**江苏信息职业技术学院毕业论文**

题目**基于Restful API的项目实施管理系统**

系 别： 物联网工程学院(信息安全学院)

专 业： 软件技术

学 号： 1617203205

姓 名： 陈小洋

指导老师： 王 辉

2018年 11月25 日

# 前言

在互联网飞速发展的网络时代，各种各样的工作都依赖于计算机，其目的主要是为了提高工作效率从而带来更大的收益。而各个企业在第三产业发展的浪潮中，也逐渐融入高效的计算机文化中，或者说是企业的发展在很大程度上已经离不开计算机与网络。而各大企业在发展的过程中不断引进互联网技术对公司经营进行科学合理的管理，但是在互联网技术呈现级数增长的21世纪，很多企业无法及时引进最新的技术从而导致企业内部仍然使用已经过时的淘汰的技术成果，常常导致企业内部使用的系统、软件等，操作不方便系统臃肿。为了改善这一情况，广大企业都投入大量的人力财力对这一步分进行优化与升级。

开发一个基于Restful API的项目实施管理系统，也是为了提高企业的运营效率，并获得最大的收益，同时替换原本臃肿复杂的系统，节省产品的生产成本。当然这个系统针对性比较强，因为是结合实际应用而开发的一个辅助管理的系统。

该系统采用的均是目前比较靠前的一些主流的技术和项目设计框架，数据传输更是采用较为科学快捷的Json流，通过URL的形式，将每一条数据进行资源定位，使得资源访问更加地准确与快捷。

# 概要

本系统根据无锡傲越软件研发有限公司的实际应用需求，基于.Net平台、重点依靠Restful API技术实现的项目实施管理系统，该系统追求程序的高拓展性以及高效的数据信息处理速度，旨在尽可能多地提高工作者的工作体验以及工作效率，采用灵活多样化的开发方式极大地降低系统模块代码之间的耦合率。该系统通过Windows桌面应用程序和Web端相结合的方式实现了所有的业务需求，较为机密的系统核心采用Restful API技术和Web Service技术托管在公司私有服务器，全部以接口的形式供应用程序使用，以防止公司内部数据层次结构的泄露，造成不必要的经济损失。

关键字：项目实施管理系统；Restful API；Basic身份验证；SQL Server 2016；MVC框架；web Service；Ajax

# ABSTRACT

The system is based on the actual application requirements of WuXi AoYue company, based on the .Net platform, and the project implementation management system based on Restful API technology. The system pursues high scalability of the program and efficient data processing speed. Designed to maximize the worker's work experience and productivity, and flexible and versatile development methods greatly reduce the coupling ratio between system module codes. The system realizes all the business requirements through the combination of the Windows desktop application and the Web. The more confidential system core is managed by the Restful API technology and the Web Service technology on the company's private server, all of which are used by the interface. To prevent the leakage of the company's internal data hierarchy, resulting in unnecessary economic losses.

Keywords: project implementation management system; Restful API; Basic authentication; SQL Server 2016; MVC framework; web service; Ajax

# 目录

[前言 II](#_Toc533681600)

[概要 III](#_Toc533681601)

[ABSTRACT IV](#_Toc533681602)

[目录 V](#_Toc533681603)

[一、系统概述 1](#_Toc533681604)

[1.1系统分析 1](#_Toc533681605)

[1.1.1背景介绍 1](#_Toc533681606)

[1.1.2需求分析 2](#_Toc533681607)

[二、系统概述与总体设计原理 3](#_Toc533681608)

[2.1项目实时管理系统模块的功能原理 3](#_Toc533681609)

[2.2项目实时管理系统模块的构成模块 3](#_Toc533681610)

[2.3规划系统文件 4](#_Toc533681611)

[2.3.1规划文件 4](#_Toc533681612)

[2.3.2选择开发工具 6](#_Toc533681613)

[三、数据库设计 7](#_Toc533681614)

[3.1数据库描述 7](#_Toc533681615)

[3.2数据库分析 7](#_Toc533681616)

[3.3数据库概念结构设计 7](#_Toc533681617)

[3.3.1系统基本表 7](#_Toc533681618)

[3.3.2项目信息表 8](#_Toc533681619)

[3.3.3单位信息表 8](#_Toc533681620)

[3.3.4人员信息表 9](#_Toc533681621)

[3.4数据库逻辑结构设计 9](#_Toc533681622)

[3.4.1系统基本表 9](#_Toc533681623)

[3.4.2项目信息表 11](#_Toc533681624)

[3.4.3单位信息表 12](#_Toc533681625)

[3.4.4人员信息表 13](#_Toc533681626)

[3.5数据库系统参数配置 14](#_Toc533681627)

[3.6数据访问层设计 15](#_Toc533681628)

[3.6.1 用户登录模块 16](#_Toc533681629)

[3.6.2 项目信息模块 17](#_Toc533681630)

[3.6.3 单位信息模块 18](#_Toc533681631)

[3.6.4 人员信息模块 18](#_Toc533681632)

[四、系统编码实现 19](#_Toc533681633)

[4.1Restful API接口设计 19](#_Toc533681634)

[4.1.1 Restful API接口说明 19](#_Toc533681635)

[4.1.2 Restful API接口设计 20](#_Toc533681636)

[4.2用户登录模块 21](#_Toc533681637)

[4.2.1登录模块 21](#_Toc533681638)

[4.2.2注册小模块 22](#_Toc533681639)

[4.2.3服务器配置 24](#_Toc533681640)

[4.3项目信息模块 25](#_Toc533681641)

[4.4单位信息模块 29](#_Toc533681642)

[4.5人员信息模块 30](#_Toc533681643)

[4.6附属功能模块 31](#_Toc533681644)

[五、系统测试 32](#_Toc533681645)

[5.1测试工具 32](#_Toc533681646)

[5.2功能性测试、非功能性测试 32](#_Toc533681647)

[5.2.1 功能性测试 32](#_Toc533681648)

[5.2.2 非功能性测试 34](#_Toc533681649)

[六、发布及管理 36](#_Toc533681650)

[6.1本地发布 36](#_Toc533681651)

[6.1.1 本地发布概述 36](#_Toc533681652)

[6.1.2 本地文件发布 36](#_Toc533681653)

[6.2.3 本地IIS目标IP发布 38](#_Toc533681654)

[6.2 远程服务器发布 39](#_Toc533681655)

[6.2.1远程服务器发布概述 39](#_Toc533681656)

[6.2.2远程服务器环境搭建 39](#_Toc533681657)

[6.2.3远程服务器连接 40](#_Toc533681658)

[结束语 41](#_Toc533681659)

[致谢 42](#_Toc533681660)

[参考文献 43](#_Toc533681661)

# 一、系统概述

## 1.1系统分析

### 1.1.1背景介绍

随着计算机硬件的不断更新与性能优化，以及大数据时代的到来，无论是生活、工作、医疗、教育都越来越重视效益。所谓效益，是指效率和收益的结合体，在单位时间内完成相应的工作，并取得高于成本的回报。脱离效益，所做的工作以及前期的付出，甚至是前期的投资，都是没有意义的，属于一种亏损状态。

在追求高效益的时代背景下，仅仅依赖于人力是无法满足生产需求的，而此时计算机软件则体现出巨大的优势，根据生产需求，编写相关软件在短时间内完成人力无法完成的工作量，同时由于计算机严谨的计算机逻辑，可以实现零失误，极大提高了效率，效率提高的同时结合科学合理的运营模式或者相关后续的数据使用，基本上能够获得极大的效益。

随着网络技术的不断发展，越来越多的企事业单位开始注重管理层的管理效益，当今市面上的比如钉钉、协同已经算是不错的管理考核系统，虽然已经很成熟了，但是仅限于普通办公，不能满足无锡傲越软件研发有限公司的实际应用需求。结合该公司现状，目前已有系统存在以下几个问题：

1. 成本偏高，采用Win Form框架与传统的Access数据库相结合的方式，随着企业规模的扩大，维护成本渐趋升高。
2. 人力资源浪费，由于不具备数据录入以及导出，在数据管理这方面投入了过多的人力资源，大大延缓了工作效率，同时由于是多人操作，带来了极大的出错风险，容错率极低。
3. 现有系统功能不完善，稳定性不够高，系统功能冗余复杂，操作不便。常发生由于操作不正确导致程序崩溃或者死机等状况。
4. 现有系统诸多方面设计不合理，导致程序不易拓展与维护。原本轻便简洁的数据库变得臃肿起来，各数据表之间关系错综复杂关系不够清晰，违反数据库的范式设计法则。程序框架单一，拓展性极差。

