

相册管理系统的设计与实现



学 院 计算机科学与技术

专 业 计算机科学与技术(双)

年 级 2005

姓 名 易腾飞

指导教师 张坤龙

2009 年 6 月 15 日

摘 要

随着数码相机和电脑的日渐普及,人们迫切需要一款功能强大且易用的相册管理软件将相片有效地管理起来。

论文在认真分析相册管理系统的需求的基础上,将相册管理系统分为相册管理子系统和相片管理子系统,设计了相册管理系统的类和数据文件。主要的类包括相片类、相册类、相册管理类和相册存储类等四个类。数据文件方面,定义了相册目录文件、类别文件、搜索结果文件。

相册管理系统的实现主要采用了XML的相关技术。相册管理子系统提供了相片导入、相册属性操作和增加、删除、修改相册的功能。相片管理子系统除了提供相片属性操作和增加、删除、修改、显式相片的功能外,还提供了相片搜索功能。

关键词: 相册管理系统; C#; XML

ABSTRACT

Album Management System (AMS) serves to manage photographs in a efficient manner. An AMS which is convenient, multifunctional while delicate is in high demand.

Basing upon the precise requirement analysis of AMS, this study divide it into two subsystems, that is, album management and photograph management. We design the class and data documentations. Major classes embrace photograph, album, album manager as well as album storage. With respect to the data documentations, we define the album contents, category together with search results.

We develop the AMS via XML-related technologies. As illustrations, four data documents are in XML formats. Our system offers photo input, album property operation in conjunction with functions of adding, deleting, revising. In addition, we provide the photo search option.

Key words: Album Management System; C#; XML

目 录

第一章 绪论	1
1.1 开发背景	1
1.2 研究意义	1
1.3 相册管理软件现状	2
1.4 论文内容组织	2
第二章 相册管理系统的需求分析.....	4
2.1 系统总体需求	4
2.2 用例需求	4
2.3 程序运行流程	5
2.4 系统功能分析	6
2.5 系统的运行和开发环境	7
第三章 相册管理系统的设计.....	9
3.1 类的设计	9
3.2 数据文件的设计	11
3.3 界面设计	14
第四章 相册管理系统的实现.....	16
4.1 C#操作 XML.....	16
4.2 相册管理子系统的实现	18
4.3 相片管理子系统的实现	19
第五章 结论	23

5.1 总结	23
5.2 展望	23
参考文献.....	24
外文资料	
中文译文	
致谢	

第一章 绪论

1.1 开发背景

随着数码相机和电脑的日渐普及，人们可以随时随地用数码相机记录生活中的每一个瞬间，并把数码相片保存在电脑里。与传统相机相比，数码相机的存储空间的可重复利用性，使人们在拍照的时候可以随心所欲。随着电脑的普及，很多人都选择把数码相机拍的照片存放在自己的电脑里。因此，人们迫切需要一款功能强大且易用的相册管理软件将人们记录的生活瞬间有效地管理起来。

目前，已有一些软件具备电子相册的功能，各有各的特点，功能、界面设计都各有千秋，但是也各有不足之处。本论文的设计方向就是综合各个软件的优点，摒弃其缺点，注重相册管理系统的实用性、安全性、稳定性，提供一个方便快捷、易用、功能强大的相册管理系统。

相册管理系统能满足人们对图像信息加工的时效性，多样性提出的要求。统计表明，人类从外界获得的信息约 75% 是从图像取得。多媒体技术的发展，万维网的普及，使得在计算机上的对图像的使用和处理日益重要。大量的图片是以数字图像的格式存在的，这使得图片的处理和传递很方便。通过互联网，图片瞬间即可被传送到世界的任何地方。但传统相片的使用因此受到局限。传统相机从拍摄相片到获得相片的周期通常是要以天计的。这种时延在数字技术发达的今天，对用户常常是难以容忍的。传统相册的另一个问题是使用不灵活。它的查询是以串行查询的线性方式进行的，查询效率低，并且相片的插入、更改等使用不方便。于是，利用计算机的强大处理功能，帮助用户快速准确地获取图像信息就成了电子相册系统的主要任务。电子相册使用的是数字图像，它的来源丰富，处理容易。利用多媒体技术，电子相册采用的是非线性查询的方式，直接访问到所需要的相片。特别是与数码相机等录入设备配合使用时，能够达到“即拍即得”的使用效果。随着数码相机逐渐深入家庭使用当中，对相册管理系统的需求量也在迅速增加。

1.2 研究意义

与传统相册相比，电子相册有着得天独厚的优点：

(1) 欣赏方便: 传统的相册在多人欣赏时只好轮流进行，而电子相册可以很多人同时欣赏。

(2) 交互性强: 可以像 VCD 点歌一样，将相册分成不同的标题，比如说“我的童年”、“我的大学生活”等。

(3) 储存量大: 随着电脑存储技术的发展，一块普通的硬盘就能存放数十万张照片，而一本传统相册最多只能存储几十张照片。

(4) 易于保存: 只要电脑硬盘不出现致命性故障，相册就一直会存在，不会变

色、发黄、损坏，更不会丢弃。

(5)欣赏性强:以高科技处理照片，可以对相片进行裁切、增加各种精美的相框，可以使用手机、mp3 等手持设备查看，真正可以做到随时随地随处进行浏览。

(6)成本低廉:电子相册都是保存在电脑中，基本费用也只用考虑电脑的成本。

(7)方便:只要打开电脑就能浏览，不需要拿着个厚厚的相册到处跑。

1.3 相册管理软件现状

目前，相册管理系统的应用十分广泛。已有的应用较为广泛的相册管理系统有 FotoEver、Picasa、iSee 等。

FotoEver 是一款功能强大的专用相册管理软件。它界面简洁、操作简便，除了提供本地相册管理功能外，还提供了墙纸屏保管理、网上相册管理、视频影音管理等功能。

Picasa 是 Google 开发的一款相册管理软件。Picasa 能自动组织电脑上的所有图片，形成相册，并提供幻灯片、修正红眼、剪切、发送 E-mail 等功能。Picasa 友好的浏览界面使用户能更方便地找到需要的图片。

iSee 同样是一款功能强大且应用广泛的相册管理软件。其提供的相册管理功能和图片处理功能异常强大，比其它相册管理软件具有更多实用的功能，如自动合成相册程序、风格特效等。功能方面 iSee 是佼佼者，但界面设计方面不够美观，这一点不如 Picasa。

