**天气查询、日历、备忘录及学习监督系统**

**系统设计说明书**

**软件名称：天气查询、日历、备忘录及学习监督系统**

**开发团队：西安交通大学—**

**指导教师：**

|  |  |
| --- | --- |
| **DOC ID:** |  |
| **VERSION:** |  |
| **DATE:** | 2020-04-19 |

目 录

[1引言 2](#_Toc403212451)

[1.1编写目的 2](#_Toc403212452)

[1.2背景 2](#_Toc403212453)

[1.2.1 Web Service 2](#_Toc403212454)

[1.2.2 WCF服务的概况 3](#_Toc403212455)

[1.2.3 Entity Framework的概况 3](#_Toc403212456)

[1.2.4 Android的概况 4](#_Toc403212457)

[2 软件总体设计 7](#_Toc403212458)

[2.1软件目标功能描述 7](#_Toc403212459)

[2.2软件架构 7](#_Toc403212460)

[2.3软件功能模块 9](#_Toc403212461)

[3 开发环境概述 10](#_Toc403212462)

[3.1 eclipse简介 10](#_Toc403212463)

[Eclipse的主要组成 11](#_Toc403212464)

[Eclipse SDK 11](#_Toc403212465)

[3.2 eclipse下载与配置 11](#_Toc403212466)

[3.3 WCF层次搭建 12](#_Toc403212467)

[4 数据库设计 15](#_Toc403212468)

[4.1 MySQL简介 15](#_Toc403212469)

[4.2 数据库需求设计 16](#_Toc403212470)

# 

# 1引言

## 1.1编写目的

编写天气查询、日历、备忘录与学习监督系统的系统设计说明书的目的，是为了软件需求方（学生）和开发方能够更好的明确软件预期达到的目的。经过系统分析对使用的需求进行的详细分析后，编写了本篇说明，明确了需求。使开发人员能够更好的投入到软件开发当中去。

本报告的预期读者为用户、业务或需求分析人员、测试人员、用户文档编写人员、项目开发人员。

## 1.2背景

### 1.2.1 Web Service

Web Services是由企业发布的完成其特定商务需求的在线应用服务,其他公司或应用软件能够通过Internet来访问并使用这项在线服务。

它是一种构建应用程序的普遍模型,可以在任何支持网络模型的操作系统中实施运行;是一种新的web。

应用程序分支，是自包含、自描述、模块化的应用，可以发布、定位、通过web调用。WebService是一个应用组件,它逻辑性的为其他应用程序提供数据与服务.各应用程序通过网络和规定的一些标准数据格式（Http，XML，Soap）来访问WebService,通过WebService内部执行得到所需结果.Web Service可以执行从简单的请求到复杂商务处理的任何功能。一旦部署以后，其他WebService应用程序可以发现并调用它部署的服务。

### 1.2.2 WCF服务的概况

Windows通信基础（Windows Communication Foundation，WCF）是基于Windows平台下开发和部署服务的软件开发包（Software Development Kit，SDK）。WCF为服务提供了运行时环境（Runtime Environment），使得开发者能够将CLR类型公开为服务，又能够以CLR类型的方式使用服务。理论上讲，创建服务并不一定需要WCF，但实际上，使用WCF却可以使得创建服务的任务事半功倍。 WCF是微软对一系列产业标准定义的实现，包括服务交互、类型转换、封送（Marshaling）以及各种协议的管理。正因为如此，WCF才能够提供服务之间的互操作性。

WCF还为开发者提供了大多数应用程序都需要的基础功能模块，提高了开发者的效率。WCF的第一个版本为服务开发提供了许多有用的功能，包括托管（Hosting）、服务实例管理（Service Instance Management）、异步调用、可靠性、事务管理、离线队列调用（Disconnected Queued Call）以及安全性。同时，WCF还提供了设计优雅的可扩展模型，使开发人员能够丰富它的基础功能。事实上，WCF自身的实现正是利用了这样一种可扩展模型。W C F的大部分功能都包含在一个单独的程序集System.ServiceModel.dll中，命名空间为System.ServiceModel。WCF是.NET 3.0的一部分，同时需要.NET 2.0的支持，因此它只能运行在支持它的操作系统上。目前，这些操作系统包括Windows Vista（客户端和服务器）、Windows XP SP2和Windows Server 2003 SP1以及更新的版本。

