本科毕业设计（论文）

基于三层架构与 SOA 的 TBM 系统设计

The Design of TBM System Base on Three-Layer Structure and SOA

Recommend
火龙果软件

摘　　要

随着科学技术的不断发展，世界正趋于一体化、信息化和统一化。软件行业是加速这个过程的一种有力工具。就一个特定的项目来说，如何最大限度的提高代码的复用性和对外服务性已经成为评价一个项目好坏的标准之一。TBM（Training Base Management）系统就是基于三层架构和 SOA 体系而开发的一款针对实训基地日常业务处理的管理系统。并且本系统将从现存的门禁系统中获得门事件数据，来实现基地内的统一、高效、半自动化的管理。

本文深入的分析了软件的分层架构技术与 SOA 技术中 Web 服务的应用。在讨论分层架构时，重点关注分层架构的技术特点和 TBM 系统应用三层架构技术的原因，并且根据业务需求的原因，将大部分的业务逻辑处理都封装成 Web 服务，供不同的基地、不同平台调用。由此引出实现 SOA 的一种方式------Web Service 技术，它能提高了系统的互操作性、封装性、普遍性，降低了系统的耦合程度，使系统易于集成、部署简单。在设计系统的表现形式上，综合考虑不同用户的操作方便性，TBM 系统采用 C/S、B/S 兼有的模式。对于普通员工，由于对系统的操作不是很频繁，并且人员众多，他们的操作页面是 Web 程序。对于管理员工（教师和有职务的同学），由于每天有大量的业务要进行处理，为了增加使用的方便性，他们的操作页面是 winForm 程序。本文主要设计了 winForm 程序表现形式，对于 Web 程序将待下一期完成。

TBM 系统在经过模拟运行和测试后，发现大大提高了基地日常管理工作的效率，并且减少了人工操作可能造成的错误记录。而且，本系统的真正意义是提供了一套完整的 Web 服务，实现跨平台、跨基地的真正意义上的服务复用。

关键词：三层架构；Web Service；面向对象；WinForm

Recommend
火龙果软件

The Design of TBM System Base on Three-Layer Structure and SOA

Abstract

With the continuous development of science and technology, the world is becoming integrated, information-based, and unified. Software industry is a powerful tool to speed up this process. To a specific project, how to maximize code reuse and improve the external services has become one of the criteria to evaluate it good or bad. TBM (Training Base Management) system is a daily business process management system, which is based on three-layer architecture and SOA system for the training base. And this system will get door events from the existing access control system to achieve a unified, efficient and semi-automated management in training base.

This paper in-depth analyzes of the layered architecture of software technology and the web services applications in SOA technology. In the discussion of layered architecture, focusing on the technical characteristics of layered architecture and the reasons why TBM system applies three-layer architecture. Because of the business needs, most of the business logic processing are packaged as web services for different platform, different base to invoke. This leads to one way of realizing SOA technology-----Web Service, which enhances the interoperability, packaging, universality but reduces the degree of coupling of the system. That makes the system easy to integrate and simple to deploy. Form in the design of the system's performance, considering the operating convenience of different users, TBM system adopts both C/S, B/S models. For ordinary workers, because of their operations to the system is not so frequent, and the huge numbers of staff, their operation page is Web program. For the management staff (teachers and some students), because of a large number of business work to be processed every day, in order to increase ease of using, their operation page is WinForm program. In this paper, mainly designed WinForm program, and the Web program will be completed on schedule.

After the simulation run and test, we find that TBM system greatly enhances the efficiency but reduces manual caused errors of daily management in training base. Moreover, the true meaning of this system is to provide a complete set of wet services, which achieves the true sense of service reusing crosses different platforms and different bases.

Key Words：Three-layer architecture；Web Service；Object-oriented；WinForm

Recommend
火龙果软件

目　　录

摘　　要 I
Abstract II
引　　言 1
1　相关技术 2
1.1　分层构架 2
1.1.1　层次结构 3
1.1.2　分层架构优点 5
1.2　SOA 技术 6
1.2.1　Web 服务 7
1.2.2　Web 服务特点 8
1.3　TBM 系统技术优势 9
2　TBM 系统需求分析 10
2.1　功能需求 11
2.2　应用需求 13
3　TBM 系统架构设计 15
3.1　技术架构设计 15
3.2　表现形式 16
4　TBM 系统应用设计 18
4.1　公共对象设计 18
4.2　功能模块设计 19
4.2.1　Entity 模块 19
4.2.2　SQLDAL 模块 26
4.2.3　BLL 模块 31
4.2.3　Web Service 封装 32
4.2.4　UI 模块 35
4.3　后台服务程序设计 37
5　设备子模块与权限子模块实现 38
5.1　设备子模块 38
5.2　账户及权限子模块 45
结　　论 50
参 考 文 献 51

Recommend
火龙果软件

致　　谢 52

Recommend
火龙果软件

引　　言

随着人类科学技术的迅猛发展，更行各业应用的软件也越来越趋于庞大。当一个应用软件规模较小时，采用什么样的开发模式几乎不影响软件的成功交付。但当开发一个规模庞大的软件产品时，采用好的开发方法的作用就显而易见了。软件系统架构技术正式在这样一种环境下诞生的，它不仅能够帮助系统的设计和开发，还会直接影响代码的复用性和强壮性，这些都往往决定了软件开发的成败。而且在当今，跨平台的、跨地域的复杂信息交流与互动已经成为明显的趋势，如何最大限度的减少软件开发的成本，加大程序之间或功能之间的复用性，显然已经成为了一个十分热点的研究内容。在这种情况下，面向服务的编程思想（SOA）应运而生，它使得不同平台、地域间的计算机可以方便的交流数据，共用服务，减少单独维护己方代码和业务的负担。

本文论述设计的 TBM（Training Base Management）系统，是一款基于三层架构和 SOA 的实训基地管理系统，以实现实训基地内高效、便捷、准确的半自动化管理。系统之所以采用三层架构的设计思想，是因为三层架构可以最大限度的剥离层与层之间的依赖关系，增强本层的内聚，实现层之间的松耦合，并且三层架构增加了代码的复用性和扩充性，为以后业务需求的变化带来了便利。基于 SOA 是因为目前实训基地分为三个分基地，我们希望提供一套完整的 Web 服务供不同的基地调用，恰好 WebService 作为实现 SOA 的一种技术，充分满足我们的业务需求，故系统在逻辑层之上都进行了 Web 服务封装，以实现服务的共享化。

本文的选题“基于三层架构与 SOA”是充分提取 TBM 系统的技术优势而确定的。目前应用这两种技术开发的软件也有很多，但是针对软件学院实训基地管理的系统却没有。这篇论文就是要将利用这两种技术开发的 TBM 系统的设计论述清楚，让读者对这个具有实际意义的项目有基本的了解。

为了将 TBM 系统的设计展示清晰，本文共分以下 5 个部分：

第一章：相关技术　　（包括：层次架构和 SOA 技术）

第二章：TBM 系统需求分析（包括：功能需求和应用需求）

第三章：TBM 系统架构设计（包括：技术架构设计和表现形式设计）

第四章：TBM 系统应用设计

论文的最主要章节，包括公共对象设计、功能模块设计（包括：Entity 模块、SQLDAL 模块、BLL 模块、WebService 模块和 UI 模块）和后台服务程序设计。

第五章：设备子模块和权限子模块的实现

Recommend
火龙果软件

1 相关技术

1.1 分层构架

构架（Architecture，原意为建筑学设计艺术与科学），在 RUP 中的定义为：软件系统的构架是指系统重要构件的组织或结构，这些重要构件通过接口与不断减小的构件与接口所组成的构件进行交互；构架是系统在其所处环境中的最高层次的概念。软件系统的构架是通过接口交互的重要构件（在特定时间点）的组织或结构，这些构件又由一些更小的构件和接口组成。

分层是软件设计者在设计复杂软件系统时最常采用的一种技术。早在 20 世纪 90 年代，基于客户端/服务端（Client/Server）架构的系统不断增加，使得分层的概念开始流行起来，在这个时候，两层结构占有很重要的位置：客户端是一些用户界面和应用代码，而服务端则通常是一个关系数据库，如常见的使用 VB、PowerBuilder 和 Delphi 语言开发的访问数据库的应用程序都属于这种两层结构。这些工具使基于数据的应用程序开发变得非常的容易，只需要通过简单的拖曳和属性填写就可以实现界面组件和数据库之间的连接。例如在银行中应用很广的大型主机/终端方式。

两层的体系结构一直到现在还广泛存在，如果软件应用的主要功能是实现关系数据的展现和简单的更新等操作，那么客户端/服务端结构的软件系统能够非常好地满足要求。但是问题随着业务逻辑的出现接踵而至：如何进行业务规则的处理；数据校验、计算等诸如此类的问题。这时候，通常是将这些需求实现到客户端，也就是将业务逻辑绑定到了用户界面代码中间。当业务逻辑变得越来越复杂的时候，客户端的代码变得越来越难于维护，更甚者，绑定业务逻辑的客户端代码复制得到处都是，致使简单的需求改变却不得不找遍所有的客户端代码来进行修改。当然，另外一种方式是将业务逻辑作为存储过程绑定到数据库中间。然而，存储过程同样导致了软件系统对数据库的依赖；人们之所以采用关系数据库是因为 SQL 语言是独立于特定数据库的数据查询标准，这使得更换更合适的数据库厂商变得非常的容易。

在客户端/服务端应用变得越来越流行的时候，面向对象领域在不断的发展壮大过程中。针对客户端/服务端应用的缺点，面向对象社区提出了三层体系结构：表示层负责处理界面，业务层负责业务逻辑，再加上持久层（数据库或者数据源）。但由于两层架构转变到三层架构并不是那么容易，因此三层体系理论在很长一段时间内没有得以发展。但随着互联网的发展，人们需要将客户端/服务端应用能够通过浏览器进行访问。然而，如果所有的业务逻辑都在客户端，那么客户端业务逻辑必须要重新实现一套来支

Recommend
火龙果软件

持基于 WEB 的界面，而一个设计良好的三层体系结构只需要增加一个新的表示层就可以满足这个要求。

1.1.1 层次结构

层（layer）：对模型中同一抽象层次上的包进行分组的一种特定方式。通过分层，从逻辑上将子系统划分成许多集合，而层间关系的形成要遵循一定的规则。通过分层，可以限制子系统间的依赖关系，使系统以更松散的方式耦合，从而更易于维护。

由于英文中 Layer 和 Tier 都有层的概念，人们在讨论分层的时候往往混淆了 Layer 和 Tier，在这里我们来区分一下 Layer 和 Tier 的概念，如图 1.1 所示。

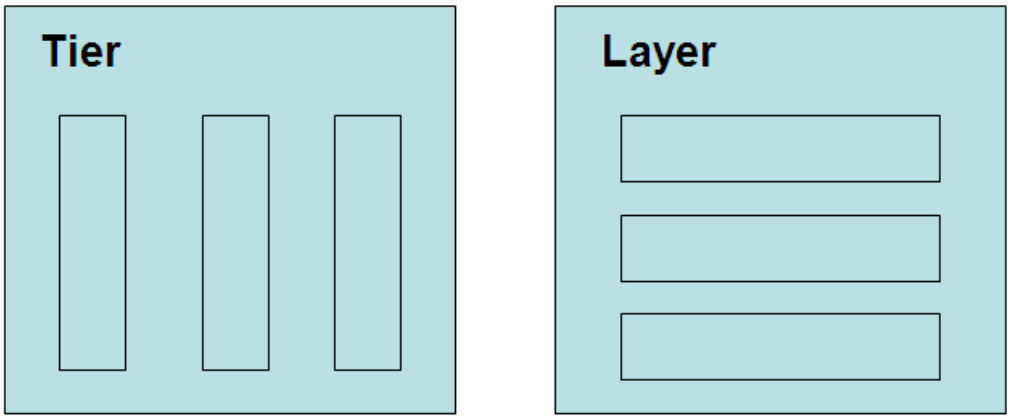


图 1.1 Layer 与 Tier 差异比较

值得注意的是，垂直和水平的划分某种程度上来说只是一个视角问题，其中揭示了物理上和逻辑上的区别。人们习惯将 Tier 表示为物理上的划分，也就是说，客户端/服务器端系统通常被表示为 two-tier 的系统，它们是物理上的划分：客户端是一个桌面系统，而服务端就是服务器。而 Layer 是表示系统所划分的 Layer，并不需要运行在不同的机器上。业务逻辑层可以运行在桌面系统机器上也可以运行在数据库服务器上。在这种情况下，整个系统就有两个节点而又三个 Layer。

除了物理与逻辑上的区分外，Layer 一般还暗示了“下面的 Layer 为上面的 Layer 提供服务”。请注意我们下面提到的层，指的都是 Layer。

（1） 表示层

表示层用来处理用户和软件之间的交互，如现在比较常见的客户端图形化界面程序，基于 HTML 和网页浏览器（以下简称浏览器）的 Web 程序等。它主要的责任是向用户展现信息以及处理用户请求，例如鼠标单击、内容输入、HTTP 请求等。并且在制

Recommend
火龙果软件

作表示层时要尽量符合用户的操作习惯，并具备一定的美观性。它位于三层架构的最顶层，如图 1.2 所示。

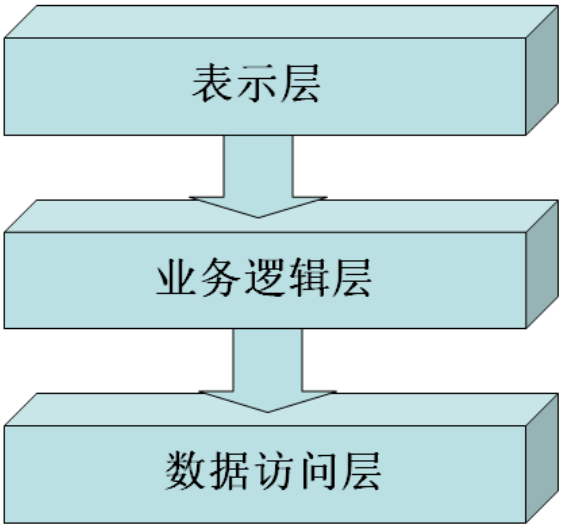


图 1.2 三层体系结构

（2） 业务逻辑层

业务逻辑层，又叫做领域逻辑层。其职责包括处理业务逻辑和存储业务数据，校验从表示层传过来的数据，通过表示层提交的命令来执行相应的业务逻辑。

（3） 数据访问层

数据访问层是用来存取业务状态数据的。数据访问层通过与系统其它层进行通信来完成应用的调用。其职责包括事务监控、消息系统、数据源等。在大多数的软件应用中，数据访问层最基本的功能就是存储持久化数据到数据库中，而关系数据库（RDBMS）则是目前最为流行的数据库。图 1.2 中箭头所示的方向，表示各个层之间的数据调用和依赖关系。这个依赖暗示了分层系统的特性规则：一层中的组件只能与同一级别中的对等实体或较低级别中的组件交互。

可以看到，业务逻辑层不会访问表示层中的内容，同样，数据层也不会访问业务逻辑中的内容。这个规则使得在同一个基础架构上实现不同的表示层成为可能，同时使得表示层的修改并不会影响到更深层次的实现。而业务逻辑层和数据层之间的关系就没有这么简单了，它依赖于数据层所使用的架构模式。三层架构作为分层架构的一种表现形式，目前被广泛的应用。这不仅降低了层与层之间的耦合度，还使系统更易于维护和更新，下面我们就来具体的认识一下分层架构的优缺点。

Recommend
火龙果软件

1.1.2 分层架构优点

分层架构的目的就是为了隔离各个层之间的耦合，并划分各个层之间的职责关系。有经验的开发者在进行架构设计时往往适当地采用分层来搭建基础的应用环境，为接下来的开发和维护做良好的铺垫。分层架构具有以下优点：

（1） 松耦合

由于层与层之间的依赖关系被最大程度地剥离，并且层之间的高内聚，以及层与层之间通过接口交互而剥离了对接口实现的依赖，使得系统解决方案的维护和增强变得更容易。比如，可以将原有 winForm 应用程序切换到 Web 页面，改变表示层的实现而不需要改变业务逻辑层和数据层的实现；又或者，将数据库从 SQLServer 切换到 Oracle 时，对表示层的影响减小到最低。