鉴于以上存在的问题，该公司决定对该系统做整体的更新换代。

### 1.1.2需求分析

无锡傲越软件研发有限公司是本地的一家大型企业，主要从事于承包各地的软件外包开发以及测试业务，为了应对不断扩大的企业规模以及在未来的发展中能够抓住机遇与面临新的挑战，该公司力争降低生产成本提高企业生产效益。

经过公司内部对公司业务现状分析以及对未来发展规划分析，要求新开发的系统必须具备以下主要功能：

1. 对系统特殊模块操作权限具备一定的控制，即用户分级别。
2. 添加项目信息以及人员信息，实时追踪项目实施进程。
3. 浏览所有项目报表，可手动导出项目明细。

在系统新能方面该公司要求：

1. 系统健壮性要好，杜绝死机与崩坏的现象。
2. 系统反应迅速，尽量避免卡顿延时现象。
3. 考虑到公司的业务发展，程序要具备一定的可拓展性，即随时随地增加新的模块和功能。
4. 低维护成本。

该公司预计系统开发预算5000，工期3个月。

# 二、系统概述与总体设计原理

## 2.1项目实时管理系统模块的功能原理

基于Restful API的项目实施管理系统的实现原理非常清晰，其主要目的是实现项目实施的进度管理，其次附带一些辅助性的功能，其目的是提高用户的用户体验以及应用程序的多元化、丰富应用程序的内涵、充实应用程序的内容。其各个主要模块的功能原理如下：

* 用户登录模块：本模块主要分为登录和注册两部分

1. 登录:本部分采用Basic认证，通过提交登录信息,应用程序调用Restful API接口，通过Ajax请求将接口将票据信息加入到请求的Head里面，将票据信息随着请求一起发送到服务端去，验证登录。
2. 注册：本部分基于MVC框架实现简易的Web应用，通过https的双向认证完成注册（基于系统安全性考虑），并将注册信息保存到数据库中。

* 项目信息模块：本模块是项目实施管理系统的最为重要核心模块，所有的执行方案全部通过调用Restful API接口的方式实现。
* 单位信息模块：本模块大部分功能是基于Restful API实现的，但少部分内容是直接通过封装的类库文件直接实现的。类库采用C++语言编写，主要考虑到C++语言的处理速度会相对较快。
* 人员信息模块：本模块由于涉及的内容较少，经过考虑全部由封装的类库文件直接实现（类库文件有C#编写）。

## 2.2项目实时管理系统模块的构成模块

基于Restful API的项目实施管理系统构成并不复杂，有四大主要模块构成，其次附带一个服务接口，大致的模块构成结构如图2.2-1项目实施管理系统模块结构所示。

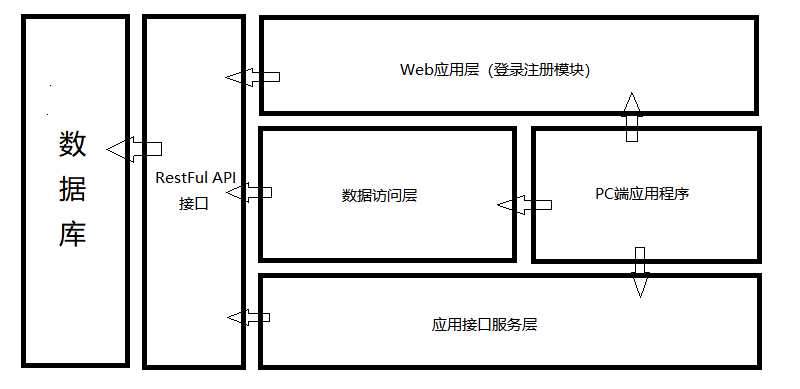


图2.2-1项目实施管理系统模块结构

## 2.3规划系统文件

### 2.3.1规划文件

系统地进行规划是一个好项目的开始，是任何项目的第一步工作，基于系统规划，能够很好地避免开发过程中由于周期太长从而导致最后偏离最原始的需求。所以针对本项目，在前期阶段就对所有需要的步骤以及文件做了合理的规划，并尝试一步到位，后期在非必要的情况下争取不对项目文件做较大的更改。预期各个文件在Visual Studio 2017解决方案管理器中效果如图2.3.1 预期规划的项目文件所示。

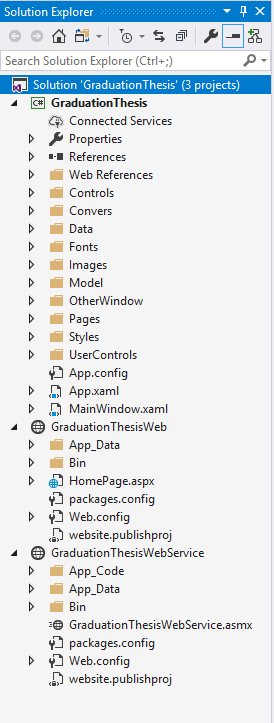


图2.3.1 预期规划的项目文件

### 2.3.2选择开发工具

#### 2.3.2.1 Visual Studio 2017 企业版

Visual Studio 2017微软主要推出了三个版本，社区版、专业版和企业版，选择企业版的主要原因是：企业版相对于其他版本而言，在诊断调试方面的优势特别突出，有利于在开发过程中，检查系统逻辑代码设计的不合理性，大大提高了提升系统性能的可能性，同时直观地反应系统运行时各个资源的使用情况，能够帮助系统开发者在开发的过程中就避免导致系统性能下降的开发方式。

#### 2.3.2.2 Microsoft SQL Server 2017 专业版

数据库种类繁多，鉴于运行环境是Windows系列的操作系统，考虑到数据库与操作系统的兼容性，数据库选择SQL Server，同时由于SQL Server具备友好的用户交互界面，在数据库的设计时能够提高开发效率，在后期也适合非专业开发人员（在非必要情况下）对数据库信息的直接管理与操作。

该公司的规模属于中小型企业，根据所开发的系统需求，从系统功能、系统并发问题、数据库的安全性和稳定性等诸多因素考虑，SQL Server也比较适合。

#### 2.3.3.3 Blend for Visual Studio 2017 企业版

Blend for Visual Studio 2017 企业版主要用于WPF框架的友好UI界面设计支持，就像Photo　shop是职业编辑图片素材一样，Blend for Visual Studio 2017 是职业设计UI界面的。之所以选择它作为开发工具之一，是因为传统的UI控件样式单一不够美观，不适应已经熟悉友好操作界面的广大用户，同时也不符合软件开发艺术。

# 三、数据库设计

## 3.1数据库描述

数据库，顾名思义“数据的仓库”，是数据集中管理的计算机科学的重要技术分支。广泛应用于各行各业，逐渐取代了传统的数据存储方式。因为数据库具有数据结构化、最低冗余度、易于拓展和编制、易于数据集中管理等优势，几乎所有的应用程序和基于数据的各类系统都是建立在数据库之上的。

本系统基于公司实际需求与前期各方面影响因素的分析采用了SQL Server数据库，作为一种常见的关系型数据库，SQL Server很好得满足了开发需求，同时满足了实际应用，能够有序合理地存储海量数据。

## 3.2数据库分析

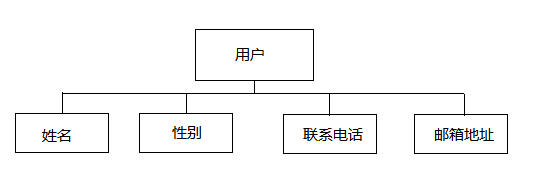
本系统基于数据库GraduationThesis进行开发，该数据库存储主要存储三大类数据信息，主要包括基本资料、项目相关进程以及资金投入。共计7张数据表，同时后续鉴于数据储备增大，系统会适当地自动增加新表。

## 3.3数据库概念结构设计

鉴于本系统是基于数据库的基础之上建立的，因此有一个健壮完善的数据库尤为重要，而设计一个健壮优秀的数据库最重要的就是数据库概念结构设计，可以说数据库的概念结构设计将会影响后期的数据库性能以及系统的性能，以下通过实体E-R图简述各个表的结构层次。

### 3.3.1系统基本表

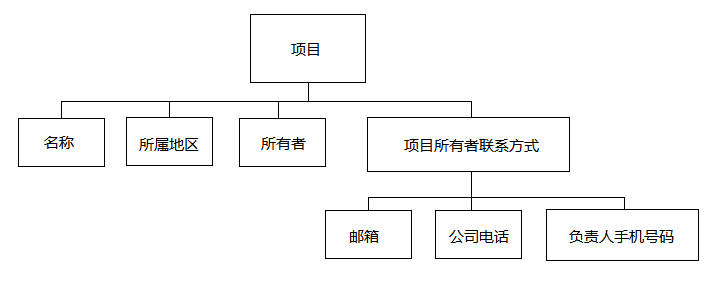
系统基本表主要分为：注册表（User\_RegisteInfo）、登陆验证表（User\_Login）、系统权限验证表（System\_ Jurisdiction），其中包含一个用户实体，详细请参考3.3.1-1用户实体E-R图。