服务:服务（Services）是公开的一组功能的集合。服务可以是本地的，也可以是远程的，可以由多个参与方使用任意技术进行开发。服务与版本无关，甚至可以在不同的时区同时执行。服务内部包含了诸如语言、技术、平台、版本与框架等诸多概念，而服务之间的交互，则只允许指定的通信模式。服务的客户端只是使用服务功能的一方。理论上讲，客户端可以是任意的Windows窗体类、ASP.NET页面或其他服务。客户端与服务通过消息的发送与接收进行交互。消息可以直接在客户端与服务之间进行传递，也可以通过中间方进行传递。WCF中的所有消息均为SOAP消息。注意WCF消息与传输协议无关，这与Web服务不同。因此，WCF服务可以在不同的协议之间传输，而不仅限于HTTP。WCF客户端可以与非WCF服务完成互操作，而WCF服务也可以与非WCF客户端交互。不过，如果需要同时开发客户端与服务，则创建的应用程序两端都要求支持WCF，这样才能利用WCF的特定优势。因为服务的创建对于外界而言是不透明的，所以WCF服务通常通过公开元数据（Metadata）的方式描述可用的功能以及服务可能采用的通信方式。元数据的发布可以预先定义，它与具体的技术无关（Technology-Neutral），例如采用基于HTTP-GET方式的WSDL，或者符合元数据交换的行业标准。一个非WCF客户端可以将元数据作为本地类型导入到本地环境中。相似的，WCF客户端也可以导入非WCF服务的元数据，然后以本地CLR类与接口的方式进行调用。

