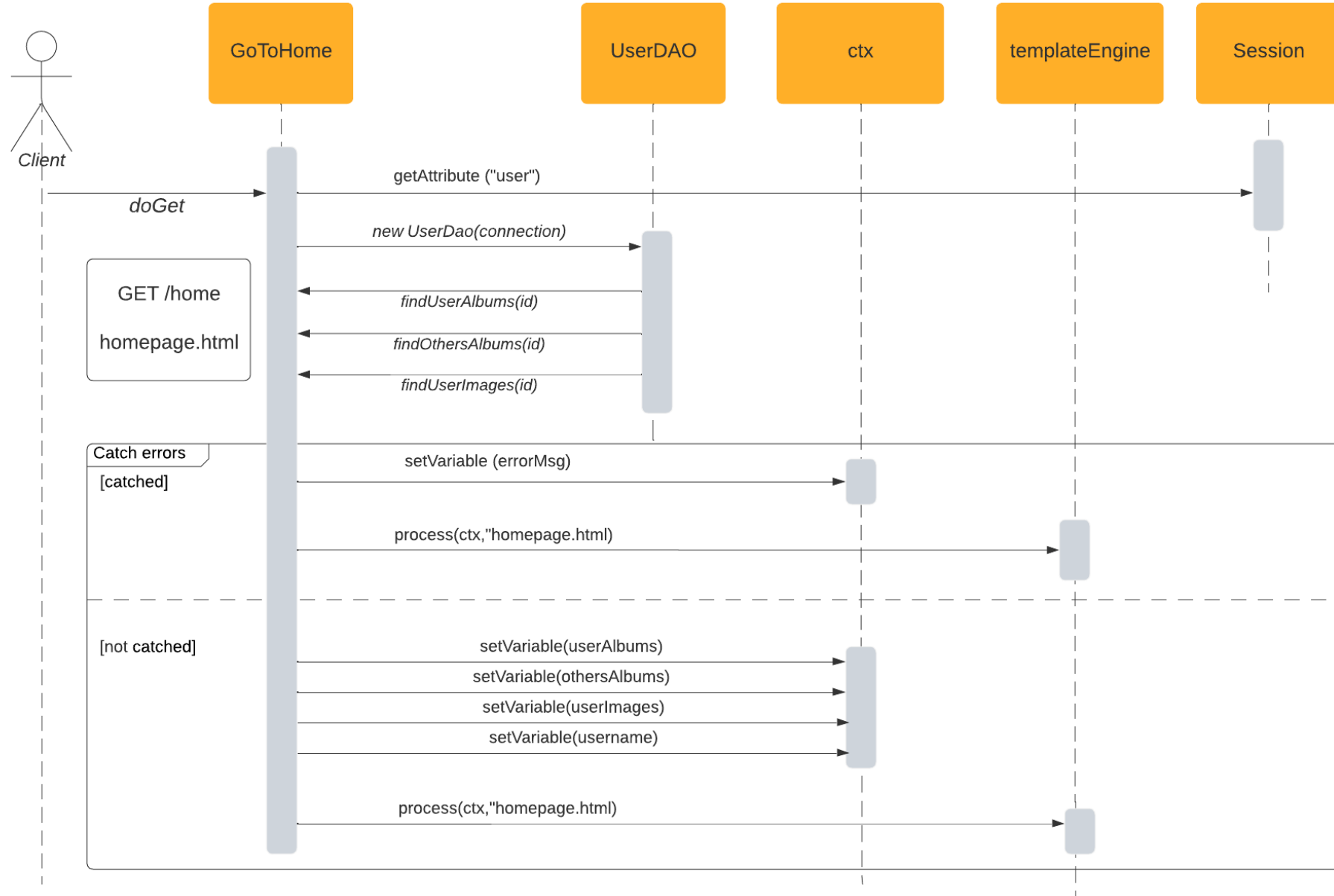
EVENT LOGIN



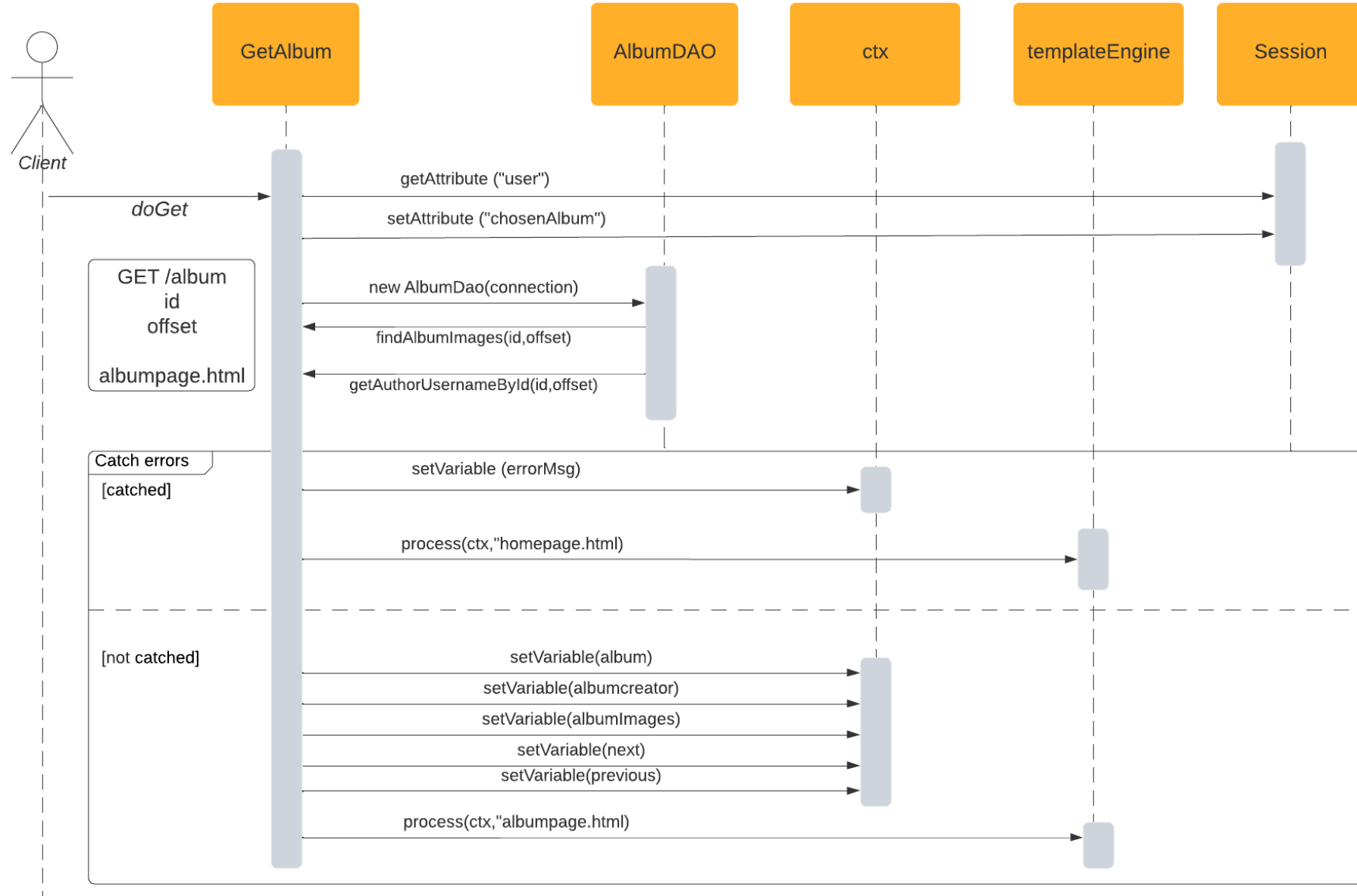
