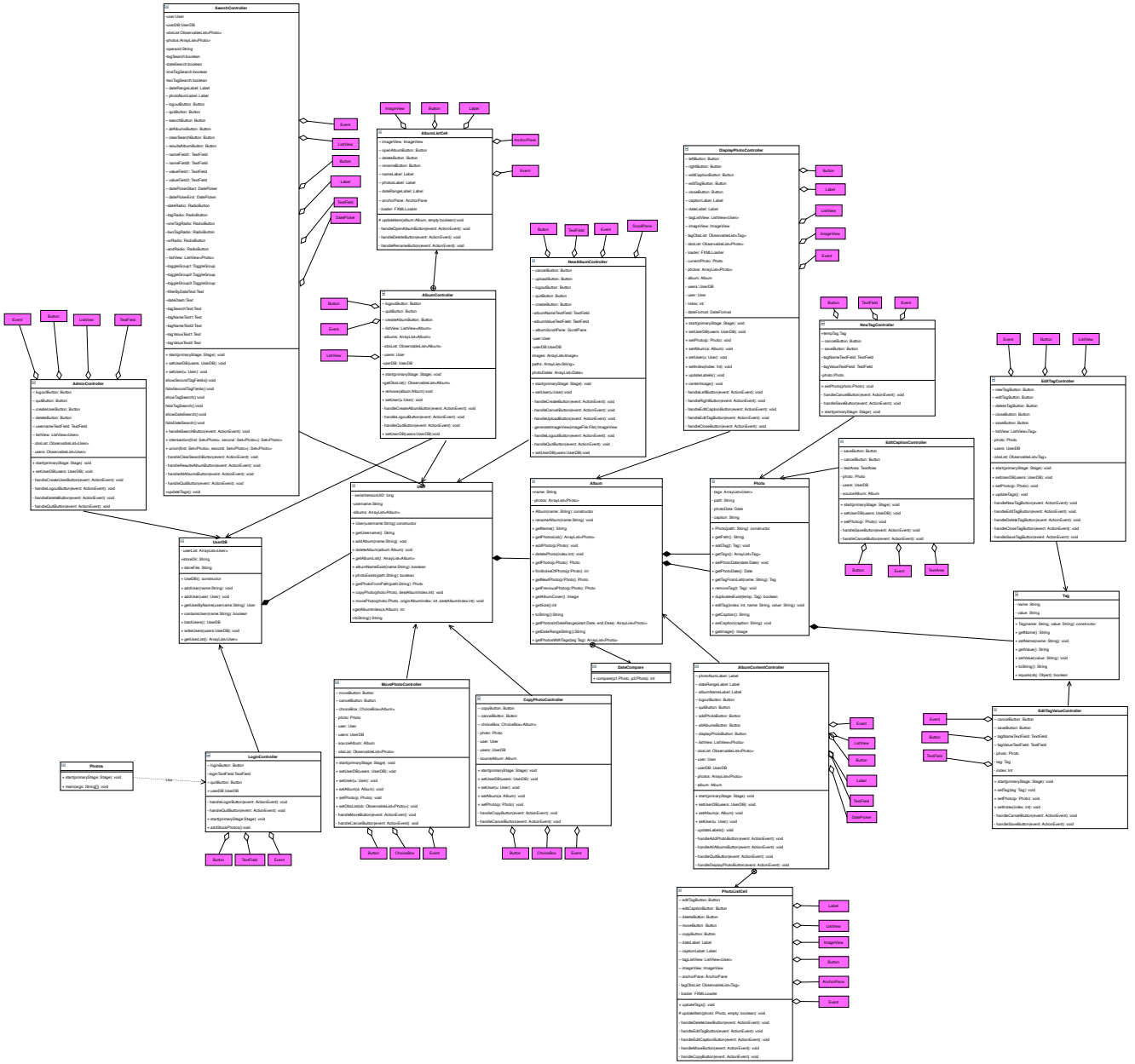
SearchController
-user:User
-useDB:UserDB
-dbList:ObservableList<Photo>
-photos:ArrayList<Photo>
-operand:String
-tagSearch:boolean
-dateSearch:boolean
-oneTagSearch:boolean
-twoTagSearch:boolean
-dateRangeLabel: Label
-photoNumLabel: Label
-logoutButton: Button
-quitButton: Button
-searchButton: Button
-allAlbumsButton: Button
-clearSearchButton: Button
-resultsAlbumButton: Button
-nameField1: TextField
-nameField2: TextField
-valueField1: TextField
-valueField2: TextField
-datePickerStart: DatePicker
-datePickerEnd: DatePicker
-dateRadio: RadioButton
-tagRadio: RadioButton
-oneTagRadio: RadioButton
-twoTagRadio: RadioButton
-orRadio: RadioButton
-andRadio: RadioButton
-listView: ListView<Photo>
-toggleGroup1: ToggleGroup
-toggleGroup2: ToggleGroup
-toggleGroup3: ToggleGroup
-filterBy:DateTextTest
-dateOath: Text
-tagSearchText: Text
-tagNameText1: Text
-tagNameText2: Text
-tagValueText1: Text
-tagValueText2: Text
+ start(primaryStage: Stage): void
+ setUserDB(users: UserDB): void
+ setUser(u: User): void
showSecondTagFields():void
hideSecondTagFields():void
showTagSearch():void
hideTagSearch():void
showDateSearch():void
hideDateSearch():void
+ handleSearchButton(event: ActionEvent): void
+ intersection(first:Set<Photo>, second: Set<Photo>): Set<Photo>
+ union(first: Set<Photo>, second: Set<Photo>): Set<Photo>
- handleClearSearchButton(event: ActionEvent): void
- handleResultsAlbumButton(event: ActionEvent): void
- handleAllAlbumsButton(event: ActionEvent): void
- handleQuitButton(event: ActionEvent): void
updateTags(): void
Event
ListView
Button
Label
TextField
DatePicker
AlbumListCell
-imageView: ImageView
-openAlbumButton: Button
-deleteButton: Button
-renameButton: Button
-nameLabel: Label
-photosLabel: Label
-dateRangeLabel: Label
-anchorPane: AnchorPane
-loader: FXMLLoader
updateItem(album:Album, empty:boolean):void
- handleOpenAlbumButton(event: ActionEvent): void
- handleDeleteButton(event: ActionEvent): void
- handleRenameButton(event: ActionEvent): void
ImageView
Button
Label
AnchorPane
Event
AlbumController
-logoutButton: Button
-quitButton: Button
-deleteAlbumButton: Button
-listView: ListView<Album>
-albums: ArrayList<Album>
-dbList: ObservableList<Album>
-users: User
-useDB: UserDB
+ start(primaryStage: Stage): void
+getDbList(): ObservableList<Album>
+ removeAlbum(Album):void
+ setUser(u: User): void
- handleCreateAlbumButton(event: ActionEvent): void
- handleLogoutButton(event: ActionEvent): void
- handleQuitButton(event: ActionEvent): void
+ setUserDB(users:UserDB):void
Button
Event
ListView
NewAlbumController
-cancelButton: Button
-uploadButton: Button
-logoutButton: Button
-quitButton: Button
-createButton: Button
-albumNameTextField: TextField
-albumValueTextField: TextField
