



Album
- albumName : String ~ picsName : ArrayList<String>
+ Album() + Album(albumName : String) + getAlbumName() : String + setAlbumName(albumName : String) : void + getPicsList() : ArrayList<String>

MyImage
- location : String - people : String - month : int - day : int - year : int - caption : String - picturePath : String
+ MyImage() + MyImage(location : String, people : String) + MyImage(picturePath : String) + MyImage(tagLocation : String, tagPeople : String, tagTime : int [], caption : String, picturePath : String) + getDay() : int + getMonth() : int + getYear() : int + getLocation() : String + getPeople() : String + getCaption() : String + getPicturePath() : String + setDay(i : int) : void + setMonth(i : int) : void + setYear(int i) : void + setLocation(location : String) : void + setPeople(people : String) : void + setCaption(caption : String) : void + setPicturePath(picturePath : String) : void

PhotoAlbum
<u>+ screen1ID : String</u> <u>+ screen1File : String</u> <u>+ screen2ID : String</u> <u>+ screen2File : String</u> <u>+ screen3ID : String</u> <u>+ screen3File : String</u> <u>+ screen4ID : String</u> <u>+ screen4File : String</u> <u>+ primarySTAGE : Stage</u>
<u>+ start(primaryStage : Stage) : void</u> <u>public static void main(String[] args)</u>

PhotoSearch
<u>+ movePhoto(MyImagesSrc : ArrayList<MyImage>, MyImagesDest : ArrayList<MyImage>, p : MyImage) : void</u>
<u>+ searchDataRange(withinRange : ArrayList<MyImage>, year1 : int, month1 : int, day1 : int, year2 : int, month2 : int, day2 : int) : void</u>
<u>+ inRange(year1 : int, month1 : int, day1 : int, year2 : int, month2 : int, day2 int, i : MyImage) : boolean</u>
<u>+ compareDates(year1 : int, month1 : int, day1 : int, year2 : int, month2 : int, day2 : int) : int</u>
<u>+ searchAllFields(a2 : ArrayList<MyImage>, location : String, year : String, month : String, day : String, people : String) : void</u>
<u>+ numMatchingPeople(names1 : String, names2 : String) : int</u>
<u>+ namesNoDups(str : String) : boolean</u>

User
- userName : String - albumList : ArrayList<Album> - imageUrl : ArrayList<MyImage>
+ User() + User(userName : String) + getUserName() : String + setUserName(String userName) : void + getAlbumList() : ArrayList<Album> + getImageList() : ArrayList<MyImage>

Utility
<u>+ output(ArrayList<User> users) : void</u> <u>+ input() : ArrayList<User></u> <u>+ newestPhoto(i : int, j : int) : String</u> <u>+ oldestPhoto(i : int, j : int) : String</u> <u>+ setDateTaken(mi : MyImage) : void</u> <u>+ extractDate(f : File) : String[]</u> <u>+ dateRange(userNum : int, albumNum : int) : String</u> <u>+ printUserList() : void</u>

<<Interface>> ControlledScreen
+ setScreenParent(screenPage : ScreensController) : void

ScreensController
- screens : HashMap<String, Node> <u>- deleteUser, deleteAlbum, openAlbum,</u> <u>sameUser, sameAlbum, transferPhoto :</u> <u>AnchorPane</u>
+ ScreensController() + addScreen(name : String, screen : Node) : void + getScreen(name : String) : Node + loadScreen(name : String, resource : String) : boolean + setScreen(NAME : String) : boolean + unloadScreen(name : String) : boolean + <u>showDeleteUserScene() : void</u> + <u>showDeleteAlbumScene() : void</u> + <u>showLoadAlbumScene() : void</u> + <u>showSameAlbumScene() : void</u> + <u>showSameUserScene() : void</u> + <u>showTransferPhotoScene() : void</u> + <u>showOpenAlbumWaningScene() : void</u>

AdminViewController
myController : ScreensController <u>+ userObs : ObservableList<String></u> + usersList : ListView<String> - newUser : TextField user : User deleteIndex : int + users : ArrayList<User> users
+ initialize(url : URL, rb : ResourceBundle) : void + setScreenParent(screenParent : ScreensController) : void - goToLogout(event : ActionEvent) : void - createUser(event : ActionEvent) : void - deleteUser(event : ActionEvent) : void - quit(event : ActionEvent) : void

