

# AO Mobile Media Architecture (Release 7)

## COMPONENT: SMS

### Provided Interface: SendSMS

`sendPhotoSMS(); // "Send Photo by SMS"`

### Provided Interface: ReceiveSMS

`ImageData processImageData(ImageAccessor imageAccessor, String photoName, String albumname) throws InvalidImageDataException, PersistenceMechanismException;`  
`processCopy(AddPhotoToAlbum copyPhotoToAlbum);`

### Provided Interface: ManageSMS

`setFromSMS(boolean fromSMS);`

## COMPONENT: MediaCopy

### Provided Interface: ManageCopy

`playMusic(); // "Play"`  
`viewPhoto(); // "View"`  
`copyPhoto(); // "Copy"`  
`saveCopyPhoto(); // "Save Photo"`  
`copyMusic();// "Copy"`

### Required Interface: InitPhotoView

`setPhotoScreen();`

### Required Interface: initMusicScreen

`setMediaName();`

## COMPONENT: FavouriteMedia

### Provided Interface: ManageFavourite

`setFavouriteCommand(); // "Set Favourite" //PhotoListScreen.ControlPhoto`  
`setFavourite(); // "Set Favourite" //BaseController.ControlPhoto`  
`viewFavouriteCommand(); // "View Favourites" //PhotoListScreen.ControlPhoto`  
`viewFavourite(); // "View Favourites" //BaseController.ControlPhoto`  
`append(); //BaseController.ManagePhotoList`

### Required Interface: ManageImage

`updateImage(ImageData image);`  
`ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;`

## COMPONENT: MediaSorting

### Provided Interface: CountMediaView

`playMusic(); // "Play" //MediaControler.ControlMedia`  
`viewPhoto(); // "View" //MediaControler.ControlMedia`

**Provided Interface: SortMedia**

```
sortMediaCommand(); // "Sort by Views" //MediaListScreen.ControlPhoto
sortMedia(); // "Sort by Views" //MediaController.ControlPhoto
append(); //MediaListController.ManagePhotoList
```

**Required Interface: ManageImage**

```
updateImage(ImageData image);
ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;
```

**COMPONENT: PhotoAspect**

**Provided Interface: ViewPhoto**

```
viewPhotoCommand();//"View" //MediaControler.ControlMedia
viewPhoto(); // "View" //MediaControler.ControlMedia
```

**Required Interface: initPhotoView**

```
initPhotiScreen();
setPhotoScreen();
```

**Required Interface: ManageMediaInfo**

```
Image getImageFromRecordStore(String recordStore, String imageName) throws ImageNotFoundException, PersistenceMechanismException;
```

**COMPONENT: MusicAspect**

**Provided Interface: PlayMusic**

```
playMusicCommand();//"Play" //MediaControler.ControlMedia
playMusic(); // "Play" //MediaControler.ControlMedia
```

**Required Interface: initMusicScreen**

```
initMusicScreen();
setMediaName();
```

**Required Interface: ManageMediaInfo**

```
InputStream getMusicFromRecordStore(String recordStore, String musicName) throws ImageNotFoundException, PersistenceMechanismException;
```

**COMPONENT: VideoAspect**

**Provided Interface: PlayVideo**

```
playVideoCommand(); //"PlayVideo"
playVideo(); //"PlayVideo"
```

**Required Interface: initVideoScreen**

```
initVideoScreen();
setMediaName();
```

**Required Interface: ManageMediaInfo**

```
InputStream getVideoFromRecordStore(String recordStore, String musicName) throws ImageNotFoundException, PersistenceMechanismException;
```

**COMPONENT: MediaCapture**

### Provided Interface: CaptureMedia

```
captureVideo(); // "Capture Video"
capturePhoto(); // "Capture Photo"
captureVideoCommand(); // "Capture Video"
capturePhotoCommand(); // "Capture Photo"
```

### Required Interface: initCaptureScreen

```
initCaptureMediaScreen();
```

## COMPONENT: ExceptionHandling

### Provided Interface: HandleExceptions

```
ImageData[] getImages(String recordName) //AlbumData.ManagePhotoInfo
deleteImage(String imageName, String storeName); //AlbumData.ManagePhotoInfo
resetImageRecordStore(); //AlbumData.ManagePhotoInfo
ImageData getImageInfo(String imageName); //ImageAcessor.PersistPhoto
loadAlbums(); // ImageAcessor.PersistPhoto
addImageData(String photoname, String path, String albumname); //ImageAcessor.PersistPhoto
ImageData[] loadImageDataFromRMS(String recordName); //ImageAcessor.PersistPhoto
boolean deleteSingleImageFromRMS(String storeName, String imageName); //ImageAcessor.PersistPhoto
deletePhotoAlbum(String albumName); //ImageAcessor.PersistPhoto
```