1.4 论文内容组织

本相册管理系统是综合了各大图片处理软件的优点而进行设计开发的。主要是针对了用户的使用习惯，在使用的方便性、易操作性上做了大量的工作。以下是本系统的一些功能简介。

(1)相册功能:增加、删除、修改相册及其属性等功能。

(2)相片功能:导入相片、修改相片的属性、删除相片、顺旋、逆旋、另存、下一张、上一张、放大、缩小、搜索。

下面介绍本文内容的详细组织结构。

(1)第一章：绪论，主要介绍论文的主要研究背景和意义、研究现状。

(2)第二章：系统需求分析，主要是相册管理系统的需求分析，得出系统的性能要求、用例需求、程序运行流程、功能模块，并确定了系统所采用的开发环境。

(3)第三章：系统的设计，主要包括系统的类的设计、数据文件的设计和界面设计。

(4)第四章：系统的具体实现，主要介绍系统 C#操作目录文件（xml 文件）的方法和各个功能模块的具体实现。

(5)第五章：结论，主要说明本系统的意义和不足以及对未来工作的展望。

第二章 相册管理系统的需求分析

2.1 系统总体需求

软件工程中包含需求、设计、编码和测试四个阶段，其中需求工程是软件工程第一个也是很重要的一个阶段，它直接关系到软件产品的方向，所以需求分析的质量至关重要。软件的需求分析必须要有对原业务的一个深入了解、提取、抽象、升华的过程^[1]。

软件的需求分析是从用户的业务中提取出软件系统能够帮助用户解决的业务问题，通过对用户业务问题的分析，规划出我们的软件产品^[2]。这个步骤是对用户业务需求的一个升华，是一个把用户业务管理流程优化，转化为软件产品，从而提升管理而实现的质的飞跃，这一步是否成功，直接关系到开发出来的软件产品能否得到用户认可，顺利交付给客户，客户能否真正运用我们的产品帮助他解决业务或管理问题。

本系统与其它相册管理软件相比，必须具备如下优点：

(1)方便：操作简便，符合人性设计，傻瓜程度高，使一般用户很容易上手。

(2)功能齐全：相册管理系统已经大量开始使用，本相册系统必须在功能上更加齐全、更加方便，才能吸引用户使用。

(3)界面精美：在界面设计上必须突出“吸引用户眼球”这个特点，应该尽量将界面设计得美观、简洁、易用。

2.2 用例需求

用例是系统提供的功能块，换句话说来说用例演示了人们如何使用系统。通过用例观察系统，能够将系统实现与系统目标分开，有助于了解最重要的部分，满足用户要求和期望，而不会沉浸于实现细节^[3]。通过用例，用户可以看到系统提供的功能，先确定系统范围再深入开展项目工作。

具体用例分析如图 2-1 所示。

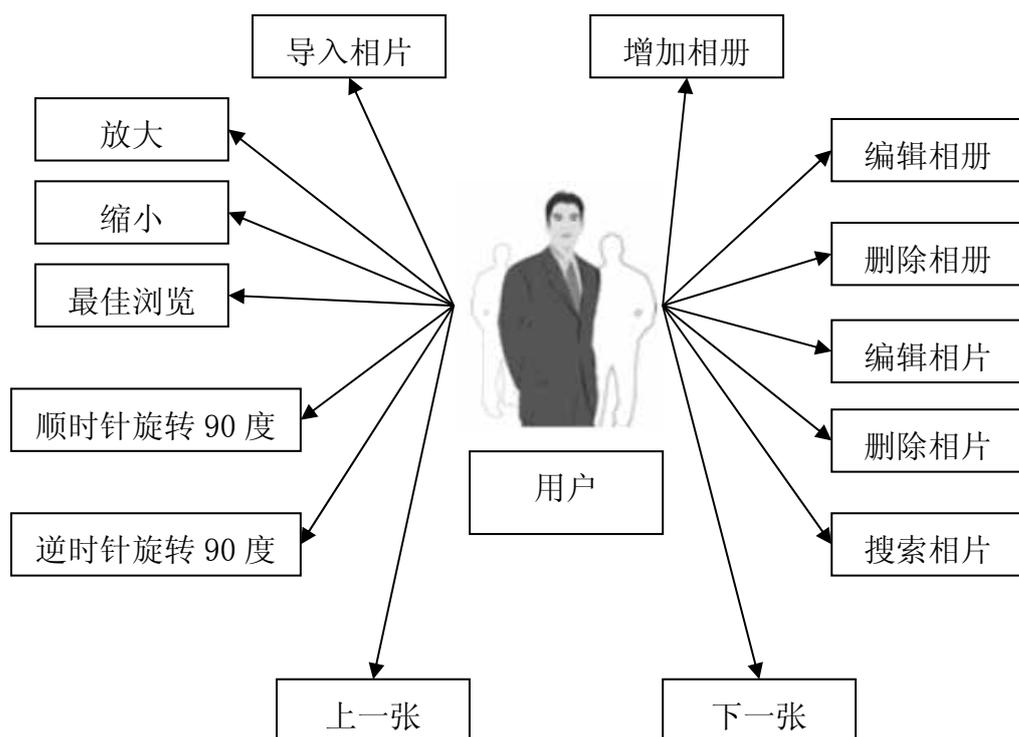


图 2-1 相册管理系统用例图

2.3 程序运行流程

系统进入后，自动读取用户电脑中已存的相册，在相册目录中显示各个相册的，系统默认选择相册目录中的第一个相册显示，并在照片浏览框中显示该相册的第一张相片。用户可以根据需要增加、删除和修改相册。用户可以向各个相册中导入相片。用户每单击一个相册项，相片浏览框就会显示该相册的第一张相片，用户还可以选择下一张或上一张相片。对于每张正显示的相片，用户可以进行“编辑相片”、“删除相片”、“顺旋”、“逆旋”、“另存”、“放大”、“缩小”、“搜索”等操作。

