

## 第3章 系统设计

在明确了需求的详细内容后，需要对软件进行系统设计。其中设计内容分为“系统架构设计、系统功能详细设计、数据库设计”三个部分。系统架构设计包含系统平台整体的物理架构设计、系统软件分层设计以及各个业务子系统的架构设计。系统功能详细设计是针对系统平台核心功能的具体业务流程设计以及内部代码类模块的结构设计。数据库设计是为了满足数据需求而进行的数据库建模以及数据处理汇总流程的设计。

### 3.1 系统架构设计

#### 3.1.1 系统整体架构设计

由于本平台的软件设计规模比较大（物理架构图如图 3.1 所示），业务需求涉及到“嵌入式设备的控制和数据采集”、“校园网内的端到端长连接通信”、“数据并发处理汇总”、“多种类型数据的基础服务”、“移动端 APP”、“移动端网站”、“PC 端网站”等多个方面<sup>错误!未找到引用源。</sup>。所以在确定了需求边界之后，将系统进行了拆分，化整为零。将系统分为了六个层次，从下到上分别为“仪表设备层”、“网关数据采集控制层”、“Socket 网络通信层”、“数据处理层”、“Web 服务层（MicroService）”、“终端展示层（Mobile Hybrid APP、HTML5 Mobile Website、PC RIA WebApplication）”，分层模型的架构图如图 3.2 所示。



图 3.1 物理架构图

分层架构模式使用“职责分离”的思想，对系统的每层业务功能职责进行明确划分<sup>错误!未找到引用源。</sup>。1、“仪表设备层”负责监测“电耗、水耗、气耗、温度、湿度、风速、照度”等数字量信息。2、“网关数据采集控制层”负责对“仪表设备”进行数据采集、控制指令下发、设备状态监测等。3、“Socket 网络通信层”负责将网关采集到的数据和用户的控制指令通过网络进行实时传输。4、“数据处理层”负责对“Socket 网络通信层”传送的数据进行并发的汇总计算，得出数据分析图表和报表。5、“Web 服务层（MicroService）”负责把“数据处理层”处理后的结果数据通过 Web 服务的方式发布出来，供移动端和 PC 端调用。6、“终端展示层（Mobile APP、HTML5 Mobile Website、PC RIA WebApplication）”调用“Web 服务层”的数据服务，进行数据查询和修改，通过移动和 PC 等终端把系统提供给用户使用。

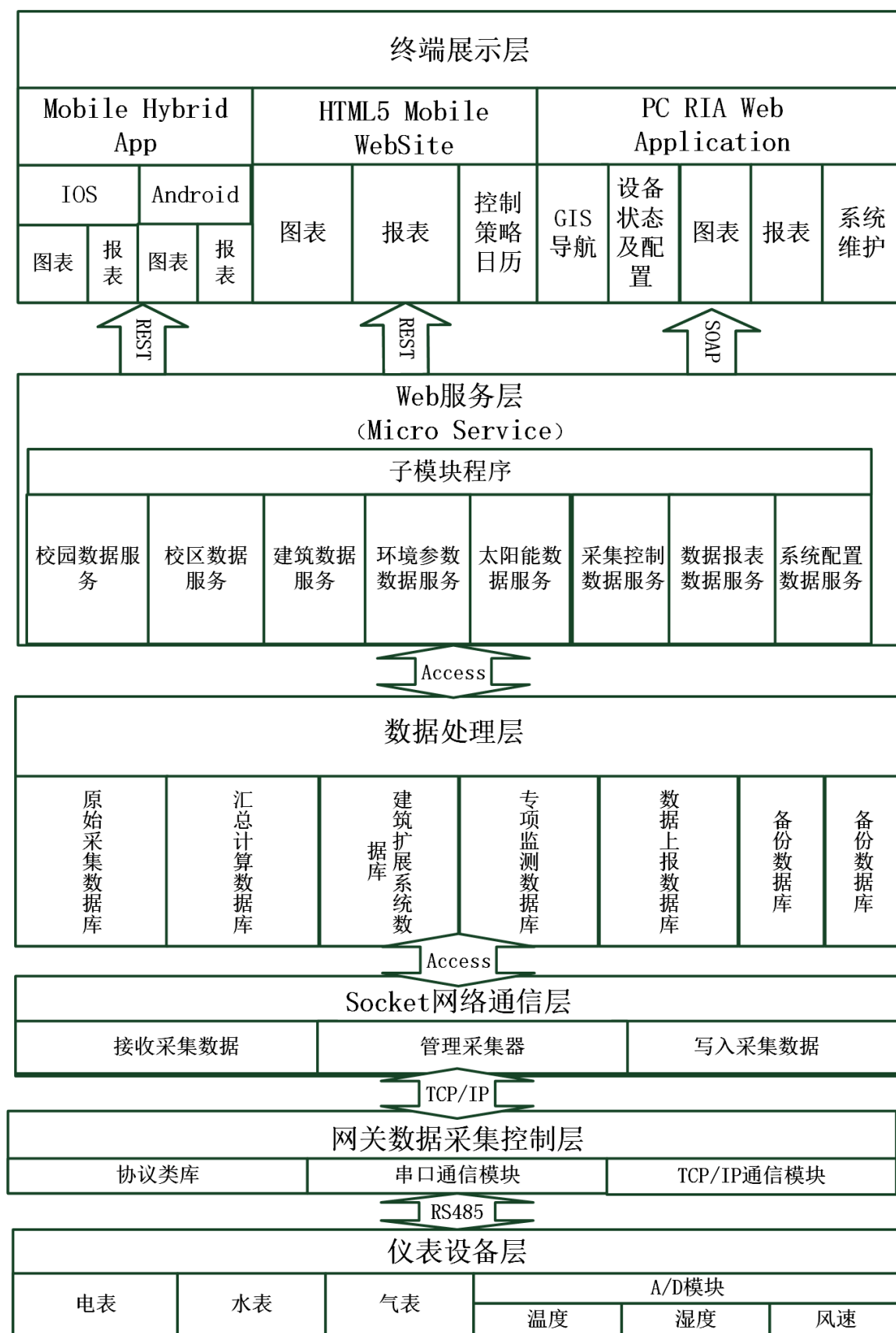


图 3.2 分层模型架构图

采用分层架构思想的优点如下：

- (1) 明确的功能边界；
- (2) 让开发人员专注每层的功能实现；
- (3) 重构或者扩展一个层次，不会影响其它的层次；
- (4) 每层的模块可以进一步划分；
- (5) 降低系统测试的难度；

系统整体使用 N 层架构,对于这 N 层又可以通过“四个子系统”和“两个模块”来实现。其中“终端展示层”包含了“Mobile Hybrid APP、HTML5 Mobile Website、PC RIA Web Application”三个子系统。“仪表设备层、网关数据采集控制层、Socket 网络通信层”用一个“数据采集控制”子系统进行了实现。这个四个子系统与用户交互有关，具有独立的业务功能体验。而“数据处理层”和“Web 服务层”是两个基础服务模块，它为四个子系统提供业务支撑。四个子系统又具有它独有的架构体系，下面分别阐述下“Mobile Hybrid App”和“HTML5 Mobile Website”、“PC RIA Web Application”、“数据采集控制子系统”的架构设计。

