**总结六**

1. 完成photo的删除、添加到设备、传输到mac和刷新等功能。
2. 自定义toolbarView，用于动态显示toolbarVeiw内部按钮以及各个按钮的排版。
3. 给相应按钮添加点击方法，并响应方法中发送通知，在对应的详情界面接收通知并进行相应的按钮点击事件响应操作。并导入传输框架功能代码。
4. 刷新功能的实现：直接通过通知传送过来的information的对象方法刷新相应album下的图片信息（难点：刷新动画的实现，主要是自定义一个刷新动画的View，之后在刷新的过程中将刷动画view设置成rootBox的contentView即可，由于rootBox的来回重复切换contentview，因此继承自NSBox后自定义了一个Box，直接调用.h提供的pushView: 和 popView方法可以简单方便的对其contentview进行管理和操作）。
5. 添加到设备功能的实现：主要用到了继承自IMBBaseTransfer类的IMBBetweenDeviceHandler类进行到设备的功能实现。先调用初始化方法: initWithSelectedArray:selectAry categoryModel:model srcIpodKey:desIpodKey: withPlaylistArray: 方法传入要传输的初始信息，之后调用startTransfer方法开始传输。之后实现下列代理方法监听文件的传输状态和过程: - (void)transferPrepareFileStart:(NSString \*)file(传输开始) - (void)transferPrepareFileEnd - (void)transferProgress:(float)progress（传输进度） - (void)transferComplete:(int)successCount TotalCount:(int)totalCount （传输完成） - (BOOL)transferOccurError:(NSString \*)error（传输失败）
6. toMac功能的实现：
7. 完成Other界面的初步显示
8. 导入SystemFile功能框架代码
9. 在界面上显示文件系统

**难点：**

1. 对象的内存管理。有时出现些莫名的crash，大多原因还是出在对象的释放时刻不合适而造成的。
2. 多设备间的各个功能操作，由于打开的每个设备详情页面是用的同一个类，因此通知响应的方法将会同时接收到，因此需要在通知传送的时候传递一个ipod，用于判断不同的设备信息，以至在正确的页面做出正确的响应。
3. 拖拽toDevice功能的实现：tableView和collectionView的拖拽进另一个window的功能实现比较难，到现在还没实现这个拖拽操作。