## COMPONENT: AlbumListScreen

### Required Interface: ControlAlbum

```
newAlbum(); // "New Album"
deleteAlbum(); // "Delete Album"
back(); // "Back"
exit(); // "Exit"
```

### Required Interface: ControlMediaList

```
viewAlbum(); // "Select"
```

### Provided Interface: ManageAlbumList

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

- The method deleteAll was not included as part of this interface because it is only called by the AlbumListScreen class.
- The method initMenu was not included because I considered it too specific to be architectural.

## COMPONENT: AddMediaToAlbumScreen

### Required Interface: ControlMedia

```
saveNewItem(); // "Save Item"
cancel();
```

### Provided Interface: ManageMedia

```
String getItemName();
String getPath();
```

```
setLabePath(String label);
setItemName(String itemName);
String getItemType();
byte[] getCapturedMedia();
setCapturedMedia(byte[] capturedMedia);
```

**COMPONENT: NewLabelScreen**

**Required Interface: ControlLabel**

```
saveLabel(); // "Save"
cancel();
```

**Required Interface: ControlAlbum**

```
saveNewAlbum(); // "Save"
cancel();
```

**Provided Interface: ManageLabel**

```
String getLabelName();
int getFormType();
```

**COMPONENT: MediaListScreen**

**Required Interface: ControlMedia**

```
newMediaItem(); // "Add"
deleteMediaItem(); // "Delete"
editLabel(); // "Edit Label"
back(); // "Back"
```

**Provided Interface: ManageMediaList**

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

- The method initMenu was not included because I considered it too specific to be architectural.

**COMPONENT: PhotoViewScreen**

**Required Interface: ControlCopy**

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

**Required Interface: ControlSMS**

**COMPONENT: NetworkScreen**

**Required Interface: ControlSMS**

```
sendSMSNow(); // "Send Now"
```

**Provided Interface: ManageNetwork**

```
String getRecPhoneNum();
String getRecPort();
```

**COMPONENT: PlayMusicScreen**

**Required Interface: ControlMusic**

```
start(); // "Start"  
back(); // "Back"  
stop(); // "Stop"
```

**Required Interface: ControlCopy**

**Provided Interface: ManageMusic**

```
startPlay();  
pausePlay();
```

**COMPONENT: PlayVideoScreen**

**Required Interface: ControlVideo**

```
start(); // "Start"  
back(); // "Back"  
stop(); // "Stop"
```

**Required Interface: ControlCopy**

**Provided Interface: ManageVideo**

```
setVisibleVideo();  
startVideo();  
stopVideo();
```

**COMPONENT: CaptureMediaScreen**

**Required Interface: ControlCapture**

```
startVideo(); // "Start"  
stopVideo(); // "Stop"  
takePhoto; // "Take Photo"  
back(); // "Back"
```

**Provided Interface: ManageCapture**

```
byte[] getByteArrays();  
pauseCapture();  
startCapture();  
byte[] takePicture();
```

**COMPONENT: CaptureMediaController**

**Provided Interface: ControlCapture**

```
startVideo(); // "Start"  
stopVideo(); // "Stop"  
takePhoto; // "Take Photo"  
back(); // "Back"
```

**Required Interface: ManageCapture**

```
byte[] getByteArrays();
```

```
pauseCapture();
startCapture();
byte[] takePicture();
```

**Required Interface: ManageMedia**

```
byte[] getCapturedMedia();
String getItemName();
String getPath();
setCapturedMedia(byte[] capturedMedia);
setItemName(String itemName);
setLabePath(String label);
```

**Required Interface: ManageMediaInfo**

```
addVideoData(String videoname, String albumname, byte[] video) throws InvalidImageDataException, PersistenceMechanismException;
String[] getAlbumNames();
```

