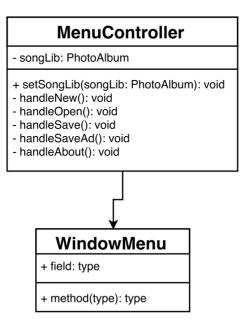


## - dialogStage: Stage - onClicked: boolean - initialize(): void + setDialogStage(dialogStage: Stage): void + isOkClicked(): boolean - handleOk(): void - handleCancel(): void + start(primaryStage: Stage): void DeleteWarning + field: type + method(type): type

# - userTable: TableView<User> - userList: TableColumn<User, String> - userID: Label - access: Label - photoAlbum: PhotoAlbum - initialize(): void - showUserInfo(selectedUser: User): void - handleDeleteUser(): void - handleNewUser(): void - handleOk():void - handleOk():void - MadminPage - Hield: type - method(type): type



## **PhotoListController**

## **EditorController**

captionField: TextField
 tagsField: TextField
 givenPhoto: Picture

+ setState(songLib: PhotoAlbum, inputAlbum:

Album, inputPhoto: Picture): void

initialize(): voidhandleOk(): voidhandleCancel(): void

- sanitizeInput(dirtyInput: String): String

+ start(primaryStage: Stage): void

## **Editor**

+ field: type

+ method(type): type

## **PhotoAlbum**

albumData: ObservableList<Album>userData: ObservableList<User>

+ userList: ArrayList<User> + currentUser: User

+ fileName: String

+ userLogins: ArrayList<String>

+ userDir: String- openingStage: Stage- windowMenu: BorderPane

+ writeUserList(): void

+ getUserData(): ObservableList<User>

+ start(openingStage: Stage): void

+ initRootLayout(): void

+ showPrimaryDisplay(returnedPhoto: Picture): void

+ showLogin(): void

+ showAdminPage(): void + showAddUserDialog(): String

+ showPhotoList(selectedPhoto: Picture): void

+ showSlideshow(Album: zCurrentAlbum,

currentPicture: Picture): void

+ showPhotoEditDialog(currentAlbum: Album,

photoToEdit: Picture): void

photoTable: TableView<Picture> photoList: TableColumn<Picture, String> photoCaption: Label - thumbnail: ImageView · defaultImage: Image · photoAlbum: PhotoAlbum - currentAlbum: Album + stage: Stage - initialize(): void - showPhotoDetails(photo: Picture): void - handleDeletePhoto(): void - handleAddPhoto(): void - handleManualSlideshow(): void - handleEdit(): void **PhotoList** + field: type + method(type): type

## **SearchResultsController**

- photoTable: TableView<Picture>

- photoList: TableColumn<Picture, String>

photoCaption: Label
 thumbnail: Label
 defaultImage: Image

photoAlbum: PhotoAlbumresults: ObservableList<Picture>

+ stage: Stage

- initialize(): void

- showPhotoDetails(photo: Picture): void

- createAlbumFromResults(): void

- handleManualSlideshow(): void

- handleLogOut(): void

## SearchResults + field: type + method(type): type

+ showSearchResults(matches: ObservableList<Picture>): void + showDeleteDialog(): boolean + getSongFilePath(): File + setSongFilePath(file: File): void

+ setSongFilePath(file: File): void + getPrimaryStage(): Stage

+ loadSongDataFromFile(file: File): void + saveSongDataToFile(file: File): void

+ main(args: String[]): void

## SlideshowController

- caption: Label - tags: Label

- date: Label

- thumbnail: ImageView - defaultImage: Image - buttonBar: ButtonBar - photoAlbum: PhotoAlbum

- photo: Picture - position: int

- numberOfPhotos: int modifiedDate: String - sdf: SimpleDateFormat

- initialize(): void

- showPhotoDetails(photo: Picture): void

handlePrevious(): void
 handleNext(): void

- handleBackToPhotoList(): void

## SlideshowPage

+ field: type

+ method(type): type

## User

- login: StringProperty

- photoz: ObservableList<Picture>

albumz: ObservableList<Album>albums: ArrayList<StringProperty>

- albumStrings: ArrayList<String>

serialVersionUID: long

## StartPageController

- photoAlbum: PhotoAlbum - usernameField: TextField

- setPhotoAlbum(photoAlbum: PhotoAlbum): void

- handleExit(): void - handleLogin(): void

## StartPage

+ field: type

+ method(type): type

## Picture

- name: String

- serialVersionUID: long - zname: StringProperty

- album: String

- tags: ArrayList<String>

- caption: String - cal: Calender - photoFile: File + phoDir: String

+ getZname(): StringProperty

+ setZname(newZname: String): void

+ getDate(): Date + getCal(): Calender

+ changeAlbum(newAlbum: String): void

+ getAlbum(): String + getName): String + getTags(): String

+ setTags(tagList: String): void

+ getPhotoFile(): File + setCaption(s: String): void + getCaption(): String

## **Album**

- name: String

nameProperty: StringPropertyphotos: ObservableList<Picture>

+ Album(name: String): void

+ getName(): String

+ getNameProperty(): String

+ setName(newName: String): void

+ getPhotos(): ObservableList<Picture>

+ getLogin(): String

+ setLogin(login: String): void + loginProperty(): StringProperty

+ getAlbums(): ObservableList<Album> + createAlbum(newName: String): boolean

+ createAlbum(name: String, results:

ObservableList<Picture>): void

+ deleteAlbum(oldName: String): void

+ checkNameAvailability(newAlbumName: String):

boolean

+ setAlbumStrings(): void + populateAlbums(): void

+ readPicture(name: String): Picture + writePicture(p: Picture): void + searchByTags(query: String):
ObservableList<Picture>

+ searchByDate(dateQuery: String):

ObservableList<Picture>

+ getAllPhotoData(): ObservableList<Picture>
+ toString(): String

## **AlbumList**

+ field: type

+ method(type): type