

基于 LINQ 技术的小区物业管理系统^①

吴伶琳

(健雄职业技术学院 软件与服务外包学院, 太仓 215411)

摘要: 为了适应小区物业业务管理扩展及服务的需求, 本文利用 B/S 体系架构和 LINQ to SQL 技术, 设计并实现了小区物业管理系统. 该系统为高效的物业管理提供了信息化平台, 减轻了物业管理人员的工作. 文中简要概括了该系统的系统结构、开发模式及核心功能. 开发实践证明, 基于 LINQ 的数据库访问编程模型简单快捷, 降低了编程难度, 提高了编程效率.

关键词: 物业管理; LINQ; 应用; B/S 架构; LINQ to SQL

Residential Property Management System Based on LINQ

WU Ling-Lin

(School of Software and Service Outsourcing, Chien-shiung Institute of Technology, Taicang 215411, China)

Abstract: In order to meet the needs of service and the extension of residential property business management, the paper mainly realizes a residential property management system which based on B/S model and adopting LINQ to SQL technology. The system provides an information platform for the efficient management of property and reduces the work of the property managers. In this paper, the system architecture, development pattern and the core functions of the system is briefly introduced. It is proved by implementation that this kind of programming pattern is not only simple and quickly, but also reduces the difficulty of programming and raises programming efficiency.

Key words: residential property management; LINQ; application; B/S structure; LINQ to SQL

随着我国经济的快速发展以及人民生活水平的不断提高, 人们对周围环境及社区服务水平提出了更高的要求. 计算机技术的飞速发展及广泛应用, 使得传统的物业公司的业务处理方式必然被以信息技术为基础的信息化管理方式所取代. 开发基于网络的小区物业管理系统, 将传统的人工管理转变为计算机软件辅助管理的高效方式, 将解决物业公司面临的占地大、规模广, 用户多、变化快, 缴费多、汇总难的问题, 有助于提高物业管理的信息化水平, 从而更好的为社区居民服务. 网络化与智能化的管理服务在当前与今后一段时间内已成为物业管理企业竞争制胜的关键筹码^[1].

本文将介绍运用 LINQ 技术实现 B/S 架构的小区物业管理系统, 该系统选用 Visual Studio 2010 作为开发平台, 采用 SQL Server 2008 作为数据库存储数据.

1 系统需求分析

通过走访相关物业公司及小区业主, 笔者对该系统的需求进行详细分析与研究, 最终确定要开发的小区物业管理系统能够满足以下目标:

- (1) 实现小区公告信息的发布与浏览.
- (2) 满足对小区用户、管理员等人员基本信息的管理.
- (3) 满足对楼宇、房屋、区域等信息的管理及查询, 如能按照业主姓名、房号、房屋类型、面积等多种条件进行查询.
- (4) 提供小区业主对服务投诉、设备报修等各类服务.
- (5) 满足对物业费、停车费等费用的收缴、统计及查询.

^① 基金项目:江苏省教育厅高等教育教改研究课题(2011JSJG307);太仓市科技基础设施建设项目(太科字[2012]59号)

收稿时间:2013-04-28;收到修改稿时间:2013-05-27

(6) 实现对停车场、停车位的信息管理。

2 系统设计

2.1 系统体系结构

本系统利用 B/S(浏览器/服务器)架构、采用微软的.NET 技术构建了小区物业管理系统. 用户界面完全通过 IE 浏览器实现, 一部分事务逻辑在前端实现, 但是主要事务逻辑在服务器端实现^[2].

为了满足系统需求, 在 Microsoft .NET Framework 的三层架构下进行设计和处理所有的应用程序, 即数据访问层、业务逻辑层和表示层. 数据访问层使用 LINQ 技术来封装对数据库的各类操作, 如针对数据的增、删、改、查等; 业务逻辑层围绕小区物业管理的业务流程实现具体处理, 将各种业务封装在类中; 而表示层则给用户程序的访问接口, 主要由 ASP.NET 窗体来实现. 系统的体系结构如图 1 所示.