（2） 伸缩性

这个特性依赖于各个层实现的方式。当一个设计合适的分层系统为了满足业务增长的需要来提高系统的性能时，可以在压力大的层增加机器来增加其负载能力。显而易见的一个例子就是，原有的系统在进行数据库操作时，需要每一个客户端都建立一个数据库连接，而三层架构通过数据库连接池机制，可以利用少量的数据库资源来满足大量的用户请求的需要。另外，通过各个层的集群能够实现大容量的应用服务，将层分布在多个物理层上可以改善可伸缩性、容错和性能。

（3） 重用性

重用性和松耦合是有联系的，都是为了达到同一个实现能够满足多种应用需求的目的。例如，业务逻辑层能够被 Web、WAP 等多种表示层实现调用；同样，业务逻辑层在调用数据层时并不需要了解其对数据库的依赖。

（4） 扩展性

通过分层，可以实现新功能的增加而不会影响其它层的实现。只有在影响原有实现的接口时，才有可能影响到各个层之间的关系。同时，具有定义明确的层接口以及交换层接口的各个实现的能力提高了可测试性。

但是，分层架构也带来了一些损失，这主要是以下几个方面：

（1） 影响性能

穿越各层（而不是直接调用组件）所需的额外开销会对性能造成不利的影响。例如，客户端访问数据库中的数据时，通常需要经过多个层次，非常耗费性能，如何尽量减少数据库访问是分层应用系统首要解决的问题。例如，过于严格的分层会禁止表示层与数据访问层无法直接交互，导致开发表示层复杂的应用程序时可能需要更长的时间。此时

Recommend
火龙果软件

如果要弥补由此而带来的性能损失，可以使用松散的分层方法。通过这种方法，较高层可以直接调用较低层。

（2） 复杂性

层的使用有助于控制和封装大型应用程序的复杂性，但增加了简单应用程序的复杂性。对较低级别接口的改变可能会渗透到较高级别，尤其是在使用了松散的分层方法的情况下可能性更大。多层架构实际是将以前系统中的显示功能、业务运算功能和数据库功能完全分开，杜绝彼此的耦合与影响，从而实现松耦合和良好的可维护性。

可以看到，软件系统分层越多，它给系统带来的复杂性也就越高，同时对软件开发人员的要求也就越高。软件系统在设计阶段就需要考虑层次的划分和各个层之间的职能关系。否则在分层体系结构上出现某种缺陷，如各层之间出现了自下而上的依赖关系时，一旦业务逻辑层发生了变化，其带来的影响有可能涉及到整个软件系统。并且，软件划分的层次越多，所带来的沟通和调试问题也会越多。

在软件系统的规模较小，业务逻辑简单的情况下，适当的简化分层架构会带来开发效率和沟通上的提高。当然，这个架构的选择也依赖于具体的应用环境，开发者应当慎重选择其软件系统架构。

1.2 SOA 技术

面向服务的体系结构（service-oriented architecture，SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。

这种具有中立的接口定义（没有强制绑定到特定的实现上）的特征称为服务之间的松耦合。松耦合系统的好处有两点，一点是它的灵活性，另一点是，当组成整个应用程序的每个服务的内部结构和实现逐渐地发生改变时，它能够继续存在。而另一方面，紧耦合意味着应用程序的不同组件之间的接口与其功能和结构是紧密相连的，因而当需要对部分或整个应用程序进行某种形式的更改时，它们就显得非常脆弱。

对松耦合的系统的需要来源于业务应用程序需要根据业务的需要变得更加灵活，以适应不断变化的环境，比如经常改变的政策、业务级别、业务重点、合作伙伴关系、行业地位以及其他与业务有关的因素，这些因素甚至会影响业务的性质。我们称能够灵活地适应环境变化的业务为按需（On demand）业务，在按需业务中，一旦需要，就可以对完成或执行任务的方式进行必要的更改。

Recommend
火龙果软件

虽然面向服务的体系结构不是一个新鲜事物，但它却是传统的面向对象的模型的替代模型，面向对象的模型是紧耦合的，已经存在二十多年了。虽然基于 SOA 的系统并不排除使用面向对象的设计来构建单个服务，但是其整体设计却是面向服务的。由于它考虑到了系统内的对象，所以虽然 SOA 是基于对象的，但是作为一个整体，它却不是面向对象的。不同之处在于接口本身。SOA 系统原型的一个典型例子是通用对象请求代理体系结构（Common Object Request Broker Architecture，CORBA），它已经出现很长时间了，其定义的概念与 SOA 相似。

然而，现在的 SOA 已经有所不同了，因为它依赖于一些更新的进展，这些进展是以可扩展标记语言（eXtensible Markup Language，XML）为基础的。通过使用基于 XML 的语言（称为 Web 服务描述语言（Web Services Definition Language，WSDL））来描述接口，服务已经转到更动态且更灵活的接口系统中，非以前 CORBA 中的接口描述语言（Interface Definition Language，IDL）可比了。

Web 服务就是实现 SOA 的一种方式，这里，我们主要探讨 Web 服务的原理与应用。

1.2.1 Web 服务

Web Service 是构筑在 XML 与 Soap 技术之上的一种典型的异构分布技术。自 Web Service 技术出现以來，其开放性的标准让各个不同系统间能够互相交换资料。这种良好的安全开放性使得以 Web Service 构建独立安全系统成为发展的新趋势。为实现不同商业企业间的应用系统提供动态安全集成，体现服务共享的真正价值。

Web Services 是由企业发布的用于完成其特别业务需求的在线应用服务，其他企业、合作伙伴的应用软件能够通过 Intranet、Internet 来动态地访问并使用这些在线服务。它是技术与市场发展的必然结果，是在 IBM、微软等许多平台供应商、软件提供商共同推动下，并在 W3C（万维网联盟）的工作流、数据安全性等方面的规范下发展起来的标准，各个供应商对其的支持程度超过了以往的任何跨供应商的软件产品。

对于 Web Services 来说它就是一组服务，是较完整的地将安全、认证等基本功能为一体的服务平台，并具有一系列相关的技术标准，这些标准使其具有作为一个服务平台的完整性和优越性。也可认为是存在于 Web 服务器上的一组程序，这组程序被封装成一个暗箱，对外提供一个能通过 Web 进行调用的 API 接口，可用编程来调用它，其执行结果被回传到客户端。

总之我们可以简单地把它理解为：Web Services 是一个集部署、发现、事务、安全、认证等基本功能为一体的服务平台，并具有一系列相关的技术标准，这些标准使其具有作为一个服务平台的完整性和优越性。尽管不同的商业公司在战略上开展 Web Services

Recommend 火龙果软件

的出发点不尽相同，但都确立了软件变服务的商业模式，并着力于解决服务的智能化以及个性化等问题。

1.2.2 Web 服务特点

（1） 面向服务

Web Services 把一切都看作服务，这种服务可以在网络上通过消息传递机制动态地被发现、组织和重用。

（2） 互操作性与松散耦合

通过 SOAP 消息机制远程调用进行应用交互，任何 Web 服务都可以与其他 Web 服务进行交互，避免了在 CORBA、DCOM（分布式组件对象模型）和其它协议之间转换的麻烦，开发者可以使用任何语言来编写 Web 服务，无需变更他们的开发环境。Web Services 使用者与 Web Services 提供者之间是松散耦合。

（3） 封装性

Web Services 对外封装成由 WSDL 描述的服务，屏蔽了业务逻辑的复杂性、实现技术的多样性和开发平台的异构性。

（4） 普遍性

Web 服务技术基于一系列开放标准。Web 服务使用 HTTP 和 XML 进行通信，因此，任何支持这些技术的设备都可以拥有和访问 Web 服务，具有使用上的广泛性。

（5） 集成性和简单性

可以使用任何语言（如 C、C++、VB、VC 等）来编写 Web 服务，开发者无需更改他们的开发环境就可以生产和使用 Web 服务。Web 服务技术不仅易于理解，并且 IBM、微软等大的供应商所提供的开发工具能够让开发者快速创建、部署 Web 服务，已有的 COM（Component Object Model 组件对象模型）组件、Java Bean 等也可方便地转化为 Web 服务。

由于 Web Services 技术建立在标准性与开放性基础之上，彻底打破了以前封闭式的实现方法，通过 Web Services 技术，使用不同语言开发、运行在各个不同平台上的客户端应用可以无缝地获取所需应用，达到资源效率的最大化。

SOA 的概念并非什么新东西，SOA 不同于现有的分布式技术之处在于大多数软件商接受它并有可以实现 SOA 的平台或应用程序。SOA 伴随着无处不在的标准，为企业的现有资产或投资带来了更好的重用性。SOA 能够在最新的和现有的应用之上创建应用；SOA 能够使客户或服务消费者免予服务实现的改变所带来的影响；SOA 能够升级单个服务或服务消费者而无需重写整个应用，也无需保留已经不再适用于新需求的现有

Recommend
火龙果软件

系统。总而言之，SOA 以借助现有的应用来组合产生新服务的敏捷方式，提供给企业更好的灵活性来构建应用程序和业务流程。

1.3 TBM 系统技术优势

TBM 系统是基于三层架构及 SOA 技术开发的。这样，TBM 系统就综合了两种技术的优势。当然，我们也不是说采用这两种体系结构配合的方式开发出的系统是最好的，但综合考虑实训基地对软件的应用需求问题，我们觉得基于这两种技术所开发的系统是很有意义的。这里我们只讨论系统技术上的优势：

首先，既然 TBM 是采用三层架构开发的，那么它一定具有所有采用架构技术开发的软件的优势，即：

（1）开发人员可以只关注整个结构中的其中某一层；

（2）可以很容易的用新的实现来替换原有层次的实现；

（3）可以降低层与层之间的依赖

（4）有利于标准化；

（5）利于各层逻辑的复用。

这就使我们代码开发模块化、颗粒化、层次化。能及时、方便的应对各种需求变化，更加便于今后的维护和更新。

其次，TBM 系统也是基于 SOA 技术开发的。这里，我们主要应用 Web 服务技术。我们知道，采用 Web Service 解决方式时，需要确定提供给其他应用程序的 API（应用程序接口），生成相应的 WSDL（Web 服务描述语言）文档，然后将异种分布式组件封装成 Web Service。使用基于 XML 的 SOAP 相互通信，便可实现应用系统的继承。对于我们的系统，将所有逻辑层方法都封装成 Web Service 方法，即对外提供的 API，供不同基地、不同平台的应用程序调用，各基地只需根据自身业务需求实现本身独有的应用程序。已实现底层代码共享、服务共用的目的。

最后，系统部署方便、简单。我们只需在服务器端发布一个用于 Web 服务功能的网站，那么基地内所有的客户端程序（无论什么平台、什么语言，只要可与 XML 进行交互）便可轻松的获得 Web 服务，进行基地事务管理。

综上，TBM 系统在采用三层架构及 SOA 技术后，无论从开发、安装及部署上，都大大的方便了实训基地的管理事务需求，具有较大的意义。

Recommend
火龙果软件

2 TBM 系统需求分析

一个软件的开发意义，在于它的实用性，这个实用性是指它存在的价值及意义。要么它可以带来巨大的商业利润、要么它可以将大量的人力劳动转变为相对自动的计算机处理程序。TBM 系统就是为了满足第二点需求，用以实现实训基地内部系统化及半自动化管理运作的需求。为了实现系统的自动考勤功能，TBM 系统和现有的门禁系统要做数据的交互，即从门禁系统中读出员工的刷卡记录，应用到本系统中进行考勤管理。系统的总体应用需求功能如 2.1 图所示：

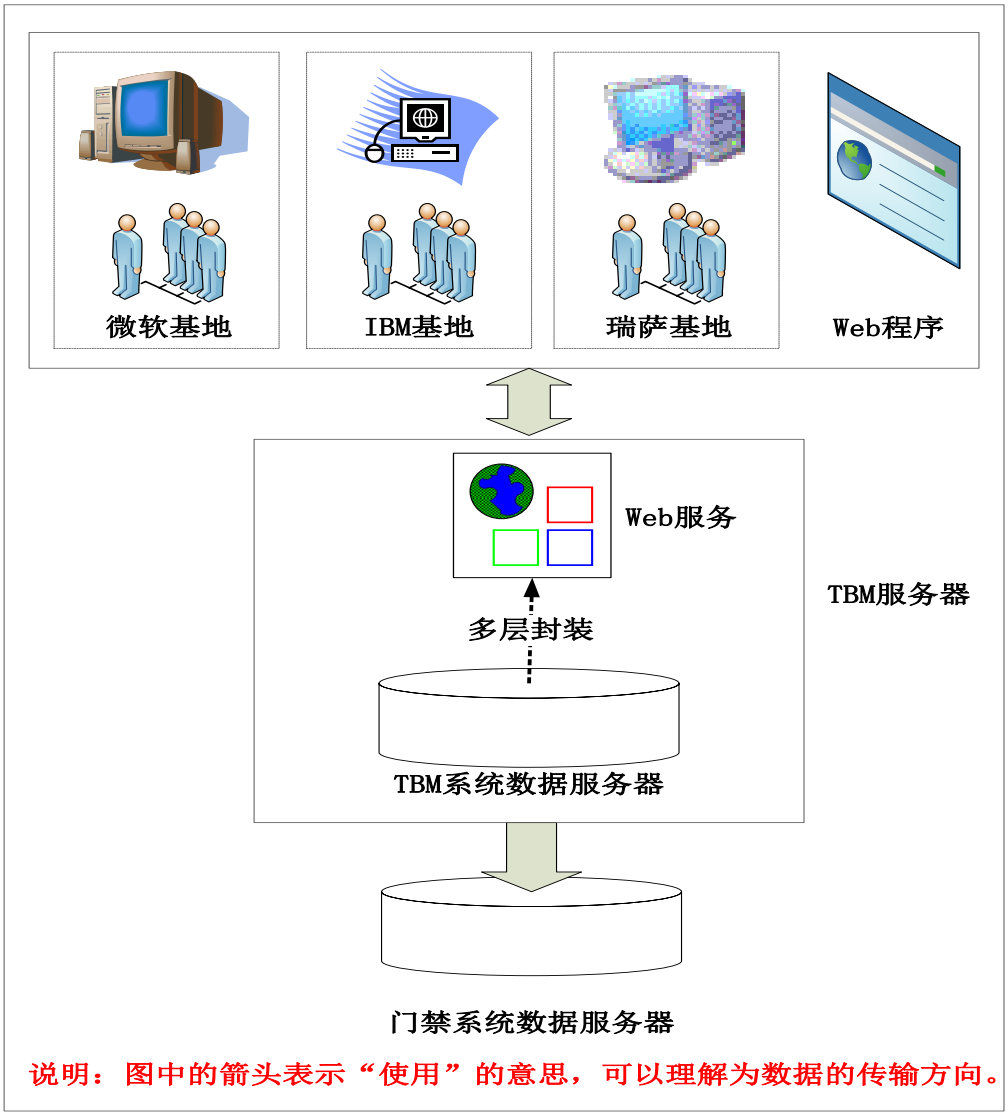


图 2.1 系统总体需求功能结构图

Recommend
火龙果软件

目前，实训基地共分为三个基地：IBM 基地、微软基地及瑞萨基地。所有的日常管理工作都是由各个职能部门的管理者来手动完成（如考勤、编排值日表等），工作量巨大。这不仅耗费了大量的人力，效率低下，还使管理不能统一化与规范化。TBM 系统开发的目的，就是希望通过提供一个通用的后台处理、服务平台，为各个基地提供数据支持及交互。普通员工通过 Web 浏览器端进行操作（即传统的 B/S 结构），而管理员工（例如指导教师、经理助理、HR、运营部部长和 Club 部长等）则操作每个基地内独有的管理界面（即传统的 C/S 结构），这个界面程序不基于任何平台及运行环境。这样，只要在后台服务器端发布一个可访问的 Web 服务，就能供至少三个独立的基地进行对数据操作的需求。同时，通过一个通用的网站，使普通员工可以方便的进行各种申请及信息查看。已达到事物申请与处理的最大化自动性与统一性。

下面我们将通过功能需求和应用需求两个方面来全面的分析 TBM 系统的需求。

2.1 功能需求

在实训基地日常管理中，以人力资源管理、考勤管理、值日管理、设备管理等环节最为重要。TBM 系统就是要实现数据录入、数据维护等一系操作，将原来的人工记录操作转变为程序处理的方式进行，来大幅度提高工作效率和管理质量。系统的总体功能模块图 2.2 所示：

