

Gestione di immagini

HTML

Gestione di immagini

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.

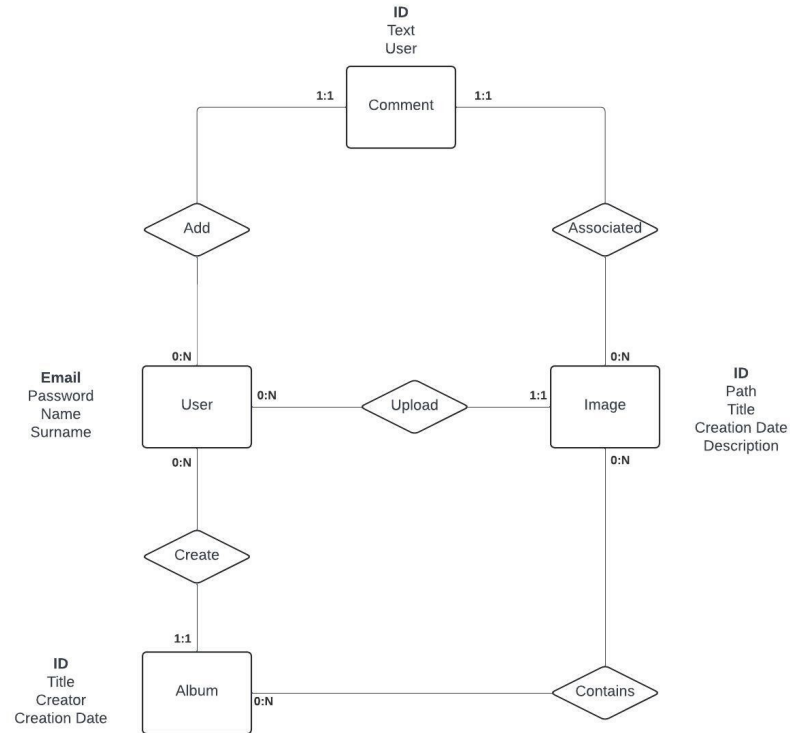
Se il numero di immagini non è un multiplo di 5 la tabella deve avere sempre 5 celle, lasciando quelle più a destra vuote. Le miniature sono ordinate da sinistra a destra per data decrescente. Se l'album contiene più di cinque immagini, sono disponibili comandi per vedere il precedente e successivo insieme 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.

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.

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 DEFAULT  
(curdate()),  
  PRIMARY KEY (`ID`),  
  KEY `creatore_idx` (`Creator`),  
  CONSTRAINT `creator` FOREIGN KEY  
(`Creator`) REFERENCES `User` (`Email`) ON  
DELETE RESTRICT ON UPDATE CASCADE  
)
```

```
CREATE TABLE `Comment` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `Text` varchar(255) NOT NULL,  
  `User` varchar(45) NOT NULL,  
  `Image` int NOT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `utente_idx` (`User`),  
  KEY `image_idx` (`Image`),  
  CONSTRAINT `image` FOREIGN KEY  
(`Image`) REFERENCES `Image` (`ID`) ON  
DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `user` FOREIGN KEY  
(`User`) REFERENCES `User` (`Email`) ON  
DELETE CASCADE ON UPDATE CASCADE  
)
```

Local database schema

```
CREATE TABLE `Image` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `Title` varchar(45) NOT NULL,  
  `CreationDate` date NOT NULL DEFAULT  
(curdate()),  
  `Description` varchar(255) DEFAULT  
NULL,  
  `Path` varchar(255) NOT NULL,  
  `User` varchar(45) NOT NULL,  
  PRIMARY KEY (`ID`),  
  KEY `utente_idx` (`User`),  
  CONSTRAINT `utente` FOREIGN KEY  
(`User`) REFERENCES `User` (`Email`) ON  
UPDATE CASCADE  
)
```

```
CREATE TABLE `User` (  
  `Email` varchar(45) NOT NULL,  
  `Password` varchar(45) NOT NULL,  
  `Name` varchar(45) NOT NULL,  
  `Surname` varchar(45) NOT NULL,  
  PRIMARY KEY (`Email`)  
)
```

Local database schema

```
CREATE TABLE `ImageOfAlbum` (  
  `Album` int NOT NULL,  
  `Image` int NOT NULL,  
  KEY `fk_ImageOfAlbum_1_idx` (`Image`),  
  KEY `fk_ImageOfAlbum_2_idx` (`Album`),  
  CONSTRAINT `fk_ImageOfAlbum_1` FOREIGN KEY (`Image`)  
REFERENCES `Image` (`ID`) ON DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `fk_ImageOfAlbum_2` FOREIGN KEY (`Album`)  
REFERENCES `Album` (`ID`) ON UPDATE CASCADE  
)
```


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 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.