### 3.1.2 “Mobile Hybrid App & HTML5 Mobile Website”架构设计

(1) “Mobile Hybrid App”和“HTML5 Mobile Website”由一套架构实现，一套代码打包为两个系统<sup>错误!未找到引用源。</sup>。发挥 HTML5 的兼容性，把一套代码打包到“Mobile Hybrid App”、“HTML5 Mobile Webstie”、“WeChat”三个平台使用<sup>错误!未找到引用源。</sup>，如下图 3.3。

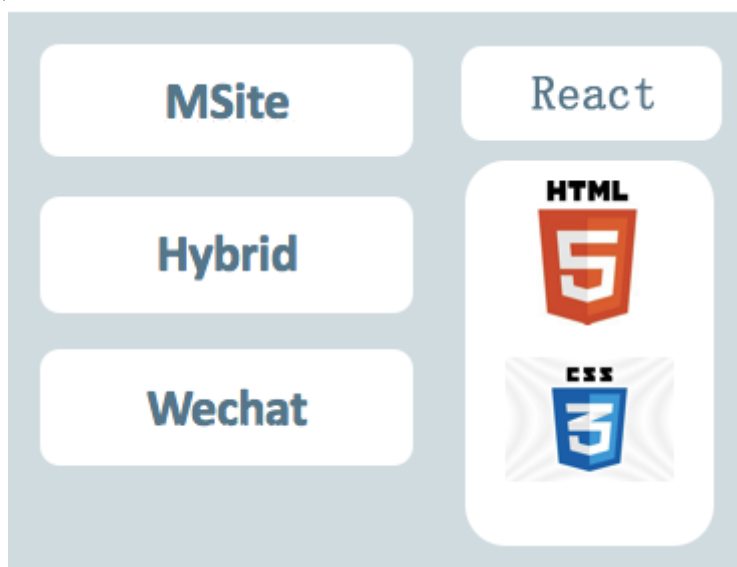


图 3.3“Mobile App & WebSite”打包部署功能图

(2) “Mobile Hybrid App & HTML5 Mobile Website”采用组件化开发的思想，先根据业务功能开发“UI 组件”、“BC 组件”及一些核心的库，然后将这些组件组

装成不同平台的系统，就像将相同的零件组装成不同汽车（卡车、汽车、跑车），如图 3.4。

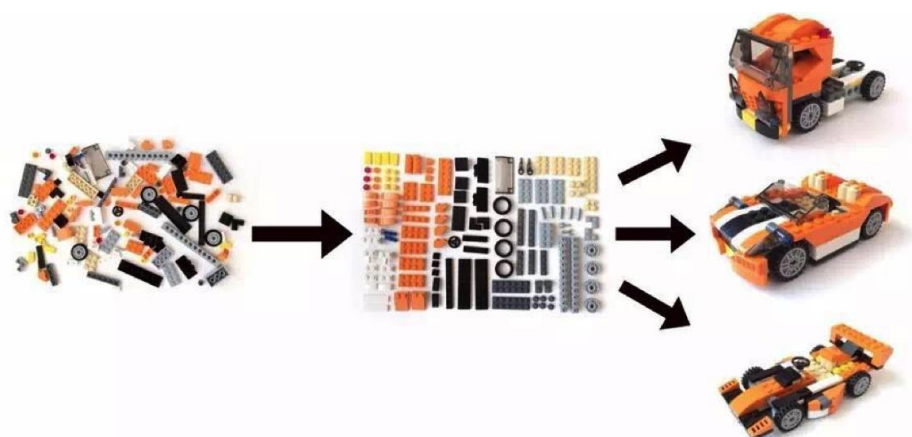


图 3.4 组装汽车图

(3) “Mobile Hybrid App”和“HTML5 Mobile Website”架构图一共分为“Native”、“Hybrid”、“H5 Framework”、“Bussiness Layer”四个层次，如图 3.5 所示。<sup>错误!未找到引用源。错误!未找到引用源。错误!未找到引用源。</sup>“Native”指“IOS 和 Android”的 API。“Hybrid”指 Native 和 HTML5 的桥接库。“H5 Framework”是实现“App 和 Website”的业务功能框架，其中包括核心公共库“缓存、路由、HTTP 发送请求客户端、通用工具、动作事件”等功能模块，UI 组件（“按钮、面板、图表”等组件）和业务组件（“导航栏”、“Banner”等组件）。“Bussiness Layer”指使用组件实现具体的“电耗”、“水耗”、“气耗”、“能耗公示”等业务功能。另外，“Application Manager”指应用控制器，用于实现各层之间的业务控制。最后使用“BrowserSync”、“HotLoader”、“ES2015”、“Babel”、“Webpack”、“JSLint”、“CSSLint”、“PostCSS”等组件完成项目的工程化，实现“多浏览器同步调试”、“热加载”、“JSES2015 的转译”、“单页应用打包”、“JS 语法检查”、“CSS 语法检查”、“PostCSS 语法预处理”等功能。<sup>错误!未找到引用源。</sup>

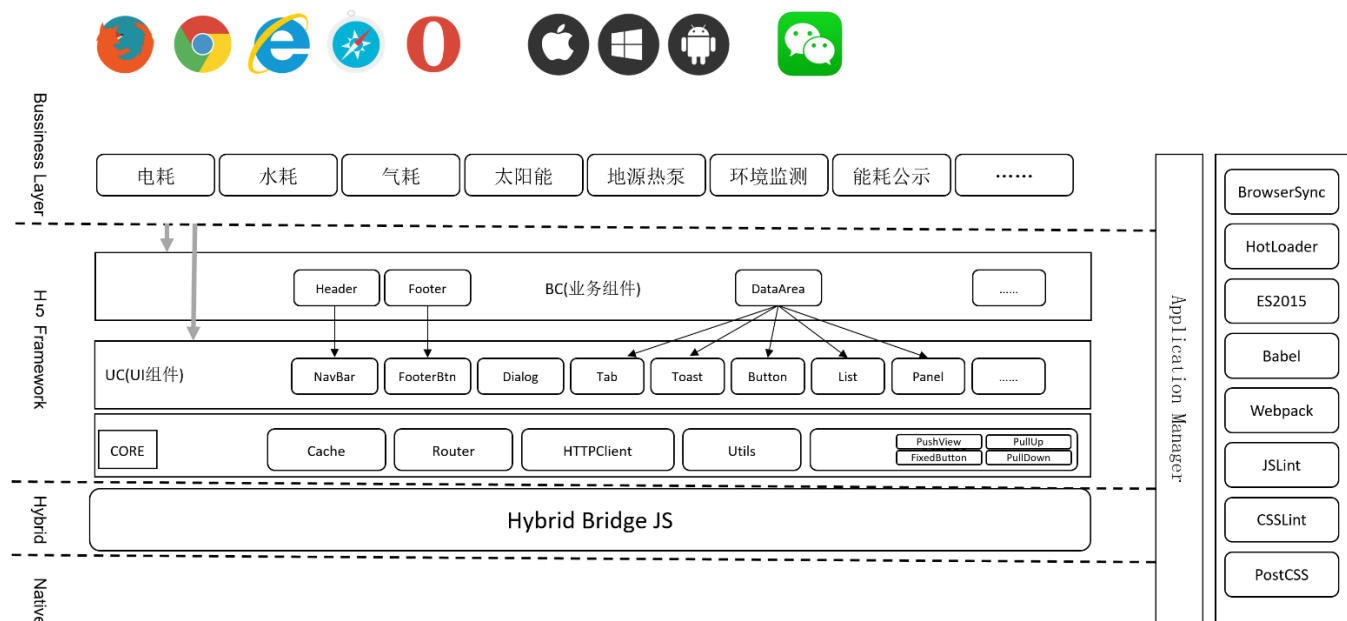


图 3.5“Mobile Hybrid App & HTML5 Mobile Website”架构图

(4) Hybrid 桥接层分为“HTML5 APP”、“WebView”、“Native API”、“Mobile OS”四个模块，它们之间的交互如图 3.6 所示。“HTML5 APP”通过“HTML API、Task”和 WebView 进行交互。WebView 作为“HTML5 APP”的渲染引擎，使用“Tasks 库”和“Native API”进行交互，同时通过“OS API”可以和手机操作系统进行交互。“Native API”通过“OS API”和手机操作系统进行交互。