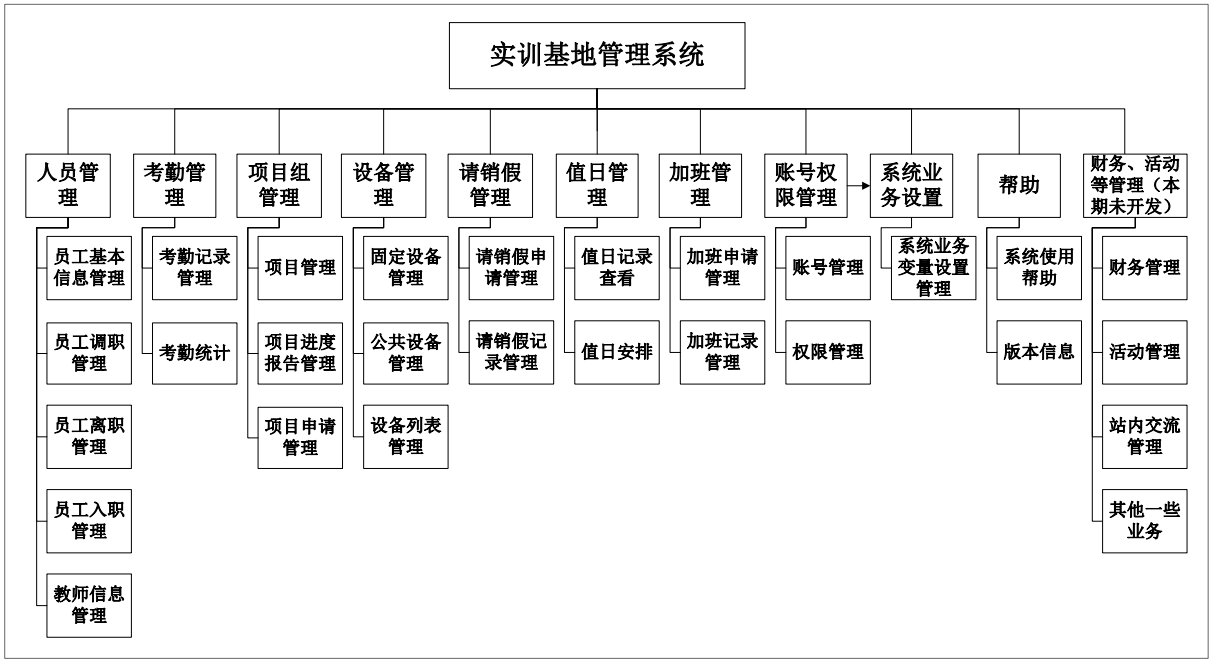


图 2.2 系统功能模块图

Recommend
火龙果软件

综合分析系统需要完成的功能，TBM 系统主要需要实现以下功能：人员管理，考勤管理，项目组管理，设备管理，请销假管理，值日管理，加班管理，账户及权限管理和系统管理。

（1）人员管理

人员管理涉及员工的申请注册、基本信息管理、离调职管理、教师基本信息管理和各种业务单子打印等功能。

对于一名新的员工，他首先要进行网上注册或直接找 HR 进行信息的录入。当他入职后，他可以进行调职申请、离职申请等操作。在管理程序端，HR 或教师可以审批他的申请，并予以通过或拒绝，并打印一些申请单。

（2）考勤管理

考勤管理是 TBM 系统需求较多的模块，涉及对人员上下班查岗、更新考勤状态等操作，来达到考勤自动判断的功能。该模块通过读取门禁系统数据库中的数据获得 TBM 系统所需的业务数据，从而进行逻辑判断与处理，分析一个员工是不是迟到、早退、矿工等状态，并保存进记录。并且，该模块还涉及考勤统计与查询，打印考勤记录等操作。（目前门禁系统的数据获得还未处理，我们开发了一个模拟程序，来获得所需数据）。

（3）项目组管理

项目组是各基地的基本组成单位。这里的项目组管理包括项目管理，项目申请管理，项目进度报告的汇报、审批、查看等功能。来达到指导教师对其指导的项目组，合理监控与指示的作用，从而保证各项目组与教师的及时交流，降低项目开发风险。

（4）设备管理

实训基地内拥有大量的实训设备，如何对这些设备进行组织化管理是这个模块要完成的重点工作。设备分为固定设备和公共设备。对于固定设备，这部分完成固定设备的分配、报修、变更申请及审批等操作。对于公共设备，这部分提供对公共设备的申请与审核操作，并且允许设备管理员进行设备状态的查询，系统可以人性化的提示该设备是否被借出，是否超期未归还等功能。同时，对于所有设备，提供设备的增加、状态改变等功能。最大限度的满足设备管理员的工作业务需求。

（5）请销假管理

实训类似于正常的工作，作为实训基地的内的所有员工，他们会有请假的需求。这样，TBM 系统要规范的、统一的处理所有的请销假申请。并且，管理者进行审批，予以假期或销假。对于普通员工，涉及请假、销假申请操作。对于管理者，涉及请销假记录查看与审批的操作。

（6）值日管理

Recommend
火龙果软件

一个正常运作的基地，必须需要正确而合理的值日安排，来保证卫生和财产安全。这样，该模块涉及到值日安排、值日表查看、值日记录填写等操作。对于普通员工，可以查看近期值日排班情况、填写值日单。对于管理者，可以查看值日记录，打印值日单等操作。

（7）加班管理

当项目进度落后或者任务量增加时，各个项目组或个人有可能需要加班。这样，该模块涉及到加班的申请与审核。并且在管理者不参与的情况下，系统能自动审批一些加班申请记录（当申请的人数小于系统要求的人数时，予以拒绝）。

（8）账户及权限管理

对于一个完整的系统，会涉及很多角色和权限。此模块主要负责处理账号及权限的对应与分配。可以添加一个账户、删除一个账户、更改账户访问、读写权限等。

（9）系统管理

系统运行需要一些业务参数，如：上班时间、下班时间、系统自动判断时间、每天值日人数、节假日等。该模块主要负责一些业务时间的设置、节假日的设置与其他业务参数的设置与更改。使系统的业务是可变化的，不是固定的。

注：系统最初设计还包括模拟财务管理、活动管理、站内交流等功能。但由于项目组人员变化较大，这些模块没有实现。只实现了上述的 9 个模块，但该 9 个模块已经可以全面的展示 TBM 系统的复杂性。还有，本次开发只实现了系统的管理程序，即 WinForm 窗体程序，而供普通员工操作访问的 web 程序还没有开发，将待下一期继续完成（系统的核心在后台的管理程序，本期已经开发完毕）。

2.2 应用需求

如果说系统的功能需求可以将一个概念性的东西转化为实际代码，那么系统的应用需求就是为系统做美化处理。我们只有知道应用需求，才能做出满足使用者操作方便的实用软件，特别是对于管理类软件来说，“可操作性”是重要的一个部分。这里的可操作性指：

（1）部署简易型

前面已经介绍过，系统是 B/S、C/S 混合的结构。如何使系统部署简易是十分必要的。目前，我们需要在服务器端部署 Web 服务，然后客户端（Web 程序及 WinForm 程序）就可以简单的访问我们的服务，进行业务操作了。这是 TBM 系统的一个主要优势。

（2）功能强大性

Recommend
火龙果软件

对于上述开发的 9 的管理模块，我们进行了详细的需求分析，综合得出了可以满足基地管理的解决方案，做到最大化的减少人工操作，增加管理效率。

（3）界面良好性

TBM 系统根据业务需求，有些模块使用多文档形式呈现，有些模块使用窗体形式呈现，使用户在使用时舒适、方便。并且，系统统一了页面的风格、与操作方式，使用户可以很快的掌握操作方式。在处理输入时，我们尽可能简化用户的输入信息，并且给与足够的提示，使用户可以清晰地知道自己将要进行的操作。

（4）帮助详尽性

对于一款管理类软件，详尽的帮助文档必不可少，我们将帮助文档总结成章，详细的介绍软件的操作方式和用户需要了解的一些模块知识。让用户不仅知道怎么用，还知道这么用的原因。

（5）安全性

TBM 系统既然是集 B/S、C/S 于一体的综合软件产品，管理系统的安全性必须得到保证，我们利用 SOA 技术本身具有的安全验证机制，帮助辅助建立 TBM 系统的安全处理方法。

（6）响应快速性

既然是一款基于 Web 服务的应用程序，它的响应速度自然受到了使用者的关注。我们通过优化数据访问层，降低连接数目，来尽量减少连接池的使用负荷，增加系统的响应敏捷度。

Recommend
火龙果软件

3 TBM 系统架构设计

一个系统在充分分析了需求后，我们需要为它建立一个完整的系统构架，以方便设计者和开发者清晰的了解系统的总体形式及展现方式。由于 TBM 系统涉及同学们的网上注册、申请、入职后的考勤管理、值日编排、设备管理等业务工作。并且系统使用者主要分为在实训基地的同学和实训基地的管理者（包括有职务的同学和老师）。故系统采用 C/S、B/S 兼有的模式。TBM 系统将同基地现有的门禁系统进行结合，选取有用的刷卡记录作为考勤依据。并且系统将在后台进行判断，实现员工考勤状态的自动修改等一系列自动操作。下期待加入新薪资管理后，TBM 系统就可以通过考勤记录实现虚拟工资的统计及发放了（虚拟工资可作为评价同学实训效果的一种策略与标准）。

3.1 技术架构设计

系统采用分层设计的思想。并对传统的三层架构模式进行了扩充（将逻辑层大多数函数封装成 Web 服务，供上层调用）。这样系统分为三层：数据访问层、业务逻辑层和表示层（WebUI 及 WinFormUI 两种展现形式）。数据访问层主要对数据库数据进行操作，包括增加数据，修改数据和删除数据。业务逻辑层主要针对业务流程来实现系统的功能，其中封装的 Web 服务对逻辑层方法进行进一步封装，实现基于 SOAP 与 XML 的信息传输，供上层不同的应用程序调用。表示层把业务逻辑表现在 Web 程序和 WinForm 程序上，以完成实训基地管理系统所要完成的功能。其技术架构如图 3.1 所示：

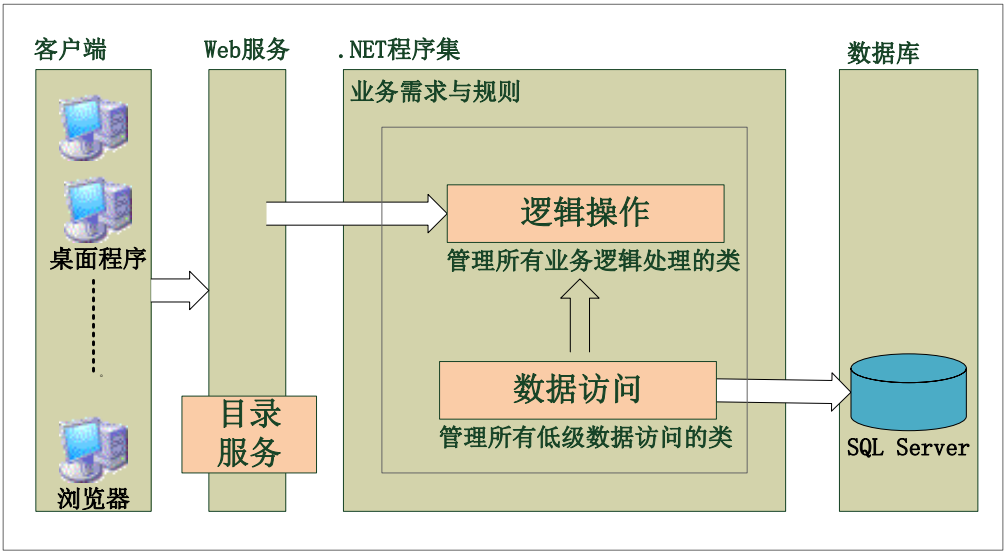


图 3.1 系统技术架构图

Recommend
火龙果软件

将技术架构图转化为功能结构图 3.2，能清楚的看到 TBM 系统的层次结构。

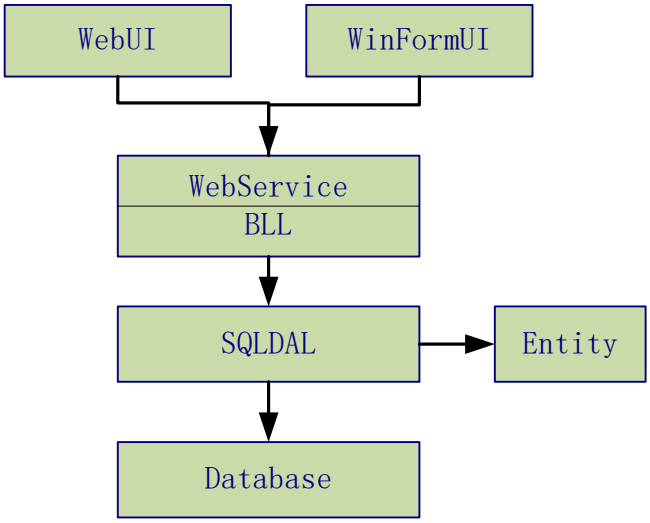


图 3.2 系统功能结构图

从 3.2 图中以看出，TBM 系统由下至上主要分三层，即数据访问层、业务逻辑层和表现层（我们将 Web 服务层归属于逻辑层，因为在 TBM 系统中这两者的关联性较大）。分别命名为 SQLDAL、BLL 和 WebUI 与 WinFormUI。在数据访问层中，有一个特殊的程序集-----实体集 Entity，它的设计满足面向对象的思想，将关系数据库中的实体表与程序的对象类一一对应起来。

这样，TBM 系统在开发的过程中就可以按照分层的顺序，逐层开发，逐层实现、由高层方法调用底层接口，实现较高层次与较低层次的松散耦合。

3.2 表现形式

由于系统的使用者涉及基地的所有实训同学和部分管理人员，在设计系统表现形式时，要综合如部署简易型、使用方便性、操作可行性等间接要求，既要达到使用者的无障碍使用，又要达到系统的资源充分利用。综合以上考虑，系统在表现形式上采取 B/S 与 C/S 综合的混合结构。

对于所有普通实训员工，操作的界面是 web 浏览器，这样做的目的是因为为这些用户（人数较多）分配一个 WinForm 的客户端意义不大，而且这些用户对系统的操作也不是十分频繁的，为其提供一个可供访问和操作的网页操作完全可以满足业务需求。对基地运行的管理者（有职务的同学及教师），这些人操作的页面大多是 WinForm 窗体客户端，因为这类人需要频繁的进行考勤查看、项目审核、值日分配、设备管理等操作。

Recommend
火龙果软件

让他们一旦操作的时候就打开浏览器有些不方便，这样，为这类人提供一个与服务器实时交互的 WinFrom 应用程序比操作浏览器的方式更加便利。

上面讨论了使用上的需求，那么我们再来从技术上的需求分析下为什么采用这两种表现形式：

（1）B/S（Browser/Server）结构即浏览器和服务器结构

浏览器和服务器结构。它是随着 Internet 技术的兴起，对 C/S 结构的一种变化或者改进的结构。在这种结构下，用户工作界面是通过 WWW 浏览器来实现，极少部分事务逻辑在前端（Browser）实现，但是主要事务逻辑在服务器端（Server）实现，形成所谓三层 3-tier 结构。这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本（TCO）。以目前的技术看，局域网建立 B/S 结构的网络应用，并通过 Internet/Intranet 模式下数据库应用，相对易于把握、成本也是较低的。它是一次性到位的开发，能实现不同的人员，从不同的地点，以不同的接入方式（比如 LAN, WAN, Internet/Intranet 等）访问和操作共同的数据库；它能有效地保护数据平台和管理访问权限，服务器数据库也很安全 。

（2） C/S（Client/Server）结构即即客户机/服务器模式

客户机和服务器结构。它是软件系统体系结构，通过它可以充分利用两端硬件环境的优势，将任务合理分配到 Client 端和 Server 端来实现，降低了系统的通讯开销。