AlbumViewController

~ myController : ScreensController

- location : TextField

- people : TextField

- month : TextField

- day : TextField

- year : TextField

- bigScreen : ImageView

- miniImages : GridPane

- captionText : TextArea

- cycleF : Button

- cycleB : Button

- quit : Button

- month2 : TextField

- day2 : TextField

- year2 : TextField

- secondary : GridPane

- createAlbumName : TextField

+ albumIndex : int

+ column : int

+ row : int

+ currSelect : int

+ savedImages : ArrayList<MyImage>

+ searchResults : ArrayList<MyImage>

+ picturePath : ObservableList<String>

+ init : int

~ imageView : ImageView

~ image : Image

~ file : File

~ grid_x, grid_y : int

- quit(event : ActionEvent) : void

+ handleClick(event : MouseEvent) :
void

+ initialize(url : URL, rb : ResourceBundle) :
void

+ setScreenParent(screenParent :
ScreensController) : void

- void goToUser(event : ActionEvent) : void

- addPhoto(event : ActionEvent) : void

+ imageAdder() : void

+ setDisplay() : void

- createImageView(IMAGEFILE : File) :
 ImageView
- removePhoto(event : ActionEvent) : void
- transferPhoto(event : ActionEvent) : void
 - reloadAlbum() : void
- openAlbum(event : ActionEvent) : void
- updateCaption(event : ActionEvent) : void
- updateFields(event : ActionEvent) : void
- clearFields(event : ActionEvent) : void
 - searchDataRangeButton(event :
 ActionEvent) : void
- searchFields(event : ActionEvent) : void
- cycleForward(event : ActionEvent) : void
 - cycleBack(event : ActionEvent) : void
- createAlbum(event : ActionEvent) : void
 - removeAll() : void
 - clearTexts() : void
 - setTexts(currSelect : int) : void
 - setCaption(currSelect : int) : void
- saveCopy(a1 : ArrayList<MyImage>, a2 :
 ArrayList<MyImage>) : void
 - removeFile(f : String) : String
- getImageIndex(currSelect :int) : int

DeleteAlbumController
+ noButton, yesButton : Button
- noButtonEvent(event : ActionEvent) : void
- yesButtonEvent(event : ActionEvent) : void

DeleteUserController
+ noButton, yesButton : Button
- noButtonEvent(event : ActionEvent) : void
- yesButtonEvent(event : ActionEvent) : void

LoadAlbumController
+ noButton, yesButton : Button + albumList : ListView<String> <u>+ albums : ObservableList<String></u> ~ myController : ScreensController
+ setScreenParent(screenParent : ScreensController) : void + initialize(url : URL, rb : ResourceBundle) : void - noButtonEvent(event : ActionEvent) : void - yesButtonEvent(event : ActionEvent) : void

LoginViewController
~ myController : ScreensController <u>+ users : ArrayList<User></u> <u>+ currUser : int</u> - userName : TextField
+ initialize(url : URL, rb : ResourceBundle) : void + setScreenParent(screenParent : ScreensController) : void - goToAdmin(event : ActionEvent) : void - goToUser(event : ActionEvent) : void - login(event : ActionEvent) : void - quit(event : ActionEvent) : void

OpenAlbumWaningController
+ okButton : Button
- okButtonEvent(event : ActionEvent) : void

SameAlbumNameController
+ okButton : Button
- okButtonEvent(event : ActionEvent) : void

SameUserNameController
+ okButton : Button
- okButtonEvent(event : ActionEvent) : void

TransferPhotoController
~ myController : ScreensController + noButton : Button + yesButton : Button - albumList : ListView<String> + <u>destIndex : int</u> + <u>albums : ObservableList<String></u>
+ setScreenParent(screenParent : ScreensController) : void + handleClick(arg0 : MouseEvent) : void - yesButtonEvent(event : ActionEvent) : void + initialize(url ; URL, rb : ResourceBundle) : void

UserViewController
~ myController : ScreensController - createAlbum : TextField - renameAlbum : TextField - albumList : ListView<String> - numImages : ListView<String> - oldestPhoto : ListView<String> - dateRange : ListView<String> + <u>index : int</u> + <u>albums : ObservableList<String></u> + <u>numberOfImages : ObservableList<String></u> + <u>oldest_photo : ObservableList<String></u> + <u>date_range : ObservableList<String></u>
+ handleClick(arg0 : MouseEvent) : void + setFields() : void + getAlbumList() : ListView<String> + initialize(url ; URL, rb : ResourceBundle) : void + setScreenParent(screenParent : ScreensController) : void - goToLogin(event : ActionEvent) : void - goToAlbum(event : ActionEvent) : void - quit(event : ActionEvent) : void - createAlbum(event : ActionEvent) : void - renameAlbum(event : ActionEvent) : void - deleteAlbum(event : ActionEvent) : void