sd Event : CreateAlbum

CreateAlbum	Session		GoToAlbumEditPage

1: doPOST()

1.1: getAttribute("username")

username

<<create>>

1.2: new AlbumDAO()

doPOST

AlbumDAO

2: createAlbum("newAlbum", username)

3: getAlbumsOfUser(username)

albums

4: newId = getFirstAlbumId(albums)

5: redirect(?id=newId)

sd Event : CreateComment

CreateComment | Session | CheckerUtility | GoToHomePage | GoToAlbumPage

1: doPOST()

<<create>>
1.1: new ImageDAO() → ImageDAO

POST

imagePosition
albumId
pageNumber
commentText

<<create>>
1.2: new CommentDAO() → CommentDAO

<<create>>
1.3: new AlbumDAO() → AlbumDAO

2: getAttribute("username")
username

3: checkAvailability(imagePosition, albumId, commentText)
true || false

4: [not available] redirect()

5: [available] getAlbumFromId(albumId)
album || null

6: [album == null] redirect()

7: [album != null] getImagesInAlbum(albumId)
images

8: getImageIdFromPosition()

9: createComment(imageId, username, commentText)

10: redirect(albumId, pageNumber, imagePosition)

sd Event : EditAlbum

| EditAlbum | Session | | CheckerUtility | GoToHomePage |

1: doPOST()

1.1: getAttribute("username")

username

POST

id
albumTitle
checkedImages

<<create>>
1.2: new AlbumDAO()

AlbumDAO

<<create>>
1.3: new ImageDAO()

ImageDAO

2: checkAvailability(id, albumTitle)

true || false

3: [parameters not available] redirect()

4: getAlbumFromId(id)

album

5: [username != album.getCreator_username] redirect()

6: getImagesOfUser(username)

userImages

7: findSelectedImages(checkedImages, userImages)

critical : atomic operation

8: updateAlbum()

9: updateAlbumTitle(albumTitle, id)

10: deleteAllImagesInAlbum(id)

loop : for all selected images

11: addImageToAlbum(imageId, albumId)

12: redirect()

sd Event : GoToAlbumEditPage

| | GoToAlbumEditPage | | Session | CheckerUtility | GoToHome | context | TemplateEngine |

1: doGET()

<<create>>
1.1: new AlbumDAO() → AlbumDAO

GET
id

<<create>>
1.2: new ImageDAO() → ImageDAO

2: getAttribute("username")

username

3: checkAvailabbility(id)

true || false

4: [id not available] redirect()

5: [id available] getAlbumFromId(id)

album

6: checkSameUsername(album.getCreator_username, username)

7: [different usernames] redirect()

8: getImagesOfUser(username)

imageList

9: getImagesOfAlbum(id)

containedImageList

10: computeAlreadySelectedImages()

11: setVariable("imagesMap", isContainedList)

12: setVariable("albumTitle", album.getTitle)

13: process("album_edit.html", context, writer)

sd Event : GoToAlbumPage

Participants: GoToAlbumPage, AlbumDAO, ImageDAO, CommentDAO, Session, CheckerUtility, GoToHomePage, context, TemplateEngine

Note (GET):
```
GET
id
page
image
```

Note: Database errors not shown, they redirect to a 502 page

1: doGET()

1.1: new AlbumDAO() «create» → AlbumDAO

1.2: new ImageDAO() «create» → ImageDAO

1.3: new CommentDAO() «create» → CommentDAO

2: getAttribute("username") → Session

username (return)

3: checkAvailability(id) → CheckerUtility

true || false (return)

4: [id not available] getFirstAlbumOfUser()

4.1: getAlbumsOfUser(username)

albums || null (return)

5: [albums == null] redirect() → GoToHomePage

6: [albums != null] getLastAlbumId()

7: redirect(?id=lastAlbumId)

8: [id available] getAlbumFromId(id)

album (return)

9: getImagesInAlbum(id)

imageList (return)

10: checkAvailability(page) → CheckerUtility

true || false (return)

11: [page not available || page <= 0] redirect(?page=1)

12: [page available] calculateImagesToSend()

13: checkAvailability(image) → CheckerUtility

true || false (return)

14: getAllCommentsForImage(imageId)

comments (return)

15: setVariable(albumTitle, imagesToShow, showNext, showPrev, isImageShown, shownImage, showEditButton) → context

16: [isImageShown == true] setVariable(comments, imageId) → context

17: process("album_page.html", context, writer) → TemplateEngine

sd Event : GoToLoginPage

| GoToLoginPage | Session | GoToHomePage | context | TemplateEngine |

1: doGET()

1.1: isPresent()

true || false

1.2: [session present] redirect()

doGET
errorId

2: parseErrorId()

3: [errorId == 1 || 2] setVariable("errorMsgLogin", errorMsgLogin)

4: [[errorId == 3 || 4 || 5 || 6]] setVariable("errorMsgSignup", errorMsgSignup)

5: process("login_page.html", context, writer)

sd Event : Login

POST
/LoginCheck

username
password

1: doPOST()

<<create>>
1.1: new UserDAO()

UserDAO

LoginCheck

CheckerUtility

GoToLoginPage

Session

GoToHomePAge

2: CheckAvailability()

true

not shown:
in case parameters are
absent, redirect with error
id 5

3: CheckCredentials(username, password)

user || null

4: [user == null] redirect(/?errorId=6)

5: [user != null] setAttribute("username", user)

6: [user != null] redirect()

sd Event : Logout

| Logout | Session | GoToLoginPage |

1: doGET()

1.1: invalidate()

2: redirect()

sd Filter : LoggedFilter