综合比较 B/S 结构的节约成本、安全、方便的优势和 C/S 结构客户端对业务有一定处理能力的需求，我们将系统设计为 B/S 与 C/S 混合的结构。即普通员工使用端采取 Web 程序，管理员工使用端采取 WinForm 程序。

Recommend
火龙果软件

4 TBM 系统应用设计

4.1 公共对象设计

标识：Tools.cs
功能：处理不同数据集之间的装换
输入：object 类型的数组或 List 集合
输出：List 集合或数组
信息：由于逻辑层的方法经 WebService 封装后，原有的实体对象会被转换成满足 soap 协议和 XML 标准的传输形式。对于对象集的操作，在形式上是传输或返回 object 数组。在表示层应用的 Web 服务的时候需要将 object 数组转换成我们需要的数据集。这样这个工具类的作用就体现出来了。它实现了将 object 数组转化为 ArrayList、将 ArrayList 转换为 object 数组等一些列操作。

标识：UPConstList.cs
功能：记录 TBM 系统使用者操作权限常量的类
输入：无
输出：静态的常量权限值
信息：由于系统权限设置为 2 进制表示，每一个十进制数都对应一个权限，这样将权限概括出一个由常量组成的类，以方便其他页面及方法判断各种权限时的操作。

标识：ExcelHepler.cs
功能：处理 Excel 操作
输入：文件路径名
输出：需要打印的 Excel 文档
信息：完成一些打印 Excel 表格的抽象操作，提高代码的复用度。

标识：WordHelper.cs
功能：处理 Word 操作
输入：文件路径名
输出：需要打印的 Word 文档
信息：完成一些打印 Word 文档的抽象操作，提高代码的复用度。

Recommend
火龙果软件

标识：SQLServerDBHelper.cs
功能：通用的操作数据库语句
输入：数据库连接，SQL 语句
输出：相应的数据和状态
信息：主要是对 SQLDAL 中出现的公共运用的 SQL 访问提取其方法，以提高代码复用。

4.2 功能模块设计

对于一个分层架构的系统来说，每一层其实就可看做是系统的一个功能模块。那么 TBM 系统的功能模块可分为 Entity 模块、SQLDAL 模块、BLL 模块和 UI 模块。其中每个功能模块完成的操作都是基于上一层提供的接口的，这样功能模块与功能模块间势必要进行数据传输，有一定的交互性，但是功能模块间却是松散耦合的。下面将分别介绍 TBM 系统的 4 大功能模块的设计：

4.2.1 Entity 模块

由于 TBM 系统采用面向对象的编程思想，所以在建立实体类的时候，基本上是保持数据库中实体表与程序中实体类一一对应的关系。下面分别介绍各个业务模块的对象及实体关系：

（1） 人员管理模块的实体对象关系，如图 4.1 所示：

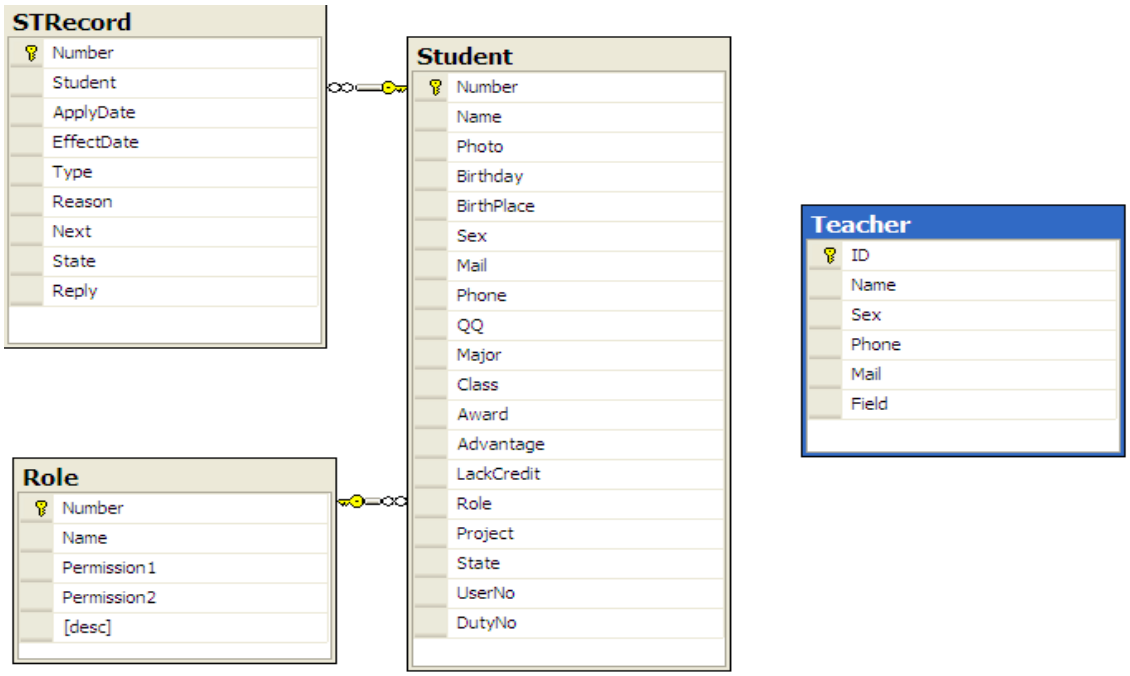


图 4.1 人员管理模块实体对象关系图

Recommend
火龙果软件

说明：人员管理模块主要涉及学生实体（Student）、角色实体（role）、调离职记录实体（STRecord）和教师实体（Teacher）。学生实体是系统最重要的实体之一。它记录了学生的基本信息和实训信息。调离职记录实体标识了每一条的调离职信息。角色实体是系统内所有角色的一个集合。教师实体是人员管理中一个特殊的实体，标识系统内的所有教师（TBM 系统将教师看成是一类特殊的使用者，角色中没有教师），它独立的存在于数据库和程序中。

（2） 考勤管理模块的对象关系，如图 4.2 所示：

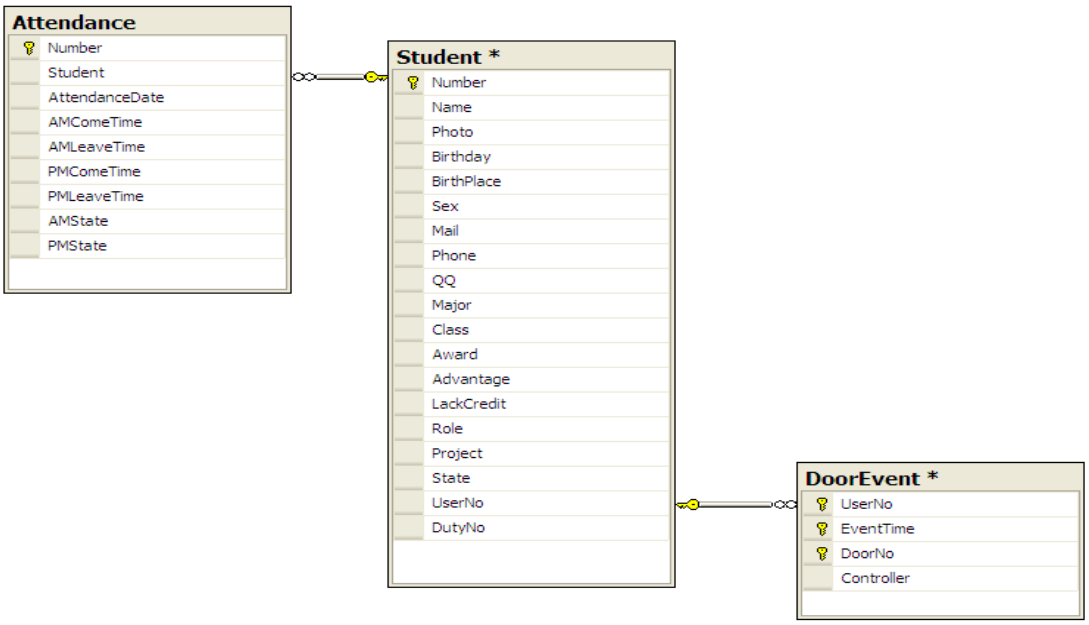


图 4.2 考勤管理模块对象关系图

说明：在此模块中，涉及考勤记录实体（Attendance）和门事件实体（DoorEvent）。考勤实体是系统另一个重要的实体。它的每一条记录都代表了一个员工一天内的考勤状态。并且，TBM 系统需要从“DoorEvent”实体中获取用户的进出门刷卡时间，用于自动判断，生成相应的考勤记录。门事件实体中的所有记录都是从现有的门禁系统中选取获得的（目前未实现数据的读取，用模拟程序模拟输入代替）。

Recommend
火龙果软件

（3） 项目组管理模块的对象关系，如图 4.3 所示：

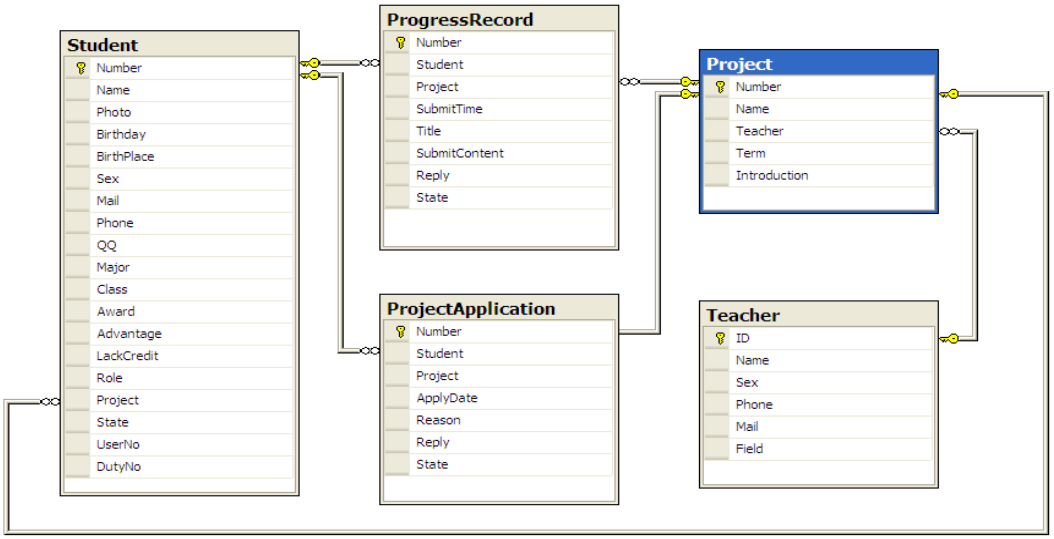


图 4.3 项目组管理模块对象关系图

在此模块中，涉及到的实体数目较多，主要有项目实体（Project）、项目进度报告实体（ProgressRecord）和项目申请实体（ProjectApplication）。并且该模块与教师实体和学生实体关系密切。只有教师可以发布及删除修改项目，这样学生才可以申请已有的项目组进行加入，才存在该项的进度报告和项目申请等实体。有些刚入职的学生是没有加入项目组的，该字段可以为空。

（4） 值日管理模块的对象关系，如图 4.4 所示：

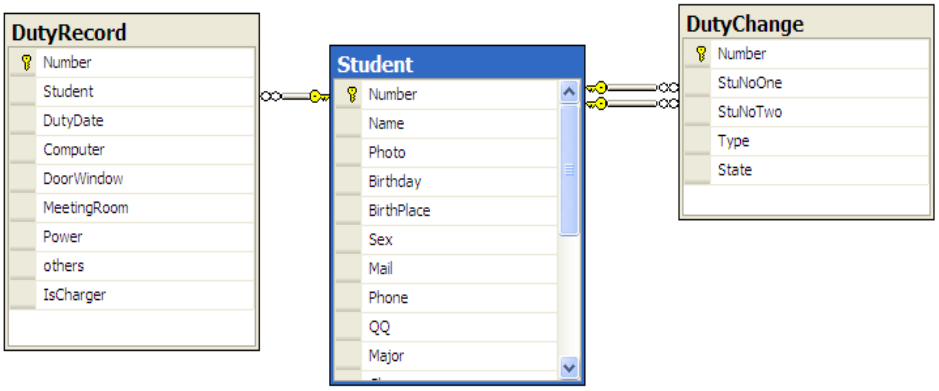


图 4.4 值日管理模块对象关系图

Recommend
火龙果软件

说明：此模块主要涉及值日记录（DutyRecord）和值日变更记录（DutyChange）两个实体。值日记录实体负责记录每天值日的具体汇报情况。值日变更实体主要负责制定每天值日的负责人（规定系统每天必须有一名值日方面的负责人）。利用这两个实体，可以进行值日记录查看和值日责任变更等操作。

（5） 设备管理模块的对象关系，如图 4.5 所示：

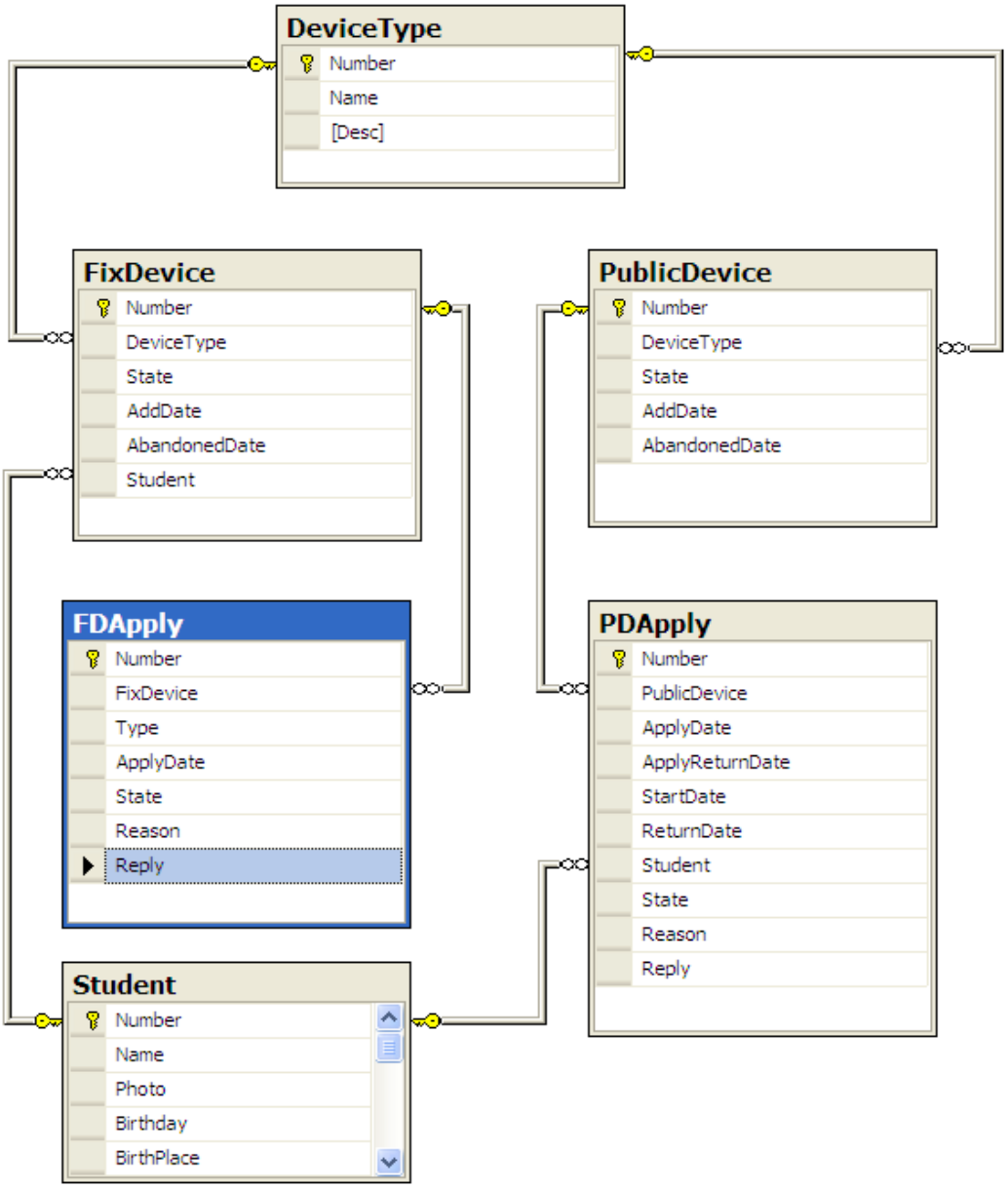


图 4.5 设备管理模块对象关系图

Recommend
火龙果软件

说明：此模块主要涉及设备类别实体（DeviceType）、公共设备实体（PublicDevice）、固定设备实体（FixDevice）、固定设备申请实体（FDApply）和公共设备申请实体（PDApply）。由于对于公共设备和固定设备的管理与数据结构是不同的，将二者分开成两种实体表示，以增加系统的健壮性（但在数据库操作上抽象成了一个便于操作的 Device 视图）。