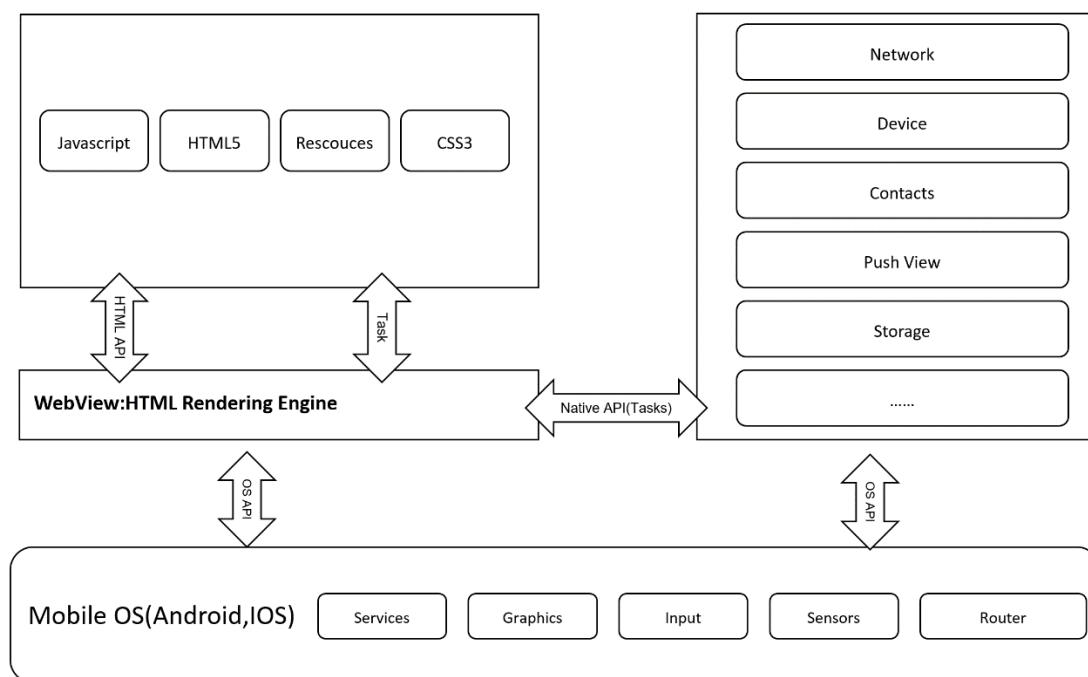


图 3.6Hybrid 桥接层交互图

(5) 组件和数据之间的交互使用的是 Flux 数据流的设计思想。Component 所在的 View 层不能直接对 React 的 State 进行操作，而需要依赖 Actions 派发动作指令来告知 Store 修改 State 状态，Store 接收 Actions 指令后对 State 做出相应的改变，View 层通过 State 进行数据更新也就产生了跟随 Store 的变化而变化的效果，如图 3.7。有关数据请求的处理数据更新也是 Flux 的设计思想，当 App 发出 AJAX 数据请求时，后台数据返回后更新 Model，然后更新 Model 数据的动作会通知修改 Store，进而 State 发生改变，浏览器执行渲染 Page 页面对应组件的动作，如图 3.8 所示。