程序流程如图 2-2 所示。

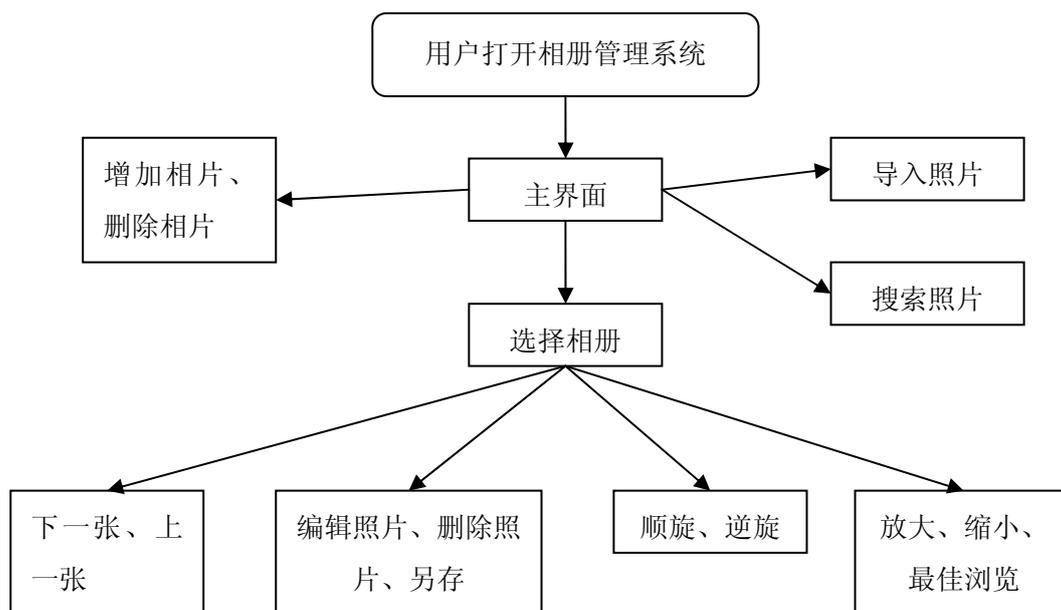


图 2-2 程序运行流程图

2.4 系统功能分析

2.4.1 系统功能结构图

本相册管理系统分为两大子系统：相册管理子系统和相片管理子系统。如图 2-3 描述的是相册管理系统功能模块。

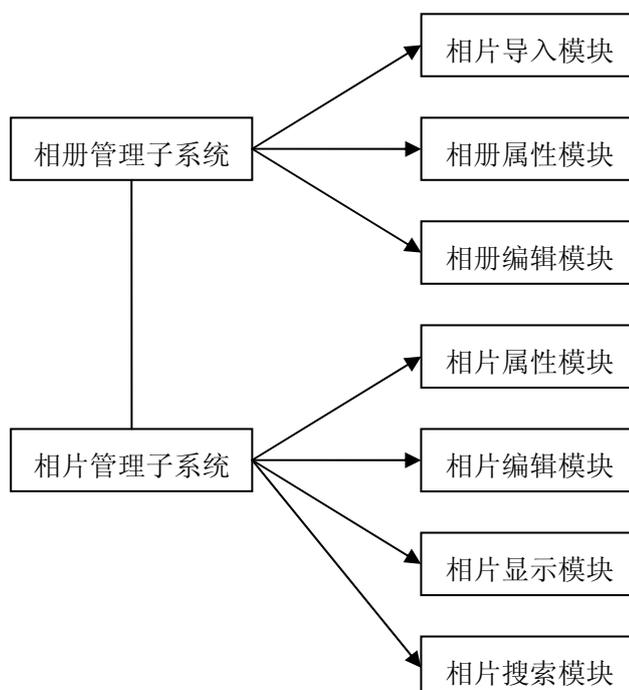


图 2-3 相册系统功能模块

2.4.2 子系统的功能说明

2.4.2.1 相册管理子系统

主要包括以下一些功能模块：

(1)相片导入模块：将选中的相片导入制定的相册。

(2)相册属性模块：主要包括添加、删除、修改相册的属性等功能。而相册属性又包括相册名称、类别、作者、详细内容、时间、地点、人物等。

(3)相册编辑模块：主要包括增加、删除、修改相册等功能。

2.4.2.2 相片管理子系统

主要包括以下一些功能模块：

(1)相片属性模块：主要包括添加、删除、修改相片的属性等功能。而相片属性又包括相册名称、类别、作者、描述、时间、地点、人物等。

(2)相片编辑模块：主要包括删除相片、顺旋、逆旋、另存等功能。

(3)相片显示模块：主要包括显示、下一张、上一张、放大、缩小等功能。

(4)相片搜索模块：根据可选条件项搜索所有相册中的相片。

2.5 系统的运行和开发环境

本相册管理系统采用 C#语言进行开发。

C#是微软公司基于.NET 平台量身订制的新一代程序开发语言。使用 C#开发基于.NET 的应用程序，具有良好的安全性和跨平台性。C#用途广泛，可以开发 Windows、ASP.NET、Web 服务，甚至使 Windows 底层的各种不同类型的应用程序^[4]。C#的程序可以通过 XML 交换数据信息。因为 XML 是纯文本的信息，这就为通过 C#的开发的应用程序的跨平台性能提供了强大的保障。C#的程序运行方式也是独特的，其应用程序完全在.NET 平台的管理下运行，这就大大地增加了应用程序完全在.NET 平台的管理下运行，这就大大增加了应用程序的稳定性和安全性。

总体看来，C#具有与 Visual Basic 一样简单的编程方法，与 Visual C++一样强大的功能，并且其跨平台性能要优于 Java。在使用 Java 开发的程序中，Bean 之间可以进行数据交互，但是任何一个 Bean 不能和其他语言开发的程序交换数据。现在 C#填补了这一空白，由于其采用 XML 作为交换数据的格式，所以 C#就具有了用 Java、COM 等其他语言编写的组件^[5]。C#语言是微软公司为它的 Microsoft.NET 计划推出的核心编程语言。该语言不仅继承 Visual Basic、Visual C++语言的优点，还几乎综合了目前所有编程语言的优点。

C#的优点^[7]：

(1)完全支持类和面向对象编程,包括接口和继承,虚函数和运算符重载的处

理。

(2)定义完整、一致的基类集合。

(3)对自动生成 xml 文档说明的内置支持。

(4)自动清理动态分配的内存。

(5)可以用用户定义的属性标记类或方法,着可以用于文档说明,对编译有一定的影响(例如,把方法标记为只在调试时编译)。

(6)对.net 基类库的完全访问权,并易于访问 windows api(有时确实需要访问,但这是不常出现的)。

(7)可以使用指针和直接内存访问,但该语言可以在没有他们的条件下工作。

(8)改变编译器选项,可以把程序编译为可执行文件或.net 组件库,该组件可以用 activex 控件(com 组件)相同的方式由其他代码调用。

(9)C#可以用于编写 asp.netweb 页面。

根据需求分析,相册管理系统采用的软硬件设施如下:

运行系统: 安装了 .Net Framework2.0 的 WINDOWS 系统;

运行硬件环境: 普通 PC 机;

开发系统: WINDOWS XP/ VISTA;

开发环境: Microsoft Visual Studio 2005;

开发语言: C#;

开发模式: Windows 窗体项目。

第三章 相册管理系统的设计

3.1 类的设计

为了利于编码的实现，在设计类的时候一定要做到结构合理、条理清晰。比如说，应将类所包含的方法设计成简要的功能性单元，用它描述和实现一个不连续的类接口部分。理想情况下，方法应简明扼要。若长度很大，可考虑通过某种方式将其分割成较短的几个方法。这样做也便于类内代码的重复使用。

在设计类的过程中也要遵循以下几个原则^[6]：

- a) 单一职责原则：对一个类而言，应该仅有一个引起它变化的原因；
- b) 开放，封闭原则：软件实体（类，模块，函数等等）应该是可以扩展的，但是不可修改的；
- c) 依赖倒置原则：高层模块不应该依赖于低层模块，二者都应该依赖于抽象；抽象不应该依赖于细节，细节应该依赖于抽象；
- d) 接口隔离原则：不应该强迫客户依赖于他们不用的方法。

因此，在相册管理系统中，鉴于功能的划分以及 C#编码的特点，设计出以下几个比较关键的类。

3.1.1 相片类

系统中设立关于相片的类 `Photograph`，用于表示单张相片。该类的属性如表 3-1 所示。

表 3-1 `Photograph` 类的属性

属性	类型	可见性	初始值	注释
<code>FileName</code>	<code>string</code>	<code>private</code>		文件路径
<code>Image</code>	<code>Bitmap</code>	<code>private</code>		相片
<code>Title</code>	<code>string</code>	<code>private</code>		名称
<code>Category</code>	<code>string</code>	<code>private</code>		类别
<code>Photographer</code>	<code>string</code>	<code>private</code>		作者
<code>Description</code>	<code>string</code>	<code>private</code>		描述
<code>Date</code>	<code>DateTime</code>	<code>private</code>	<code>DateTime.Now</code>	时间
<code>Place</code>	<code>string</code>	<code>private</code>		地点
<code>Character</code>	<code>string</code>	<code>private</code>		人物
<code>Mark</code>	<code>string</code>	<code>private</code>		打分

3.1.2 相册类

系统中设立关于相册的类 `PhotoAlbum`，用于该类的属性如表 3-2 所示。

表 3-2 `PhotoAlbum` 类的属性

属性	类型	可见性	初始值	注释
AlbumName	string	private		相册名
Category	string	private		类别
Photographer	string	private		作者
Content	string	private		详细内容
Datetaken	DateTime	private	DateTime.Now	相册日期
Lastdate	DateTime	private	DateTime.Now	上次更新日期
Place	string	private		地点
Character	string	private		人物
Count	int	private	0	相片数

我们需要对相册进行“添加相片”、“下一张”等操作。为了更方便地执行这些操作，我们考虑使用 `ICollection` 接口。`ICollection` 接口定义所有集合的大小、枚举数和同步方法，派生于 `IEnumerable`。它定义了集合类最基本的行为。`PhotoAlbum` 类继承于这一接口，并使用泛型 `<Photograph>` 来管理相册。`Collection<Photograph>` 提供了一些保护方法，允许一个类更改清除、添加、插入和设置操作的逻辑。

关键代码：

添加相片方法 `Add()`：

```
base.Add(p);
```

删除所有相片方法 `ClearItems()`：

```
if(Count>0)
```

```
{
```

```
Dispose();
```

```
base.ClearItems();
```

```
}
```

3.1.3 相册管理类

在管理相册时，必须先要在 `MainForm` 类中创建一个 `PhotoAlbum` 类的实例，然后可以向 `MainForm` 类添加各个事件处理方法对这个相册进行操作。这种方法固然可行，但它把管理相册的代码全部放在窗体类中，使得用户界面的代码变得复杂，而且限制了其他应用程序复用这些逻辑。因此，为了更好地实现代码复用，我们实现了一个 `AlbumManager` 类来处理这些操作，并且在用户界面中通过这个类与相册进行交互。这种方法很清楚地将用户界面可以完成的任务放在一个单独的类中，从而保证了 `Photograph` 和 `PhotoAlbum` 类的“纯洁性”，使它们简单地表示相片和相册这两个抽象概念。图 3-1 显示了 `AlbumManager` 类的规划。

程序清单

```
public class AlbumManager
{
    //Properties:
    //Album-管理的相册实例
    //Current-当前位置的照片
    //Index-当前位置

    //Methods:
    //MoveNext()-下一张
```

图 3-1 AlbumManager 类的规划

3.1.4 相册存储类

根据 3.1 节所介绍的相册存储结构，当我们添加相片时，需要将相片复制到该存储目录的相应相册文件夹下，并初始化相片目录(Album.xml)中的相片属性；同样，删除相片时，不仅要删除相册文件夹下的相片文件，也要对相片目录进行更新。读取和删除相册也同样要进行类似的操作。

为了实现代码复用，我们同样将从磁盘中读取、添加、删除相册和相片的方法安排在一个 AlbumStorage 类中。图 3-2 显示了 AlbumStorage 类的公共方法的基本规划。

程序清单

```
public class AlbumStorage
{
    static public PhotoAlbum ReadAlbum(string albumname)
    static public void AddPhoto(string fileName, PhotoAlbum album)
    static public void RemovePhoto(string filePath, PhotoAlbum album)
    static public void DeleteAlbum(PhotoAlbum album)
}
```

图 3-2 AlbumStorage 类的规划

3.2 数据文件的设计

为了将用户的相片集中管理，本系统在计算机硬盘的某个位置开辟一块空间用于保存相片及其相关信息。系统默认保存目录为：我的文档/EasyPhoto。EasyPhoto 内建立相册目录 xml 文档(Contents.xml)、类别 xml 文档(Category.xml)、搜索结果文档(Search.xml)和所有的相册文件夹。对于每个相册，系统设立一个