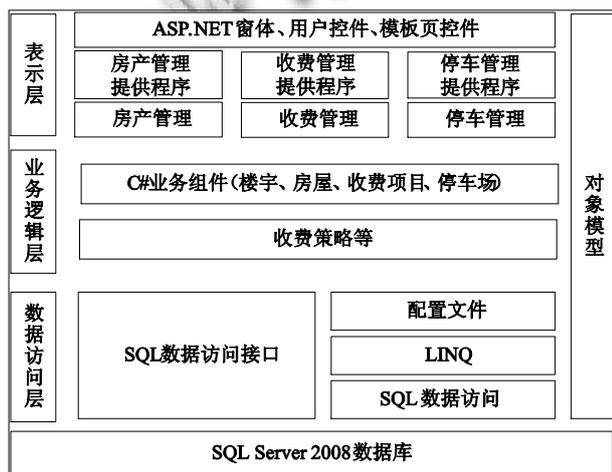


图 1 小区物业管理系统体系结构图

2.2 系统功能设计

小区物业管理系统的功能主要围绕住户服务及小区的各项业务管理进行设计, 根据实际需求本系统将小区物业管理系统分为前台和后台两大部分. 前台主要是针对普通住户进行服务的, 包括网站首页、用户登录、小区公告、用户投诉、用户报修等功能. 后台主要是物业公司的管理人员对小区的日常工作、各项信息进行管理及维护, 可以将其分为 7 个模块, 即: 房产信息管理、住户信息管理、设备信息管理、收费信息管理、停车信息管理、公共信息管理和系统管理, 功能模块如图 2 所示.

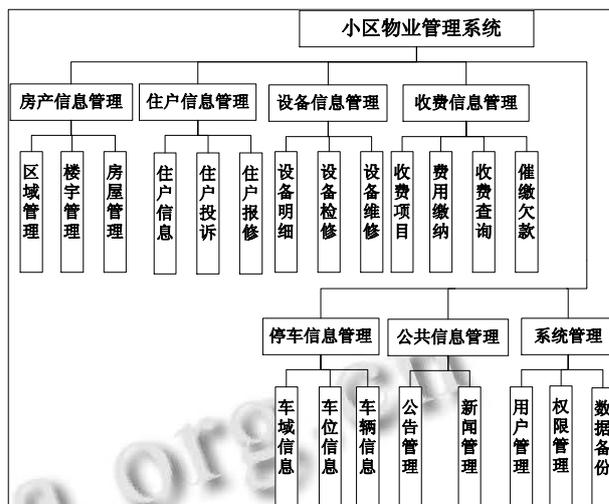


图 2 小区物业管理系统后台功能图

(1) 房产信息管理

房产信息管理模块主要用于对区域信息、楼宇信息、房屋信息的管理, 如楼宇信息的管理中实现楼宇信息录入、查询、修改、删除等功能. 房产信息管理中可记录房屋的使用状态是入住、出租或出售, 并向用户提供可供租赁及销售的房屋信息的查询.

(2) 住户信息管理

住户信息管理模块主要包括小区住户的人员信息、住户投诉、住户报修功能. 如住户信息管理中可以对住户的基本信息进行增加、删除、修改、查询等.

(3) 设备信息管理

设备信息管理模块主要实现对小区所拥有的设备进行添加与维护, 同时还要对住户所提出的报修进行即时的处理, 反映报修项目的原因及上报时间等信息.

(4) 收费信息管理

收费信息管理模块主要用于对住户的水电煤气、停车费、物业费费用的收缴, 并对收费超出期限进行设置, 当有住户未能按时缴纳费用时, 系统会进行提醒.

(5) 停车信息管理

停车信息管理模块主要对于停车位的基本信息、住户的车辆信息、停车场的信息进行管理.

(6) 公共信息管理

公共信息管理模块主要是用来向小区内的住户发布公告和新闻, 方便小区内的住户对信息进行浏览.

(7) 系统管理

系统管理模块主要用于对登录用户的信息进行管理,

分配用户的使用权限,并定期对系统数据进行备份。

以上各个模块有着相对独立性,可以分别加以设计实现,另一方面,模块之间可以通过信息或调用关系等互相关联,共同构成一个整体^[3]。

3 系统实现的关键技术

3.1 安全机制

3.1.1 身份验证

小区物业管理系统采用 ASP.NET 成员资格实现用户权限的管理。ASP.NET 成员资格是微软提供的一套通用的用户权限管理模块。通过它可以创建、删除、编辑用户,对用户角色进行管理,实现用户登录、密码找回等功能^[4]。本系统中笔者将 ASP.NET 成员资格与 ASP.NET 窗体身份验证一起使用,创建了一个完整的用户身份验证系统。

3.1.2 数据完整性和一致性控制

通过定义主键、外键、检查约束、非空约束、规则等方式来保证数据的完整性,在应用程序级和数据库级都进行并发控制的方法来保证数据的一致性^[5]。SQL Server 2008 提供了强大的并发控制机制,如可以通过事务处理来保证操作的完整性,确保信息的一致性。而 LINQ 技术提供了对并发数据处理的支持,通过两者的合理应用,建立起完整的数据一致性体系。