Pages (views), view components, events, actions

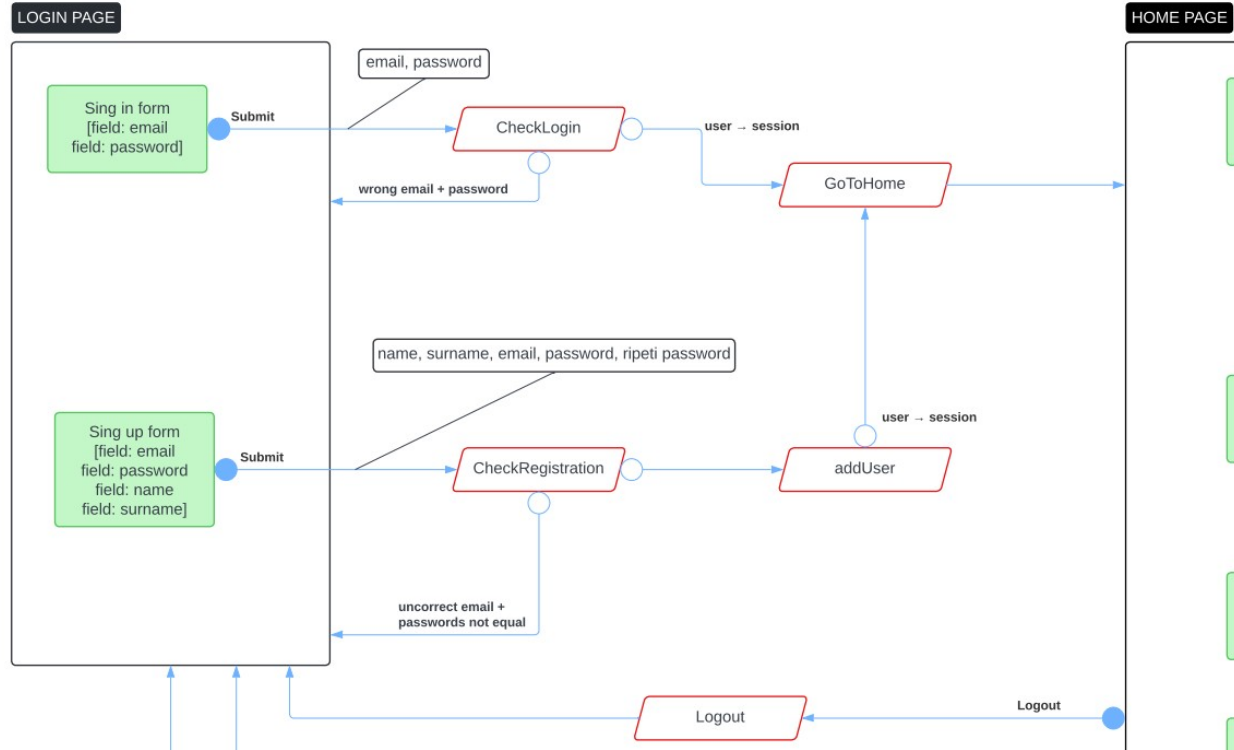
Se il numero di immagini non è un multiplo di 5 la tabella deve avere sempre 5 celle, lasciando quelle più a destra vuote. Le miniature sono ordinate da sinistra a destra per data decrescente. Se l'album contiene più di cinque