### 1.2.3 Entity Framework的概况

在.Net Framework SP1 微软包含一个实体框架（Entity Framework），此框架可以理

解成微软的一个 ORM 产品。用于支持开发人员通过对概念性应用程序模型编程（而不是

直接对关系存储架构编程）来创建数据访问应用程序。目标是降低面向数据的应用程序所

需的代码量并减轻维护工作。

Entity Framework 应用程序有以下优点：

应用程序可以通过更加以应用程序为中心的概念性模型（包括具有继承性、复杂成员

和关系的类型）来工作。

应用程序不再对特定的数据引擎或存储架构具有硬编码依赖性。可以在不更改应用程

序代码的情况下更改概念性模型与特定于存储的架构之间的映射。开发人员可以使用可映

射到各种存储架构（可能在不同的数据库管理系统中实现）的一致的应用程序对象模型。

多个概念性模型可以映射到同一个存储架构。 语言集成查询支持可为查询提供针对概念

性模型的编译时语法验证。

实体框架 Entity Framework 是 ADO.NET 中的一组支持开发面向数据的软件应用程

序的技术。在 EF 中的实体数据模型（EDM）由以下三种模型和具有相应文件扩展名的映

射文件进行定义。

概念架构定义语言文件 (.csdl) -- 定义概念模型。

存储架构定义语言文件 (.ssdl) -- 定义存储模型（又称逻辑模型）。

映射规范语言文件 (.msl) -- 定义存储模型与概念模型之间的映射。

实体框架使用这些基于 XML 的模型和映射文件将对概念模型中的实体和关系的创

建、读取、更新和删除操作转换为数据源中的等效操作。EDM 甚至支持将概念模型中的实

体映射到数据源中的存储过程。它提供以下方式用于查询 EDM 并返回对象：

LINQ to Entities -- 提供语言集成查询 (LINQ) 支持用于查询在概念模型中定义的

实体类型。

Entity SQL -- 与存储无关的 SQL 方言，直接使用概念模型中的实体并支持诸如继

承和关系等 EDM 功能。

查询生成器方法 --可以使用 LINQ 风格的查询方法构造 Entity SQL 查询。

### 1.2.4 Android的概况

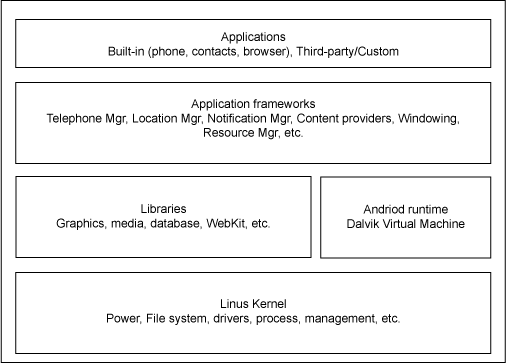
Android 有丰富的功能，因此很容易与桌面操作系统混淆。Android 是一个分层的环境，构建在 Linux 内核的基础上，它包括丰富的功能。UI 子系统包括：

* 窗口
* 视图
* 用于显示一些常见组件（例如编辑框、列表和下拉列表）的小部件

Android 包括一个构建在 WebKit 基础上的可嵌入浏览器，iPhone 的 Mobile Safari 浏览器同样也是以 WebKit 为基础。

Android 提供多种连接选项，包括 WiFi、蓝牙和通过蜂窝（cellular）连接的无线数据传输（例如 GPRS、EDGE 和 3G）。Android 应用程序中一项流行的技术是链接到 Google 地图，以便在应用程序中显示地址。Android 软件栈还提供对基于位置的服务（例如 GPS）和加速计的支持，不过并不是所有的 Android 设备都配备了必需的硬件。另外还有摄像支持。

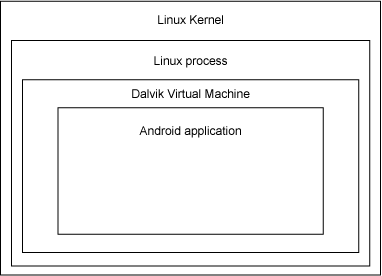
过去，移动应用程序努力向桌面应用程序看齐的两个领域分别是图形/媒体和数据存储方法。Android 通过提供对 2D 和 3D 图形的内置支持，包括 OpenGL 库，解决了图形方面的挑战。由于 Android 平台包括流行的开源 SQLite 数据库，因此缓解了数据存储的负担。图 1 显示一个简化的 Android 软件层次结构。

**图 1. Android 软件层次结构**  




**应用程序架构**

如前所述，Android 运行在 Linux 内核上。Android 应用程序是用 Java 编程语言编写的，它们在一个虚拟机（VM）中运行。需要注意的是，这个 VM 并非您想象中的 JVM，而是 Dalvik Virtual Machine，这是一种开源技术。每个 Android 应用程序都在 Dalvik VM 的一个实例中运行，这个实例驻留在一个由 Linux 内核管理的进程中，如下图所示。

**图 2. Dalvik VM**

Android 应用程序由一个或多个组件组成：

**活动** 具有可视 UI 的应用程序是用活动实现的。当用户从主屏幕或应用程序

动器选择一个应用程序时，就会开始一个动作。

**服务** 服务应该用于任何需要持续较长时间的应用程序，例如网络监视器或更新检查应用程序。

**内容提供程序**

可以将内容提供程序看作数据库服务器。内容提供程序的任务是管理对持久数据的访问，例如 SQLite 数据库。如果应用程序非常简单，那么可能不需要创建内容提供程序。如果要构建一个较大的应用程序，或者构建需要为多个活动或应用程序提供数据的应用程序，那么可以使用内容提供程序实现数据访问。