EVENT SIGNUP



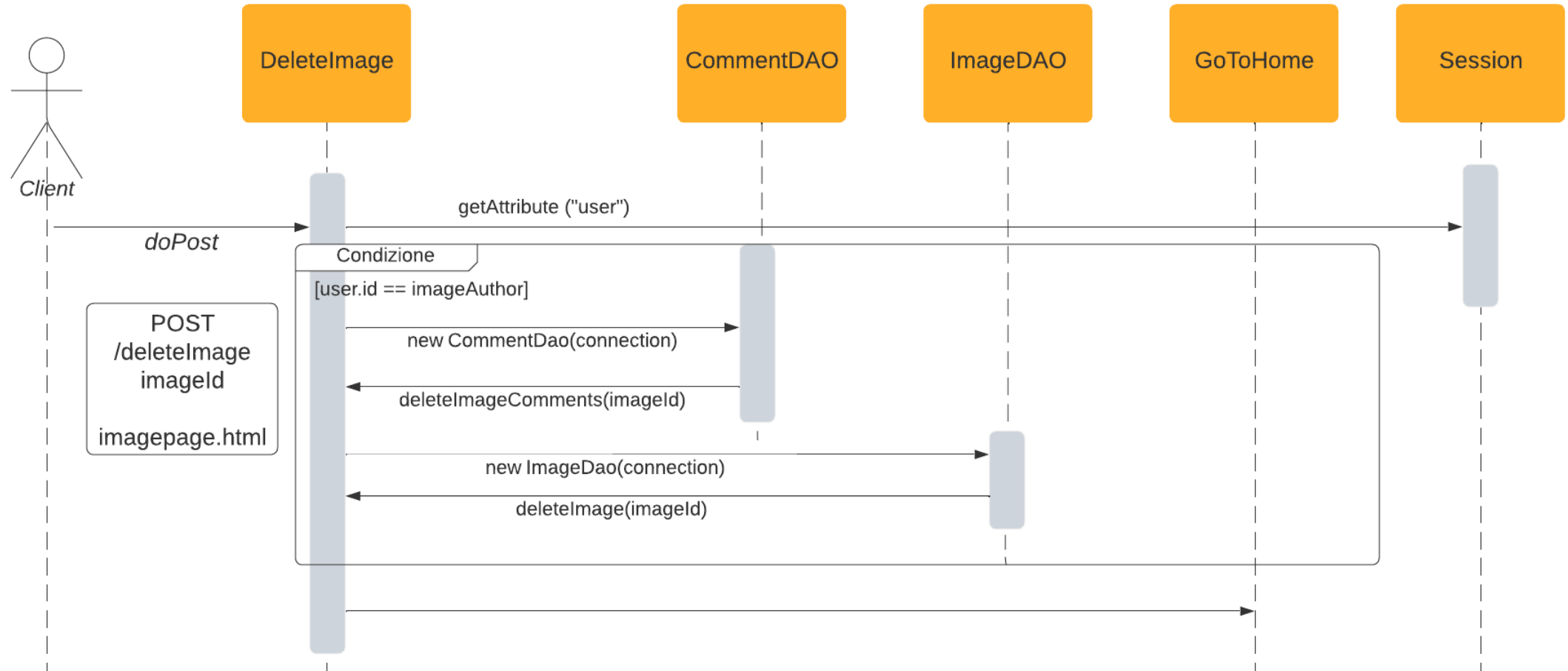
EVENT HOME