-albumScrollPane: ScrollPane
-user:User
-useDB:UserDB
images: ArrayList<Image>
paths: ArrayList<String>
photoDates: ArrayList<Date>
+ start(primaryStage: Stage): void
+ setUser(u:User):void
- handleCreateButton(event: ActionEvent): void
- handleCancelBtn(event: ActionEvent): void
- handleUploadBtn(event: ActionEvent): void
- generateImageView(imageFile:File):Image
- handleLogoutBtn(event: ActionEvent): void
- handleQuitBtn(event: ActionEvent): void
+ setUserDB(users:UserDB):void
Button
TextField
Event
ScrollPane
DisplayPhotoController
-leftButton: Button
-rightButton: Button
-editCaptionButton: Button
-editTagButton: Button
-closeButton: Button
-caption:Label: Label
-dateLabel: Label
-tagListView: ListView<User>
-imageView: ImageView
-tagDbList: ObservableList<Tag>
-dbList: ObservableList<Photo>
-loader: FXMLLoader
-currentPhoto: Photo
-photos: ArrayList<Photo>
-album: Album
-users: UserDB
-user: User
-index: int
-dateFormat: DateFormat
+ start(primaryStage: Stage): void
+ setUserDB(users: UserDB): void
+ setPhoto(p: Photo): void
+ setAlbum(a: Album): void
+ setUser(u: User): void
+ setIndex(index: int): void
+ updateLabels() : void
+ centerImage() : void
- handleLeftButton(event: ActionEvent): void
- handleRightButton(event: ActionEvent): void
- handleEditCaptionButton(event: ActionEvent): void
- handleEditTagButton(event: ActionEvent): void
- handleCloseButton(event: ActionEvent): void
Button
Label
ListView
ImageView
Event
NewTagController
+tempTag:Tag
-cancelButton: Button
-saveButton: Button
-tagNameTextField: TextField
-tagValueTextField: TextField
-photo:Photo
+ setPhoto(photo:Photo): void
- handleCancelButton(event: ActionEvent): void
- handleSaveButton(event: ActionEvent): void
+ start(primaryStage: Stage): void
Button
TextField
Event
AdminController
-logoutButton: Button
-quitButton: Button
-createUserButton: Button
-deleteButton: Button
-usernameTextField: TextField
-listView: ListView<User>
-dbList: ObservableList<User>
-users: ObservableList<User>
+ start(primaryStage: Stage): void
+ setUserDB(users: UserDB): void
- handleCreateUserButton(event: ActionEvent): void
- handleLogoutButton(event: ActionEvent): void
- handleQuitButton(event: ActionEvent): void
Event
Button
ListView
TextField
User
-serialVersionUID: long
-username:String
-albums: ArrayList<Album>
+ User(username:String): constructor
+ getUsername(): String
+ setUsername(name:String): void
+ addAlbum(name:String): void
+ deleteAlbum(album:Album): void
+ getAlbumList(): ArrayList<Album>
+ albumNameExists(name:String): boolean
+ photoExists(path:String): boolean
+ getPhotoFromPath(path:String): Photo
+ copyPhoto(photo:Photo, destAlbumIndex:int): void
+ movePhoto(photo:Photo, origAlbumIndex: int, destAlbumIndex:int): void
+ getAlbumIndex(a:Album): int
+ toString():String