**广播接收器**

Android 应用程序可用于处理一个数据元素，或者对一个事件（例如接收文本消息）做出响应。

Android 应用程序是连同一个 AndroidManifest.xml 文件一起部署到设备的。AndroidManifest.xml 包含必要的配置信息，以便将它适当地安装到设备。它包括必需的类名和应用程序能够处理的事件类型，以及运行应用程序所需的许可。例如，如果应用程序需要访问网络 — 例如为了下载一个文件 — 那么 manifest 文件中必须显式地列出该许可。很多应用程序可能启用了这个特定的许可。这种声明式安全性有助于减少恶意应用程序损害设备的可能性。

# 2 软件总体设计

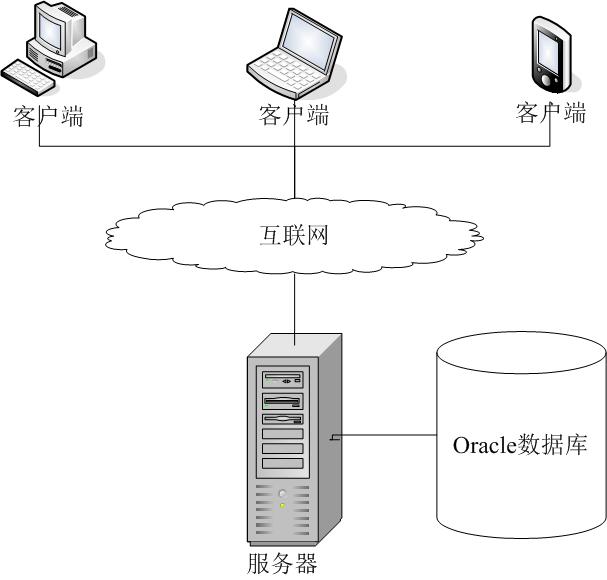
## 2.1软件目标功能描述

我们开发本软件是为了方便学生随时查询天气状况、添加任意时间的备忘录和日程安排以及提高自身学习效率。本软件的使用者以学生为主。

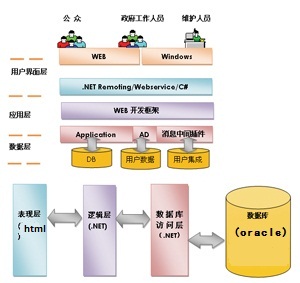
软件由手机客户端或电脑客户端用户与服务端数据库来搭配，并使用第三方公司提供的open API接口。客户端分为九个大模块，分别为：登陆、注册、检索、发布、用户管理、高级管理、版块管理、数据库维护。各模块需调用的API不再赘述。

## 2.2软件架构

平台架构引用

******

技术架构引用



如上图技术架构适用于客户端为PC。客户端为移动设备时，整体框架照旧，仅需将三层结构的最上层修改为移动设备适配项，开发可使用Eclipse或andriod studio。

## 2.3软件功能模块

客户端

登录、注册、

天气、日历及学习监督系统

用户

管理员

发布通知

数据库管理

视图管理

用户管理

登录及接收通知

查询天气

创建备忘录

添加学习锁屏任务

设定闹钟

注销登录

用户注册生成特定账户

查询日历

查询或修改已有闹钟

查询或修改已有备忘录

数据库转储

数据库更新

用户增删改及辅助密码找回

视图板块增删改

查询或修改已有学习锁屏任务

推荐每日音乐

ALARM

用户登录接口、

服务器

个人备忘录、闹钟及锁屏任务增删改

数据库检索接口

视图管理

用户管理

显示天气

用户注册接口

发布通知

视图版块增删改

接收通知

提醒用户日程安排

操作数据库

增加数据

查询数据库

服务器

# 3 开发环境概述

## 3.1 eclipse简介

[](http://baike.baidu.com/albums/23576/23576.html#0$3a86813d16326e36bba16702)

eclipse-galileo

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过[插件](http://baike.baidu.com/view/18979.htm" \t "_blank)组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括 Java 开发工具（Java Development Tools，JDT）。

