

# OO Mobile Media Architecture (Release 1)

## COMPONENT: AlbumListScreen

### Required Interface: ControlPhoto

```
newPhotoAlbum(); // "New Photo Album"
deleteAlbum(); // "Delete Album"
viewAlbum(); // "Select"
```

### Provided Interface: ManageAlbumList

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

## COMPONENT: AddPhotoToAlbumScreen

### Required Interface: ControlPhoto

```
saveNewPhoto(); // "Save Add Photo"
cancel();
```

### Provided Interface: ManagePhoto

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

## COMPONENT: NewAlbumScreen

### Required Interface: ControlPhoto

```
saveNewPhotoAlbum(); // "Save"
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) throws InvalidImageDataException, PersistenceMechanismException;
createNewPhotoAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteImage(String imageName, String storeName) throws PersistenceMechanismException, ImageNotFoundException;
deletePhotoAlbum(String albumName) throws PersistenceMechanismException;
String[] getAlbumNames();
Image getImageFromRecordStore(String recordStore, String imageName) throws ImageNotFoundException, PersistenceMechanismException;
String[] getImageNames(String recordName) throws UnavailablePhotoAlbumException;
resetImageData() throws PersistenceMechanismException;
```

## COMPONENT: AlbumData

### Provided Interface: ManagePhotoInfo

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

### Required Interface: PersistPhoto

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

## COMPONENT: ImageAcessor

### Provided Interface: PersistPhoto

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

### Required Interface: ManagePhotoInfo

```
Hashtable getImageInfoTable();
```