（6） 请销假管理模块的对象关系，如图 4.6 所示：

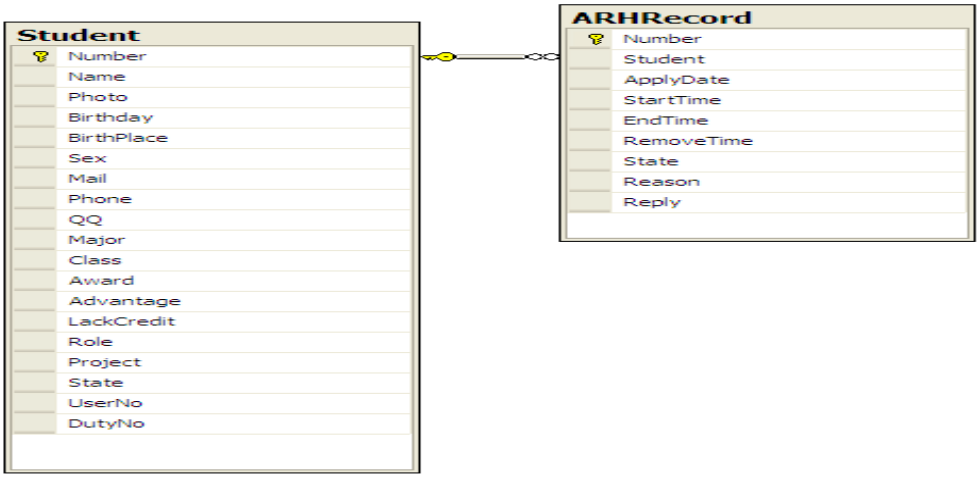


图 4.6 请销假管理对象关系图

说明：该模块涉及请销假记录实体（ARHRecord）。这个实体记录了一位学生的请假时间、请假开始时间、请假截止时间等信息。用以表示一个学生某次的请销假记录情况。并且该实体与考勤记录实体密切相关。

（7） 加班管理模块的对象关系，如图 4.7 所示：

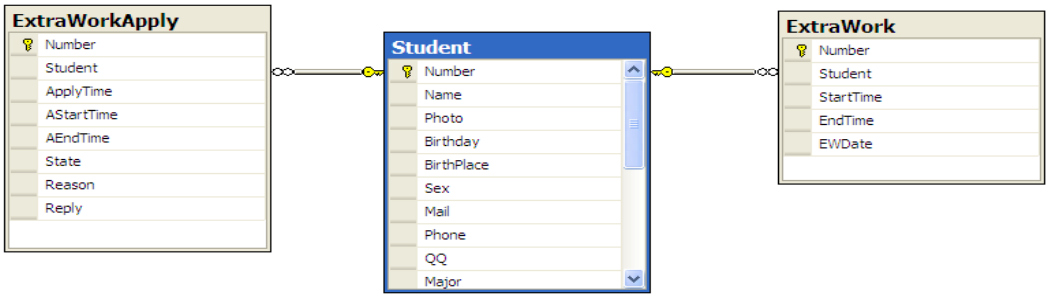


图 4.7 加班管理对象关系图

Recommend
火龙果软件

说明：此模块主要涉及加班记录（ExtraWork）和加班申请记录（ExtraWorkApply）两个实体。加班记录实体保存的是一条条的加班记录，加班申请实体保存的是每条加班申请记录。待系统增加薪资管理模块后，加班记录实体会与薪资模块中的部分实体密切相关。

（8） 账号及权限模块的对象关系，如图 4.8 所示：

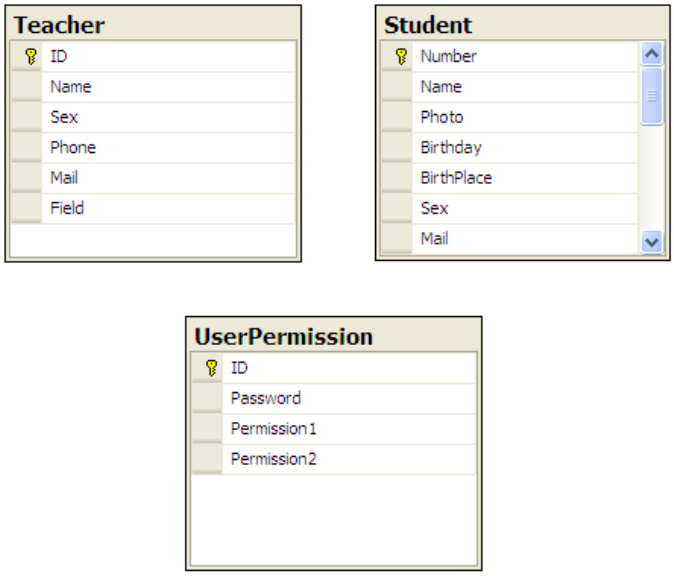


图 4.8 账号及权限对象关系

说明：该模块以 UserPermission 实体为主。该实体记录了一个账号的账户名、密码和权限。权限用于判断用户操作时所具有的操作权限。这个实体跟教师实体、学生实体间关联很小，即：这个实体中存在的对象与另两个实体中存在的对象没有任何关系。

（9） 系统设置模块的对象关系，如图 4.9 所示：

图 4.9 系统设置对象关系图

Recommend
火龙果软件

说明：此模块涉及系统业务设置实体（SystemSet）和假期实体（Holiday）。SystemSet 实体记录一些 TBM 系统正常运行要使用的业务常量，比如上班时间、下班时间、每天值日人数等。这样就使得系统的业务有选择性的，不是定死的。Holiday 实体记录了所有的假期，这样考勤模块、请销假、值日模块就可以根据假日再进行其模块的处理了，使系统的真实性更强。

将上述所有的数据实体转化为开发程序中对应的实体类，那么就得到了 TBM 系统中处于最底层的实体类程序集，如图 4.10 所示：

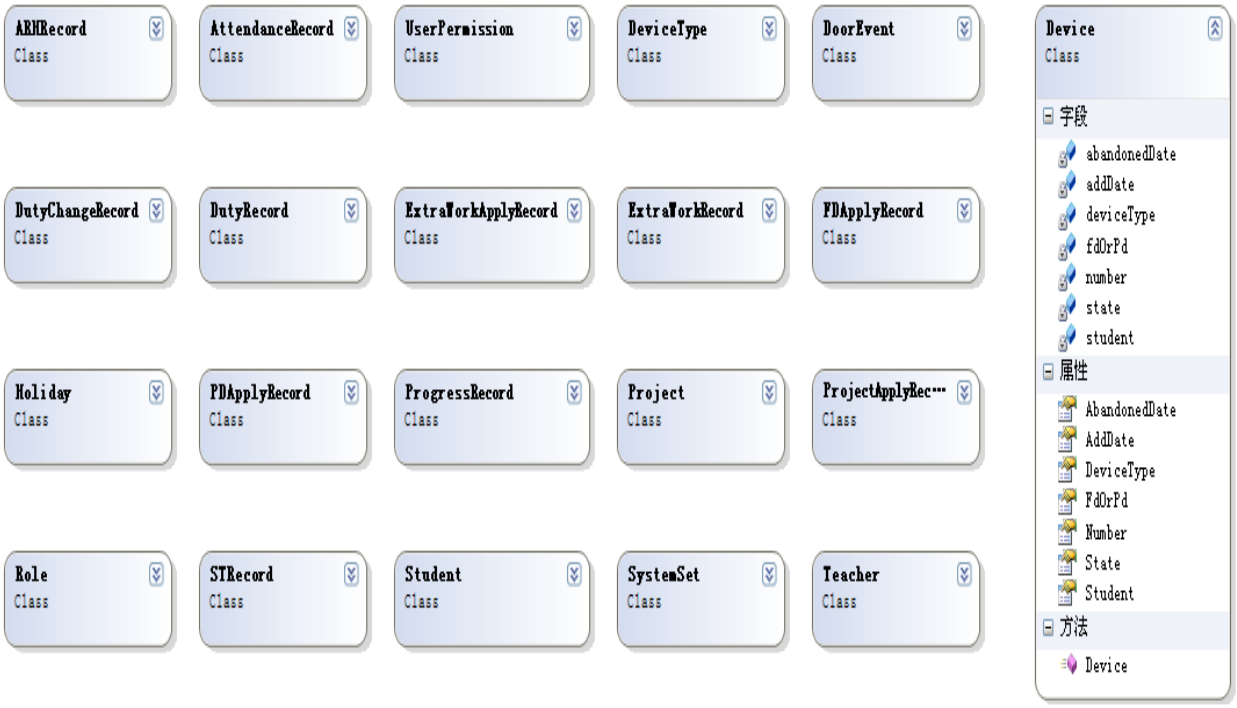


图 4.10 TBM 系统实体类程序集

在搭建出实体类程序集后，我们将需要对数据实体的操作就转化为了对程序中实体类的操作，这充分体现了关系型数据库的优势与面向对象编程思想的优势。我们在程序中只需处理实体类，而不必关心对应的数据表了。需要说明的是，由于设备模块的特殊性，在映射固定设备和公共设备成实体类时，我们发现如果映射成两个实体类，那么会有很多冗余的操作。为了操作便捷与易懂，我们将固定设备表和公共设备表抽象出了视图 Devices，然后将该视图映射成实体类 Device。至此，系统的实体类集合搭建完毕，TBM 系统共包含 21 个实体类，以后的操作都是基于这些实体类进行的。

Recommend
火龙果软件

4.2.2 SQLDAL 模块

SQLDAL 模块主要涉及数据访问，通过 SQL 的数据访问接口来提取数据库的数据和传入数据到数据库中存储。各个领域对象的 SQLDAL 主要涉及两个部分：一、单记录的操作，提取一条记录存储在 Entity 中，程序操作完毕后，把修改后的数据更新到数据库中。二、多记录操作，通过 ArrayList 等数据集的操作来提取数据和更新数据到数据库中。

（1） 人员管理模块，数据访问类如图 4.11 所示：

图 4.11 人员管理数据访问类图

StudentDAL:提供了和数据库交互的操作，包括插入学生的基本信息，从数据库加载记录信息以及更新数据库记录信息等功能。

STRecordDAL:提供请销假记录的增删改查操作。

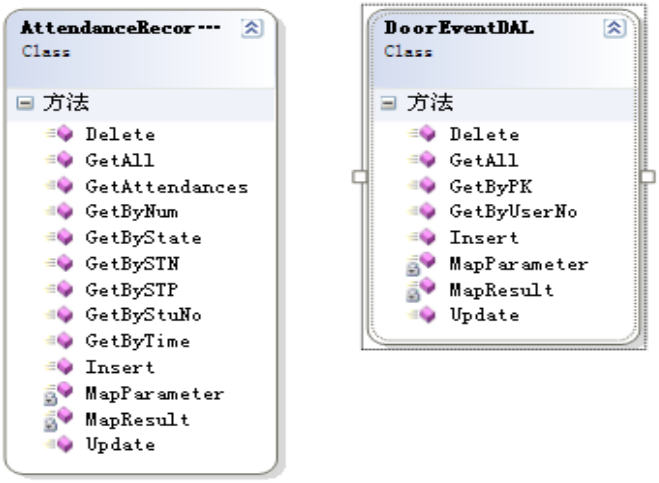
RoleDAL:提供角色对象的增删改查操作。

TeacherDAL:提供教师的增删改查操作。

说名：其中 Delete 、GetAll 、GetByNum 、Insert 、Update 是最基本的增删改查操作。而 GetByProjectNo、GetByRoleNo 等是基于每个实体类的独特的简单数据业务处理操作。MapParameter 函数作用是将各个不同的实体类的属性变成其执行的存储过程的参数。MapResult 函数的作用是将存储过程返回的参数变成程序中实体类的属性。在接下来的分析中不在重复论述。