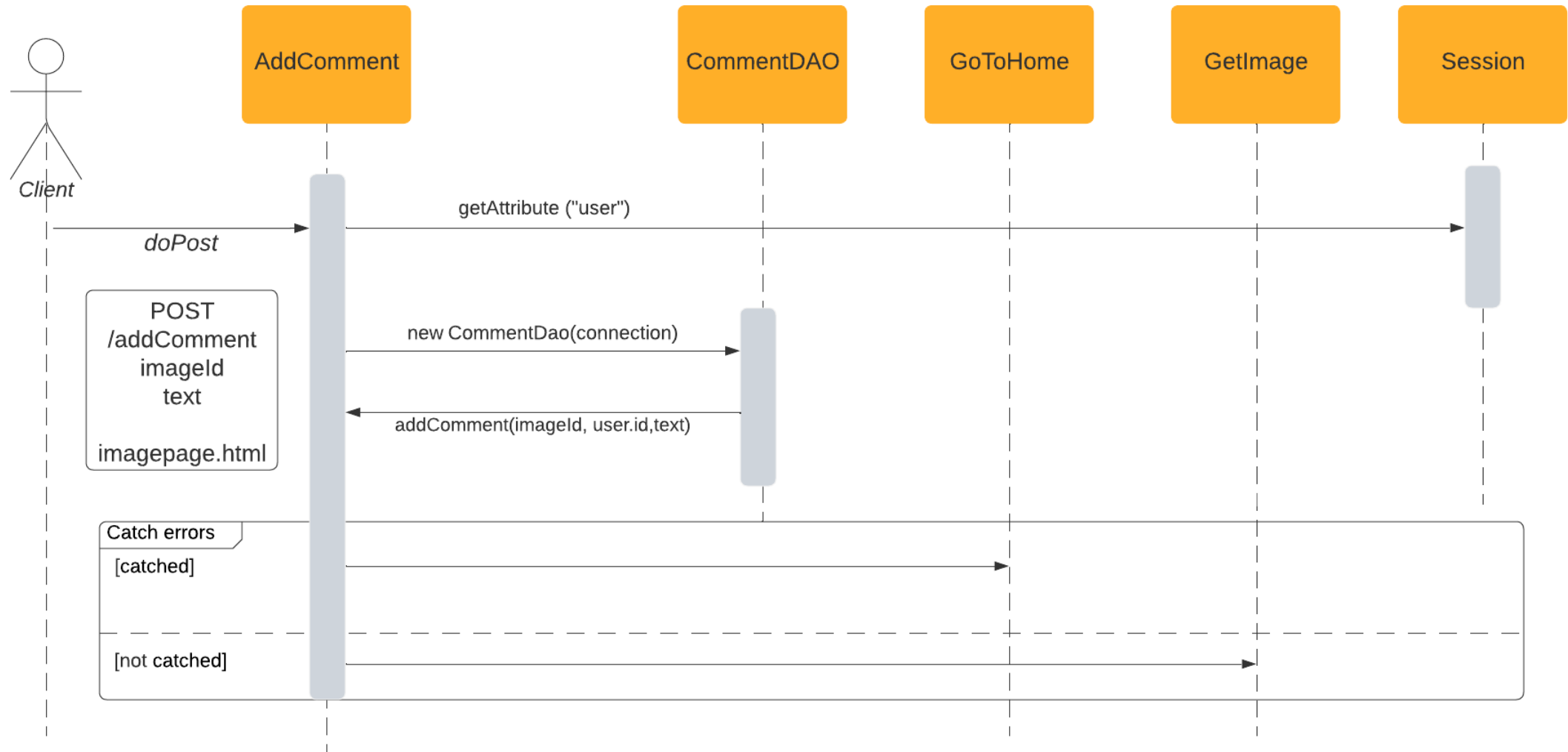
EVENT GETALBUM



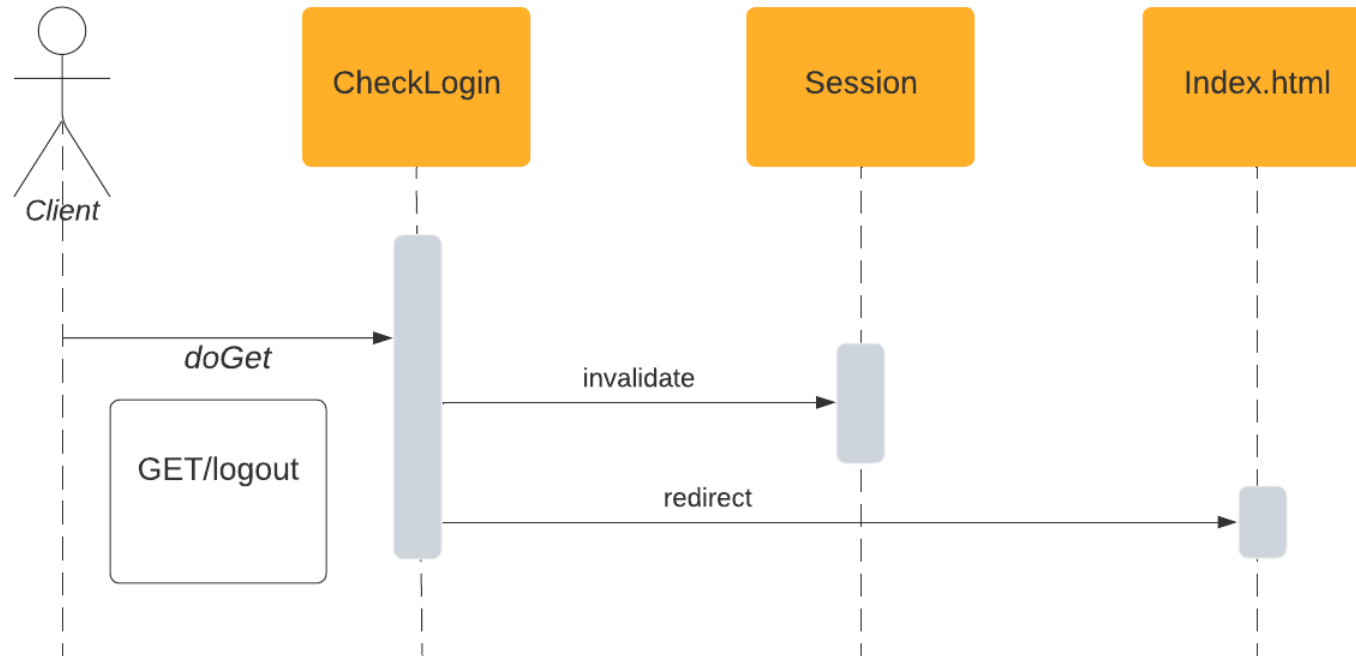