3.3.1-1用户实体E-R图

### 3.3.2项目信息表

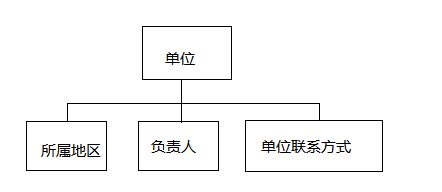
项目信息表主要分为：项目基本信息表（Project\_Info）、项目实施管理表（Project\_Manage），其中包含一个项目实体，详细请参考3.3.2-1项目实体E-R图。



3.3.2-1项目实体E-R图

### 3.3.3单位信息表

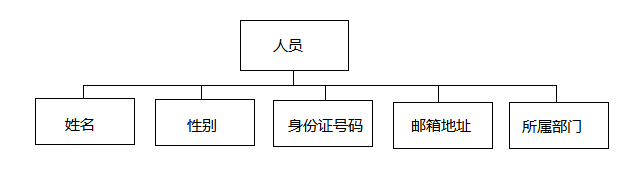
单位信息表中包含一个单位实体，详细请参考图3.3.3-1单位实体E-R图。



3.3.3-1单位实体E-R图

### 3.3.4人员信息表

人员信息表中包含一个人员实体，详细请参考3.3.4-1人员实体E-R图。



3.3.4-1人员实体E-R图

## 3.4数据库逻辑结构设计

关于数据库的具体逻辑设计，通过表格的形式简介明了地展示GraduationThesis数据库中的各个表的具体逻辑结构以及各个表之间地逻辑关系。

### 3.4.1系统基本表

（1）注册表（User\_RegisteInfo），主要记录用户的基本信息，具体设计如表3.4.1-1注册表（User\_RegisteInfo） 所示。

表3.4.1-1 注册表（User\_RegisteInfo）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 说明 | 列名 | 数据类型 | 是否为空 | 备注 |
| 用户编号 | UserRegiste\_ID | int | 否 | 主键，关联System\_ Jurisdiction验证用户权限 |
| 登录编号 | UserLogin\_ID | nvarchar(10) | 否 | 关联User\_Login表的登录编号,添加登录信息 |
| 姓名 | UserRegiste\_Name | nvarchar(50) | 否 |  |
| 性别 | UserRegiste\_Sex | nvarchar(10) | 否 |  |
| 联系电话 | UserRegiste\_Phone | nvarchar(50) | 否 |  |
| 邮箱地址 | UserRegiste\_Email | nvarchar(50) | 否 |  |

（2）登陆验证表（User\_Login），主要记录用户注册时注册的用户名和密码，用于登陆验证，具体设计如表3.4.1-2 登陆验证表（User\_Login）所示。

表3.4.1-2 登陆验证表（User\_Login）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 说明 | 列名 | 数据类型 | 是否为空 | 备注 |
| 登录编号 | UserLogin\_ID | nvarchar(10) | 否 |  |
| 登录名 | UserLogin\_Name | nvarchar(50) | 否 |  |
| 密码 | UserLogin\_Password | nvarchar(50) | 否 |  |

（3）系统权限验证表（System\_ Jurisdiction），主要记录用户权限的信息，控制登陆用户的最高权限，实现系统的级别控制，具体设计如表3.4.1-3系统权限验证表（System\_ Jurisdiction）所示。

表3.4.1-3 系统权限验证表（System\_ Jurisdiction）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 说明 | 列名 | 数据类型 | 是否为空 | 备注 |
| 用户编号 | UserRegiste\_ID | int | 否 |  |
| 用户类型 | UserLogin\_SystemType | int | 否 | 系统控制权限识别 |

### 3.4.2项目信息表

（1）项目基本信息表（Project\_Info），主要记录项目的基本信息，具体设计如表3.4.2-1 项目基本信息表（Project\_Info）所示。

表3.4.2-1 项目基本信息表（Project\_Info）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 说明 | 列名 | 数据类型 | 是否为空 | 备注 |
| 项目编号 | Project\_ID | int | 否 | 主键自增 |
| 项目名称 | Project\_Name | nvarchar(50) | 否 |  |
| 项目所属地区 | Project\_Area | nvarchar(50) | 否 |  |
| 项目所有者 | Project\_Master | nvarchar(50) | 否 | 项目委托公司名称 |
| 联系方式 | Project\_Email | nvarchar(50) | 否 | 电子邮箱 |
| 联系方式 | Project\_Telephone | nvarchar(50) | 否 | 公司电话 |
| 联系方式 | Project\_Phone | nvarchar(50) | 否 | 负责人联系方式 |

（2）项目实施管理表（Project\_Manage）,主要记录项目的管理信息，包括项目负责部门以及项目负责人等，同时统筹项目资金等相关信息，具体设计如表3.4.2-2项目实施管理表（Project\_Manage）所示。

表3.4.2-2项目实施管理表（Project\_Manage）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 说明 | 列名 | 数据类型 | 是否为空 | 备注 |
| 项目编号 | Project\_ID | int | 否 | 主键，与项目基本信息表（Project\_Info）中关联，便于检索数据。 |
| 项目部门编号 | Department\_ID | int | 否 | 关联单位信息表，对项目的各个分部进行合理科学的规划与管理 |
| 项目负责人 | MainSalesMan\_ID | int | 否 | 关联人员信息表 |
| 项目业务员1 | SalesMan\_IDone | int | 否 | 关联人员信息表 |
| 项目业务员1 | SalesMan\_IDtwo | int | 否 | 关联人员信息表 |
| 项目业务员1 | SalesMan\_IDthree | int | 否 | 关联人员信息表 |

### 3.4.3单位信息表

单位信息表（Company\_Info），主要记录单位详细信息，对公司的各个分部部门单位集中管理，便于公司后期的规模扩大以及业务的发展，详细设计如表3.4.3-1 单位信息表（Company\_Info）所示。

表3.4.3-1 单位信息表（Company\_Info）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 说明 | 列名 | 数据类型 | 是否为空 | 备注 |
| 项目部门编号 | Department\_ID | int | 否 | 主键，关联项目实施管理表，为其提供详细信息，对项目进行管理 |
| 项目部门所属地区 | Company\_Area | nvarchar(50) | 否 |  |
| 项目部门负责人 | Company\_MainSalesMan | nvarchar(50) | 否 |  |
| 项目部门联系方式 | Company\_Email | nvarchar(50) | 否 |  |

### 3.4.4人员信息表

人员信息表（Staff\_Info），主要记录单位员工的基本信息，对员工信息集中管理，便于部门单位的人员变动以及工程项目任务的快速分配，详细设计如表3.4.4 人员信息表（Staff\_Info）所示。

表3.4.4 人员信息表（Staff\_Info）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 说明 | 列名 | 数据类型 | 是否为空 | 备注 |
| 人员编号 | Staff\_ID | nvarchar(50) | 否 | 主键，员工唯一标识，由注册时间以及六位随机码唯一确定 |
| 姓名 | Staff\_Name | nvarchar(50) | 否 |  |
| 性别 | Staff\_Sex | nvarchar(10) | 否 |  |
| 身份证号码 | Staff\_CardID | nvarchar(50) | 否 |  |
| 手机号码 | Staff\_Phone | nvarchar(50) | 否 |  |
| 邮箱地址 | Staff\_Email | nvarchar(50) | 否 |  |
| 所属部门 | Department\_ID | int | 否 | 关联单位信息表 |

## 3.5数据库系统参数配置

数据库配置文件主要存储数据库所在服务器的IP地址（ServerIP）、数据库名称（DataBaseName）、数据库登陆用户名（LoginName）、数据库密码（Password），应用程序通过配置文件中的配置信息连接数据库，一方面提高应用程序的安全行，另一方面提高了应用程序的迁移性以及数据库文件的迁移性，能够极大地保证数据地的安全性，同时便于后期的数据库的维护与优化。