### Eclipse的主要组成

　　Eclipse是一个开放源代码的软件开发项目，专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。它主要由**Eclipse项目**、**Eclipse工具项目**和**Eclipse技术项目**三个项目组成，具体包括四个部分组成——Eclipse Platform、JDT、CDT和PDE。JDT支持Java开发、CDT支持C开发、PDE用来支持插件开发，Eclipse Platform则是一个开放的可扩展IDE，提供了一个通用的开发平台。它提供建造块和构造并运行集成[软件开发工具](http://baike.baidu.com/view/973702.htm" \t "_blank)的基础。Eclipse Platform允许工具建造者独立开发与他人工具无缝集成的工具从而无须分辨一个工具功能在哪里结束，而另一个工具功能在哪里开始。

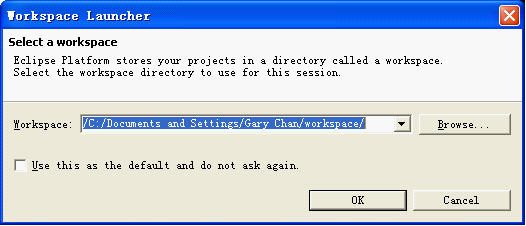
### Eclipse SDK

Eclipse SDK（软件开发者包）是Eclipse Platform、JDT和PDE所生产的组件合并，它们可以一次下载。这些部分在一起提供了一个具有丰富特性的开发环境，允许开发者有效地建造可以无缝集成到Eclipse Platform中的工具。Eclipse SDK由Eclipse项目生产的工具和来自其它开放源代码的[第三方软件](http://baike.baidu.com/view/287257.htm" \t "_blank)组合而成。Eclipse项目生产的软件以 GPL发布，第三方组件有各自自身的许可协议。

## 3.2 eclipse下载与配置

下载Eclipse  
Eclipse的下载网页是[http://www.eclipse.org/](http://www.eclipse.org/downloads/index.php)[downloads/index.php](http://www.eclipse.org/downloads/index.php)。

**安装Eclipse**  
安装Eclipse异常简单，把zip包中的eclipse目录解压缩到C盘根目录中即可。

**运行Eclipse**  必须安装了J2SE 1.4以上的SDK或者JRE来运行Eclipse。  
一切就绪之后，双击“C:eclipseeclipse.exe”便能够启动Eclipse了。首次启动Eclipse，它会让你配置工作区，一般选择默认选项即可：   


## 3.3 WCF层次搭建

基于整个解决方案模型已经构建好的前提下，创建服务契约: WCF 采用基于契约的交

互方式实现了服务的自治，以及客户端和服务端之间的松耦合。WCF 包含四种类型的契约：

服务契约、数据契约、消息契约和错误契约，这里着重于服务契约。从功能上讲，服务契

约抽象了服务提供的所有操作；而站在消息交换的角度来看，服务契约则定义了基于服务

调用的消息交换过程中，请求消息和回复消息的结构，以及采用的消息交换模式。通过在

接口上应用 System.ServiceModel.ServiceContractAttribute 特性将一个接口定义成服务契约。在应用 ServiceContractAttribute 特性的同时，还可以指定服务契约的名称和命名空间，如下：

namespace LabSystem.Service

{

[ServiceContract]

public interface I LabSystem Client

{

[OperationContract]

void DoWork();

[OperationContract]

bool AddUserType(UserType userType);

} }

当服务契约成功创建时，我们需要通过实现服务契约来创建具体的 WCF 服务，并通过

自我寄宿的方式寄宿服务，WCF 服务寄宿通过一个特殊的对象完成：ServiceHost，松耦合

是 SOA 的一个基本的特征，WCF 应用中客户端和服务端的松耦合体现在客户端只须要了解WCF 服务基本的描述，而无须知道具体的实现细节，就可以实现正常的服务调用。WCF 服