immagini, sono disponibili comandi per vedere il precedente e successivo insieme 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.

Pages (views), view components, events, actions

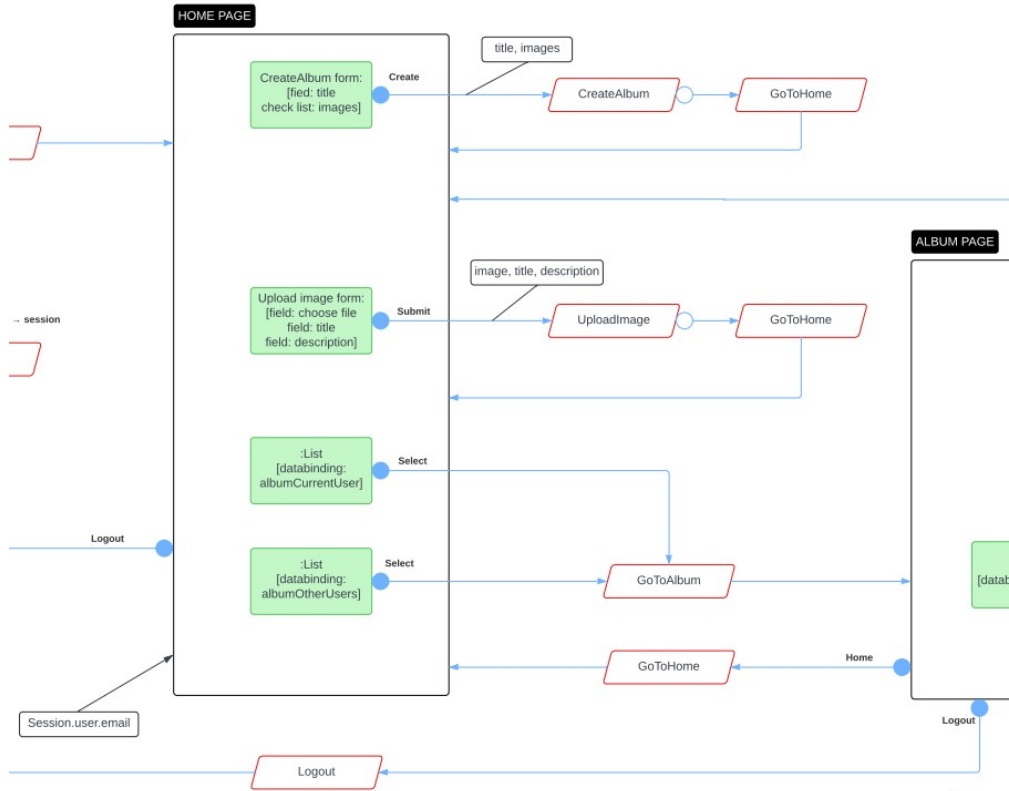
Completamento delle specifiche

- La HOME PAGE contiene una form per il caricamento delle immagini