数据库配置文件如下代码所示：

<appSettings>

<!--数据库部署IP地址-->

<add key="ServerSQLip" value="10.21.5.48"/>

<!--数据库名称-->

<add key="DataBaseName" value="GraduationThesis"/>

<!--数据库登录名-->

<add key="UserLoginName" value="pc-miner"/>

<!--数据库登录密码-->

<add key="Passwords" value="pc-miner123"/>

</appSettings>

## 3.6数据访问层设计

数据访问层又称持久层，其功能主要是针对数据库做一下高效的操作与访问，最基本的即实现各个数据表之间的Select、Insert、Delete、Update操作，由于常规的直接操作会导致数据访问缓慢响应速度极差的问题，因此采用加入ORM元素的方式，提高数据的访问速度，那么这就涉及到对象与数据表之间的映射关系，以及各个对象实体的持久化。

在本项目中SQLHelper中的cs文件（详情参见图3.6-1 SQLHelper文件规划）即实现应用程序的数据库访问层，供应用程序调用，具体作用请参考表 3.6-2 SQLHelper文件各个类文件说明的简要说明，具体功能实现的流程请参考各个主要部分。

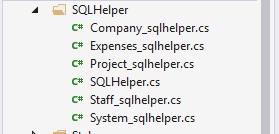


图3.6-1 SQLHelper文件规划

表3.6-2 SQLHelper文件各个类文件说明

|  |  |  |
| --- | --- | --- |
| 序号 | 类名 | 主要作用 |
| 1 | SQLHelper | 读取数据库连接字符串 |
| 2 | System\_sqlhelper | 操作系统基本表 |
| 3 | Project\_sqlhelper | 操作项目信息表 |
| 4 | Company\_sqlhelper | 操作单位信息表 |
| 5 | Staff\_sqlhelper | 操作人员信息表 |
| 6 | Expenses\_sqlhelper | 操作项目收益分成表 |

数据访问层的重要性不言而喻，它是.Net框架的核心知识体系，一个合理科学的数据访问层将会给应用程序后期的维护带来极大的便利，同时对项目本身的扩大也有着至关重要的作用。另一方面也会极大地提高系统的效率。它最主要的作用莫过于通过对数据的访问，达到对物理数据库中的各个表、视图等的访问与操作，通常应用程序对数据库的访问有以下几种方式：

1. 事务脚本：直面存储过程
2. ORM：通过各个对象与实体之间的映射关系
3. 表模型：以单个数据表为基本单位直接进行访问

上述三种方式，显然第三种方式简单而且容易实现，但是并不能满足本项目的需求，所以本项目采用了第二种方式，即加入ORM元素，通过个对象与实体之间的映射关系实现了本项目的数据访问层。

### 3.6.1 用户登录模块

本模块主要用于验证用户的操作权限的，操作权限分为三种：

1. 无权操作：非内部人员（非授权人员）禁止登录系统
2. 查看权限：一般用户，能够查看非敏感信息，同时做出简单的操作
3. 全局权限：高级用户，能偶查看所有信息，同时拥有修改权限

用户登录模块主要包含注册模块（小模块）以及登录模块，本模块基本实现流程参见图3.6.1-1 用户登录模块实现流程。

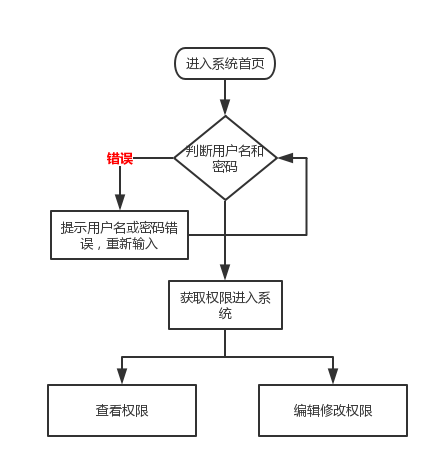


图3.6.1-1 用户登录模块实现流程

### 3.6.2 项目信息模块

本模块主要提供对项目相关信息的Select、Insert、Update、Delete等操作，由于系统设计到操作权限的问题，所以在各个操作上会有一定的约束条件（即为条件操作，在此处设计时是以全局权限进行分析与设计的，关于涉及权限问题参考用户登录模块具体了解各个权限的操作范围，此处不在负赘），本模块通过Project\_sqlhelper.cs以及最基本的SQLHelper.cs实现，将各个操作封装成一个类文件，方便在各处调用，包括Restful API接口中也是直接调用以达到操作数据库的目的，同时在单元测试中也带来了极大的便利。

### 3.6.3 单位信息模块

本模块主要提供对单位相关信息的Select、Insert、Update、Delete等操作，主要辅助项目和人员的集中有序管理，本模块通过Company\_sqlhelper.cs以及最基本的SQLHelper.cs实现对数据库的操作，在Company\_sqlhelper中偏重于Select和Insert，主要是由于项目信息模块对单位信息的高度依赖，因此会比较偏重于Select和Insert功能多一点。

### 3.6.4 人员信息模块

本模块主要提供对人员相关信息的Select、Insert、Update、Delete等操作，其主要作用除了为项目信息提供业务员和分责人之外，还为项目收益分成提供参考依据，同时为公司的员工管理带来便利。

# 四、系统编码实现

## 4.1Restful API接口设计

### 4.1.1 Restful API接口说明

#### 4.1.1.1 Restful API的基本概念与认知

Restful API是2000年Roy Fielding的博士提出的，他在论文中提到："我这篇文章的写作目的，就是想在符合架构原理的前提下，理解和评估以网络为基础的应用软件的架构设计，得到一个功能强、性能好、适宜通信的架构。REST指的是一组架构约束条件和原则。" 如果一个架构符合REST的约束条件和原则，我们就称它为Restful架构。

#### 4.1.1.2 如何理解Restful架构

通过对REST原则的解析，Restful架构主要包括以下几个概念：

* 资源与URI：只要有引用的必要即为资源，而资源可以任何一个实体，也可以是任何一个抽象概念，同时每一个资源都有唯一的URL与之对应。
* 统一资源接口：无论什么资源都遵循统一的接口原则，接口使用标准的HTTP方法如Get、Put、Post等，并遵循这些方法的语意。
* 资源的表述：资源在外界可以有不同的表现显示，是客户端于服务端的一种资源表达，即资源的表述。常见的表述方式有xmal、json等，基于json的快捷与方便解析所以被广泛应用于开发领域。
* 资源的链接：REST是使用的HTTP方法进行资源的操作，然而如果仅仅将其理解为CURD的Web数据库架构显然是不够的。这种模式的核心为“超媒体即应用状态引擎”（使用超媒体来增强资源的连通性），侧重于给资源的表述提供链接，而不是专注于"资源的CRUD"。
* 状态的转移：即应用状态与资源状态的转移，客户端与服务端的交互是无状态的，但是每一次请求中都包含处理该请求的所需的一切信息。

### 4.1.2 Restful API接口设计

基于Restful架构，简单实现了Restful API接口的开发，将每一个项目实体作为资源放置在服务端并提供唯一的URL供客户端操作，当然作为项目实体包含的小实体也提供了更加精确的URL唯一确定。

具体的Restful API接口详情见表4.1.2 Restful API部分接口一览表。

表4.1.2 Restful API部分接口一览表

|  |  |  |  |
| --- | --- | --- | --- |
| 组别 | HTTP | API地址 | 描述 |
| 账号API | POST | users/login | 用户登录（同时返回AccessToken） |
| 项目API | GET | projects/{projectid} | 查询单个项目 |
| GET | projects | 模糊查询项目 |
| GET | projects/{projectid}/ids | 查询项目所有人员的ID |
| 人员API | GET | users/{ userid} | 查询单个人员 |
| GET | users | 模糊查询人员 |
| GET | users /{ userid }/ sections | 查询人员所在部门的ID |
| 部门API | GET | sections/{sectionsid} | 查询单个部门 |
| GET | sections | 模糊查询部门 |
| GET | sections /{sectionsid}/users | 查询部门所有人员的ID |

## 4.2用户登录模块

### 4.2.1登录模块

登录模块采用无边框窗口设计，在重构windows窗体的基础上加入了三维立体元素，杜绝了传统的无边框窗体形如纸片的状态，增加了其作为窗口的视觉质感。效果图如图4.2.1-1登录界面所示。



图4.2.1-1登录界面

登录模块采用Basic认证，他的主要原理并不复杂，解释起来就是加密用户提交的信息，然后生成票据，之后的每一次请求的时候将票据带过来验证，从而达到提高安全性的目的。

登录界面的重构主要修改如下Windows窗体属性值(属性修改详情请参考表4.2.1 属性修改一览表)从而达到如图所示的效果同时通过设置窗体的Radius[X\Y]从而控制阴影覆盖面积。