3.2 LINQ 技术

LINQ 是 Language Integrated Query 的英文缩写,中文意思是语言集成查询。LINQ 是一系列的编程接口,是微软公司在 Visual Studio 2008 和 .NET Framework 3.5 版本中的一项突破性创新,并在对象领域和数据领域架设了一座桥梁。LINQ 的主要设计目标是大幅度提高开发数据访问应用程序的效率,降低实现数据访问应用程序的难度,并且增加相应的程序的功能扩展性^[6]。

.NET Framework 3.5 提供了 3 种 LINQ Provider,分别是 LINQ to OBJECT、LINQ to ADO.NET 和 LINQ to XML^[7]。本系统主要使用的 LINQ to SQL 是 LINQ to ADO.NET 的子类之一,它可以将关系数据库的对象(如数据表、视图等)映射到编程语言中定义的对象模型来实现对数据的相关操作,即在关系数据库的数据表和编程语言对象之间创建了一个对象关系映射层。这个对象关系映射将数据源的关系对象和 LINQ to SQL 对象模型一一对应,为使用和管理关系对象提供

了基本元素^[8]。程序开发人员需先改变对象模型的相关值,并执行其中的方法提交到系统方可改变数据库中的数据。

4 系统实现

本系统的实现中大量使用了 LINQ 技术,现以房产信息管理模块为例,具体介绍使用 LINQ 创建数据源和向楼宇表(Pavilion)中插入、更新、删除数据的过程。图 3 为小区物业管理系统的使用界面。



图 3 小区物业管理系统的房产信息管理模块截图

4.1 应用 LINQ 创建小区物业管理系统的类模型

将“LINQ to SQL”类添加到小区物业管理系统项目中,命名为 Houses.dbml 并保存在 App_Code 文件夹中。系统会自动打开对象关系设计器,通过将小区物业管理系统中的数据对象(如数据表、视图)从服务器资源管理器中拖动到该设计器中,Microsoft Visual Studio 2010 会自动生成包含相关属性、方法的 HousesDataContext 类和实体类,与数据库的相关对象相关联,不需要用户编写太多的代码。在以后的编程中,可以直接使用 HousesDataContext 类对数据对象进行访问。图 4 是本系统数据库主要数据表的 ORM 映射关系图。

连接该数据库的字符串保存在 Web.config 配置文件中,其主要代码如下:

```
<configuration>
  <appSettings/>
  <connectionStrings>
    <addname="HousesConnString"
      connectionString="Data Source=.;Initial
```

```
Catalog=Houses;uid=sa;pwd=sa"
providerName="System.Data.SqlClient" />
    </connectionStrings>
</configuration>
```

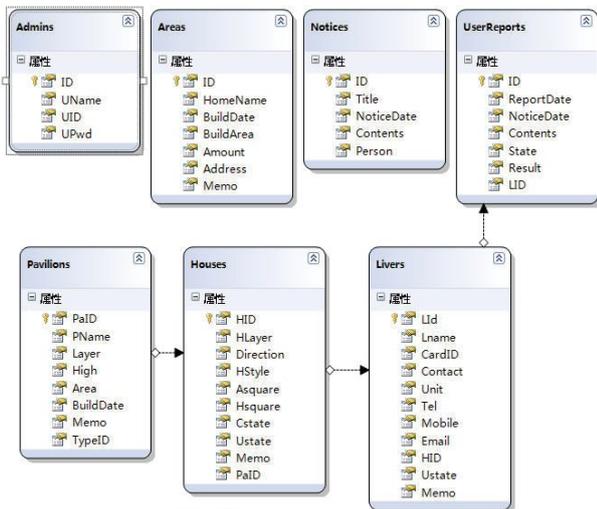


图 4 系统中主要数据表 ORM 映射关系图

4.2 应用 LINQ 实现数据插入

首先需要创建一个 HousesContext 对象来管理与数据库的通信；然后将业务逻辑层传递过来的一个 Pavilion 对象插入 HousesContext 类中类型为 Table<Pavilion> 的 tb_Pavilion 表中；最后，调用 SubmitChanges() 方法将该 Pavilion 对象提交到数据库中。主要代码如下：

```
public string AddPavilion (Pavilion pavilion)
{
    //生成一个 HousesContext 对象 ctx 来管理与数据库的通信
    HousesDataContext ctx = new
        HousesDataContext();
    //将需要增加的对象添加到 HousesContext 对象里面, 该对象来自业务逻辑层
    ctx.Pavilion.InsertOnSubmit(pavilion);
    try
    {
        //通过 HousesContext 对象将修改保存到数据库中
        ctx.SubmitChanges();
        return "插入成功";
    }
}
```

```
catch
{
    return "插入失败";
}
}
```