- Per il caricamento delle immagini è obbligatorio solo il titolo
- Come user name viene usato l'indirizzo e-mail

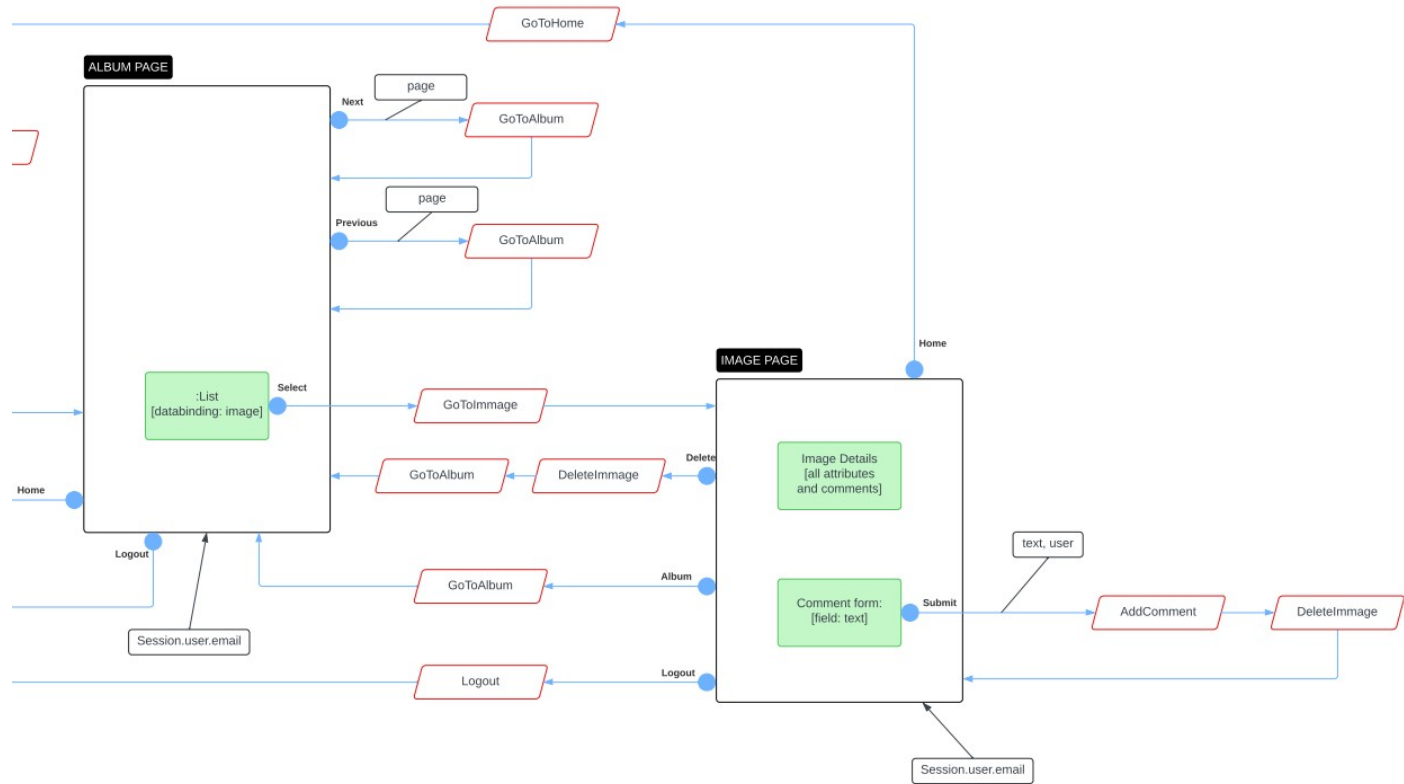
Application design



Application design



Application design



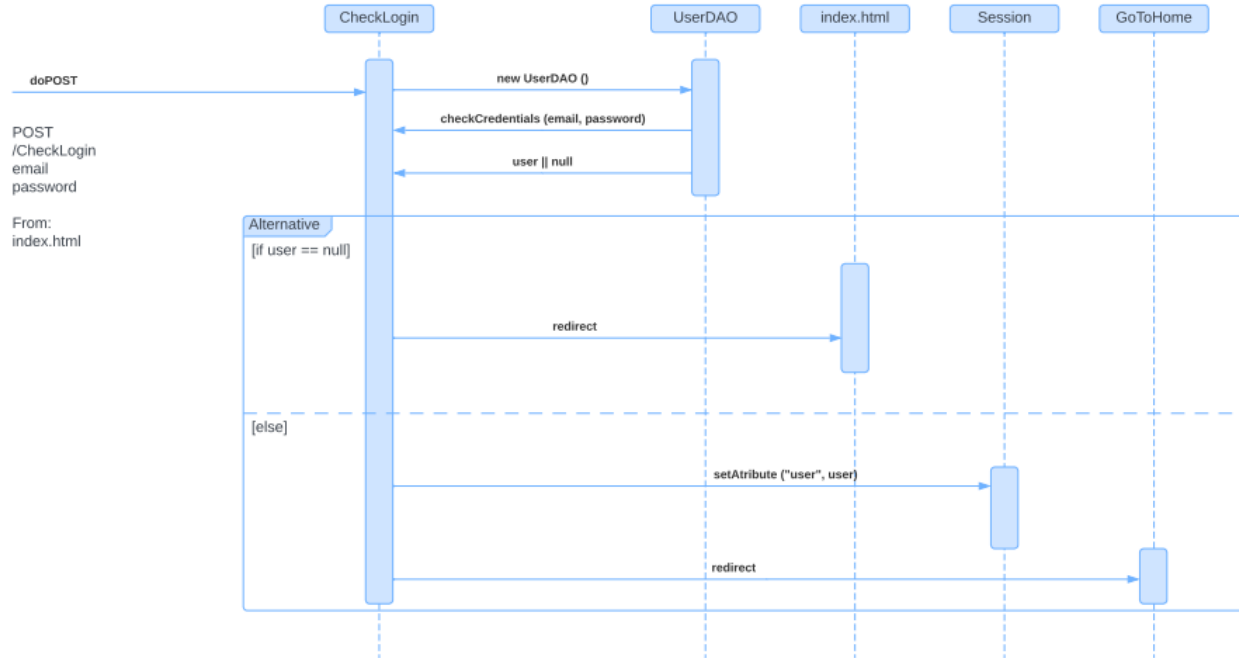
Components

- Model objects (Beans)
 - User
 - Comment