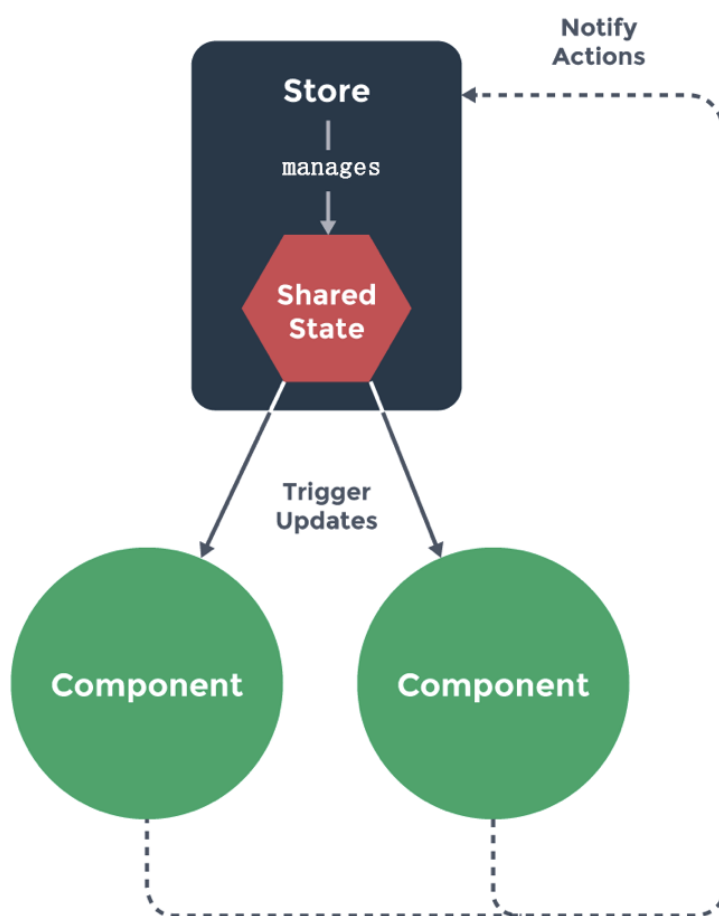


图 3.7 组件和数据交互图

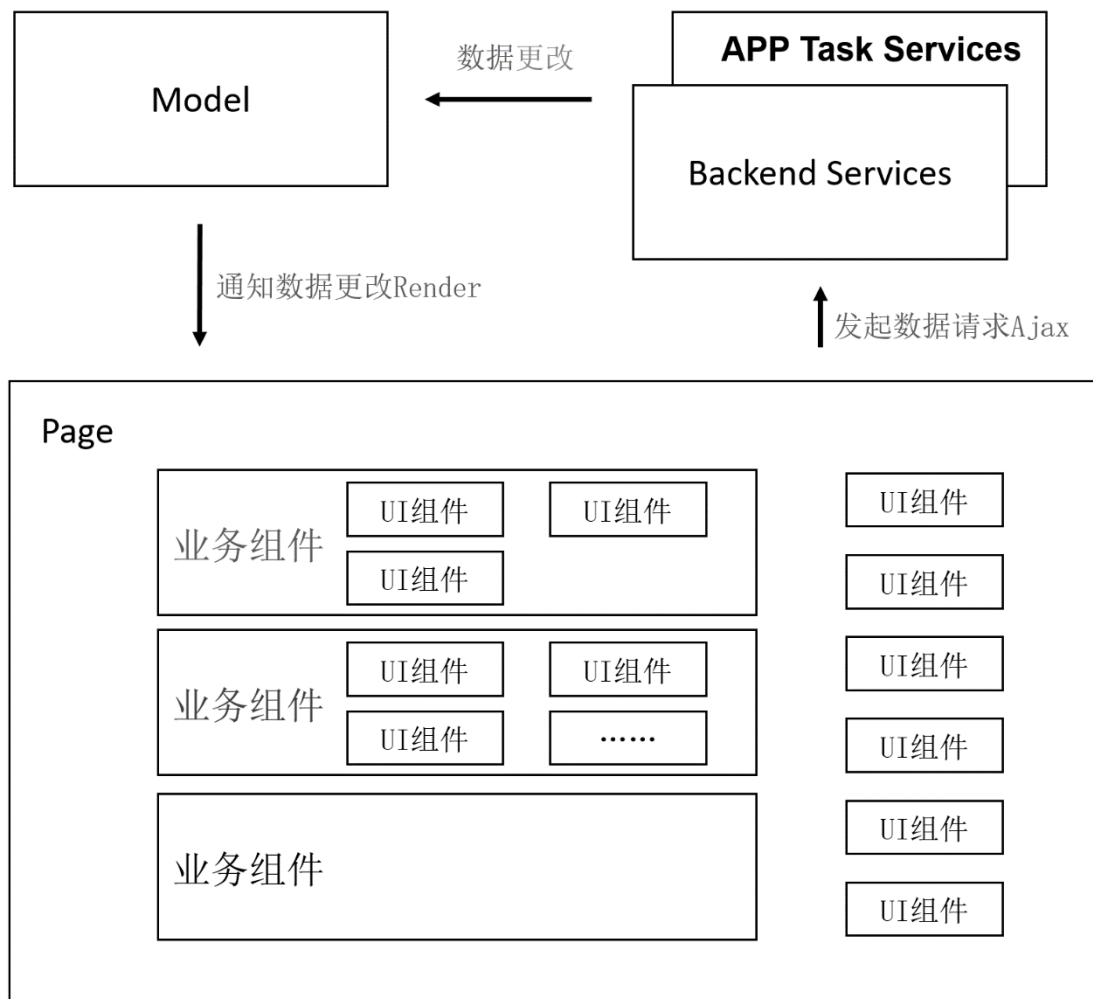


图 3.8Ajax 数据请求图

(6) 业务组件 (BC) 的工作流程是启动后探测它的运行环境是 Native 还是 Browser。如果是 Native 调用 Task 与 Native 进行交互, 否则调用纯“HTML5 API”实现 HTML5 页在 Browser 端的渲染, 如图 3.9。



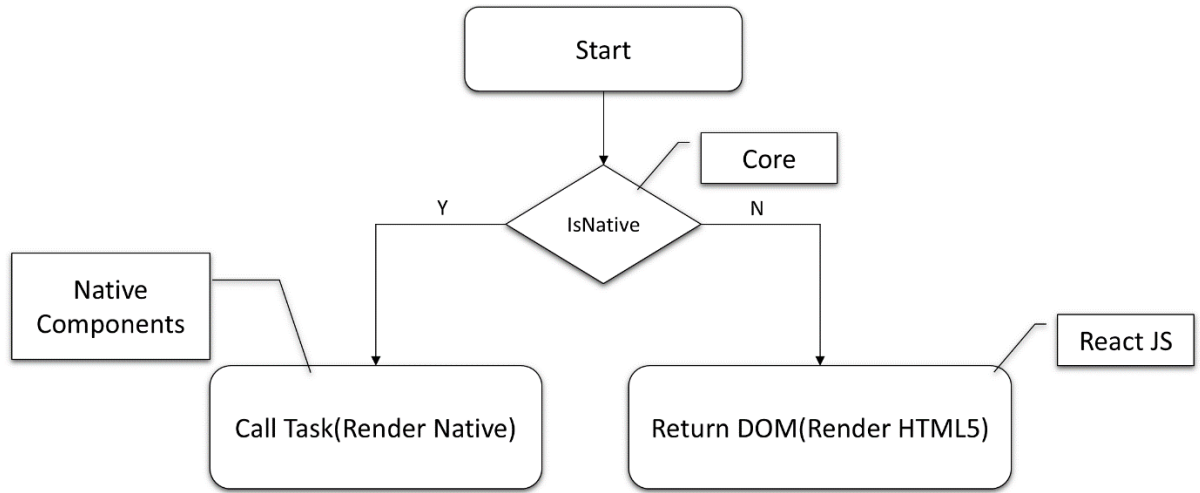


图 3.9 业务组件工作流程

“Mobile Hybrid App”和“HTML5 Mobile Website”，使用同一套代码，可根据场景自动打包至 Hybrid 或者 Online Server。并且在打包时压缩前端代码,最大化优化性能。通过 Webpack 将众多的“JS”、“CSS”、“图片”、“模板”等资源，打包为静态优化后的一个“JS”和一个“CSS”及压缩后的图片资源<sup>错误:未找到引用源。</sup>，如图 3.10 和图 3.11。