4.3 应用 LINQ 实现数据更新

通过修改 LINQ to SQL 对象的值，成员可以进行数据行的更新。在此之前，先要通过查询找到需要更新的特定记录，并修改该记录的相关字段。更新后必须调用 SubmitChanges() 方法，才能更新数据库中的记录，否则数据库中的内容不会被修改。主要代码如下：

```
public string UpdatePavilion(Pavilion pavilion)
{
    //获取需要更新的楼宇的编号
    int paID = pavilion.paID ;
    //生成一个 HousesContext 对象 ctx 来管理与数据库的通信
    HousesDataContext ctx = new
        HousesDataContext();
    //查询需要修改的对象
    var pa = (from pavilion in ctx.Pavilion
        where pavilion.paID == paID
        select pavilion).SingleOrDefault();
    //修改查询出来的对象
    pa.Name = pavilion.Name;
    pa.layer = pavilion.layer;
    pa.high = pavilion.high;
    pa.area = pavilion.area;
    pa.buildDate = pavilion.buildDate;
    pa.memo = pavilion.memo;
    try
    {
        //将修改的对象保存到数据库中
        ctx.SubmitChanges();
        return "更新成功";
    }
    catch
    {
        return "更新失败";
    }
}
```

4.4 应用 LINQ 实现数据删除

删除数据行的操作可以通过删除 LINQ to SQL 对象来实现,主要通过调用 DeleteAllOnSubmit()方法来删除实体集合.例如在 Pavilion 表中,将业务逻辑层传递过来的实体集合删除,主要代码如下:

```
public string DeletePavilionBypaID(string paID)
{
    //如果楼宇内有住户,请先删除房间信息,代码省略
    .....
    //生成一个 HousesContext 对象 ctx 来管理与数据库的通信
    HousesDataContext ctx = new
        HousesDataContext();
    //查询出需要删除的对象
    var pa=from item in ctx.Pavilion
        where item.paID == paID
        select item;
    ctx.Pavilion.DeleteAllOnSubmit(pa);
    try
    {
        //通过 HousesContext 对象将修改保存到数据库
        ctx.SubmitChanges();
        return "删除成功";
    }
    catch
    {
        return "删除失败";
    }
}
```

5 结束语

作为新一代的数据访问技术, LINQ 优秀的语法查

询机制对开发人员提高项目效率有很大的现实意义^[9]. 本文介绍的小区物业管理系统的数据库访问层代码全部采用 LINQ 技术来实现,因此在编写这些代码时可以获得编译时类型检查和 IntelliSense 支持,提高了编程效率,大大节约了开发时间和成本.小区物业管理系统运行以来,整体程序运行良好,在社区居民中反响良好,有效地提高了物业管理的信息化水平,减轻了物业管理人员的工作负担,提高了他们的工作效率和管理水平.

参考文献

- 1 刘娟.基于.NET 的小区物业管理系统的设计与实现[硕士学位论文].上海:华东师范大学,2010.
- 2 王飞,李虹,韩志勇,韩建宁.基于 GIS 的煤矿地理信息系统的研究与开发.微计算机信息,2008,24(8-1):230-231.
- 3 Mladenic D. Personal web watcher. Design and Implementation. Department of Intelligent Systems, Institute,1996.
- 4 魏宗仪.基于.NET 的小区物业管理系统设计[硕士学位论文].山东:山东大学,2009.
- 5 文春生,罗勇.基于 WebService 技术的教务管理系统的设计与实现.计算机工程与科学,2004,26(9):79-81.
- 6 王的强,张晓娜,周静.LINQ 入门及应用.北京:清华大学出版社,2009:3-5.
- 7 解本巨,李宗颜,宫生文.LINQ 从基础到项目实践.北京:化学工业出版社,2010:4-6.
- 8 牟永敏,丁媛,朱绪利,李良杰.基于 LINQ 技术的在线考试系统.北京信息科技大学学报(自然科学版),2011,26(2):19-24.
- 9 王小花,李红霞,袁媛.基于 LINQ to SQL 的实验室网上预约系统的设计与实现.计算机与现代化,2012(11):81-84.