Recommend
火龙果软件

（2） 考勤管理模块，数据访问类如图 4.12 所示：



4.12 考勤模块数据访问类

AttendanceRecordDAL：进行考勤记录的增删改查操作。

DoorEventDAL:进行门事件记录的增删改查。

（3） 项目组管理模块，数据访问类如图 4.13 所示：

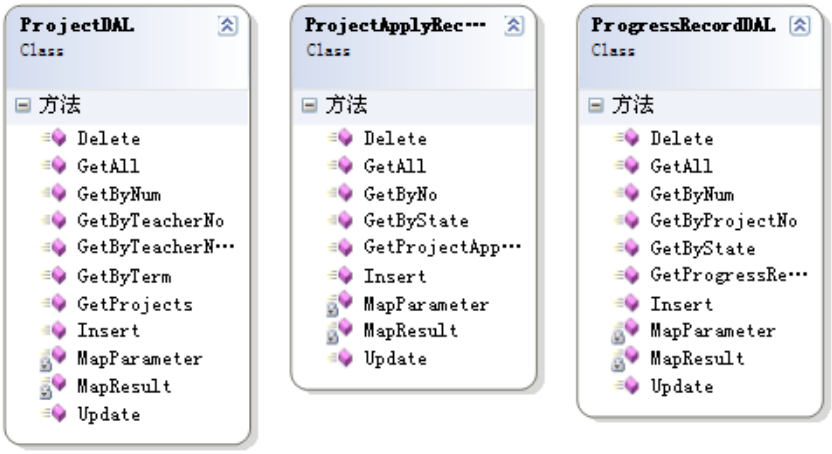


图 4.13 项目组管理数据访问类

ProjectDAL:进行项目的增删改查操作。

ProjectApplyRecordDAL:进行项目申请的增删改查操作。

ProgressRecordDAL:进行项目进度报告的增删改查操作。

Recommend
火龙果软件

（4） 设备管理模块，数据访问类如图 4.14 所示：

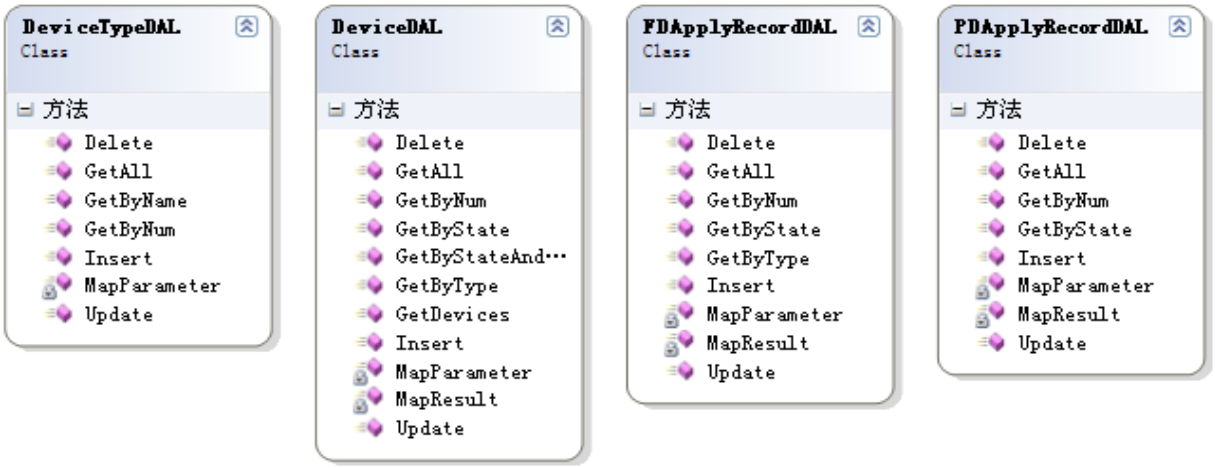


图 4.14 设备模块数据访问类

DeviceDAL:进行设备的增删改查操作。

DeviceTypeDAL:进行设备类别的增删改查操作。

FDApplyRecordDAL:进行固定设备申请的增删改查操作。

PDApplyRecordDAL;进行公共设备申请的增删改查操作。

（5） 请销假管理模块，数据访问类如图 4.15 所示：

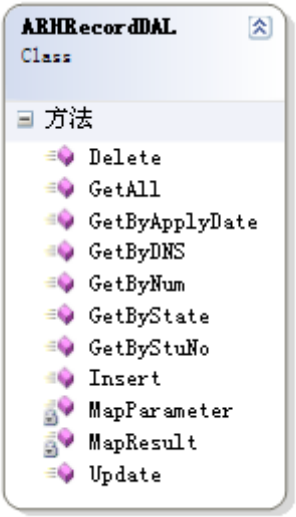


图 4.15 请销假模块数据访问类

ARHRecordDAL:进行请销假记录的增删改查操作。

Recommend
火龙果软件

（6） 值日管理模块，数据访问类如图 4.16 所示：

图 4.16 值日模块数据访问类

DutyRecordDAL:进行值日记录的增删改查操作。

DutyChangeRecordDAL:进行值日变更记录的增删改查操作。

（7） 加班管理模块，数据访问类如图 4.17 所示：

图 4.17 加班管理数据访问类

ExtraWorkRecordDAL:进行加班记录的增删改查操作。

ExtraWorkApplyRecordDAL:进行加班申请记录的增删改查操作。

Recommend
火龙果软件

（8） 账号及权限模块，数据访问类如图一 4.18 所示：

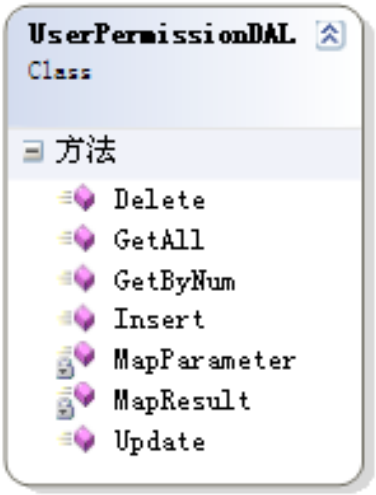


图 4.18 帐号及权限模块数据访问类

UserPermissionDAL:进行账号及权限的增删改查操作。

（9） 系统设置模块，数据访问类如图 4.19 所示

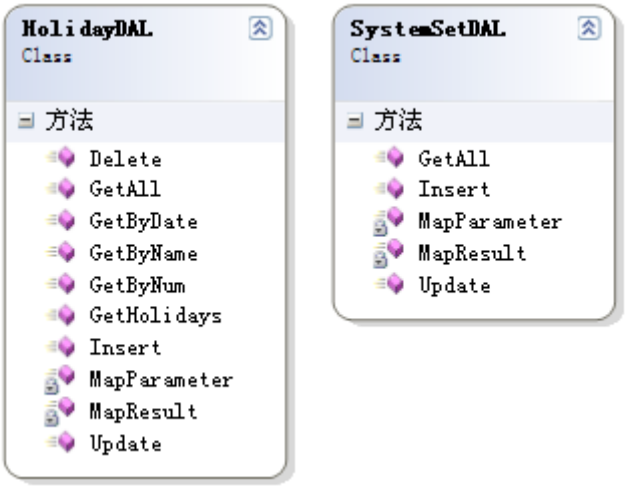


图 4.19 系统设置模块数据访问类

SystemSetDAL:进行业务变量的设置操作。

HolidayDAL:进行假期的增删改查操作。

Recommend
火龙果软件

4.2.3 BLL 模块

BLL 模块即 TBM 系统的业务处理模块。主要负责完成系统的逻辑业务处理。利用数据访问层提供的基本函数接口，进行封装、集成等操作，以实现更加复杂的业务需求。该模块将每一个 DAL 类对应一个逻辑处理层的类，已实现功能的原子性与独立性。

（1） 人员管理模块

人员管理主要涉及员工及教师的基本信息的处理和维护。这些信息包括基本信息，详细信息，工作经历，参与项目，学历信息，以调离职信息等。除了直接将数据访问层的函数封装之外，还增加了以下业务逻辑函数：

bool RegistCheck(ArrayList students){…}　　//批量审核学生注册信息

bool STRecordCheck(ArrayList STRecords){…}　//批量审核学生调离职申请

（2） 考勤管理模块

考勤管理主要涉及人员的自动考勤记录，考勤状态的改变，考勤信息统计，打印考勤记录表等操作。该模块涉及大量的逻辑处理操作，以确保考勤状态判断与更改的正确性。除了直接将数据访问层的基本函数封装外，还添加以下业务逻辑函数：

bool UpdateStudentState(Student student) {…} //更新一个同学的考勤状态

bool UpdateAllState() {…}　　//更新全部员工的考勤状态

bool UpdateAllStateByTask(int task) {…}　　//有参的更新全部员工的考勤状态

（3） 项目组管理模块

项目组管理模块主要涉及项目的添加，修改和删除；项目进度报告的填写，老师的审阅；项目申请的填写，老师的批准等一系列操作。除了直接将数据访问层的基本函数封装外，还添加以下业务逻辑函数：

bool ProjectApplyCheck(ArrayList projectApplyRecords) {…}//项目申请的批量审批

bool ProgressCheck(ArrayList progressRecords) {…}　　//项目进度报告的批量审批

（4） 设备管理模块

设备管理模块主要负责设备的添加，设备的维护，设备状态的改变，设备申请的审阅等操作。除了直接将数据访问层的基本函数封装外，还添加以下业务逻辑函数：

bool DeviceCheck(ArrayList devices) {…} //设备申请的批量审核

（5） 请销假管理模块

请消假管理模块主要负责实训基地内同学们请假申请及审批，销假等操作。该模块的逻辑要牵扯到考勤模块。除了直接将数据访问层的基本函数封装外，还添加以下业务逻辑函数：

bool ARHRecordCheck(ArrayList ARHRecords){…}//请销假的批量审批

Recommend 火龙果软件

bool MakeAttendanceRecord(ARHRecord record){…}//为批准的学生生成假期内的考勤记录

bool RemoveHoliday(ARHRecord record){…} //进行销假操作

（6） 值日管理模块

值日管理模块负责值日生的编排，值日记录的查看，值日责任转让等操作。除了直接将数据访问层的基本函数封装外，还添加以下业务逻辑函数：

bool InitDutyNo() {…}　　//为新注册的同学分配值日号码

bool ChangeDutyNo(){…} //为值日的同学更改值日号码

bool CreateDutyRecord(int personNum) {…}//根据值日人数创建当天的值日记录

（7） 加班管理模块

加班管理模块主要负责加班申请审批，加班记录查看等操作。除了直接将数据访问层的基本函数封装外，还添加以下业务逻辑函数：

bool ExtraWorkRecordCheck() {…} //根据审批结果自动生成加班记录

bool ExtraWorkApplyCheck(ArrayList records){…} //加班申请的批量审核（有参）

bool ExtraWorkApplyCheck() {…} //加班记录的批量审核（无参）

（8） 账号及权限模块

账号及权限模块主要负责系统登陆账号的管理。进行帐号的添加，修改，删除，更改账号权限等一系列操作。该模块没有复杂的函数，但是该模块贯穿于整个项目，系统的大多数页面都要进行访问权限的判断，从而给予不同的操作权限。

（9） 系统设置模块

系统设置模块主要负责系统业务变量的设置，节假日的设置，更改等操作。对于一个以业务处理为主的管理系统而言，灵活的业务处理方式势必会大大增加系统的灵活性。除了直接将数据访问层的基本函数封装外，还添加以下业务逻辑函数：

bool AddAllWeekend(int year) {…} //添加一年的所有周末

bool DeleteYearRecord(int year) {…}//删除一年的所有周末

bool IsWorkDay() {…}　　//判断当天是否是工作日

4.2.3 Web Service 封装

TBM 系统的一大优势就是为基地内的所有分基地提供了一套完整的 Web 服务。我们将大多数的逻辑层接口都对外封装成 WebService 方法，这样我们的底层程序与表示层交换的数据就变成了满足 SOAP 协议与 XML 标准，平台无关的数据流。各个基地可以根据需要开发自己的管理应用程序。

Recommend 火龙果软件

发布 Web 服务时，有一个重要的文件 Service.cs 。为了区分不同模块对外提供的 Web 服务，在 Service.cs 文件中，应用统一的命名前缀来进行区分。这样在表示层调用的时候，就可以使用“服务对象.前缀_函数名（）”的形式访问 Web 服务了。模块前缀缩写如图 4.20 所示：

1.学生WebService 缩写：Stu
2.教师WebService 缩写：Tea
3.调离职WebService 缩写：STR
4.角色WebService 缩写：Rol
5.考勤WebService 缩写：Att
6.门事件WebService 缩写：DE
7.项目WebService 缩写： Pro
8.项目申请WebService 缩写：PA
9.项目进度WebService 缩写：Prog
10.设备基本信息WebService 缩写：Dev
11.设备类别WebService 缩写：DT
12.固定设备申请WebService 缩写：FDA
13.公共设备申请WebService 缩写：PDA
14.请销假WebService 缩写：ARH
15.值日记录WebService 缩写： Dut
16.值日变更WebService 缩写：DC
17.加班记录WebService 缩写：EW
18.加班申请WebService 缩写：EWA
19.账户权限WebService 缩写：UP
20.系统设置WebService 缩写：SS
21.假期WebService 缩写：Hol

图 4.20 TBM 系统 WebService 缩写

为了使 WebService 函数对外传递的数据符合 SOAP 协议，并且是 XML 语言可解析的对象。在定义一个返回类型为自定义类的对象时，函数声明方式必须如下所示：

```
/// <summary>
///  更新一个学生记录
/// </summary>
/// <param name="student">需要更新的学生对象</param>
/// <returns>操作是否成功</returns>
[XmlInclude(typeof(Student))]          ----------→说明按Student类型解析与传递
[WebMethod]                            ----------→说明是WebService方法
public bool Stu_Update(Student student)
{
    return stuBLL.Update(student);
}
```


Recommend 火龙果软件

按照以上的方法，提供的服务越多，就意味着实现表示层表现程序时可越少写逻辑操作。这样，将大部分的逻辑层方法都封装成 WebService 方法，提供上层调用。TBM 系统提供的部分服务如表 4.1 所示：

表 4.1　TBM 系统对外提供的部分服务列表

名称	类型	说明
ARH_Add	bool	增加一条请销假记录
ARH_ARHRecordCheck	Bool	请销假申请记录的批量审批
ARH_MakeAttendanceRecord	Bool	为批准的生成假期内考勤记录
ARH_Remove	Bool	删除一条请销假记录
ARH_RemoveHoliday	Bool	进行销假
Att_UpdateAllState	Bool	更新全部同学考勤状态
Att_UpdateAllStateByTask	Bool	有参的更新全部同学考勤状态
Att_UpdateStudentState	Bool	更新一个同学的状态
DC_GetByState	ArrayList	由状态获得一些值日变更记录
DE_GetByPK	DoorEvent	根据主键获得一条门事件记录
Dev_DeviceCheck	Bool	公共、固定设备的批量审批
Dut_ChangeDutyNo	Bool	为值日后的同学更改值日号码
Dut_CreateDutyRecord	Bool	创建当天的值日记录
Dut_InitDutyNo	Bool	为新注册的同学分配值日号
EWA_ExtraWorkApplyCheck	Bool	有参，加班申请的批量审核
EWA_ExtraWorkApplyCheckAll	Bool	无参的加班申请的批量审核
FDA_Add	Bool	增加一条固定设备申请记录
Hol_AddAllWeekend	Bool	添加一年的所有周末
Hol_DeleteYearRecord	Bool	删除一年的所有假期
Hol_IsWorkDay	Bool	判断当天是否为工作日
PA_GetByNo	ProjectApplyRecord	根据编号获得
PDA_Update	Bool	更新一条申请记录
Pro_Add	Bool	增加一个项目
Prog_ProgressCheck	Bool	进度报告批量审核
SS_Update	Bool	更新一个系统设置对象
STR_STRecordCheck	Bool	调离职记录的批量审批
Stu_GetStudents	ArrayList	根据自定义条件查询
Tea_GetAll	ArrayList	获得所有的教师
UP_Update	Bool	更新一个账号的基本信息

Recommend
火龙果软件

4.2.4 UI 模块

（1） winForm-UI 框架设计，如图 4.21 所示：

① 框架设计思路

整个页面分为两个部分：

菜单栏：置于 Win 界面的上部，链接到程序的所有功能模块。

详细功能展现区：位于 win 界面的下部，负责每个模块及功能的展现窗体，是 winUI 的核心。在该部分区域中，有的是采用窗体的形式展现的，有的是采用多文档的形式展现的，采用哪种方式主要取决于那个模块的页面需求。

图 4.21 WinForm 程序页面框架图

② 操作流程

单击页面上部分的九大模块之一，将展开显示其各模块所包含的内容链接；

点击内容链接，将在页面的主窗体部分显示其对应的窗体；

在页面主体窗体部分，根据需要进行选择或点击等操作；

退出系统时，选择系统管理中的“退出系统”功能，也可以点‘X’退出。

Recommend
火龙果软件

（2） web-UI 框架设计，如图 4.22 所示：

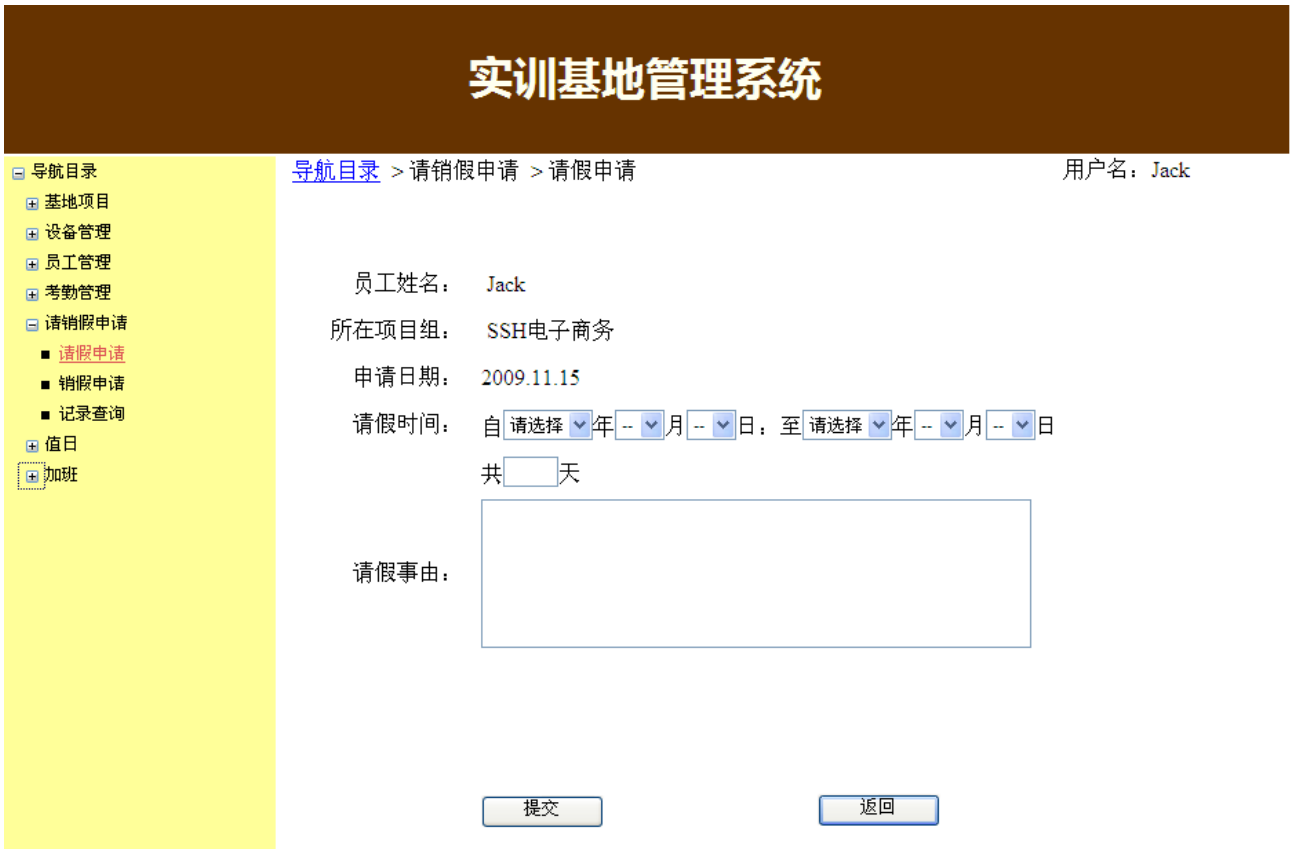
① 框架设计思路

整个页面分为三个部分：

名称：top，页面顶层，用于展示系统的名称、LOGO 等；

名称：left，页面左部，导航栏部分，显示七大模块及各模块的相关子菜单；

名称：main，页面的主要内容，通过导航栏链接显示详细信息，是 WebUI 的核心。



4.22 Web 程序页面框架图

② 操作流程

单击页面 left 部分的七大模块之一，将展开显示其各模块所包含的内容链接；

点击内容链接，将在页面的 main 部分显示其对应的页面；

在页面 main 部分，根据需要进行选择或点击等操作。

Recommend
火龙果软件

4.3 后台服务程序设计

后台服务程序运行在数据库服务器上，主要完成系统的自动处理操作。由于该处理程序不涉及太多与操作者的交互，将此程序设计成以控制台的形式展现，并为其指定一些操作命令，方便服务器管理员的操作。并且，作为服务器管理员，只有登录后才能进行一些操作，以此来增加系统使用的安全性。后台服务程序运行界面如图 4.23 所示：