**Provided Interface: initCaptureScreen**

```
initCaptureMediaScreen();
```

**COMPONENT: PlayVideoController**

**Provided Interface: ControlVideo**

```
start(); // "Start"
back(); // "Back"
stop(); // "Stop"
```

**Provided Interface: ControlCopy**

```
copyMusic();// "Copy"
```

**Required Interface: ManageVideo**

```
setVisibleVideo();
startVideo();
stopVideo();
```

**Required Interface: ManageMedia**

```
String getPath();
setItemName(String itemName);
setLabePath(String label);
```

**Required Interface: ManageAlbumList**

```
repaintListAlbum(String[] names);
```

**Required Interface: ManageMediaInfo**

```
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
String[] getAlbumNames();
MediaData getMediaInfo(String imageName) throws ImageNotFoundException;
```

**Provided Interface: initVideoScreen**

```
initVideoScreen();
setMediaName();
```

**Required Interface: HandleCommand**

```
postCommand(Command command);
```

**COMPONENT: PlayMusicController**

**Provided Interface: ControlMusic**

```
start(); // "Start"
back(); // "Back"
stop(); // "Stop"
```

**Provided Interface: ControlCopy**

```
copyMusic();// "Copy"
```

**Required Interface: ManageAlbumList**

```
repaintListAlbum(String[] names);
```

**Required Interface: ManageMedia**

```
String getPath();
setItemName(String itemName);
setLabePath(String label);
```

**Required Interface: ManageMusic**

```
startPlay();
pausePlay();
```

**Required Interface: ManageMediaInfo**

```
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
String[] getAlbumNames();
MediaData getMediaInfo(String imageName) throws ImageNotFoundException;
```

**Provided Interface: initMusicScreen**

```
initMusicScreen();
setMediaName();
```

**Required Interface: HandleCommand**

```
postCommand(Command command);
```

**COMPONENT: SMSController**

**Provided Interface: ControlSMS**

```
sendPhotoSMS(); // "Send Photo by SMS"
sendSMSNow(); // "Send Now"
```

**Required Interface: ManageAlbumList**

```
repaintListAlbum(String[] names);
```

**Required Interface: ManageSMS**

```
setFromSMS(boolean fromSMS);
setImage(byte[] img);
```

**Required Interface: ManageNetwork**

```
String getRecPhoneNum();
String getRecPort();
```

**Required Interface: ManageMediaInfo**

```
String[] getAlbumNames();
MediaData getMediaInfo(String imageName) throws ImageNotFoundException;
byte[] loadMediaBytesFromRMS(String recordName, int recordId) throws PersistenceMechanismException;
```

**Required Interface: HandleCommand**

```
postCommand(Command command);
```

**COMPONENT: BaseController**

**Required Interface: ManageAlbumList**

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

**Required Interface: ManageMediaInfo**

```
String[] getAlbumNames();
```

**Provided Interface: HandleCommand**

```
postCommand(Command command);
```

**COMPONENT: MediaController**

**Provided Interface: ControlMedia**

```
newMediaItem(); // "Add"
saveNewMediaItem(); // "Save Item"
deleteMediaItem(); // "Delete"
editLabel(); // "Edit Label"
saveLabel(); // "Save"
cancel(); // "Cancel"
back(); // "Back"
```

**Provided Interface: ManageImage**

```
updateImage(ImageData image);
showImageList(String recordName);
ImageData getImageInfo(String imageName) throws ImageNotFoundException, NullAlbumDataReference;
```

**Required Interface: ManageLabel**

```
String getLabelName();
```

**Required Interface: ManageMedia**

```
String getItemName();
String getPath();
String getItemType();
```

**Required Interface: ManageMediaInfo**

```
InputStream getVideoFromRecordStore(String recordStore, String musicName) throws ImageNotFoundException, PersistenceMechanismException;
MediaData[] loadMediaDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
InputStream getMusicFromRecordStore(String recordStore, String musicName) throws ImageNotFoundException, PersistenceMechanismException;
addNewMediaToAlbum(String label, String path, String album) throws InvalidImageDataException, PersistenceMechanismException;
deleteMedia(String imageName, String storeName) throws PersistenceMechanismException, ImageNotFoundException;
Image getImageFromRecordStore(String recordStore, String imageName) throws ImageNotFoundException, PersistenceMechanismException;
MediaData getMediaInfo(String imageName) throws ImageNotFoundException;
boolean updateMediaInfo(ImageData oldData, ImageData newData) throws InvalidImageDataException, PersistenceMechanismException;
```