UserDB
-userList: ArrayList<User>
+storeDir: String
+ storeFile: String
+ UserDB(): constructor
+ addUser(name:String): void
+ addUser(user: User): void
+ getUserByName(username:String): User
+ contains(username:String): boolean
+ loadUsers(): UserDB
+ writeUsers(users:UserDB): void
+ getUserList(): ArrayList<User>
Album
+name: String
-photos: ArrayList<Photo>
+ Album(name: String): constructor
+ renameAlbum(name:String): void
+ getName(): String
+ getPhotoList(): ArrayList<Photo>
+ addPhoto(p:Photo): void
+ deletePhoto(index:int): void
+ getPhoto(p:Photo): Photo
+ findIndexOfPhoto(p:Photo): int
+ getNextPhoto(p:Photo): Photo
+ getPreviousPhoto(p:Photo): Photo
+ getAlbumCover(): Image
+ getSize():int
+ toString():String
+ getPhotosInDateRange(start:Date, end:Date): ArrayList<Photo>
+ getDateRangeString():String
+ getPhotosWithTag(tag:Tag): ArrayList<Photo>
Photo
-tags: ArrayList<User>
-path: String
-photoDate: Date
-caption: String
+ Photo(path: String): constructor
+ getPath(): String
+ setTag(t: Tag):void
+ getTag(): ArrayList<Tag>
+ setPhotoDate(date:Date): void
+ getPhotoDate(): Date
+ getTagFromList(name: String): Tag
+ removeTag(t: Tag): void
+ duplicateExists(tag: Tag): boolean
+ editTag(index: int, name: String, value: String): void
+ getCaption(): String
+ setCaption(caption: String): void
+ getImage(): Image
EditCaptionController
-saveButton: Button
-cancelButton: Button
-textArea: TextArea
-photo: Photo
-users: UserDB
-sourceAlbum: Album
+ start(primaryStage: Stage): void
+ setUserDB(users: UserDB): void
+ setPhoto(p: Photo): void
- handleSaveButton(event: ActionEvent): void
- handleCancelButton(event: ActionEvent): void
Button
Event
TextArea
EditTagController
-newTagButton: Button
-editTagButton: Button
-deleteTagButton: Button
-closeButton: Button
-saveButton: Button
-listView: ListView<Tag>
-photo: Photo
-users: UserDB
-dbList: ObservableList<Tag>
+ start(primaryStage: Stage): void
+ setUserDB(users: UserDB): void
+ setPhoto(p: Photo): void
+ updateTags(): void
- handleNewTagButton(event: ActionEvent): void
- handleEditTagButton(event: ActionEvent): void
- handleDeleteTagButton(event: ActionEvent): void
- handleCloseTagButton(event: ActionEvent): void
- handleSaveTagButton(event: ActionEvent): void
Event
Button
ListView
Tag
-name: String
-value: String
+ Tag(name: String, value: String): constructor
+ getName(): String
+ setName(name: String): void
+ getValue(): String
+ setValue(value: String): void
+ toString(): String
+ equals(o): Object): boolean
DateCompare
+ compare(p1:Photo, p2:Photo): int
AlbumContentController
-photoNumLabel: Label
-dateRangeLabel: Label
-albumNameLabel: Label
-logoutButton: Button
-quitButton: Button
-addPhotoButton: Button
-editAlbumButton: Button
-displayPhotoButton: Button
-listView: ListView<Photo>