表4.2.1 属性修改一览表

|  |  |  |
| --- | --- | --- |
| 序号 | 属性名称 | 修改值 |
| 1 | AllowsTransparency | True |
| 2 | ResizeMode | NoResize |
| 3 | WindowStyle | None |

阴影效果的通过以下代码实现：

<Rectangle Fill="{DynamicResource {x:StaticSystemColors.WindowBrushKey}}" RadiusX="5" RadiusY="5">

<Rectangle.Effect>

<DropShadowEffect BlurRadius="10" ShadowDepth="0"/>

</Rectangle.Effect>

</Rectangle>

### 4.2.2注册小模块

注册模块作为附属于登录模块的一个小模块，采用简单的web技术实现，通过点击登录界面的注册账号label标签按钮直接进行跳转到sign-up界面（效果图如图4.2.2-1 项目实施管理系统注册页面所示）进行注册，同理忘记密码也是一样，忘记密码这个小模块也是通过这种方式实现，主要依靠用户提供的电子邮箱地址和邮箱返回的验证码进行密码找回（效果图如图4.2.2-2 项目实施管理系统找回密码所示）。

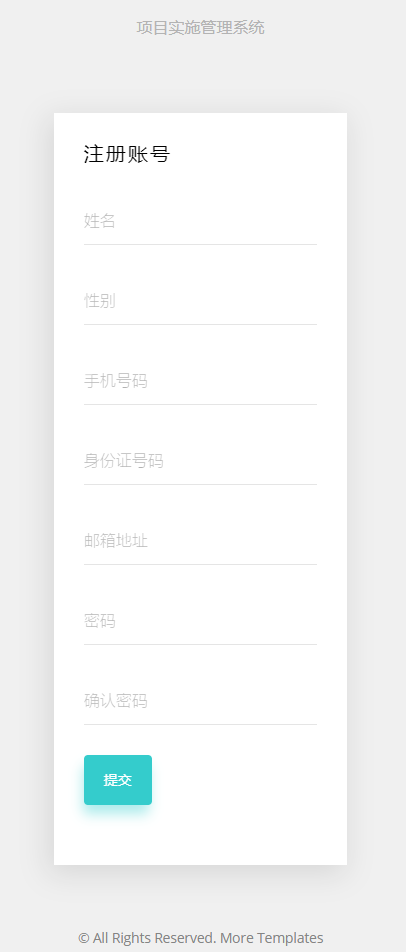
 

图4.2.2-1 项目实施管理系统注册页面 图4.2.2-2 项目实施管理系统找回密码

本部分采用Ajax技术进行数据的交互，Ajax即为异步的 JavaScript和XML,是一种现有标准的新方法，其最大的优点是在不加载整个页面的情况下可以与服务器交换数据并完成部分网页内容的更新，Ajax的工作原理如图4.2.2-3 Ajax工作原理图所示。

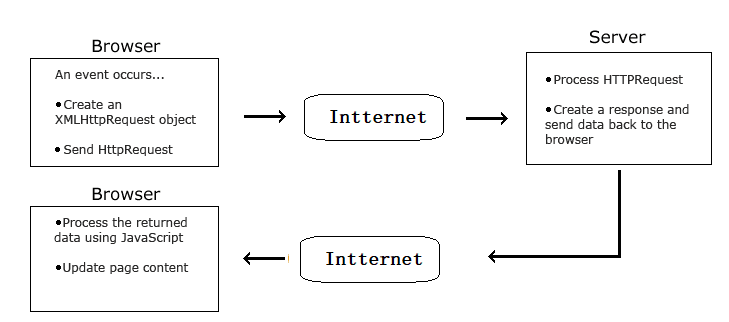


图4.2.2-3 Ajax工作原理图

Ajax技术在软件技术行业具有广泛的应用价值，其主要运用于以下几个方面：

* 在XHTML+CSS中广泛用来表达资讯
* 通过运用JavaScript操作DOM（Document Object Model）来执行动态效果
* 运用XML和XSLT操作资料
* 运用XMLHttpRequest或新的Fetch API与网页服务器进行异步资料交换

但是我们在运用的过程中必须将Ajax与Flash、Silverlight和Java Applet等RIA技术区分开来，毕竟他们拥有本质的区别，不能够混淆。

本模块都是基于Ajax技术实现的，具体实现代码如下所示：

$.ajax({

type: "POST",

url: "RegisterHelper.ashx",

data: "RegisterMessage=" RegisterMessage,

success: function (data) {

alert(GetAction(data));

}

})

### 4.2.3服务器配置

服务器配置主要通过读取数据库配置文件app.config中数据库配置参数，连接部署在远程服务器（或者远程主机）上的Reatful API服务接口和数据库，从而达到数据的流通与获取，本部分点击登录界面上的倒置尖括号调出配置界面（如图 4.2.3-1 远程服务器端口），修改参数之后确认修改即永久性修改app.config文件中的配置参数，方便下次连接，同时为了确保连接，通过连接测试确定是否连接成功（连接成功如图 4.2.3-2 连接测试成功，连接失败如图4.2.3-3 连接测试失败日志）。

图 4.2.3-1 远程服务器端口 图 4.2.3-2 连接测试成功



图4.2.3-3 连接测试失败日志

本模块主要基于Log4net进行模块设计于开发，Log4net是一个功能完善并且被广泛运用的日志记录组件，常常被用于普通的测试以及软件异常处理中记录软件异常发生情况，其功能广泛，日志信息可以很方便地通过多种形式进行记录(比如：文本文件、数据库、Windows事件日志等)，同时Log4net还为开发者提供了可记载控制要记载的日志级别(常见的几种日志级别：DEBUG（调试信息）、ERROR（一般错误）、WARN（警告）、FATAL（致命错误）、INFO（一般信息）等)。

在本模块的设计中，将所有的日志记录都以文本文件的形式进行存储，主要考虑到前期开发便于查看，而后期部署的过程中，这些程序运行日志将统一记录到数据库中，以便对系统性能的测试以及运行情况的分析。

而Log4net在使用过程中需要优先配置Config文件才能够使用，首先要在config文件的<configSections>节点中增加Section配置：

<type="System.Configuration.IgnoreSectionHandler"/>

## 4.3项目信息模块

本模块通过自定义的用户控件Grid View，大致的展示项目的实施进度（以百分比的形式展示）以及项目的一些基本情况（所属地区、项目开发周期、负责人）。通过点击查看按钮可以对项目组进一步的操作，详情如图4.3-1项目进展所示。



图4.3-1项目进展

在详情展示面板上对项目信息进行编辑修改以及保存，详情如图4.3-1 项目详情展示以及图4.3-2项目详情编辑。



图4.3-1 项目详情展示



图4.3-2项目详情编辑

对于新项目需要通过项目管理模块（如图 4.3-3项目管理）进行添加，点击添加新项中的“+”即可完成添加任务。

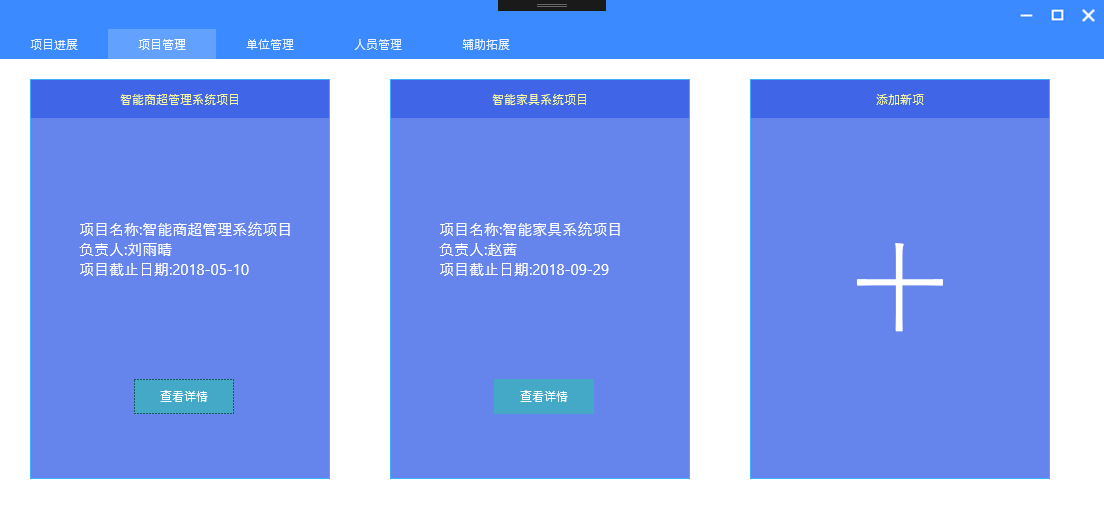


图 4.3-3项目管理

具体添加信息参见 图 4.3-4 项目添加信息界面。



图 4.3-4 项目添加信息界面

本模块由于涉及大量数据的加载，所以最主要的核心即为数据的加载，模块的设计中侧重于代码与UI分离的思想，整个系统的设计都遵循了这一理念。在这个模块中得到了最集中的体现。