**Provided Interface: HandleCommand**

```
postCommand(Command command);
```

**Required Interface: HandleCommand**

```
postCommand(Command command);
```

**COMPONENT: AlbumController**

**Provided Interface: ControlAlbum**

```
newAlbum(); // "New Album"
deleteAlbum(); // "Delete Album"
saveNewAlbum(); // "Save"
```

**Required Interface: ManageAlbumList**

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

**Required Interface: ManageLabel**

```
String getLabelName();
int getFormType();
```

**Required Interface: ManageMediaInfo**

```
createNewPhotoAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteAlbum(String albumName) throws PersistenceMechanismException;
String[] getAlbumNames();
resetMediaData() throws PersistenceMechanismException;
```

**Provided Interface: HandleCommand**

```
postCommand(Command command);
```

**Required Interface: HandleCommand**

```
postCommand(Command command);
```

**COMPONENT: PhotoViewController**

**Provided Interface: ControlCopy**

```
back(); // "Back"
copyPhoto(); // "Copy"
saveCopyPhoto(); // "Save Photo"
```

**Provided Interface: InitPhotoView**

```
initPhotoScreen();
setPhotoScreen();
```

**Provided Interface: ProcessingPhotoEvents**

```
ImageData processImageData(ImageAccessor imageAccessor, String photoName, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
processCopy(AddPhotoToAlbum copyPhotoToAlbum);
```

**Required Interface: ManageMedia**

```
String getItemName();
String getPath();
```

```
setLabePath(String label);
setItemName(String itemName);
byte[] getCapturedMedia();
setCapturedMedia(byte[] capturedMedia);
```

**Required Interface: ManageImage**

```
showImageList(String recordName);
```

**Required Interface: ManageMediaInfo**

```
addImageData(String photoname, Image imgdata, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
String[] getAlbumNames();
MediaData getMediaInfo(String imageName) throws ImageNotFoundException;
```

**Required Interface: HandleCommand**

```
postCommand(Command command);
```

**COMPONENT: MediaListController**

**Provided Interface: ControlMediaList**

```
viewAlbum(); // "Select"
```

**Required Interface: ManageMediaList**

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

**Required Interface: ManageMediaInfo**

```
MediaData[] getMedias(String recordName) throws UnavailablePhotoAlbumException;
```

**Provided Interface: HandleCommand**

```
postCommand(Command command);
```

**Required Interface: HandleCommand**

```
postCommand(Command command);
```

**COMPONENT: AlbumPhotoData**

**Provided Interface: ManageMediaInfo**

```
addVideoData(String videoname, String albumname, byte[] video) throws InvalidImageDataException, PersistenceMechanismException;
InputStream getVideoFromRecordStore(String recordStore, String musicName) throws ImageNotFoundException, PersistenceMechanismExc
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
byte[] loadMediaBytesFromRMS(String recordName, int recordId) throws PersistenceMechanismException;
MediaData[] loadMediaDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
InputStream getMusicFromRecordStore(String recordStore, String musicName) throws ImageNotFoundException, PersistenceMechanismException;
addImageData(String photoname, Image imgdata, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addNewMediaToAlbum(String label, String path, String album) throws InvalidImageDataException, PersistenceMechanismException;
createNewPhotoAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteAlbum(String albumName) throws PersistenceMechanismException;
deleteMedia(String imageName, String storeName) throws PersistenceMechanismException, ImageNotFoundException;
String[] getAlbumNames();
Image getImageFromRecordStore(String recordStore, String imageName) throws ImageNotFoundException, PersistenceMechanismException;
MediaData[] getMedias(String recordName) throws UnavailablePhotoAlbumException;
resetMediaData() throws PersistenceMechanismException;
MediaData getMediaInfo(String imageName) throws ImageNotFoundException;
```

```
boolean updateMediaInfo(ImageData oldData, ImageData newData) throws InvalidImageDataException, PersistenceMechanismException;
```