-dbList: ObservableList<Photo>
-user: User
-useDB: UserDB
-photos: ArrayList<Photo>
-album: Album
+ start(primaryStage: Stage): void
+ setUserDB(users: UserDB): void
+ setAlbum(a: Album): void
+ setUser(u: User): void
-updateLabels() : void
- handleAddPhotoButton(event: ActionEvent): void
- handleAllAlbumsButton(event: ActionEvent): void
- handleQuitButton(event: ActionEvent): void
- handleDisplayPhotoButton(event: ActionEvent): void
Event
ListView
Button
Label
TextField
DatePicker
EditTagViewController
-cancelButton: Button
-saveButton: Button
-tagNameTextField: TextField
-tagValueTextField: TextField
-photo: Photo
-tag: Tag
-index: int
+ start(primaryStage: Stage): void
+ setTag(tag: Tag): void
+ setPhoto(p: Photo): void
+ setIndex(index: int): void
- handleCancelButton(event: ActionEvent): void
- handleSaveButton(event: ActionEvent): void
Event
Button
TextField
Photos
+ start(primaryStage:Stage): void
+ main(args: String[]): void
use
LoginController
-logoButton: Button
-loginTextField: TextField
-quitButton: Button
+ useDB: UserDB
- handleLoginButton(event: ActionEvent): void
- handleQuitButton(event: ActionEvent): void
+ start(primaryStage:Stage): void
+ addStockPhotos(): void
Button
TextField
Event
MovePhotoController
-moveButton: Button
-cancelButton: Button
-chooseBox: ChoiceBox<Album>
-photo: Photo
-user: User
-users: UserDB
-sourceAlbum: Album
-dbList: ObservableList<Photo>
+ start(primaryStage: Stage): void
+ setUserDB(users: UserDB): void
+ setUser(u: User): void
+ setAlbum(a: Album): void
+ setPhoto(p: Photo): void
+ setDbList(dl: ObservableList<Photo>): void
- handleMoveButton(event: ActionEvent): void
- handleCancelButton(event: ActionEvent): void
Button
ChoiceBox
Event
CopyPhotoController
-copyButton: Button
-cancelButton: Button
-chooseBox: ChoiceBox<Album>
-photo: Photo
-user: User
-users: UserDB
-sourceAlbum: Album
+ start(primaryStage: Stage): void
+ setUserDB(users: UserDB): void
+ setUser(u: User): void
+ setAlbum(a: Album): void
+ setPhoto(p: Photo): void
- handleCopyButton(event: ActionEvent): void
- handleCancelButton(event: ActionEvent): void
Button
ChoiceBox
Event
PhotoLabelUtil
-editTagButton: Button
-editCaptionButton: Button
-deleteButton: Button
-moveButton: Button
-copyButton: Button
-dateLabel: Label
-captionLabel: Label
-tagListView: ListView<User>
-imageView: ImageView
-anchorPane: AnchorPane
-tagDbList: ObservableList<Tag>
-loader: FXMLLoader
+ updateTag(): void
# updateItem(photo: Photo, empty: boolean): void
- handleDeleteLabelButton(event: ActionEvent): void
- handleEditTagButton(event: ActionEvent): void
- handleEditCaptionButton(event: ActionEvent): void
- handleMoveButton(event: ActionEvent): void
- handleCopyButton(event: ActionEvent): void
Label
ListView
ImageView
Button
AnchorPane
Event