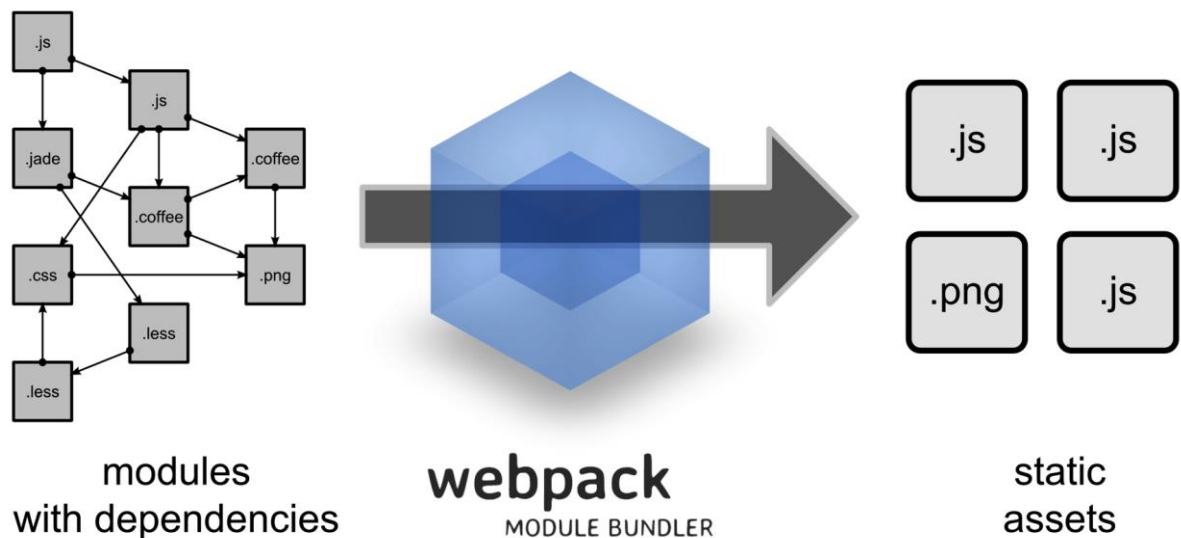


图 3.10 webpack 打包示意图

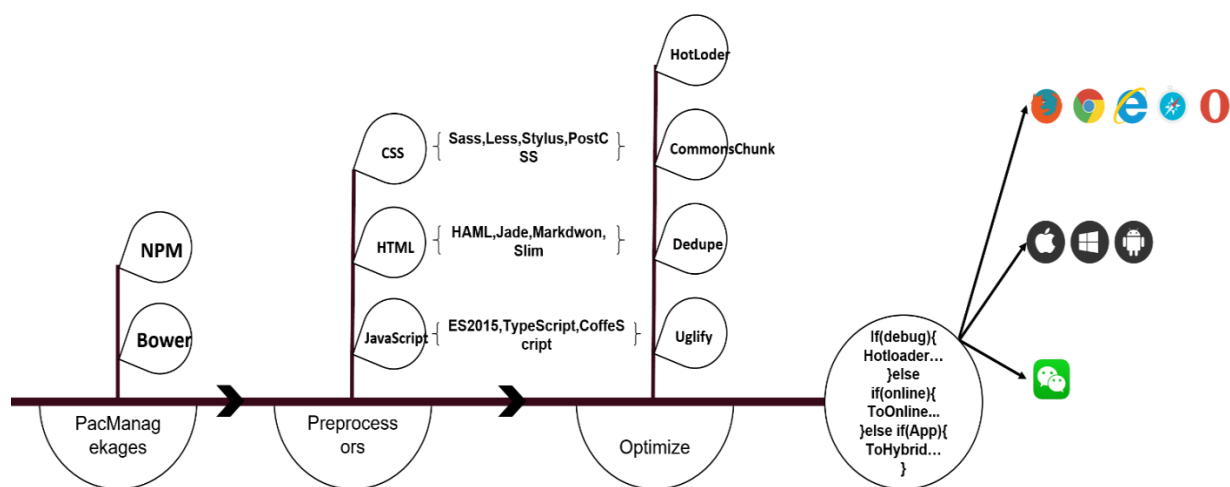


图 3.11webpack 打包流程图

### 3.1.3 “PC RIA WebApplication”架构设计

“PC RIA WebApplication”子系统包含四层，从下到上分别是“DAL 数据访问层”、“BLL 逻辑控制层”、“Web 逻辑应用层”、“Silverlight 及 Aps.net 对应的 View 层”<sup>错误!未找到引用源。错误!未找到引用源。错误!未找到引用源。</sup>。其中“DAL 数据访问层”实现系统对数据库和其它 Web 服务的访问。“BLL 逻辑控制层”实现系统数据的逻辑控制。“Web 逻辑应用层”实现系统 View 层对应的业务逻辑和部分 Web 服务的转化逻辑。“Silverlight 及 Aps.net 对应的 View 层”实现了业务的呈现，将系统通过 UI 界面展示给用户，使用户可以操作访问<sup>错误!未找到引用源。</sup>。具体如图 3.12。另外，系统还包含“Model”和“Entity”的实体类库，以实现系统数据的映射实体对象集合。最后，系统通过公用的“Common 公共类库”实现系统中核心方法和工具方法的共用，并管理整个系统的运行控制。



图 3.12PC RIA WebApplication 架构图

3.1.4 “数据采集控制子系统”架构设计

数据采集控制子系统主要包括“采集配置”、“数据采集控制”、“数据上传”三个模块。每一个模块有一个对应的子应用程序，应用程序通过协议栈中间件和数据库中间件调用对应的服务实现其功能。对应的功能逻辑架构图如图 3.13 所示。

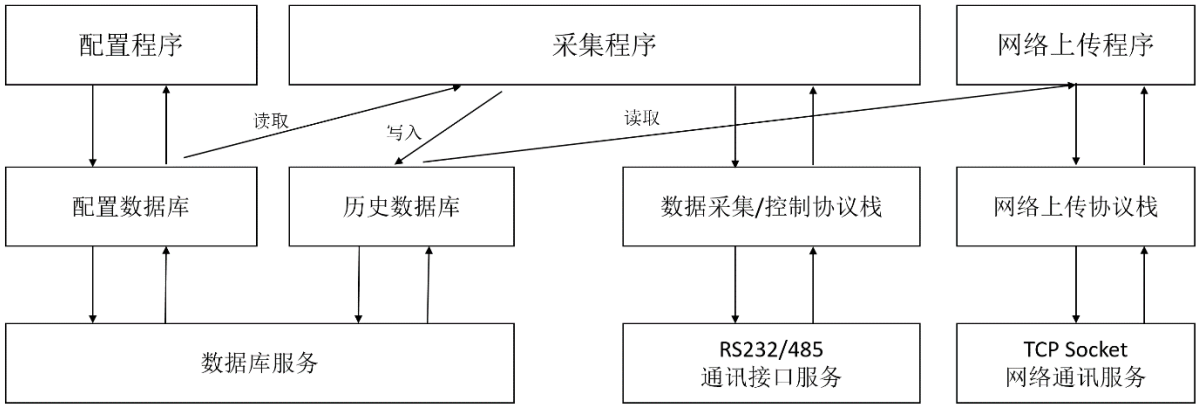


图 3.13 数据采集控制子系统功能逻辑架构图

### 3.2 系统功能详细设计

系统平台具有 16 个典型功能，分别是“电耗监测、能耗公示、太阳能发电监测、能耗监测、费用分析、专项监测（地源热泵空调监测）、环境参数监测、日程管理、GIS 导航、节能分析、系统维护、数据采集、远程控制、网络传输、数据上报、采集配置”等。它们和子系统之间的关系如图 3.14。

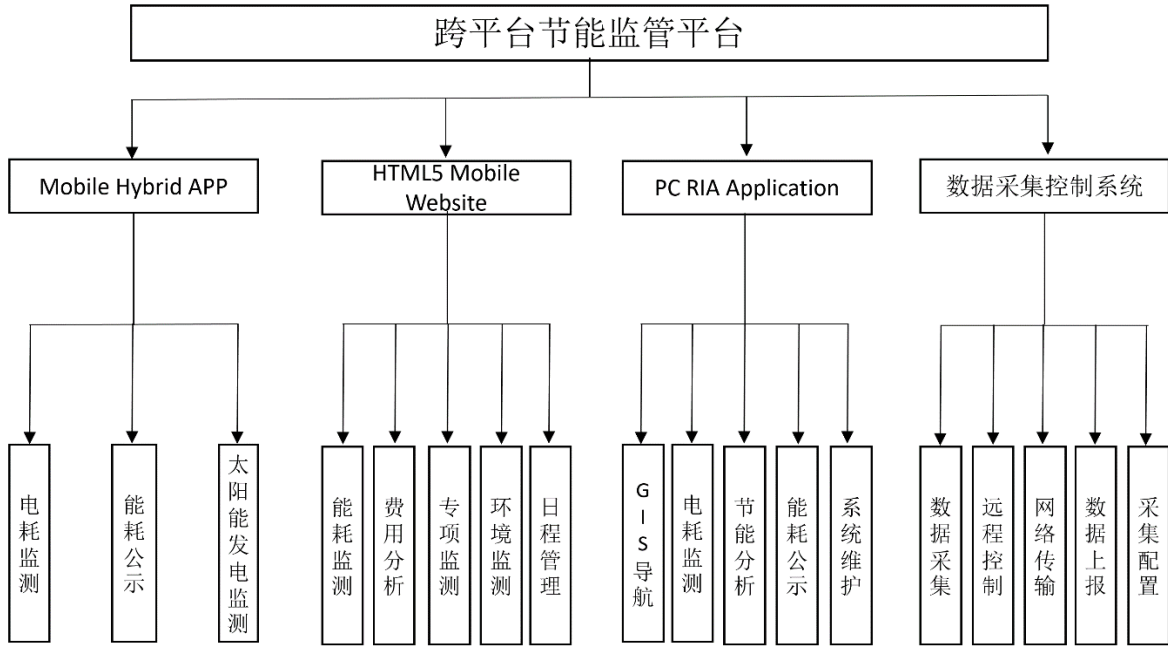


图 3.14 功能模块组织结构图

“系统功能详细设计”是从功能模块的角度详细的阐述了跨平台校园节能监管平台的功能结构，直观的描述了各个子系统和功能模块之间的关系。明确了功能边界，降低了系统开发的成本、风险和难度。