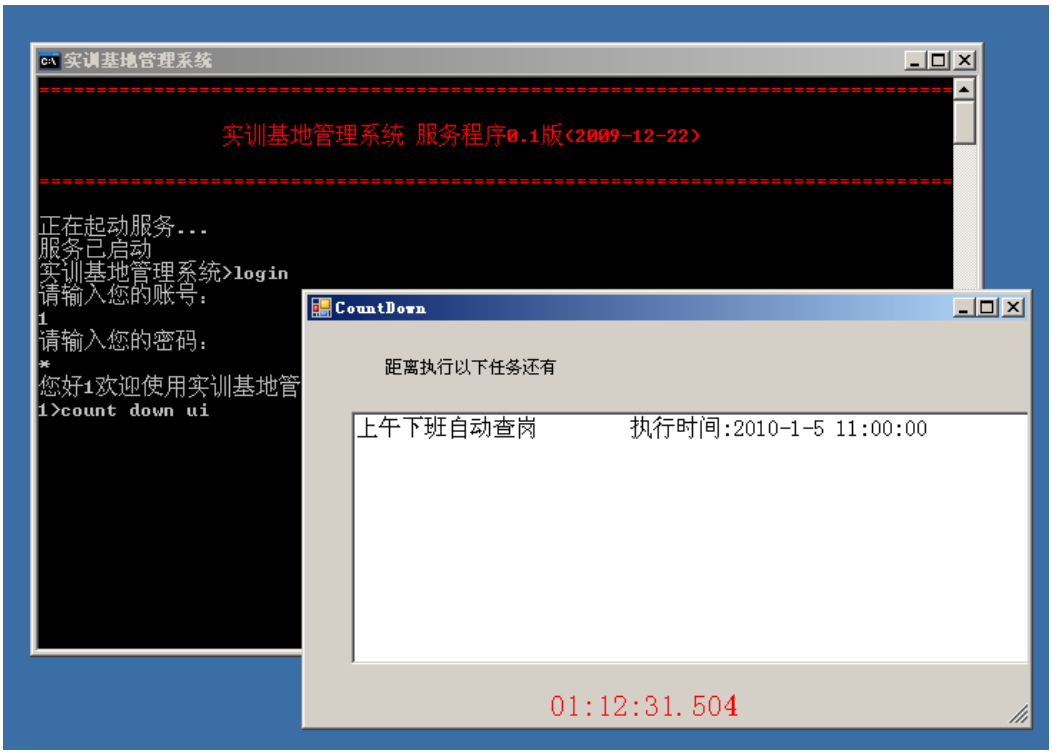


图 4.23 TBM 服务程序运行界面

TBM 系统之所以能够实现半自动的管理，既是因为这个后台服务程序的作用。通过这个程序，系统可以在规定的时间进行自动处理，减少人工操作。它可执行的自动操作有。（1）考勤状态的自动判断与更改（2）当天值日记录的自动生成（3）请假日期结束时的自动销假等。该服务程序主要分三个类：

Program 类：主函数类、完成系统的自动运行事件；

Controler 类：处理通过控制台输入的命令；

Lister 类：处理从 winForm 客户端传过来的命令（客户端暂未实现）；

Recommend
火龙果软件

5 设备子模块与权限子模块实现

在整个项目开发的过程中，我担任的主要任务是架构设计和编码任务。做完了整体的程序架构后，就涉及到具体模块的实现。我主要涉及两个模块的详细实现。

5.1 设备子模块

实训基地中有大量的实训设备，如何对这些设备进行合理的管理，是这个模块设计与实现的核心。该模块需要实现新设备的添加、设备状态更改、设备申请与审批等一系列业务操作。同时，作为设备管理员操作的主要模块，该模块还需满足使用方便性、页面良好性等需求。基于以上的要求及该模块的详细设计文档，设备子模块实现如下：

（1） 实体类

① 设备实体

在数据库中，关于设备的实体表存在 FixDevice 和 PublicDevice 两张表，但是当进行业务处理的时候，发现分别操作两张表没有操作一张表方便。于是请数据库设计的同学将这两张表抽象成了一个 Device 的设备视图，以方便在程序中的操作。视图的具体内容如图 5.1 所示：

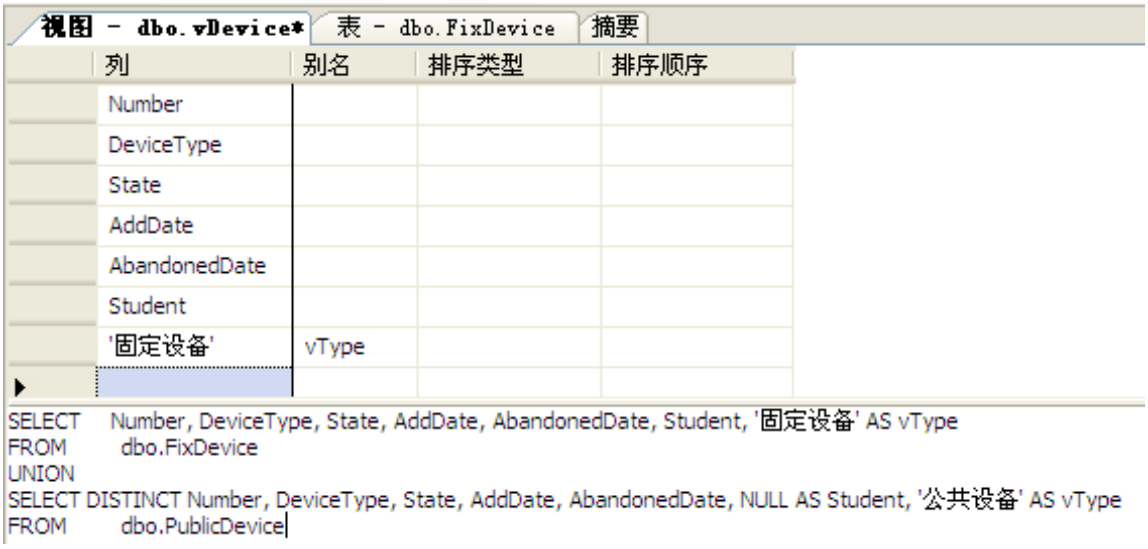


图 5.1 设备视图、

将此视图映射成 TBM 实体类中的 Device 实体类。这样在程序中操作的 Device 对象，经数据库处理时就会自动转化为对两个表的单独操作，增加对数据库实施业务性操作的方便性。

Recommend
火龙果软件

Device 实体类如图 5.2 所示：

名称	类型	修饰符	摘要
属性			
AbandonedDate	DateTime?	public	废弃日期
AddDate	DateTime?	public	添加日期
DeviceType	int	public	设备类别
FdOrPd	int	public	标示是固定设备还是公共设备0-固定；1-公共
Number	string	public	设备号
State	int	public	状态。1-闲置；2-使用(借出)；3-废弃；4-维修
Student	string	public	应用者
<添加属性>			
字段			
abandonedDate	DateTime?	private	
addDate	DateTime?	private	
deviceType	int	private	
fdOrPd	int	private	
number	string	private	
state	int	private	
student	string	private	

图 5.2 设备实体类

② 固定设备申请记录实体，如图 5.3 所示：

名称	类型	修饰符	摘要
属性			
ApplyDate	DateTime?	public	申请日期
ApplyReturnDate	DateTime?	public	申请归还日期
Number	int	public	记录编号
PubilcDevice	string	public	设备编号
Reason	string	public	申请理由
Reply	string	public	反馈意见
ReturnDate	DateTime?	public	结束使用日期
StartDate	DateTime?	public	开始使用日期
State	int	public	申请记录状态。0-未审核；1-审核通过；2-审核未通过；3-记录完结
Student	string	public	申请人
<添加属性>			

图 5.3 固定设备申请记录实体类

③ 公共设备申请记录实体，如图 5.4 所示：

名称	类型	修饰符	摘要
属性			
ApplyDate	DateTime?	public	申请日期
FixDevice	string	public	固定设备编号
Number	int	public	记录编号
Reason	string	public	申请理由
Reply	string	public	反馈意见
State	int	public	申请记录状态
Type	int	public	申请类别
<添加属性>			

图 5.4 公共设备申请记录实体类

Recommend
火龙果软件

（2） 数据访问层

设计设备实体类 Device 的目的是为了以面向对象的思想供上层方法直接操作数据库，实现对象与表的对应关系。这种关系易于理解和操作。在数据访问层中，实现直接操作数据库的方法，以 DeviceDAL 为例，具体函数列表如图 5.5 所示：

名称	类型	修饰符	摘要
方法			
Delete	bool	public	根据设备编号，删除一条记录
GetAll	ArrayList	public	查询所有设备记录
GetByNum	Device	public	根据设备编号查找一条设备记录
GetByState	ArrayList	public	根据状态查询设备记录
GetByStateAndType	ArrayList	public	根据类别与状态联合查询设备记录
GetByType	ArrayList	public	根据类别查询设备记录
GetDevices	ArrayList	public	根据设备实体模板查询
Insert	bool	public	根据Device实体对象插入一条device记录
MapParameter	SqlParameter[]	private	根据device的成员变量，生成与之对应的存储过程参数
MapResult	ArrayList	private	将数据集转化为实体集
Update	bool	public	保持设备号不变，根据设备实体更新一条记录

图 5.5 设备数据访问类

（3） 逻辑层及 WebService 方法

在逻辑层中，将基本的数据层方法封装成逻辑层函数，然后再针对业务的需求，写出此模块独特的逻辑操作函数，如固定设备和公共设备申请批量审核等函数。以 DeviceBLL 为例，具体函数列表如图 5.6 所示：

名称	类型	修饰符	摘要
方法			
Add	bool	public	添加一个设备
DeviceCheck	bool	public	公共设备和固定设备的批量审核
GetAll	ArrayList	public	查询所有设备
GetByNum	Device	public	根据设备编号查询设备
GetByState	ArrayList	public	根据设备状态查询设备
GetByStateAndType	ArrayList	public	根据状态与设备类型联合查询设备
GetByType	ArrayList	public	根据设备类型查询设备
GetDevices	ArrayList	public	根据参数查询设备
Remove	bool	public	删除一个设备
Update	bool	public	更新一个设备

图 5.6 设备逻辑访问类

Recommend
火龙果软件

然后，将这些方法封装成 WebService 的方法，使其变为对外提供的服务。封装时要注意自定义对象和 XML 实体对象间的合理显示转换。例如：上面的 Add 方法，在封装成 WebService 方法后变成：Dev_Add（）。

（4） 表示层

一个软件的可用程度和用户体验性决定了它是不是容易被用户所接受。作为实训基地管理系统，C/S 端操作的主要是有职务的老师和同学，页面不求十分美观，但是要求功能和可用程度都达到最大的优化。该模块的操作页面主要分为固定设备管理（固定设备审批、固定设备分配）、公共设备管理和设备列表。基于这些要求。对于设备页面的处理与详细编码获得了以下的操作界面：

① 固定设备分配，运行界面如图 5.7 所示：

图 5.7 固定设备分配页面

页面说明：此页面分三部分。左侧是基地内本期所有的同学列表。可供选择，并分配其设备。双击一个同学的姓名，可以显示该同学目前拥有的设备，并且可以删除他拥有的设备，如 5.7 右侧图所示。右上侧是设备类别选择区，可选择需要分配的固定设别类别。右下侧是该类别下处于闲置状态的空闲设备，选中想分配的设备，点击“确认”按钮进行非配。在 5.7 右侧图中，左侧是该同学的基本信息，右侧是其拥有的设备，可以删除对某个设备的使用权限，也可以全部清空其使用的设备。被删除或清空的设备将被置于闲置状态，可供其他人使用。

Recommend
火龙果软件

② 固定设备维修、变更审批，如图 5.8 所示：

图 5.8 固定设备审批查看

页面说明：页面分为查看以审批记录和未审批记录两部分。未审批的可以进行批量的审批操作。并且可以填写拒绝理由。

③ 公共设备管理，如图 5.9 所示：

图 5.9 公共设备管理

Recommend
火龙果软件

页面说明：页面分为已归还、为归还、未审批和已通过四部分。其中已通过包括（借出和未借出的）。各个窗口根据操作需要按某一个时间排序，比如在未归还选显卡中，我们更关注哪个借出设备的申请结束使用日期快到了，如图 5.9 所示，记录是按申结日期降序排序的。并且对过期未归还的设备可以进行过期提醒。

④ 设备列表，如图 5.10 所示：

图 5.10 设备列表及信息修改

页面说明：在设备列表页面中，可以按不同条件搜索，并且可以添加一个新的设备。双击每一行会查看该设备的详细信息并进行修改。在设备详细信息页面中，可以根据设备的不同状态给予不同的提示。

（5） 部分代码

在进行数据显示的时候，需要对数据进行分页，TBM 系统统一应用分页控件 bindingNavigator 进行操作。下面的代码详细说明了是如何利用这个空间将 DataGridView 控件显示数据分页的：

```
//分页函数的全局参数
private int pageNum=0;        //总页数
```


Recommend
火龙果软件

```
private int rowPerPage=5;      //每页5行
private int cutPage=1;         //当前第几页
private int TotalRows=0;       //总共的行数
private int rowCount=0;        //当前页数的行数
ArrayList orders =new ArrayList();//需要分页的数据集

/// 分页函数，绑定未审核的
/// <param name="ts">输入的需要分页的数据集</param>
private void PageWei(ArrayList ts)
{
    ArrayList NewObjList = new ArrayList();
    TotalRows = ts.Count;
    if (TotalRows % rowPerPage == 0)
    {
        pageNum = TotalRows / rowPerPage;
    }
    else
    {
        pageNum = (TotalRows / rowPerPage) + 1;
    }
    TotalCountWei.Text = "/ " + Convert.ToString(pageNum);
    for (int j = 0; j < TotalRows; j++)
    {
        if (j > (curPage - 1) * rowPerPage - 1 && j < curPage * rowPerPage)
        {
            NewObjList.Add(ts[j]);
        }
    }
    rowCount = NewObjList.Count;//获得分页后数组里面值的大小
    PositionWei.Text = Convert.ToString(curPage);
    if (curPage == 1 && curPage != pageNum)
    {
        MoveNextWei.Enabled = true;
        MoveLastWei.Enabled = true;
        MoveFirstWei.Enabled = false;
        MovePreviousWei.Enabled = false;
    }
    if (curPage == pageNum && curPage != 1)
    {
        MoveFirstWei.Enabled = true;
        MovePreviousWei.Enabled = true;
        MoveNextWei.Enabled = false;
        MoveLastWei.Enabled = false;
    }
```