关于界面设计，自定义TabControl、DataGrid（最主要的核心组件），对于TabControl除了对其背景的修改主要针对性地对Border属性进行了重新设计，具体设计如下代码所示：

<Border Grid.Row="1"

BorderBrush="Transparent"

Padding="0"

BorderThickness="0" Background="{TemplateBinding Background}">

<ContentPresenter ContentSource="SelectedContent" Name="PART\_SelectedContentHost"/>

</Border>

而对于DataGrid控件设计主要修改如下参数即可（详情参考表4.3-５DataGrid控件参数修改一览）。

表4.3-５DataGrid控件参数修改

|  |  |  |
| --- | --- | --- |
| 序号 | 属性名 | 属性值 |
| 1 | SnapsToDevicePixels | True |
| 2 | MinWidth | 0 |
| 3 | MinHeight | 28 |
| 4 | Foreground | #323433 |
| 5 | FontSize | 14 |
| 6 | Cursor | Hand |

## 4.4单位信息模块

通过ListBox直接展示所有已经添加的单位，选中左侧边的具体单位名称，点击图标在右侧栏展示可编辑的内容，删除或者修改（详情如图 4.4-1 单位信息所示）。



图 4.4-1 单位信息

对于新增单位，在右侧填写完整信息之后直接点击图标即可完成添加。

界面设计主要针对Treeview进行参数的添加与修改，首先在Treeview.xaml样式设计代码中添加以下代码：

<Grid.Resources>

<convers:BoolToVisible x:Key="boolToVisible"/>

<convers:VisibleToReverse x:Key="visibleToReverse"/>

</Grid.Resources>

## 4.5人员信息模块

通过Tree View以项目为分组依据对人员进行分类，点击任务姓名在右侧栏展示详细信息，同时提供编辑信息的操作（如图 4.5-1 人员信息所示）。

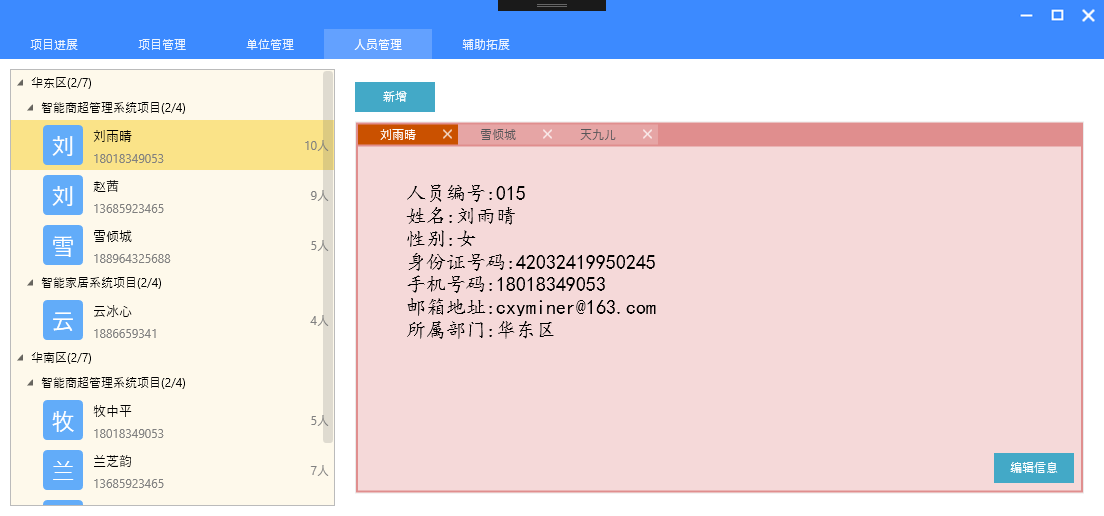


图4.5-1 人员信息

对于新增人员，本模块也提供添加操作，但是由于权限要求较高，普通用户无法操作，所见建议一般用户通过注册模块进行人员的增加，当然权限足够的用户可以直接在本模块直接新增人员信息。

## 4.6附属功能模块

辅助功能主要是为了丰富用户的使用体验以及软件的内容，简易的日历附加代办标签，新建代办添加到右侧栏的用户自定义listView中，完成代办勾选示意（如图 4.6-1 辅助功能）。



图 4.6-1辅助拓展

# 五、系统测试

## 5.1测试工具

测试工具选择了Visual Studio 2017自带的测试工具，通过Visual Studio Installer 安装升级工具，添加所需要的测试插件(如图 5.1-1 测试工具的选择与安装所示)。

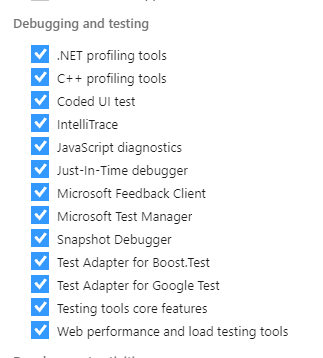


图 5.1-1 测试工具的选择与安装

## 5.2功能性测试、非功能性测试

### 5.2.1 功能性测试

按照正常的业务逻辑进行操作，之后检验结果如否偏离预期。功能性测试有7位同学组成的测评小组，使用软件添加相关数据，最后对软件生成的日志以及结果与理论预期进行比较，经过测试，该系统满足实际需求，功能全部实现。其次针对代码中各个重要参数进行监视，通常使用添加断点的方式进行测试数据代码的作用，以监测项目数据的正确性(详情参考图5.2.1-1 条件断点设置、图5.2.1-2 操作断点设置)。

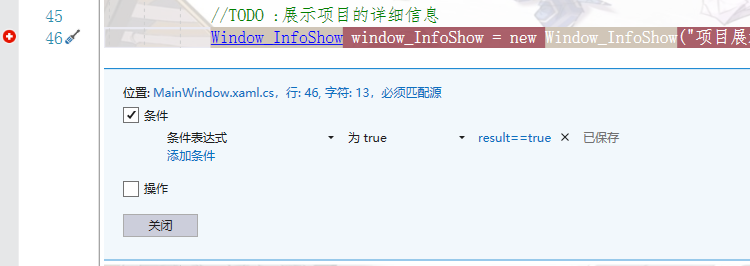


图5.2.1-1 条件断点设置

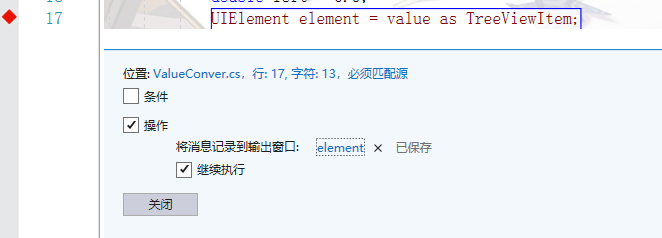


图5.2.1-2 操作断点设置

以上仅为几个简单的示例，其中条件断点设置中可添加多个条件，而在操作断点设置中亦可输出多个信息记录到窗口，同时可以控制程序是否继续进行运行。

通过断点的方式对重要的数据信息进行注意的排查，断点的排查主要采用了以下集中方式：

* 普通断点排查
* 条件断点排查(添加条件表达式、命中次数、筛选器)，对一些可能会存在异常的位置采用这样的排查，以确保应用程序代码逻辑上没有问题。
* 事件断点排查(通过将信息输出到输出窗口更加直观地看到数据值的变化)

同时利用Visual Studio自带的代码测试工具生成代码结构图，在清晰的代码结构图例中明确代码逻辑以及层次是否出现问题，同时也明确每一个对象每一个实例对应的实际调用关系。详细请参考图5.2.1-3 自定义Linq库文件代码结构图。