#### 3.2.1 数据采集控制子系统

数据采集控制子系统主要包括“采集配置”、“数据采集控制”、“数据上传”、“数据上报”四个功能。

(1) 数据采集配置功能用于设置网关机内置的基本配置信息，并维护这些信息，比如“COM 口”、“设备”、“点位”、“网络”、“采集器”、“控制计划”、“传输配置”等。此功能的实现活动图如图 3.15 所示。程序启动会自动检查配置信息，如果不存在则创建默认配置，存在则读取配置信息。最后进行配置信息的基本维护。

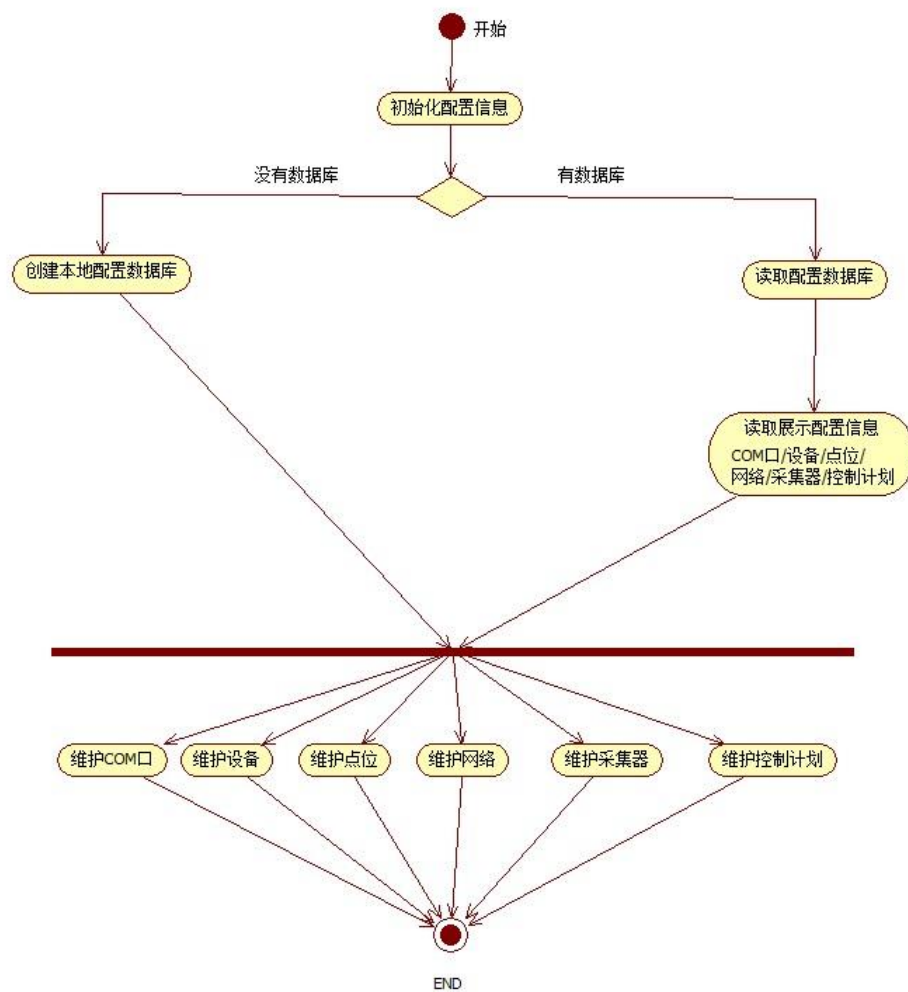


图 3.15 数据采集配置功能活动图

(2) 数据采集控制功能主要包括：

- a. 对网关机下属的各种设备进行数据采集和控制。
- b. 展示采集信息，比如“配置信息”、“采集数据读数”、“控制指令下发”、“协议报文”等过程信息，有利于用户查看网关机工作状态及排查异常。

此功能的逻辑如图 3.16 所示。程序启动首先读取配置信息，并将配置信息转化为实体对象集合。遍历实体对象集合，开启多线程，通过 COM 口或者网口进行数据采集和控制。