 - Image
 - Album
- Data Access Objects (Classes)
 - UserDao
 - checkCredentials(email, password)
 - addUser(email, password, name, surname)
 - CommentDao
 - addComment(text, email)
 - findedImage(imageId)
 - AlbumDao
 - createAlbum(title, email, images)
 - findOther(email)
 - findByUser(email)
 - findById(albumId)
 - ImageDao
 - addImage(path, title, description, email)
 - findByUser(email)
 - findFiveImages(albumId, page)
 - findByAlbum(albumId)
 - findById(imageId)
 - deleteImage(imageId)

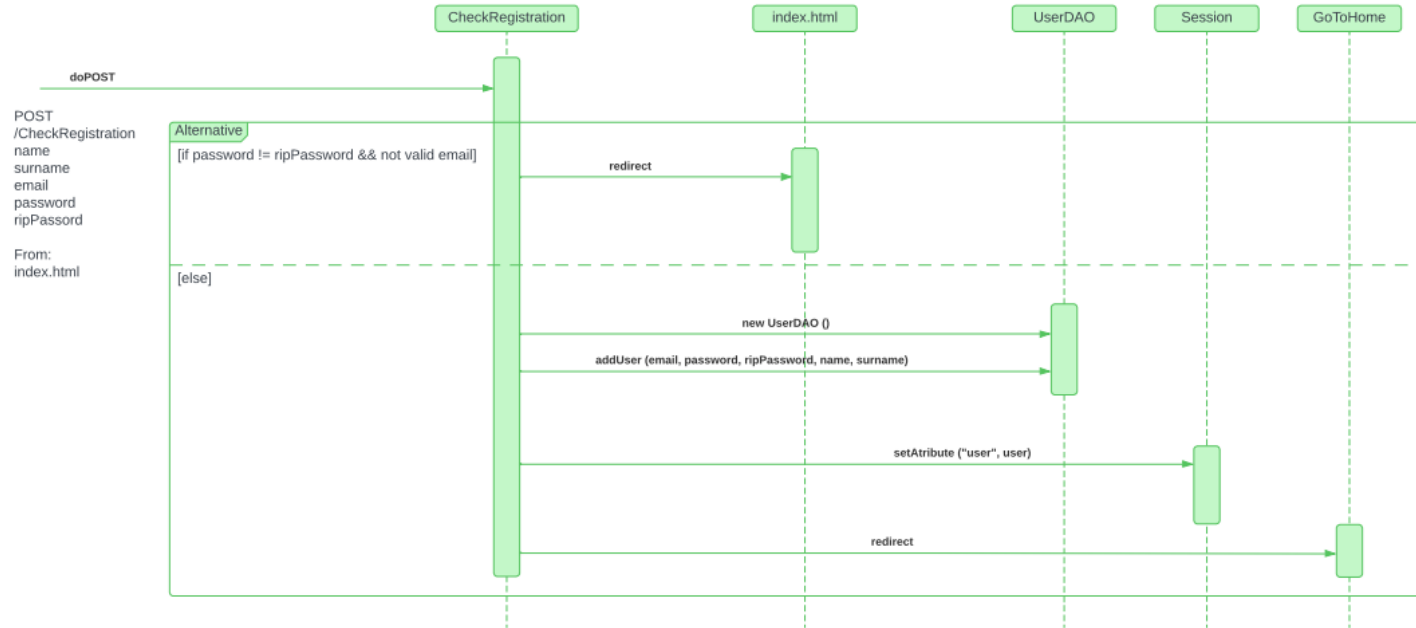
Components

- Controllers (Servlets)
 - AddComment
 - CheckLogin
 - CheckRegistration
 - CreateAlbum