Recommend
火龙果软件

```
        if (curPage != 1 && curPage != pageNum)
        {
            MoveNextWei.Enabled = true;
            MoveLastWei.Enabled = true;
            MoveFirstWei.Enabled = true;
            MovePreviousWei.Enabled = true;
        }
        if ((curPage == 1 && curPage == pageNum) || (curPage == 0))
        {
            MoveNextWei.Enabled = false;
            MoveLastWei.Enabled = false;
            MoveFirstWei.Enabled = false;
            MovePreviousWei.Enabled = false;
        }
        dgvWei.DataSource = NewObjList;     //绑定分页后的数据源
    }

    private void MovePreviousWei_Click(object sender, EventArgs e)
    {
        curPage = curPage - 1;
        PageWei(orders);
    }
    private void MoveFirstWei_Click(object sender, EventArgs e)
    {
        curPage = 1;
        PageWei(orders);
    }
    private void MoveNextWei_Click(object sender, EventArgs e)
    {
        curPage = curPage +1;
        PageWei(orders);
    }
    private void MoveLastWei_Click(object sender, EventArgs e)
    {
        curPage = pageNum;
        PageWei(orders);
    }
```

5.2 账户及权限子模块

一个完整的系统会有各种角色的使用者，就 TBM 系统而言、有普通员工、经理助理、HR、设备部长和老师等不同的使用者，如何合理的安排与分配这些人对系统的使用权限是这个模块要处理的。本模块实现了对账号的添加、删除、更改。并且可以为账

Recommend 火龙果软件

号设置权限，使其按照自身角色的不同拥有对于系统的不同操作权限。使系统的应用性更加广泛。

（1） 实体类，该模块涉及一个 UserPermission 实体，如图 5.11 所示：

名称	类型	修饰符	摘要
属性			
ID	string	public	账号
Password	string	public	密码
Permission1	int	public	权限
Permission2	int	public	
<添加属性>			
字段			
id	string	private	
password	string	private	
permission1	int	private	
permission2	int	private	

图 5.11 账号权限实体类

说明：这里的 ID 指系统的登录账号，和 Student 表中与 Teacher 表中的记录没有本质上的联系。即一个在 Student 表中存在的同学记录对应的学号，在账号表中不一定在。Permission1 和 Permission2 是系统的权限判断值，目前系统只适用了 Permission1 进行判断，Permission2 为以后扩充系统做准备。

（2） 数据访问层，该模块的数据访问类如图 5.12 所示：

名称	类型	修饰符	摘要
方法			
Delete	bool	public	根据编号删除一个账户
GetAll	ArrayList	public	查询所有账户信息
GetByNum	UserPermission	public	根据ID查询账户信息
Insert	bool	public	插入一条新账户信息
MapParameter	SqlParameter[]	private	
MapResult	ArrayList	private	
Update	bool	public	保持ID不变更新一个账户

图 5.12 账号及权限数据访问类

该模块在逻辑处理上没有复杂的函数，只是简单的增删改查功能即可完成业务的需求，但是该模块的复杂之处在于权限的设置和权限的判断，因为这需要贯穿整个系统达到权限控制的目的。我们将在下面论述。

Recommend 火龙果软件

（3） 逻辑层及 WebService 方法

逻辑层将基本的数据访问层方法封装成逻辑层方法。这里主要介绍权限的控制。作为管理端的应用程序，多数的按钮和页面都要进行权限控制。TBM 系统目前定义了近 30 个权限，实质上是用二进制的不同位代表不同的权限，表示上用十进制数（将十进制数转化为二进制数，哪一位为 1 即说明具有该位的权限）。在程序中不断运用‘与’‘或’运算来判断访问者的读写权限，进行业务操作。具体的权限类代码如图 5.13 所示：

```
public class UPConstList
{
    static public int GROUP_READ = 1;              // 组内人信息读权限
    static public int GROUP_WRITE = 2;             // 组内人信息写权限
    static public int ALL_READ = 4;                // 所有人信息读权限
    static public int ALL_WRITE = 8;               // 所有人信息写权限
    static public int GROUPSTR_READ = 16;          // 组内调离职申请读权限
    static public int GROUPSTR_WRITE = 32;         // 组内调离职申请写权限
    static public int STRALL_READ = 64;        // 对所有人掉离职申请读权限
    static public int STRALL_WRITE = 128;      // 对所有人调离职申请写权限
    static public int ALLATT_READ = 256;           // 考勤记录读权限
    static public int ALLATT_WRITE = 512;          // 考勤记录写权限
    static public int PROJECT_READ = 1024;         // 项目读权限
    static public int PROJECT_WRITE = 2048;        // 项目写权限
    static public int PROJECTAPP_READ = 4096;      // 项目申请记录读权限
    static public int PROJECTAPP_WRITE = 8192;     // 项目申请记录写权限
    static public int PROGRESS_READ = 16384;       // 进度报告读权限
    static public int PROGRESS_WRITE = 32768;      // 进度报告写权限
    static public int DEVICEManangement = 65536;   // 设备管理执行权限
    static public int ALLOWHOLIDAY = 131072;       // 请销假审批权限
    static public int ARHMANANGEMENT = 262144;     // 请销假记录查看权限
    static public int DUTY_READ = 524288;          // 值日单读权限
    static public int DUTY_WRITE = 1048576;        // 值日单写权限
    static public int EXTRAWORKRECORD = 2097152;   // 加班记录查看权限
    static public int EXTRAWORKAPPLY = 4194304;    // 加班记录审核权限
    static public int BUSINESS_Set = 8388608;      // 业务变量设置权限
    static public int ADMINISTRATOR = 1073741824;// 超管权限1073741824
}
```

图5.13 系统权限类UPConstList

比如对于一名 HR 来说，他就需要同时具备 ALL_READ、ALL_WRITE、STRALL_READ、STRALL_WRITE、ALLOWHOLIDAY、ARHMANANGEMENT、ALLATT_READ、ALLATT_WRITE 的权限。他的权限十进制数就为上述所有权限的和。

Recommend 火龙果软件

（4） 表示层

在本模块的表示层中，主要要完成账户的添加、修改和删除的操作。并且要清晰易懂。本模块的主页面如图 5.14 所示：

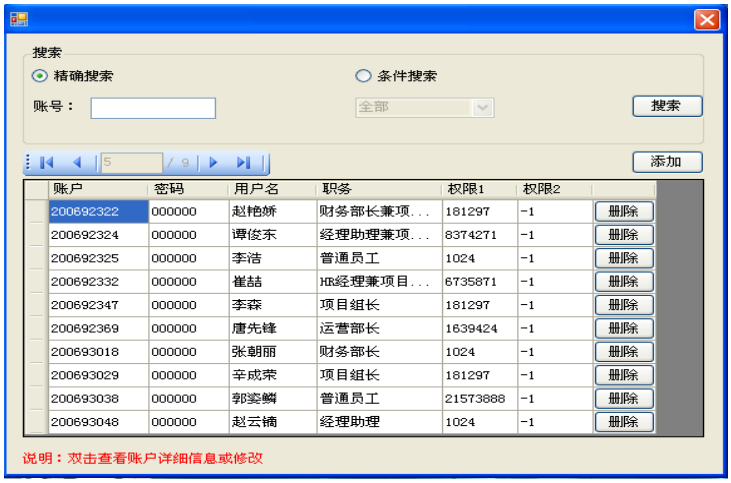


图 5.14 账户及权限主页面

页面说明：用户可以根据需求进行精确搜索或者条件搜索，双击可显示一个账户的详细信息。用户在修改账户权限时，如图 5.15 所示，可以更改他的密码和操作权限。

图 5.15 账户详细信息及修改

页面说明：其中的“读权限”代表让该账户具有访问某个页面查看信息的权限。“写权限”代表用户可以对该页面背后对应的数据表进行修改操作，比如增删改等。“管理、设置权限”代表用户既具有读又具有写权限。

Recommend
火龙果软件

（5） 部分代码

在登录成功进入首页的时候，系统需要进行权限的判断，以此来控制该用户是否对某一个模块有操作权限。以下面的代码为例,控制了对人员管理模块的访问操作：

```
#region 将所有主功能标签均初始化为 false
MenuItemPerson.Enabled = false;
MenuItemAtt.Enabled = false;
MenuItemDev.Enabled = false;
MenuItemPro.Enabled = false;
MenuItemARH.Enabled = false;
MenuItemDuty.Enabled = false;
MenuItemEWork.Enabled = false;
MenuItemUP.Enabled = false;
//MenuItemSM.Enabled = false;//模块内有所有操作者都拥有的操作，不做初始限制
#endregion
#region 判断登陆用户对人员模块的可见程度
if ((PowerPermission1 & UPConstList.ALL_READ) !=0 || (PowerPermission1 &
UPConstList.STRALL_READ) !=0)
{
    MenuItemPerson.Enabled = true;   //人员模块主标签可用
    InMItemRegistCheck.Enabled = false;//初始其他标签为不可用
    InMItemSepreate.Enabled = false;
    InMItemTrans.Enabled = false;
    InMItemInfor.Enabled = false;
    InMItemTeaInfor.Enabled = false;
    //可进行注册信息、人员信息及教师信息查看
    if ((PowerPermission1 & UPConstList.ALL_READ) != 0)
    {
        InMItemRegistCheck.Enabled = true;
        InMItemInfor.Enabled = true;
        MyWebService.Teacher teaTemp = ser.Tea_GetByNo(UserNumber);
        if (teaTemp != null)//如果该人是教师，他可以进行教师信息查看
        {
            InMItemTeaInfor.Enabled = true;
        }
    }
    //可以进行调离职信息查看
    if ((PowerPermission1 & UPConstList.STRALL_READ) != 0)
    {
        InMItemSepreate.Enabled = true;
        InMItemTrans.Enabled = true;
    }
}
#endregion
```


Recommend 火龙果软件

结　　论

在大规模的软件产品开发时，能合理的应用软件架构技术，不仅能够帮助系统的设计和开发，还会直接影响代码的复用性和强壮性，这往往决定了软件开发的成败。而且跨平台的、跨地域的复杂信息交流与互动已经成为明显的趋势，如何最大限度的减少软件开发的成本，加大程序之间或功能之间的复用性，显然已经成为了一个十分热点的研究内容。面向服务的编程思想（SOA）的产生使得不同平台、地域间的计算机可以方便的交流数据，共用服务，减少单独维护己方代码和业务的负担。本文上面所论述的 TBM 系统正式一个基于三层架构和 SOA 技术的管理系统，它实现了基地内高效、便捷、准确的半自动化管理。

本文从软件设计的角度出发，分 5 章综合论述了 TBM 系统的架构设计和两个子模块的实现。首先，从技术背景上分析。基于实训基地目前的业务需求和开发需求，我们选择了分层架构和 SOA 作为基本的理论依据来满足开发过程的独立性、开发后服务的共享性的需求。然后，综合分析了 TBM 系统的业务需求，总体介绍了 TBM 系统的运行环境和需要完成的任务。最后，详细分析了 TBM 系统的架构设计和应用设计，该模块是设计的核心，它是整个系统的骨架。文章的最后部分，还详细介绍了两个子模块的实现方式，使设计转化为实质可用的软件。

在项目组成员的共同努力下，TBM 系统管理人员使用的 WinForm 应用程序基本完工，可作为第一版本交付使用。经过模拟输入数据的测试，该系统能够真正的解决人工操作带来的一些麻烦，可以大大的提高工作效率。并且，TBM 系统和现有的门禁系统同步，用门禁系统提供的考勤刷卡时间进行逻辑处理。这样，基地对于考勤的管理就实现了半自动化。而且，我们在设计考勤模块的逻辑处理函数时，考虑各种情况的复杂组合，真正的做到考勤的记录精准、正确。对于其他模块，比如人员管理、值日管理、设备管理等，TBM 系统也完全实现了我们的业务需求，今后对于这些方面的操作，只需要进行网上申请和审批，而很少需要人与人之际的交流，大大提高了工作效率。由于使用了 Web 服务，TBM 系统可对外提供满足大多数需求的服务，供 3 个基地调用，直接应用到各自开发的程序中去。

当然，由于时间紧迫和项目组成员变动较大。TBM 系统还存在很多未完成的部分需要待下期同学继续完成，比如 web 程序端、与门禁系统数据库的自动数据交流、薪资模块、活动管理模块、站内交流模块等。相信在大家的不断努力与修改下，TBM 系统一定能够真正成为软件学院实训基地管理时的必备软件，使基地的管理发生质的变化，为同学们提供一个更加良好的实训环境。

Recommend
火龙果软件

参 考 文 献

[1] 王听,金淳兆.软件体系结构的研究与发展现状[J].计算机科学,1998,25 (3):16-19.
[2] 王晓霞,宋丽,王翠茹.软件体系结构的性能评价研究[J].计算机工程与应用,2003,32(3):22-22.
[3] Dikel. Software Architecture: Organizational Principles and Patterns[M].New Jersey:Prentice-Hall,2001.
[4] Eric Freeman 等.Head First 设计模式（中文版）[M].中国电力出版社,2007.
[5] 甄镭.NET 与设计模式[M].电子工业出版社,2005.
[6] 张逸.软件设计精要与模式[M].电子工业出版社,2007.
[7] 章利.基于.NET 的多层分布式企业应用架构的设计与实现[D].哈尔滨:哈尔滨理工大学,2007.
[8] 刘卫忠,徐重阳,蔷薇.多层客户机服务器结构分析[J].网络世界,2000.10:74-75.
[9] 陈志刚.三层客户/服务计算模型的优化研究[J].系统工程,1998.9:83-84.
[10] 徐舒亮,叶少珍.基于 Web 服务的 EAI 解决方案讨论[N].福州大学学报,2004,12.
[11] 顾宁,刘家茂,柴晓路. Web Services 原理与研发实践[M].机械工业出版社,2006.
[12] 宋振蔚. 基于面向服务架构(SOA)的企业销售系统设计与实现[D].中山大学,2006,4.
[13] 叶钰,应时等. 面向服务体系结构及其系统构建研究[J].计算机应用研究,2005,2：32-34.
[14] 李军怀,周全明,耿国华,张景. XML 在异构数据集成中的应用研究[J].计算机应用,2002,9(9):10.
[15] 孙华林,赵正文. 基于 Web Services 的面向服务架构(SOA)的探索与研究[J].信息技术,2007.

Recommend
火龙果软件

致　　谢

在实训基地进行毕业设计是机缘巧合，但我非常珍惜在这里的 3 个月，这段时间让我真正的学到了一个软件是如何从无到有的诞生过程，谢谢基地的老师和院领导们。

在 TBM 系统的开发过程中，我们面临了不少困难，但在大家的共同努力下。我们终于克服了一个又一个的难题，按时将一个合格的软件交付给了自己的努力。记得开始的 1 各月，我们在林乐春老师和刘文杰老师的带领下，从大型软件开发的基础理论学起，一步一步的了解了什么是 PMI，什么是 CMMI，这为接下来的工作奠定了坚实的理论基础。第二个月，我们开始了项目的需求和设计。这个阶段是我们项目组最困难的时候，人员变动的巨大，技术储备的不足，曾经使得大家一筹莫展。但我们没有放弃，靠着大家的努力和老师的细心指导，终于挺过了项目最艰难的一段时间。这个阶段过后的最后一个月，可以说我们的项目逐渐步入了正轨，大家有条不紊的开发着，过着忙碌的实训生活，老师经常过来检查项目进度、询问大家项目开发的困惑并给与详细的指导。功夫不负有心人，项目终于在 1 月 5 日结束了第一期的开发并测试通过，大家看着自己的成果不知道有多么高兴。

总结这三个月的生活，我要感谢的人很多。作为组长，我感谢我的组员能积极配合我的工作，和我共同努力按时完成任务。同时，我感谢林乐春老师和刘文杰老师对项目和论文的指导，项目的成功交付和他们正确的指导是分不开的。还有，感谢学院和学院领导为我们提供了如此优越的实训条件，让我们能全身心的投入到实训生活中去。在这 3 个月中，我学会了如何将一个抽象的、模糊的需求转化为代码，转化为软件，我想这是我最大的收获。以后如果再有类似规模的软件开发，我想我不会惧怕，因为从需求到设计，再从设计到实现，我都一步一步的跟下来了，我了解软件的开发过程，了解了软件是如何实现的，谢谢实训基地给我的经验。

最后，再次由衷的感谢实训基地的指导教师和院领导们。谢谢你们为我提供的良好的实训环境，提高我的素质和能力。将来我一定会再接再厉，用自己的实际行动来回报母校的培育之恩。

Recommend
火龙果软件