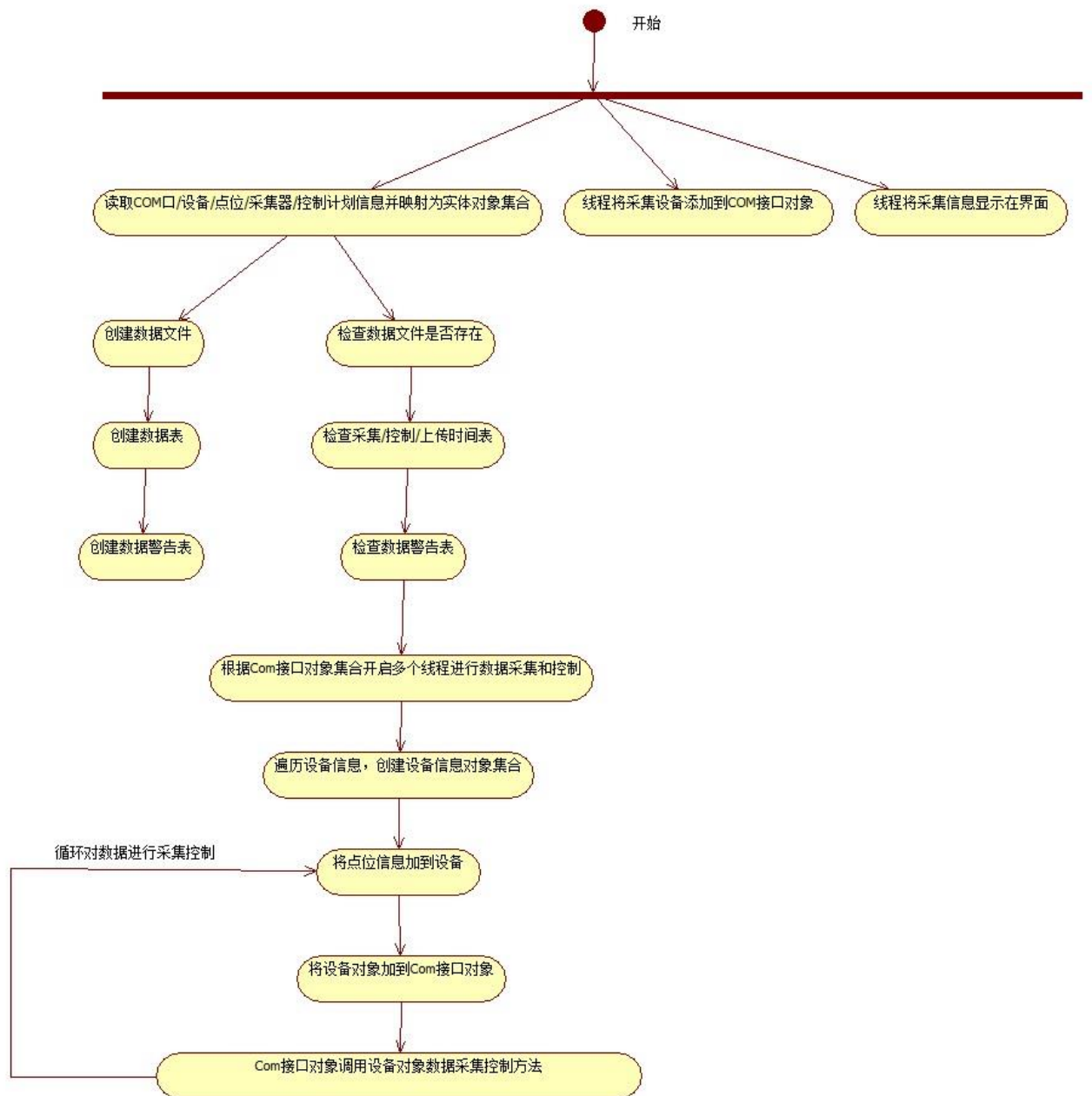


图 3.16 数据采集控制功能活动图

此模块比较复杂，为了实现“高内聚低耦合”的目的，对此功能模块抽象出了若干类如图 3.17。

a. “IDevice”接口，包含“通信编码和解码”的基本功能，实现各种采集控制设备的采集控制协议。由各种各样的设备类继承实现，比如“安科瑞电表 DTSF1352（DeviceDTSF1352 类）”、“八回路继电器（DeviceXKEight 类）”等一批设备对应的类。它们都具有“device（设备属性）”、“CollectDateTimeTicks（数据采集控制时间戳属性）”、“CommCode（通信报文属性）”、“AddMeter(添加设备点位方

法)”、“AddByte(添加采集字节并进行解码方法)”、“AddQueue(添加采集字节队列并进行解码方法)”、“GenTxbuf(获取发送报文方法)”等若干属性和方法。如果增加了新的设备，只要继承此接口进行实现即可，扩展性极好。

b. “ICommDevice”接口，包含了“报文读取和发送”的基本功能，由“CommSocket(Socket 网口通信类)”和“CommRS232(RS 串口通信类)”两个类实现。它们都具有“State(通信状态属性)”、“Start(开始方法)”、“Close(关闭方法)”、“Send(发送方法)”、“BytesToRead、ReadByte、ReadBuf(读取接受字节数据)”等方法属性。如果还有其它端口类型的通信方式，只要继承此接口并实现即可，保证了扩展性。

c. “CommInterface”设备每个通信端口的类，它依赖于“设备类”和“设备通信类”。它控制每个设备在端口中的通信过程。包含“AddDevice(添加设备进行数据采集控制方法)”、“ReadData(读取设备返回数据方法)”。

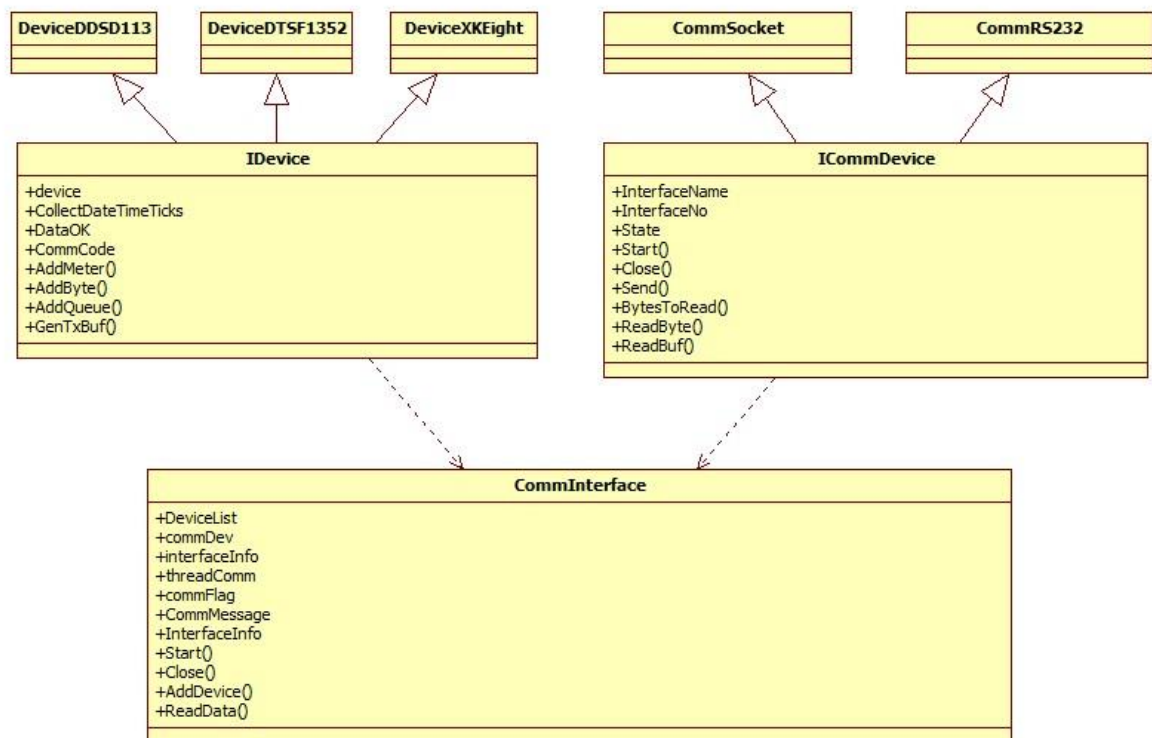


图 3.17 数据采集控制功能类图

(3) 数据采集上传主要指采集数据和异常信息的上传，以及服务端控制指令、控制计划、点位配置信息的下发。

此功能在业务逻辑上分为客户端和服务端。客户端发起 Socket 连接请求，服务器端接收到请求建立数据连接，然后完成数据相互之间的传输，如图 3.18。

此功能模块包含“SocketListener（Socket 通信监听类）”、“SocketConnect（Socket 连接类）”、“enNetType（通信状态标示）”、“NetProtocol（网络通协议类）”、“SocketCommunication（Socket 通信协议类）”、“NetGateClient（客户端类）”等，如图 3.19。其中，SocketListener 主要实现服务端网口 Socket 请求监听的功能。SocketConnect 主要实现客户端 Socket 请求发的发起功能。enNetType 用于标示每个客户端交互对象的交互状态。NetProtocol 实现了网络通信自定义的协议规则，包含“采集报文”、“控制报文”、“异常报文”、“心跳报文”的规则等。SocketCommunication 实现 Socket 收发报文的方法，是真正实现数据交互的类。NetGateClient 网关机客户端类，主要实现通信交互过程的控制，比如依赖 SocketCommunication 类实现 Socket 数据的收发，依赖 NetProtocol 类实现报文的编码和解码。