相册文件夹，用于保存相片目录(Album.xml)和所有的相片。

目录文件是整个相册管理系统的核心内容。它们都采用 XML 技术来组织。扩展标记语言 XML 是一种简单的数据存储语言，使用一系列简单的标记描述数据，而这些标记可以用方便的方式建立^[11]。之所以相册管理系统要采用 XML 进行数据存储，主要是考虑到了它有以下几个优点^[12]：

- (1) XML 通过使用一个可扩展标记集提供文档内容的更准确描述。
- (2) XML 可用标准化语法来验证文档内容。
- (3) XML 使用户与应用程序之间文件交换更容易。
- (4) XML 支持高级搜索。
- (5) XML 将文档结构与显示内容分开。
- (6) XML 改进用户响应、网络负载和服务器负载。
- (7) XML 支持 Unicode。
- (8) XML 支持文档间的高级链接。

下面将阐述这几种目录文件的具体格式设计，并给出相应的 XML 文件举例，使读者能清楚的了解每一个文件在相册管理系统中的作用。

3.2.1 相片目录文件

Album.xml 是相册的目录文件。它包含相册中所有相片的属性。图 3-3 是 Album.xml 文件的结构设计举例。

```
<?xml version="1.0" ?>
<北洋园 No="6">
  <Photo>
    <Filepath>C:\Documents and Settings\a\My Documents\EasyPhoto\北洋园\2.jpg</Filepath>
    <Title>玫瑰</Title>
    <Category>花</Category>
    <Photographer>DK</Photographer>
    <Description>beautiful</Description>
    <Date>2008-10-5 10:05:46</Date>
    <Place />
    <Character />
    <Mark>5</Mark>
  </Photo>
</北洋园>
```

图 3-3 Album.xml 文件的结构设计举例

其中，<北洋园 No="6">节点代表这是相册“北洋园”的相片目录文件。为

为了更好地管理某一相册中的相片，我们设立一个属性“**No**”作为该相册内相片的编号。“**No**”属性从 0 开始，以后每加入一张相片，便以“**No**”属性作为其文件名，并且“**No**”属性加一保存到此目录文件中。**<Photo>**节点代表一张相片，其子节点所表示的含义如表 3-3 所示。

表 3-3 <Photo>节点的所有子节点的涵义

节点名称	节点含义	节点名称	节点含义
<Filepath>	文件目录	<Date>	相片日期
<Title>	相片名称	<Place>	相片地点
<Category>	相片类别	<Character>	相片人物
<Photographer>	相片作者	<Mark>	相片评分
<Description>	相片描述		

3.2.2 相册目录文件

Contents.xml 是相册的目录文件。它除包含所有相册的属性外，还包含相片数、上次更新日期等信息。图 3-4 是 Contents.xml 文件的结构设计举例。

```
<?xml version="1.0" ?>
<Contents>
  <Album>
    <Name>北洋园</Name>
    <Category>校园</Category>
    <Photographer>DK</Photographer>
    <Count>5</Count>
    <Content />
    <Datetaken>2009-6-1 10:24:38</Datetaken>
    <Place>天大</Place>
    <Lastdate>2009-6-9 14:48:03</Lastdate>
    <Character />
  </Album>
</Contents>
```

图 3-4 Contents.xml 文件的结构设计举例

其中，<Contents>节点代表这是个相册目录文件，<Album>节点代表一个相册，其子节点所表示的含义如表 3-5 所示。

表 3-4 <Album>节点的所有子节点的涵义

节点名称	节点含义	节点名称	节点含义
------	------	------	------

<Name>	相册名称	<Datetaken>	相册日期
<Category>	相册类别	<Place>	相册地点
<Photographer>	相册作者	<Lastdate>	上次更新日期
<Count>	相片数	<Character>	相册人物
<Content>	相册内容描述		

3.2.3 类别目录文件

Category.xml 是相册的目录文件。它记录用户使用过程中添加过的相册和相片类别，并在用户编辑相册或相片属性时提供可选择的类别。图 3-5 是 Category.xml 文件的结构设计举例。

```
<?xml version="1.0" ?>
<Category>
  <C>花</C>
  <C>校园</C>
  <C>天津</C>
</Category>
```

图 3-5 Category.xml 文件的结构设计举例

其中<Category>节点代表这是一个类别目录文件，<C>代表一个类别。

3.2.4 搜索结果文件

Search.xml 文件用于保存搜索结果。其中，<Search>节点表明这是一个搜索结果文件。其子节点所表示的涵义如表 4-1 所示。

表 3-5 <Search>节点的所有子节点的涵义

节点名称	节点含义
Album	相册名称
Photo	相片名称
Filepath	文件路径

3.3 界面设计

界面设计是人与机器之间传递和交换信息的媒介。因此，在软件设计的过程中应首先设计出易于用户操作并且简洁美观的用户界面。用户界面设计在漫长的软件发展中一直没有被重视起来。其实软件界面设计就像工业产品中的工业造型设计一样，是产品的重要买点。一个友好美观的界面会给人带来舒适的视觉享受，拉近人与电脑的距离。界面设计不是单纯的美术绘画，它需要定位使用者、使用环境、使用方式并且为最终用户而设计，是纯粹的科学性的艺术设计。检验一个

界面的唯一标准就是最终用户的感受。所以界面设计要和用户研究紧密结合，是一个不断为最终用户设计满意视觉效果的过程。

经过多次修改，系统的界面设计最终采用风格采取模块化的程序样式。它由一个主界面和若干个弹出界面构成。系统的主要操作都是在主界面中进行。主界面布局最终如图 3-6 所示。弹出界面将在下一章的介绍模块实现时适当介绍。