 - DeleteImage
 - GetImage
 - GoToAlbum
 - GoToHome
 - GoToImage
 - Logout
 - UploadImage
- Views (Templates)
 - LOGIN
 - HOME PAGE
 - ALBUM PAGE
 - IMAGE PAGE

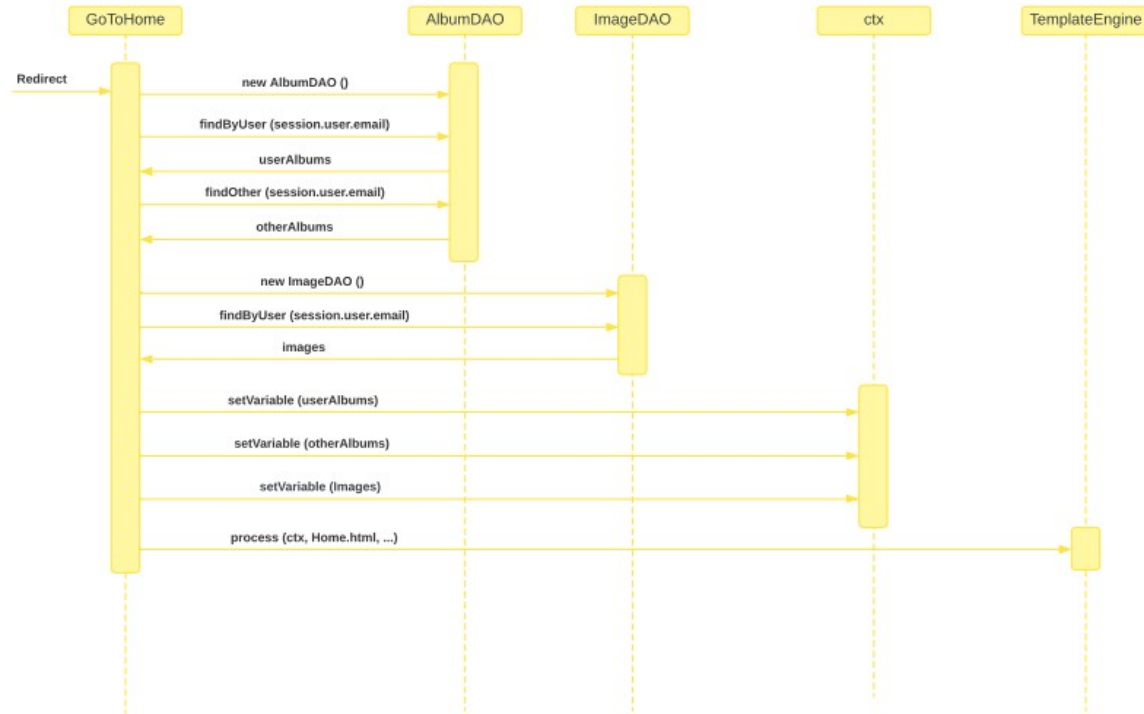
Event: login



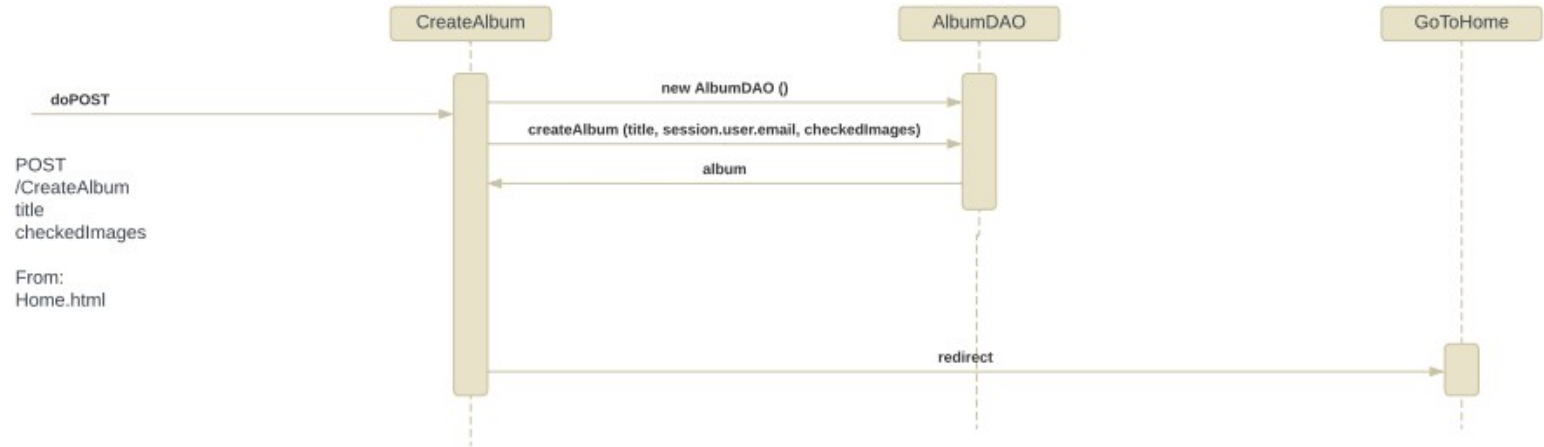
Event: Registration



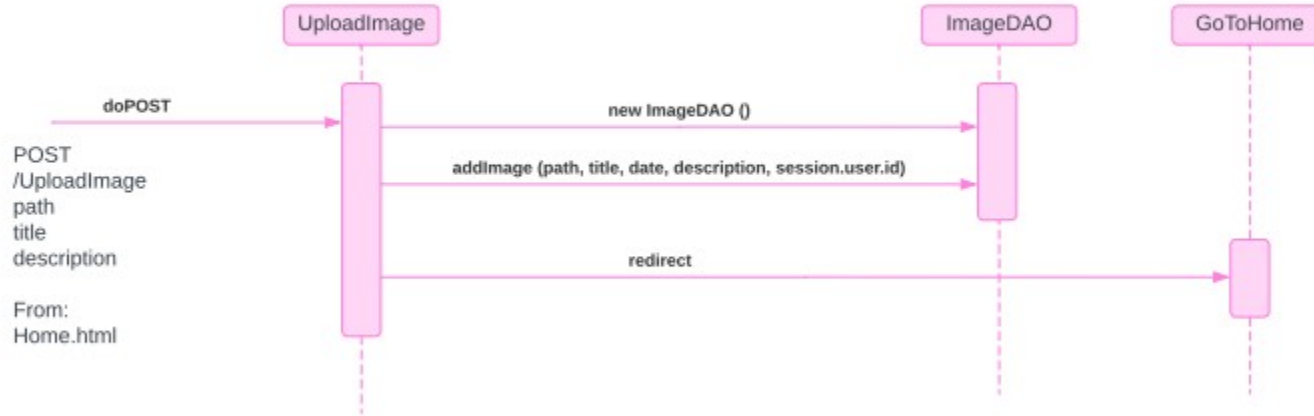