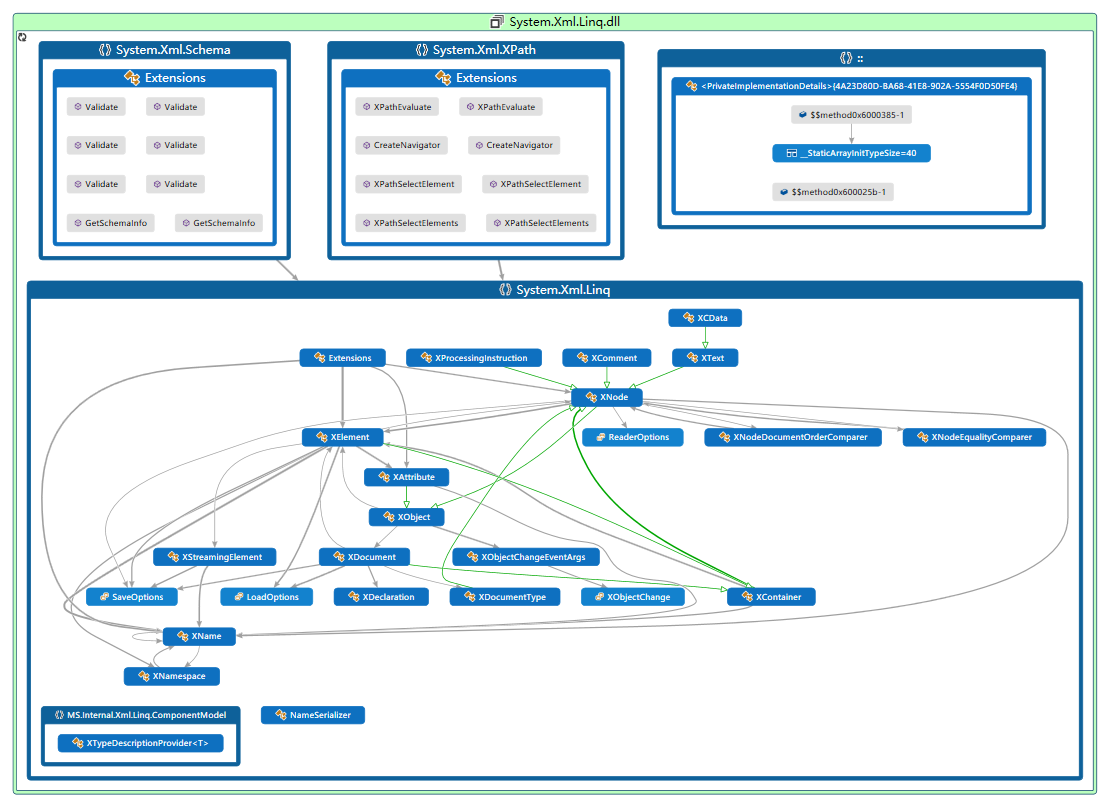


图5.2.1-3 自定义Linq库文件代码结构图

### 5.2.2 非功能性测试

主要采用比较流行的log4net框架，log4net是一个开源的日志框架，用来做非功能性测试主要是因为其能够很大程度上将程序所有的运行情况、过程、以及结果全部以文本文件的形式直接展现，方便我们详细得了解在系统业务功能实现的背后，计算机室如何一步步完成实际的操作的。

同时基于Visual Studio 2017自带的测试工具，我们还能够很直观地查看系统运行过程中对CPU、内存的使用情况，以各个线程的运行情况(如图 5.2-1 系统运性时内存、CPU的使用情况所示)。

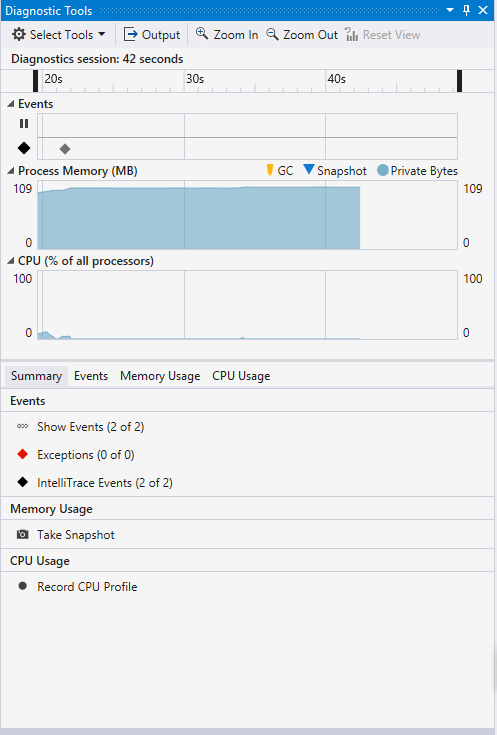


图 5.2-1 系统运性时内存、CPU的使用情况

在性能探查器中主要有以下几个参数值得关注(详情参考表5.2-2 性能探查器重要参数说明 )：

表5.2-2 性能探查器重要参数说明

|  |  |  |
| --- | --- | --- |
| 序号 | 属性名 | 备注说明 |
| 1 | Diagnostics session | 监测某段程序运行时间 |
| 2 | Process Memory | 监测内存占用情况 |
| 3 | CPU | 监测CPU使用情况 |
| 4 | Exceptions | 监测异常事件 |
| 5 | IntlliTrance Events | 监测各个点击事件 |

# 六、发布及管理

## 6.1本地发布

### 6.1.1 本地发布概述

文件发布到本地，部署在服务器主机的IIS(Windows系统自带的服务，使用前需要安装)服务上，供应用程序调用。本地发布大部分以文件的形式发布，或者可以直接通过目标IP地址直接发布。本地发布具有一定的局限性，一方面维护成本高，另一方面由于部分技术要求，非专业人士在部署时会有一定的障碍，但是参考发布部署说明文档，相对于本地IIS目标IP地址发布更简单一些。但是本地发布便于程序的调试，在开发过程中具有很大的便宜，省去了发布到服务器的繁琐过程，简化了测试过程。

## 6.1.2 本地文件发布

（1）发布部署过程图解

通过Visual Studio 2017发布网站到文件，配置文件名命名为，选择目标地址之后，在目标地址会生成相关的文件。将该文件发布到已经安装的IIS服务上，通过浏览绑定的地址，提供应用程序的接口，供应用程序调用、或者通过Web浏览器做简单的调试，发布成功之后，得到的发布文件可以在任意计算机上部署。

详细图解如下：

通过Visual Studio2017一键发布进行文件发布，文件命名为（参见图6.1.2(1).1文件Publish），文件发布格式应为文件系统（即 File System，参见图6.1.2(1).2文件发布文件格式），检查文件发布日志检查是否发布成功（参见图6.1.2(1).3 文件发布日志），如图即为发布成功。

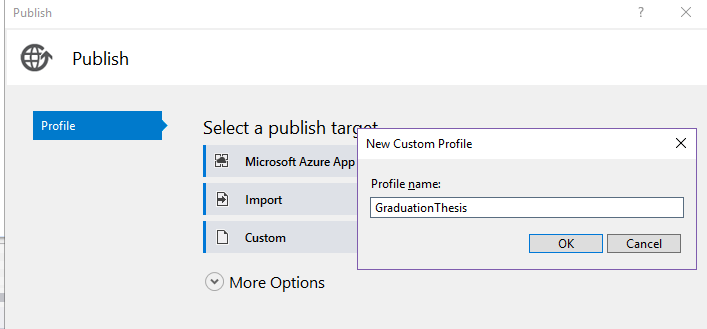


图6.1.2(1).1文件发布Publish

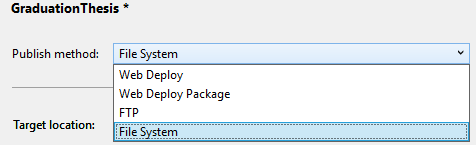


图6.1.2(1).2文件发布文件格式

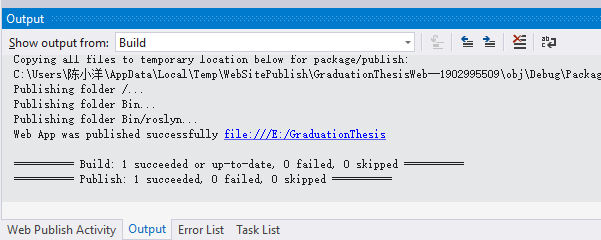


图6.1.2(1).3 文件发布日志

1. 注意事项
2. 打开IIS服务的目录浏览，否则会出现无法找到资源的错误。
3. IP地址绑定时，建议不要绑定计算机本身的IP地址，防止本机IP发生改变时出现异常，带来不必要的繁复工作。
4. 文件夹的权限要足够，尽量设置为Everyone可编辑，防止由于权限问题而导致资源不能正常访问。