#### Required Interface: PersistPhoto

```
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addMediaData(String photoname, String path, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
createNewAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteAlbum(String albumName) throws PersistenceMechanismException;
boolean deleteSingleMediaFromRMS(String storeName, String mediaName) throws PersistenceMechanismException, ImageNotFoundException, NullAlbumDataReference;
String[] getAlbumNames();
ImageData getMediaInfo(String mediaName) throws ImageNotFoundException, NullAlbumDataReference;
loadAlbums() throws InvalidImageDataException, PersistenceMechanismException;
byte[] loadMediaBytesFromRMS(String recordName, int recordId) throws PersistenceMechanismException;
ImageData[] loadMediaDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
Image loadSingleImageFromRMS(String recordName, String imageName, int recordId) throws PersistenceMechanismException;
resetRecordStore() throws InvalidImageDataException, PersistenceMechanismException;
boolean updateImageInfo(MediaData oldData, MediaData newData) throws InvalidImageDataException, PersistenceMechanismException;
addImageData(String photoname, Image imgdata, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
```

## COMPONENT: AlbumMusicData

#### Provided Interface: ManageMediaInfo

```
addVideoData(String videoname, String albumname, byte[] video) throws InvalidImageDataException, PersistenceMechanismException;
InputStream getVideoFromRecordStore(String recordStore, String musicName) throws ImageNotFoundException, PersistenceMechanismException;
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
byte[] loadMediaBytesFromRMS(String recordName, int recordId) throws PersistenceMechanismException;
MediaData[] loadMediaDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
InputStream getMusicFromRecordStore(String recordStore, String musicName) throws ImageNotFoundException, PersistenceMechanismException;
addImageData(String photoname, Image imgdata, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addNewMediaToAlbum(String label, String path, String album) throws InvalidImageDataException, PersistenceMechanismException;
createNewPhotoAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteAlbum(String albumName) throws PersistenceMechanismException;
deleteMedia(String imageName, String storeName) throws PersistenceMechanismException, ImageNotFoundException;
String[] getAlbumNames();
Image getImageFromRecordStore(String recordStore, String imageName) throws ImageNotFoundException, PersistenceMechanismException;
MediaData[] getMedias(String recordName) throws UnavailablePhotoAlbumException;
resetMediaData() throws PersistenceMechanismException;
MediaData getMediaInfo(String imageName) throws ImageNotFoundException;
boolean updateMediaInfo(ImageData oldData, ImageData newData) throws InvalidImageDataException, PersistenceMechanismException;
```

#### Required Interface: PersistMusic

```
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addMediaData(String photoname, String path, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
createNewAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteAlbum(String albumName) throws PersistenceMechanismException;
boolean deleteSingleMediaFromRMS(String storeName, String mediaName) throws PersistenceMechanismException, ImageNotFoundException, NullAlbumDataReference;
String[] getAlbumNames();
ImageData getMediaInfo(String mediaName) throws ImageNotFoundException, NullAlbumDataReference;
loadAlbums() throws InvalidImageDataException, PersistenceMechanismException;
byte[] loadMediaBytesFromRMS(String recordName, int recordId) throws PersistenceMechanismException;
ImageData[] loadMediaDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
resetRecordStore() throws InvalidImageDataException, PersistenceMechanismException;
boolean updateImageInfo(MediaData oldData, MediaData newData) throws InvalidImageDataException, PersistenceMechanismException;
```

## COMPONENT: AlbumVideoData

### Provided Interface: ManageMediaInfo

```
addVideoData(String videoname, String albumname, byte[] video) throws InvalidImageDataException, PersistenceMechanismException;
InputStream getVideoFromRecordStore(String recordStore, String musicName) throws ImageNotFoundException, PersistenceMechanismException;
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
byte[] loadMediaBytesFromRMS(String recordName, int recordId) throws PersistenceMechanismException;
MediaData[] loadMediaDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
InputStream getMusicFromRecordStore(String recordStore, String musicName) throws ImageNotFoundException, PersistenceMechanismException;
addImageData(String photoname, Image imgdata, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addNewMediaToAlbum(String label, String path, String album) throws InvalidImageDataException, PersistenceMechanismException;
createNewPhotoAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteAlbum(String albumName) throws PersistenceMechanismException;
deleteMedia(String imageName, String storeName) throws PersistenceMechanismException, ImageNotFoundException;
String[] getAlbumNames();
Image getImageFromRecordStore(String recordStore, String imageName) throws ImageNotFoundException, PersistenceMechanismException;
MediaData[] getMedias(String recordName) throws UnavailablePhotoAlbumException;
resetMediaData() throws PersistenceMechanismException;
MediaData getMediaInfo(String imageName) throws ImageNotFoundException;
boolean updateMediaInfo(ImageData oldData, ImageData newData) throws InvalidImageDataException, PersistenceMechanismException;
```

