

OO Mobile Media Architecture (Release 7)

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);
```

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

```
viewPhoto(); // "View"
```

```
playMusic();//"Play"
playVideo();//"PlayVideo"
captureVideo(); //"Capture Video"
capturePhoto(); //"Capture Photo"
newMediaItem(); //"Add"
deleteMediaItem(); //"Delete"
editLabel(); // "Edit Label"
sortMedia(); // "Sort by Views"
back();//"Back"
setFavourite(); // "Set Favorite"
viewFavourite(); // "View Favorites"
```

Provided Interface: ManagePhotoList

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

COMPONENT: PhotoViewScreen

Required Interface: ControlCopy

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

Required Interface: ControlSMS

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

Provided Interface: ManageSMS

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

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

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

Provided Interface: ManageMusic

```
startPlay();
```

```
pausePlay();
```

COMPONENT: PlayVideoScreen

Required Interface: ControlVideo

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

Required Interface: ControlCopy

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

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();
```

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();
```

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;
```

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;
```

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

```
viewPhoto(); // "View"
playMusic(); // "Play"
playVideo();// "PlayVideo"
captureVideo(); // "Capture Video"
capturePhoto(); // "Capture Photo"
newMediaItem(); // "Add"
saveNewMediaItem(); // "Save Item"
deleteMediaItem(); // "Delete"
editLabel(); // "Edit Label"
sortMedia (); // "Sort by Views"
setFavourite(); // "Set Favorite"
viewFavourite(); // "View Favorites"
saveLabel(); // "Save"
cancel(); // "Cancel"
back();// "Back"
```

Provided Interface: ManageImage

```
showImageList(String recordName, boolean sort, boolean favorite);
```

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"
```

Required Interface: ManageMedia

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

Required Interface: ManageImage

```
showImageList(String recordName, boolean sort, boolean favorite);
```

Required Interface: ManageSMS

```
boolean isFromSMS();
Image getImage();
```

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: ManageCapture

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

Required Interface: HandleCommand

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

COMPONENT: MediaListController

Provided Interface: ControlMediaList

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

Required Interface: ManageMediaList

```
append(String imageUrl);
```

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, 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: 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;
```

- The addImageData method included in this release is equivalent to the copyPhotoToAlbum method of the previous release.

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;
MediaData[] 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;
MediaData[] 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;
```