

# AO Mobile Media Architecture (Release 0)

## COMPONENT: AlbumListScreen

### Required Interface: ControlPhoto

```
newPhotoAlbum();
deleteAlbum();
viewAlbum();
```

### Provided Interface: ManageAlbumList

```
append(String albumName);
delete(int i);
repaintListAlbum(String[] names);
```

## COMPONENT: AddPhotoToAlbumScreen

### Required Interface: ControlPhoto

```
addPhoto();
cancel();
```

### Provided Interface: ManagePhoto

```
String getPhotoName();
String getPath();
```

## COMPONENT: NewAlbumScreen

### Required Interface: ControlPhoto

```
saveNewPhotoAlbum();
cancel();
```

### Provided Interface: ManageAlbum

```
String getAlbumName();
```

## COMPONENT: PhotoListScreen

### Required Interface: ControlPhoto

```
viewPhoto(); // "View"
newPhoto(); // "Add"
deletePhoto(); // "Delete"
back(); // "Back"
```

### Provided Interface: ManagePhotoList

```
append(String imageLabel);
```

## COMPONENT: PhotoViewScreen

### Required Interface: ControlPhoto

```
back(); // "Back"
```

## COMPONENT: BaseController

### Provided Interface: ControlPhoto

```
newPhotoAlbum(); // "New Photo Album"
deleteAlbum(); // "Delete Album"
saveNewPhotoAlbum(); // "Save"
viewAlbum(); // "Select"
viewPhoto(); // "View"
newPhoto(); // "Add"
saveNewPhoto(); // "Save Add Photo"
deletePhoto(); // "Delete"
cancel(); // "Cancel"
back(); // "Back"
```

### Required Interface: ManageAlbumList

```
append(String albumName);
delete();
repaintListAlbum(String[] names);
```

### Required Interface: ManagePhoto

```
String getPhotoName();
String getPath();
```

### Required Interface: ManageAlbum

```
String getAlbumName();
```

### Required Interface: ManagePhotoList

```
append(String imageLabel);
```

### Required Interface: ManagePhotoInfo

```
addNewPhotoToAlbum(String label, String path, String album);
createNewPhotoAlbum(String albumName);
deleteImage(String imageName, String storeName);
deletePhotoAlbum(String albumName);
String[] getAlbumNames();
Image getImageFromRecordStore(String recordStore, String imageName);
String[] getImageNames(String recordName);
resetImageData();
```

## COMPONENT: AlbumData

### Provided Interface: ManagePhotoInfo

```
addNewPhotoToAlbum(String label, String path, String album);
createNewPhotoAlbum(String albumName);
deleteImage(String imageName, String storeName);
deletePhotoAlbum(String albumName);
String[] getAlbumNames();
Image getImageFromRecordStore(String recordStore, String imageName);
Hashtable getImageInfoTable();
String[] getImageNames(String recordName);
resetImageData();
```

**Required Interface: PersistPhoto**

```
addImageData(String photoname, String path, String albumname);
createNewPhotoAlbum(String albumName);
deletePhotoAlbum(String albumName);
boolean deleteSingleImageFromRMS(String storeName, String imageName);
String[] getAlbumNames();
ImageData getImageInfo(String imageName);
loadAlbums();
ImageData[] loadImageDataFromRMS(String recordName);
Image loadSingleImageFromRMS(String recordName, String imageName, int recordId);
resetImageRecordStore();
```

**COMPONENT: ImageAcessor****Provided Interface: PersistPhoto**

```
addImageData(String photoname, String path, String albumname);
createNewPhotoAlbum(String albumName);
deletePhotoAlbum(String albumName);
boolean deleteSingleImageFromRMS(String storeName, String imageName);
String[] getAlbumNames();
ImageData getImageInfo(String imageName);
loadAlbums();
ImageData[] loadImageDataFromRMS(String recordName);
Image loadSingleImageFromRMS(String recordName, String imageName, int recordId);
resetImageRecordStore();
boolean updateImageInfo(ImageData oldData, ImageData newData);
```

**Required Interface: ManagePhotoInfo**

```
Hashtable getImageInfoTable();
```