### Required Interface: PersistVideo

```
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addMediaData(String photoname, String path, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
createNewAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteAlbum(String albumName) throws PersistenceMechanismException;
boolean deleteSingleMediaFromRMS(String storeName, String mediaName) throws PersistenceMechanismException, ImageNotFoundException, NullAlbumDataReference;
String[] getAlbumNames();
ImageData getMediaInfo(String mediaName) throws ImageNotFoundException, NullAlbumDataReference;
loadAlbums() throws InvalidImageDataException, PersistenceMechanismException;
byte[] loadMediaBytesFromRMS(String recordName, int recordId) throws PersistenceMechanismException;
MediaData[] loadMediaDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
resetRecordStore() throws InvalidImageDataException, PersistenceMechanismException;
boolean updateImageInfo(MediaData oldData, MediaData newData) throws InvalidImageDataException, PersistenceMechanismException;
addVideoData(String videoname, String albumname, byte[] video) throws InvalidImageDataException, PersistenceMechanismException;
```

## COMPONENT: ImageAcessor

### Provided Interface: PersistPhoto

```
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addMediaData(String photoname, String path, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
createNewAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteAlbum(String albumName) throws PersistenceMechanismException;
boolean deleteSingleMediaFromRMS(String storeName, String mediaName) throws PersistenceMechanismException, ImageNotFoundException, NullAlbumDataReference;
String[] getAlbumNames();
ImageData getMediaInfo(String mediaName) throws ImageNotFoundException, NullAlbumDataReference;
loadAlbums() throws InvalidImageDataException, PersistenceMechanismException;
byte[] loadMediaBytesFromRMS(String recordName, int recordId) throws PersistenceMechanismException;
ImageData[] loadMediaDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
Image loadSingleImageFromRMS(String recordName, String imageName, int recordId) throws PersistenceMechanismException;
resetRecordStore() throws InvalidImageDataException, PersistenceMechanismException;
boolean updateImageInfo(MediaData oldData, MediaData newData) throws InvalidImageDataException, PersistenceMechanismException;
addImageData(String photoname, Image imgdata, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
```

## COMPONENT: MusicAcessor

### Provided Interface: PersistMusic

```
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addMediaData(String photoname, String path, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
createNewAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteAlbum(String albumName) throws PersistenceMechanismException;
boolean deleteSingleMediaFromRMS(String storeName, String mediaName) throws PersistenceMechanismException, ImageNotFoundException, NullAlbumDataReference;
String[] getAlbumNames();
ImageData getMediaInfo(String mediaName) throws ImageNotFoundException, NullAlbumDataReference;
loadAlbums() throws InvalidImageDataException, PersistenceMechanismException;
byte[] loadMediaBytesFromRMS(String recordName, int recordId) throws PersistenceMechanismException;
ImageData[] loadMediaDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
resetRecordStore() throws InvalidImageDataException, PersistenceMechanismException;
boolean updateImageInfo(MediaData oldData, MediaData newData) throws InvalidImageDataException, PersistenceMechanismException;
```

## COMPONENT: VideoAcessor

### Provided Interface: PersistVideo

```
addMediaData(MediaData mediaData, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
addMediaData(String photoname, String path, String albumname) throws InvalidImageDataException, PersistenceMechanismException;
createNewAlbum(String albumName) throws PersistenceMechanismException, InvalidPhotoAlbumNameException;
deleteAlbum(String albumName) throws PersistenceMechanismException;
boolean deleteSingleMediaFromRMS(String storeName, String mediaName) throws PersistenceMechanismException, ImageNotFoundException, NullAlbumDataReference;
String[] getAlbumNames();
ImageData getMediaInfo(String mediaName) throws ImageNotFoundException, NullAlbumDataReference;
loadAlbums() throws InvalidImageDataException, PersistenceMechanismException;
byte[] loadMediaBytesFromRMS(String recordName, int recordId) throws PersistenceMechanismException;
MediaData[] loadMediaDataFromRMS(String recordName) throws PersistenceMechanismException, InvalidImageDataException;
resetRecordStore() throws InvalidImageDataException, PersistenceMechanismException;
boolean updateImageInfo(MediaData oldData, MediaData newData) throws InvalidImageDataException, PersistenceMechanismException;
addVideoData(String videoname, String albumname, byte[] video) throws InvalidImageDataException, PersistenceMechanismException;
```