务的描述通过元数据（Metadata）的形式发布出来。WCF 中元数据的发布通过一个特殊的

服务行为 ServiceMetadataBehavior 实现。配置结束生成正确的 xml 文件：

<?xml version="1.0" encoding="utf-8"?>

<configuration>

<configSections>

<!-- For more information on Entity Framework configuration, visit

http://go.microsoft.com/fwlink/?LinkID=237468 -->

<section name="entityFramework"

type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection,

EntityFramework, Version=5.0.0.0, Culture=neutral,

PublicKeyToken=b77a5c561934e089" requirePermission="false" />

</configSections>

<appSettings>

<add key="aspnet:UseTaskFriendlySynchronizationContext" value="true" />

</appSettings>

<system.web>

<compilation debug="true" targetFramework="4.5.1">

<assemblies>

<add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral,

PublicKeyToken=b77a5c561934e089" />

</assemblies>

</compilation>

<httpRuntime targetFramework="4.5.1" />

</system.web>

<system.serviceModel>

<behaviors>

<serviceBehaviors>

<behavior>

<!-- To avoid disclosing metadata information, set the values below to

false before deployment -->

<serviceMetadata httpGetEnabled="true" httpsGetEnabled="true" />

<!-- To receive exception details in faults for debugging purposes, set

the value below to true. Set to false before deployment to avoid disclosing

exception information -->

<serviceDebug includeExceptionDetailInFaults="false" />

</behavior>

</serviceBehaviors>

</behaviors>

<protocolMapping>

<add binding="basicHttpsBinding" scheme="https" />

</protocolMapping>

<serviceHostingEnvironment aspNetCompatibilityEnabled="true"

multipleSiteBindingsEnabled="true" />

</system.serviceModel>

<system.webServer>

<modules runAllManagedModulesForAllRequests="true" />

<!--

To browse web app root directory during debugging, set the value below

to true.

Set to false before deployment to avoid disclosing web app folder

information.

-->

<directoryBrowse enabled="true" />

</system.webServer>

<connectionStrings>

<add name="RestaurantEntities"

connectionString="metadata=res://\*/Entity.RestaurantEntity.csdl|res://\*/Entit

y.RestaurantEntity.ssdl|res://\*/Entity.RestaurantEntity.msl;provider=System.D

ata.SqlClient;provider connection string=&quot;data source=GATEWAY;initial

catalog=Restaurant;user

id=sa;password=sa;MultipleActiveResultSets=True;App=EntityFramework&quot;"

providerName="System.Data.EntityClient" />

</connectionStrings>

<entityFramework>

<defaultConnectionFactory

type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory,

EntityFramework">

<parameters>

<parameter value="v11.0" />

</parameters>

</defaultConnectionFactory>

</entityFramework>

</configuration>

至此，WCF配置结束。

# 4 数据库设计

## 4.1 MySQL简介

[](http://baike.baidu.com/albums/24816/24816.html#0$8b527d278cbdde50908f9d66)

MySQL是一个小型关系型数据库管理系统，开发者为瑞典MySQL AB公司。在2008年1月16号被Sun公司收购。而2009年，SUN又被Oracle收购。MySQL是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内。这样就增加了速度并提高了灵活性。MySQL的SQL“结构化查询语言”。SQL是用于访问数据库的最常用标准化语言。MySQL软件采用了GPL（GNU通用公共许可证）。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了MySQL作为网站数据库。

**数据类型**

通常的整数、浮点数、字符串和数字均以多种长度表  
示，并支持变长的BLOB（Binary Large OBject）类型。对整数字段由自动增量选项， 日期，时间字段也能很好的表示。  
**SQL兼容性**

mysql包含一些与SQL标准不同的转变， 他们的大多数被设计成是对SQL语言脚本语言的不足的一种补偿。

## 4.2 数据库需求设计

本软件需要建立的数据库表（略）

**\*本次开发使用的是eclipse3.6。**

**\*本次开发使用的是Visual Studio2013**

**\*本次开发使用MySQL 5.2版本。**