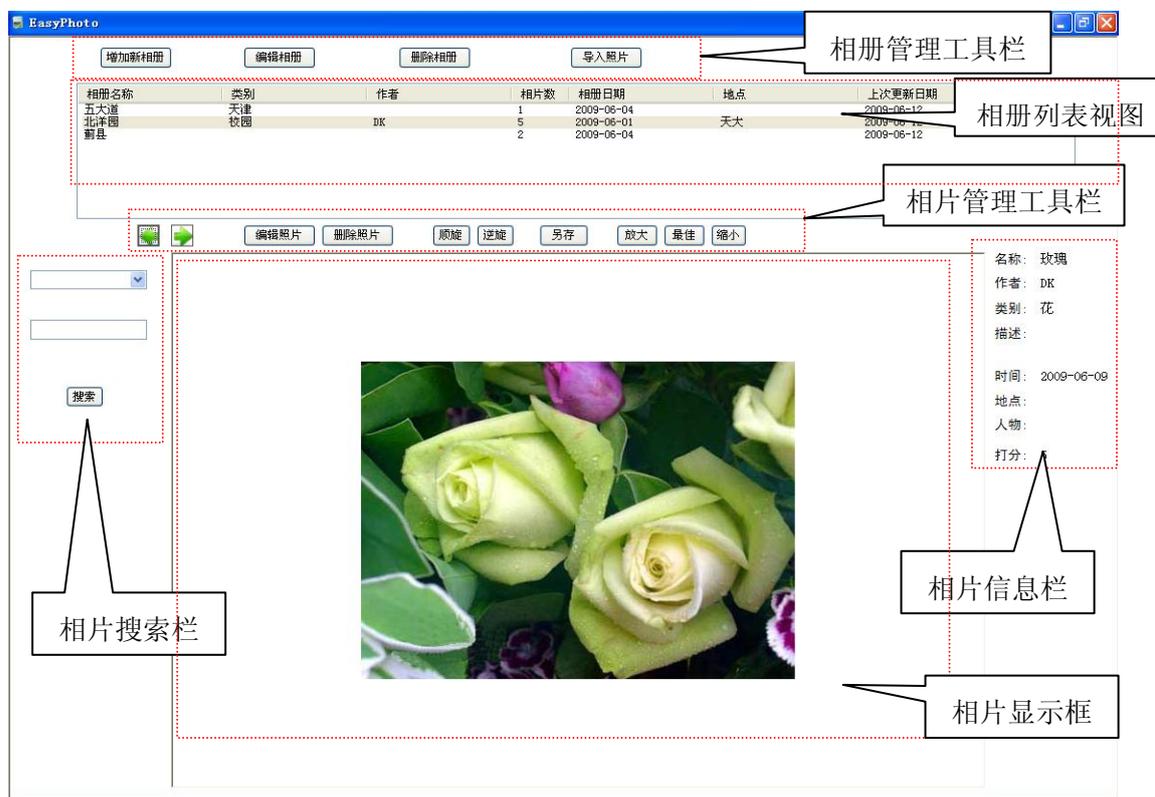


图 3-6 主界面的设计

采用此种设计方案的主要原因有两个：

(1)功能区的布局有条列、紧凑。如图 3 所示，上半部分是相册管理子系统界面，下半部分是相片管理子系统界面。上半部分包括相册管理工具栏和相册列表视图，下半部分包括相片管理工具栏、相片搜索栏、相片显示栏和相片信息栏。这种设计风格布局紧凑、功能区划分明显，使得用户很容易上手。

(2)界面简洁，便于用户使用。主界面上的主要操作区块（相册管理工具栏和相片管理工具栏）和主要显示区块（相册列表视图和相片显示框）均居于界面中央，这样用户能更方便地操作，处于用户视线范围的正中央，方便操作。

第四章 相册管理系统的实现

4.1 C#操作 XML

我们需要操作的 xml 文件主要有已知有 Contents.xml、Album.xml 和 Category.xml 三种文件，主要有插入、删除和修改三种操作。三种文件的操作基本相同，现仅对 Contents.xml 的操作方法作一介绍，另外两种文件操作方法可以类推。

4.1.1 插入节点

往节点中插入一个<Album>节点如图 4-1 所示。

```
XmlDocument doc = new XmlDocument();
doc.Load(Contentspath);
XmlNode root = doc.SelectSingleNode("Contents");//查找
XmlElement newAlbum = doc.CreateElement("Album");//创建一个节点
XmlElement newName = doc.CreateElement("Name");
newName.InnerText = album.AlbumName; //设置文本节点
newAlbum.AppendChild(newName); //添加到节点中
XmlElement newCategory=doc.CreateElement("Category");
newCategory.InnerText=类别.Text;
newAlbum.AppendChild(newCategory);
XmlElement newPhotographer=doc.CreateElement("Photographer");
newPhotographer.InnerText=作者.Text;
newAlbum.AppendChild(newPhotographer);
XmlElement newCount=doc.CreateElement("Count");
newCount.InnerText = "0";
newAlbum.AppendChild(newCount);
.....
root.AppendChild(newAlbum); //添加到节点中
doc.Save(Contentspath);
```

图 4-1 插入一个<Album>节点

4.1.2 修改节点

导入相片时，更新相册目录中相册节点的<Count>子节点和<Lastdate>子节点，如图 4-2 所示。

```
doc.Load(Contentspath);
XmlNode root2 = doc.SelectSingleNode("Contents");
XmlNodeList xnList = root2.ChildNodes; //获取 Contents 节点的所有子节点
foreach(XmlNode xn in xnList)//遍历所有子节点
{
    if (xn.SelectSingleNode("Name").InnerText == album.AlbumName)
        //如果找到
        {
            string strCount = xn.SelectSingleNode("Count").InnerText;
            album.Count = Convert.ToInt32(strCount) + 1;//转换为 int 类型再加一
            xn.SelectSingleNode("Count").InnerText = album.Count.ToString();
            //修改
            album.Lastdate = DateTime.Now;//设置为现在的时间
            xn.SelectSingleNode("Lastdate").InnerText=album.Lastdate.ToString();//修改
            break;
        }
}
xmlDoc.Save(Contentspath);//保存。
```

图 4-2 修改节点

4.1.3 删除节点

删除相册目录中的一个相册，根据<Album>节点的<Name>子节点的值判断是否删除此节点，如图 4-3 所示。

```
XmlDocument xmlDoc=new XmlDocument();//创建一个 XmlDocument 对象
xmlDoc.Load(Contentspath);//加载指定的 XML 数据
XmlNode root=xmlDoc.SelectSingleNode("Contents");//选择匹配 XPath 表达式的第一个
XmlNode
XmlNodeList xnList=root.ChildNodes;//获取节点的所有子节点
foreach(XmlNode xn in xnList)
{
    if(xn.SelectSingleNode("Name").InnerText==album.AlbumName)
    {
        root.RemoveChild(xn);//移除指定的子节点
        xmlDoc.Save(Contentspath);
        break;
    }
}
xmlDoc.Save(Contentspath);
```