Event: GoToHome



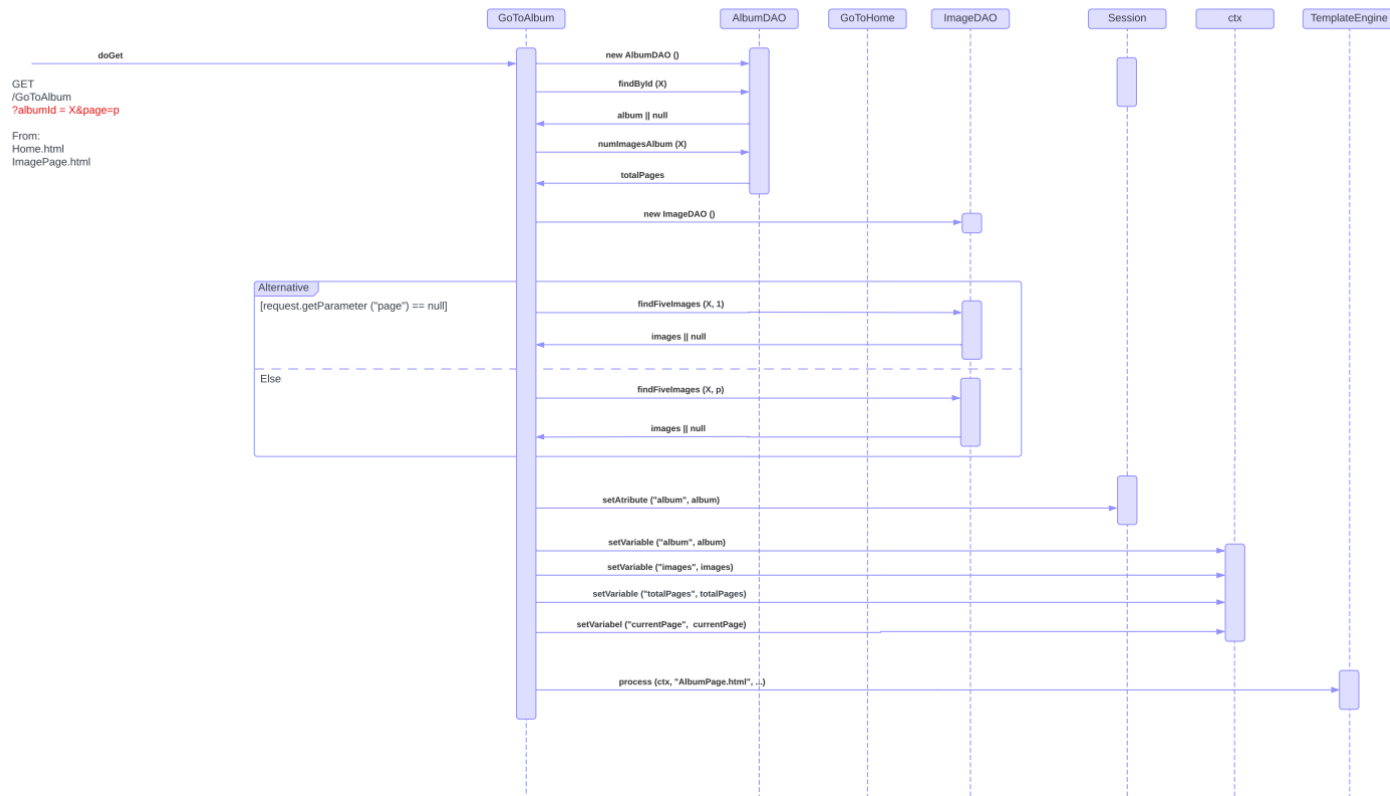
Event: CreateAlbum



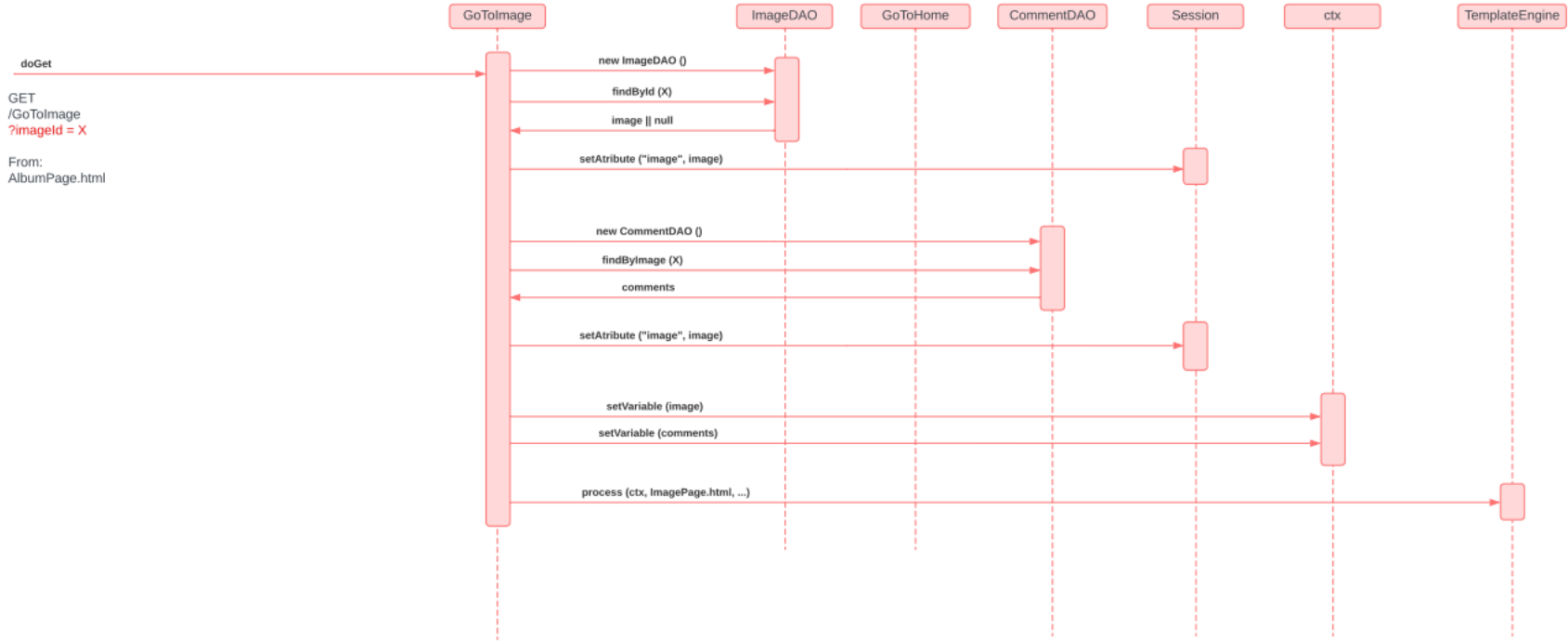
Event: UploadImage



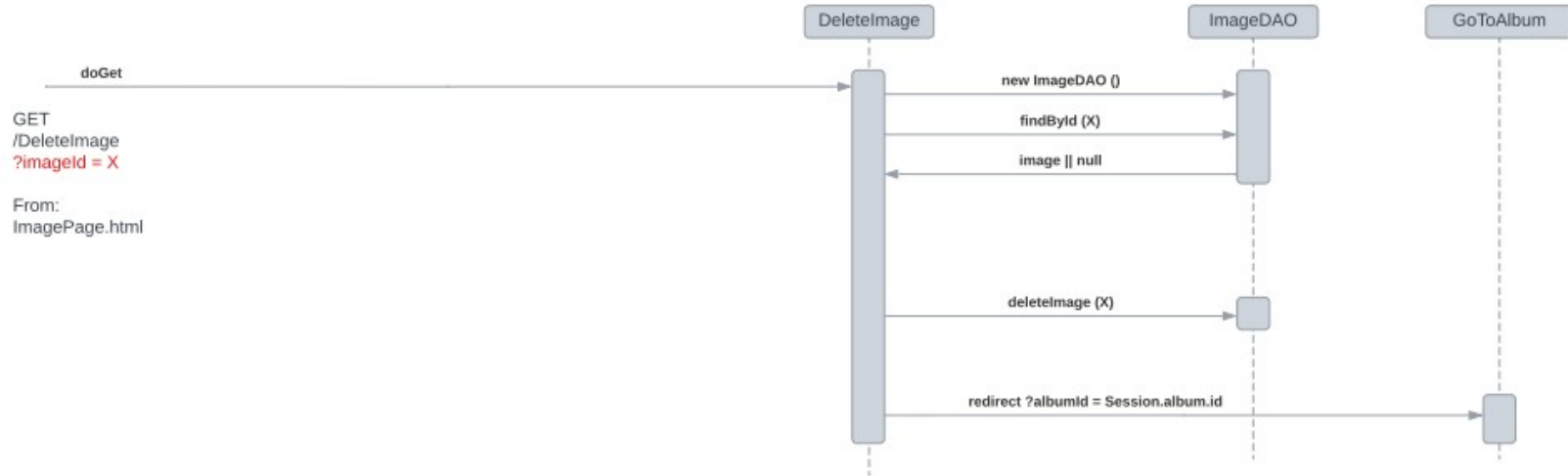
Event: GoToAlbum



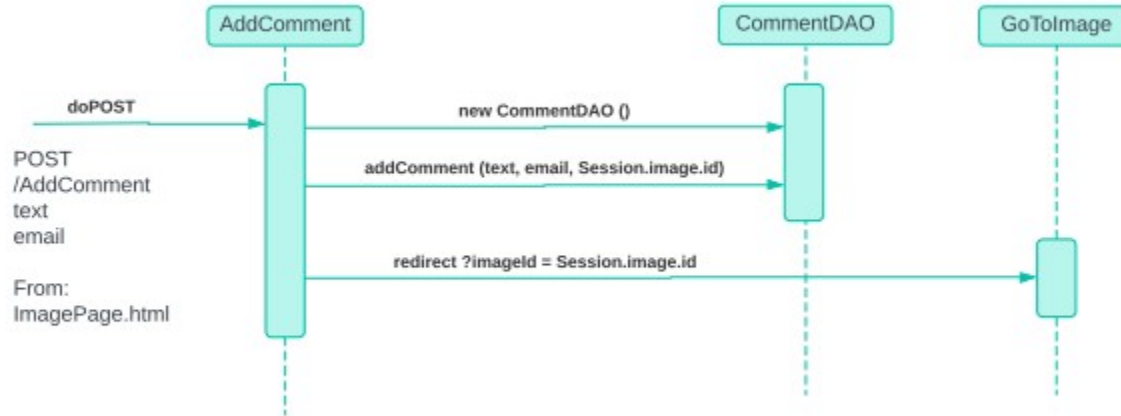
Event: GoToImage



Event: DeleteImage



Event: AddComment



Event: Logout