EVENT DELETEIMAGE



EVENT ADDCOMMENT

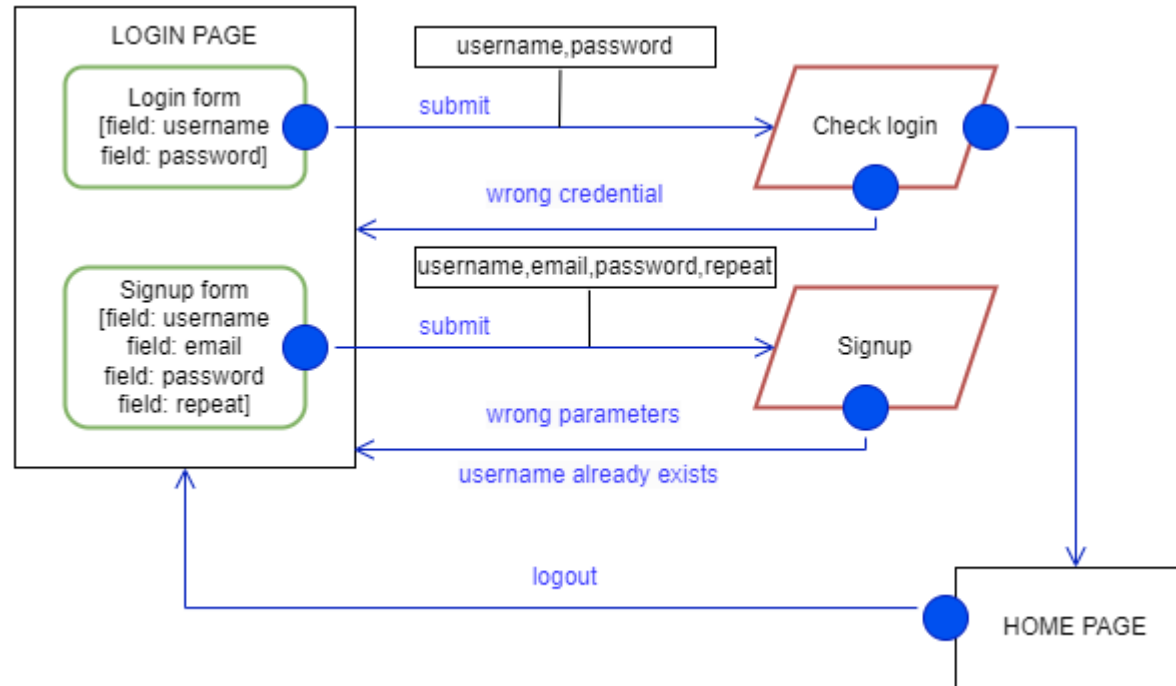


EVENT LOGOUT



Application Design - RIA

LOGIN PAGE



[HOME PAGE](#)