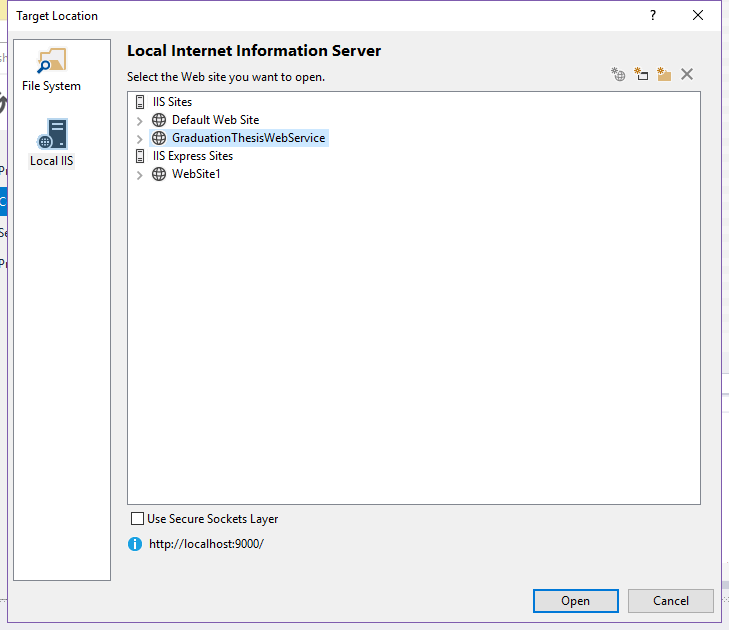
## 6.2.3 本地IIS目标IP发布

（1）发布过程图解

通过Visual Studio 2017 直接发布到目标IP地址，这种方式简易快捷，方便操作。但同时存在一定的局限性，即只能通过开发机进行发布以及部署。但是在同一网络下，在知道服务器主机IP地址的情况下能够直接发布并完成部署，但是这种方式容易出错，且不易排除，如果服务器主机存在多个服务，一旦操作失误会影响其他服务的正常使用。由于这种方式的弊端，故不建议采用这种方式，做简单的介绍也只是为了佐证前一种方式的优势。

详细图解如下：

通过Visual Studio2017一键发布到指定的IIS地址（参见图6.2.3(1).1 指定IP发布）。



参见图6.2.3(1).1 指定IP发布

（2）注意事项

1. 由于这种发布方式权限要求较高，所以Visual Studio 2017必须以管理员的方式运行，否则无法访问IIS上指定的IP地址。
2. 打开IIS服务的目录浏览，否则会出现无法找到资源的错误。
3. 发布完成时候，需要通过web浏览器检验是否发布成功。

### 6.2 远程服务器发布

### 6.2.1远程服务器发布概述

程序测试通过后，通过发布到服务器供应用程序调用，部署过程比较繁琐，但是相对于本地发布，远程服务器发布最大的优势在于两点：

1. 服务器相应速度远高于普通计算机
2. 服务器能够24小时运行，并且有专业人员进行维护，不易发生由于服务器的原因而导致应用服务瘫痪。

远程服务器发布需要购买云服务器、或者购买相关企业的物理服务器，这里采用的腾讯云服务器。对于云服务器主要有以下要求：

1. Windows系统、SQL Server环境、 .Net 4.0 或以上的运行环境，其他的配置标准配置即可，随着业务模式的扩大，可以提高相关新能。
2. 网络流量规格：6Mbps。

### 6.2.2远程服务器环境搭建

远程服务器环境的搭建主要是为了给服务应用提供运行环境与平台，而关于云服务器的各种配置参数的设置，可以参考腾讯云官方提供的说明文档进行配置。只要注意以下几个关键点即可：是否具备SQL Server环境、 .Net 4.0 或以上的运行环境以及IIS服务。

### 6.2.3远程服务器连接

使用已经注册的通讯云服务器账户，通过Windows 10自带的服务直接连接云服务器进行操作(图6.2.3.1 远程服务器连接)，或者通过腾讯云提供的云服务器后台管理直接操作，建议使用前者，后者由于过程繁琐，不易于操作。

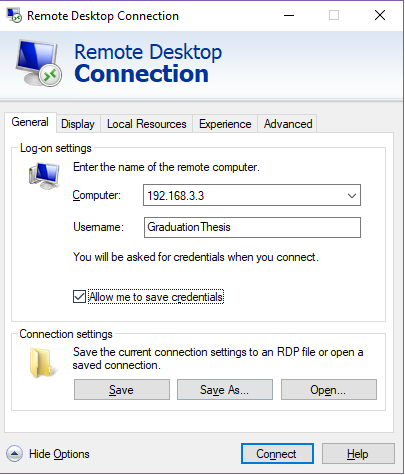


图6.2.3.1 远程服务器连接

# 结束语

经过两个多月的毕业设计与完善，基于Restful API的项目实施管理系统以及相对应的论文已经完成，在做毕业设计的过程中，我对Restful API的认识更进一步，在实际的运用过程中从理论到实践的层次对这一项技术有了质的认知，同时对Visual Studio 2017的测试用例以及postman等知名测试工具有了一个大概的了解。在UI Design方面，相对于之前的一窍不通到现在的小有心得亦是十分值得肯定欣慰的。

基于Restful API的项目实施管理系统虽然时一款PC端应用，在这个移动应用很火的时代似乎并不吃香，但是相对于办公而言，大多数企业管理者对于这种涉及很多重要的数据的工作，大多会采用PC客户端应用程序。退一步分析，这套系统是基于Restful API接口开发的应用，因此本系统的核心在于 Restful API接口的开发与设计，基于这套Restful API接口可以很快的开发出相对应的移动应用程序，并不需要做过多的开发工作，同时由于Restful API具有高度的可拓展性、开发接口的时候也特别留意这方面的注意事项，所以对接口功能的拓展也不是很困难的事情。

在整个项目完善的过程中，掌握了软件设计的基本原则，在满足用户体验需求的同时，能够更加多的注重在代码的质量上，写优质的代码、写规范的代码，精简自己的代码优化自己的代码是软件设计过程中贯彻始终的一个最基本的原则之一，同时更加应该考虑软件的健壮性与后期的可拓展性，以此提高软件的生命周期。

而基于软件设计，数据库的设计是更加值得关注的，数据库决定了软甲的基本设计模式以及数据访问模式以及数据访问速率。如何设计出优质的数据库就必须按照数据库设计的最基本的原则——“范式设计法则”进行设计。在完善本系统的过程中，从数据库的需求分析到数据库的逻辑设计再到数据库的优化以及维护，在此过程中深刻理解了三种范式设计与BC范式，同时也了解了数据库命名的规则以及后期的一些维护手段。

# 致谢

感谢我的毕业设计指导老师王辉老师在百忙之中抽出宝贵的时间对我的毕业设计进行精心的指导，让我能够在短时间里完成课题“基于Restful API的项目管理实施系统的”的毕业设计与论文，并进行论文的润色修改以及应用程序的代码规范和性能优化。

感谢学院两年来对我的大力培养，使得我在各方面能力提高的同时，懂得很多为人处世的道理以及做人的内涵，亦使我成为一个优秀的技术性人才。

感谢同学们在为数不多的时光里带我领略了精彩绝伦的大学生活以及纯洁真挚的友谊。

两个多月兢兢业业的时光一晃而过，会想起出入门槛时仿佛是在昨天一般，或许毕业设计是我在这个学校的最后一份答卷，就以此画一个满意的终止符。然后收拾行囊继续杨帆远航，获得更加辉煌璀璨的明天！

# 参考文献

[1] 张权 郭天娇，SQL查询的艺术，人民邮电出版社，2014年1月第一版

[2] [美]Sanjay Patni 著，郭理勇 译，RESTfulAPI开发实践，清华大学出版社，2018年2月第一版

[3] C#高级编程—C# 5.0 & .NET 4.5.1，清华大学出版社第九版

[4] Roy Thomas Fielding，Architectural Styles and the Design of Network-based Software Architectures，2000年

[5][英]Gary McLean Hall著 许顺强 译，C#敏捷开发实践，中国版本图书馆CIP数据核子(2016)第139459号，人民邮电出版社，2017年4月第一版

[6][英]Jon Skeet著 姚琪琳 译，深入理解C#，人民邮电出版社，中国版本图书馆CIP数据核子(2014)第027896号，2014年4月第一版

[7] 包善东，更锋利的C#代码，清华大学出版社

[8] 张子阳，.NET之美：.NET关键技术深入解析，机械工业出版社，2014年01第一版

[9] 吕皓月，APP蓝图，清华大学出版社，2015年1月第一版

[10] [美]Jeff Johnson 著 张一宁 译，认知与设计：理解UI设计准则，中国版本图书馆CIP数据核子(2011)第156335号，人民邮电出版社2011年第一版