图 4-3 删除节点

4.2 相册管理子系统的实现

相册管理子系统包括相片导入模块、相册属性模块和相册编辑模块三大模块。

4.2.1 相片导入模块

相册导入模块允许同时导入多张相片。在导入时可以选择相册，默认选择当前选择的相册。选择之后 `AlbumStorage.AddPhoto(s, Manager.Album)`，该函数将导入的相片属性导入相片目录文件(`Album.xml`)，并将相片复制到所选相册的存储目录，最后 `album.Add(filePath,photoTitle)`将相片添加到当前工作相册中。此外，还要更新相册的相片数和上次更新日期等信息。相片导入模块界面如图 4-4 所示。

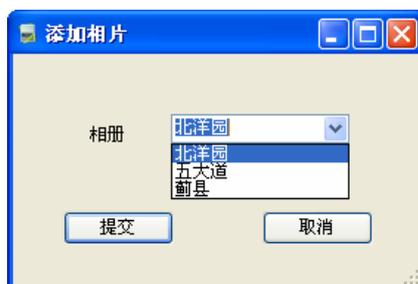


图 4-4 相片导入模块界面

4.2.2 相册属性模块

相册属性包括相册名称、类别、作者、详细内容、时间、地点、人物等。相册属性可以在增加相册时设定，也可以对已存相册进行属性设定。相册属性除了相册名是必选的，其它属性都是可选项。编辑相册窗体在打开时先读取已存在的属性，并初始化类别组合框。设定 bool 型变量记录相册被修改过的属性，如有属性被修改，才将修改结果记录到相册目录文件(Contents.xml)。相册属性模块界面如图 4-5 所示。



图 4-5 相册属性模块界面

4.2.3 相册编辑模块

主要包括增加、删除相册等功能。增加相册功能首先 `Manager=new AlbumManager()` 初始化一个相册管理类，属性设置好后 `Directory.CreateDirectory(Albumpath)` 建立相册目录，最后在相册目录 (Contents.xml) 中添加相册属性。增加相册界面与相册属性编辑界面相似。

删除相册功能先给予用户必要的提醒，以免错误删除相册，一旦删除相册后将无法恢复。接着 `Manager.Album.Clear()` 调用 `Collection<Photograph>` 接口函数清除相册，最后调用 `AlbumStorage.DeleteAlbum(Manager.Album)` 删除相册记录和相片目录。

4.3 相片管理子系统的实现

相片管理子系统包括相片属性模块、相片编辑模块和相片显示模块三大模块。

4.3.1 相片属性模块

相片属性包括相片名称、类别、作者、描述、时间、地点、人物等。相片导入相册后，可以对其进行属性设定。相片属性除了相片名是必选的，其它属性都

是可选项。编辑相片窗体在打开时先读取已存在的属性，并初始化类别组合框。设定 bool 型变量记录相片被修改过的属性，如有属性被修改，才将修改结果记录到相片目录文件(Album.xml)。图 4-6 所示的是编辑相片属性的界面



图 4-6 编辑相片属性的界面

4.3.2 相片编辑模块

主要包括删除相片、顺旋、逆旋、另存等功能。

删除相片时同样要给予用户必要的提示，以免删错相片。`Manager.Album.RemoveAt(Manager.Index)`调用 `Collection<Photograph>`的接口函数删除管理器中的相片，接着调用 `AlbumStorage.RemovePhoto(filePath, a)`删除相片记录和相片。

`Bitmap` 类的 `RotateFlip()`方法可以实现以不同的翻转方式显示图像。顺旋即是 90 度旋转(`bitmap.RotateFlip(RotateFlipType.Rotate90FlipNone)`)，逆旋即是 270 度旋转(`bitmap.RotateFlip(RotateFlipType.Rotate270FlipNone)`)。旋转完即保存到了硬盘中。

另存功能只需使用 `SaveFileDialog` 就行轻松地实现，调用 `Manager.Current.Image.Save(dlg.FileName)`保存相片。

4.3.3 相片显示模块

主要包括显示、下一张、上一张、放大、缩小等功能。

相片使用 `PictureBox` 控件显示，`SizeMode` 属性选择 `CenterImage` 模式，相片将用最合适控件的大小显示。用户可以放大、缩小相片，以便更好地查看相片。当相片不合适时，也可以点击“最佳”按钮选择 `Zoom` 模式查看相片。

上一张和下一张功能分别调用 `Manager.MoveNext()`和 `Manager.MovePrev()`即可实现。

4.3.4 相片搜索模块

此模块的操作流程是：用户首先选择相片的搜索条目，然后输入搜索条件，点击搜索按钮后弹出搜索结果列表视图，用户选择搜索结果后跳回主界面，显示选择的相片。搜索流程如图 4-7 所示。

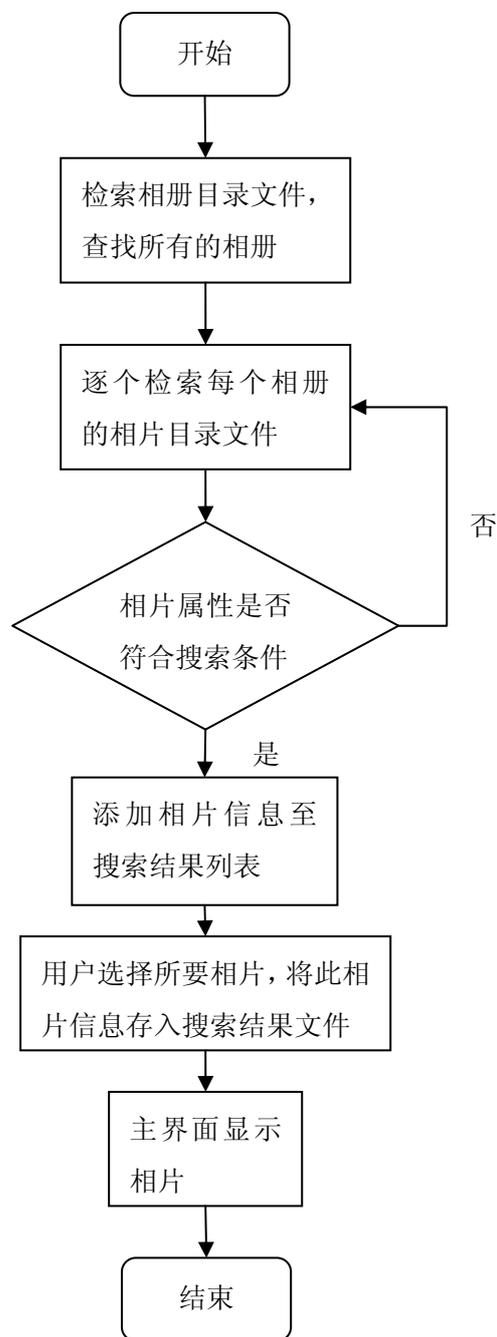


图 4-7 相片搜索模块流程图

图 4-8 是主界面上的搜索输入栏。组合框可以选择相片的各个属性，文本框用于输入搜索条件。



图 4-8 搜索输入栏

为了实现搜索功能，在此增加一个“搜索”窗体。主界面将搜索条目和搜索条件传给该窗体。“搜索”窗体接收后检索相册系统中所有的 Album.xml 文件，将符合条件的相片显示在一个列表视图中，如图 4-9 所示。

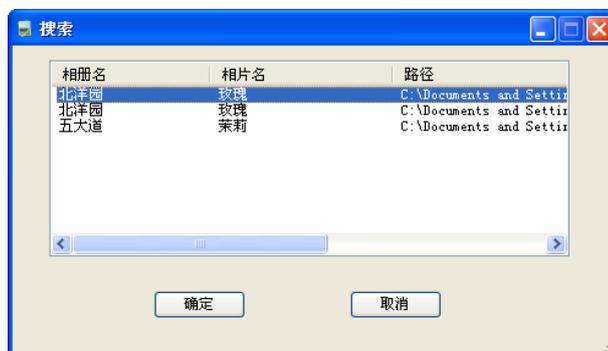


图 4-9 搜索结果显示界面

用户选择一张需要的相片，点击确定后，系统将该相片的必要信息（相册名、相片名和文件路径）存入搜索结果目录文件（Search.xml）中。接下来，主界面调用 `search_item(albumname)` 找到所在相册，然后累加 `Manager.Index`，直到当前相片的 `FileName` 属性与上述选择的路径相符。

第五章 结论

5.1 总结

本相册管理系统经过本人数月的努力，终于有了一定的成果。本系统的目的是帮助人们更好地管理自己的相片，现在这个目的基本实现了。

本系统与其它相册管理软件相比，具有非常明显的特点：方便、功能齐全、页面精美、界面精美。

方便：本系统开发前专门针对用户的使用习惯进行了调查，对用户常用的功能进行了整合，并对系统的菜单优先级进行了调整，使用户一眼就能看到自己需要的功能菜单。完全傻瓜式的引导操作，再加上简单易见的菜单，一个刚接触电脑的人也能很快掌握操作方法，因此具有方便的特点。

功能齐全：本相册系统专门研究了其他各大相册软件的功能，综合了他们的优点，把常用、受用户欢迎的一些功能集合起来，并进行了操作上的改良，方便用户使用，本系统的功能非常齐全。

界面精美：本相册系统比较注重用户的直观感受，因此在界面设计时特别注意界面的美观。

通过本次毕业设计，我也学到了很多。通过亲身实践一款软件的开发和毕业论文的写作，我更深刻地理解了软件开发的全过程，锻炼了自己的编程能力和论文写作能力，更让我看到了我的不足。总之，此次毕业设计让我受益匪浅。

5.2 展望

由于能力和时间的限制，本相册管理系统存在着许多的不足，主要问题有：

（1）相片处理功能不够强大，例如一些专业图片处理软件所具有的补光、消除红眼等功能；

（2）网络相册管理功能暂时没有实现。

以上这些功能都是今后工作需要完善的地方。相信，经过我的不懈努力，本相册管理系统将变得越来越强大。

参考文献

- [1] 卫红春等.软件工程概论[M].北京：清华大学出版社，2007.54-55.
- [2] 姜淑娟，刘迎春，荣新节等.软件工程与软件开发工具[M].北京：清华大学出版社，2004.23-25.
- [3] 林锐.软件工程与项目理解析[M].北京：电子工业出版社，2003.67.
- [4] 张立.C#程序设计编程经典[M].北京：清华大学出版社，2008.9-12.
- [5] 刘焯,季石磊等.C#编程及应用程序开发教程[M].北京：清华大学出版社，2007.15-16.
- [6] Erik Brown.Windows Forms 编程实战[M].北京：机械工业出版社,2008.84.
- [7] 王小科,吕双等.C#从入门到精通编著[M].北京：清华大学出版社,2008.46-48.
- [8] Simon Robinson, Christian Nagel.C#高级编程.北京：清华大学出版社,2005.127-128.
- [9] Scott Allen,Syed Fahad Gilani 等.C#数据库入门经典.北京：清华大学出版社,2006.54-55.
- [10] Karli Watson,Christian Nagel 等.C#入门经典.北京：清华大学出版社,2008.88-89.
- [11] H.M.Deitel,P.J.Deitel 等.XML 编程技术大全.北京：清华大学出版社，2002.23-25.
- [12] Frank P. Coyle 著. XML、Web 服务和数据革命.北京：清华大学出版社，2003.43-44.
- [13] W3 Schools.XML Tutorial[EB/OL].<http://www.w3schools.com/>,December 1999.
- [14] A.Loui and M.Wood.A Software System for Automatic Albuming of consumer Pictures[EB/OL].<http://www.php.net>,2003.
- [15] D.Wang, A.Vagnucci and C.Li, Digital image enhancement: A survey [M]. Computer Version,Graphics,and image Processing, 1983.Vol.24,363-381.

致 谢

本人学位论文工作是在我的导师张坤龙副教授的悉心指导下完成的。在整个论文工作期间，张老师对我的论文的选题、论文的结构各个环节都提供了非常多的指导。无论是在系统的开发方法和论文写作方法上，还是在实际问题的解决办法上，张老师都给了会让我受益终生的教诲。特别是导师的严谨治学态度，高效务实工作作风也令我受益非浅。

最后衷心感谢我的家人、室友和朋友，是他们对我的关心、支持和帮助才使我能够安心地做毕设，才能够顺利地完成本论文。