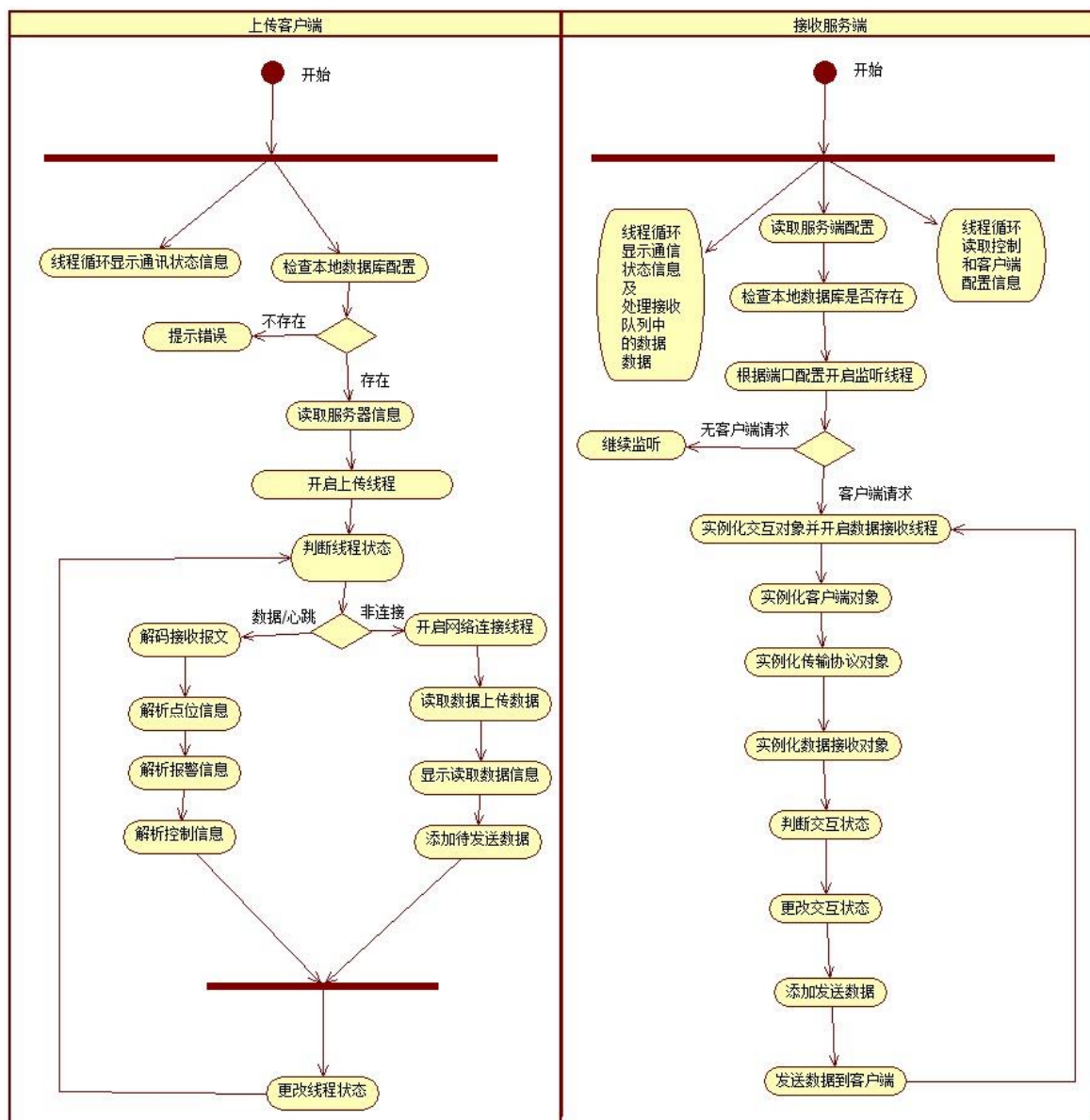


图 3.18 数据上传功能活动图

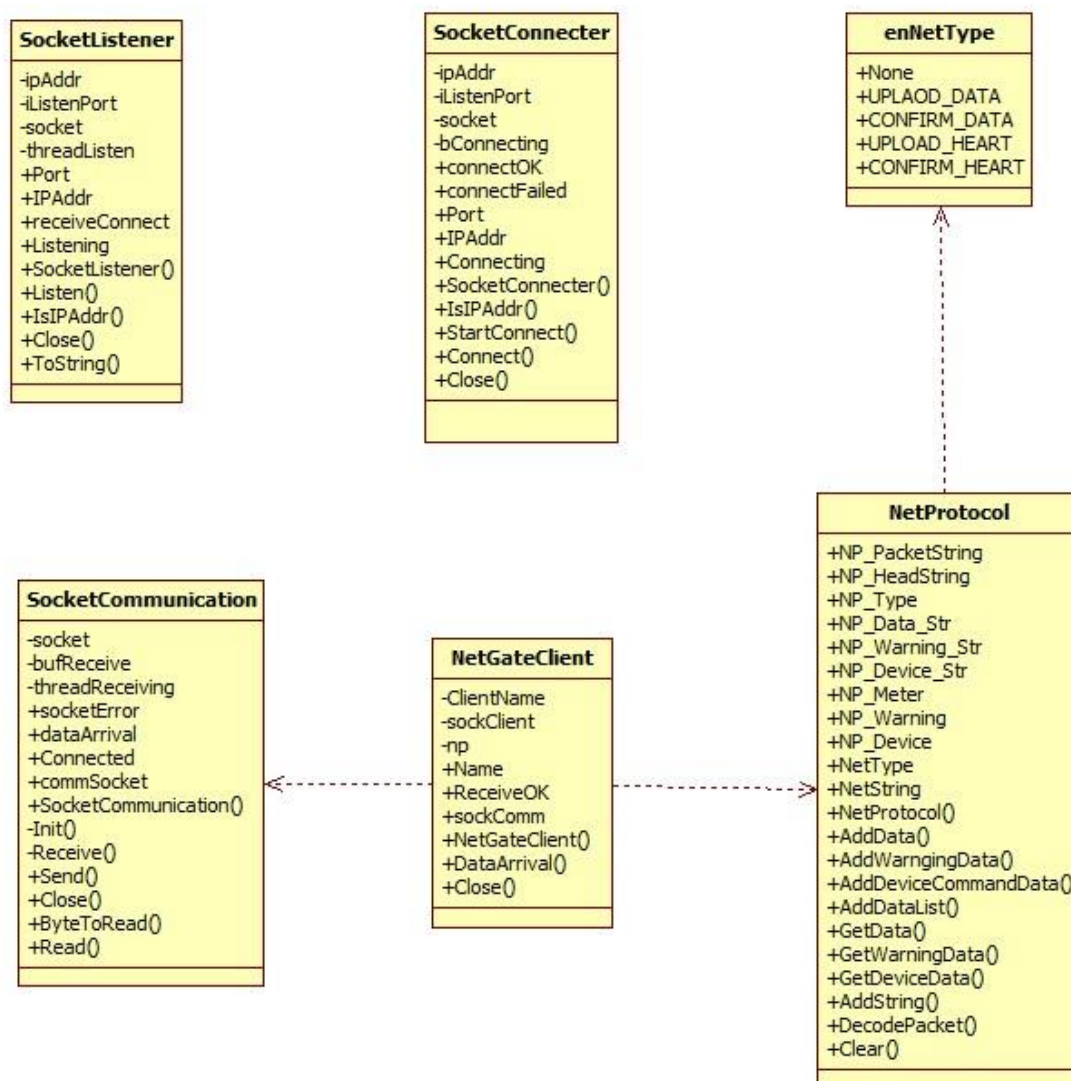


图 3.19 数据上传功能类图

(4) 数据上报功能是指系统通过作业生成需要上传的数据，然后使用客户端访问数据上传的接口服务，将数据上传到“教委平台”等，如图 3.20。

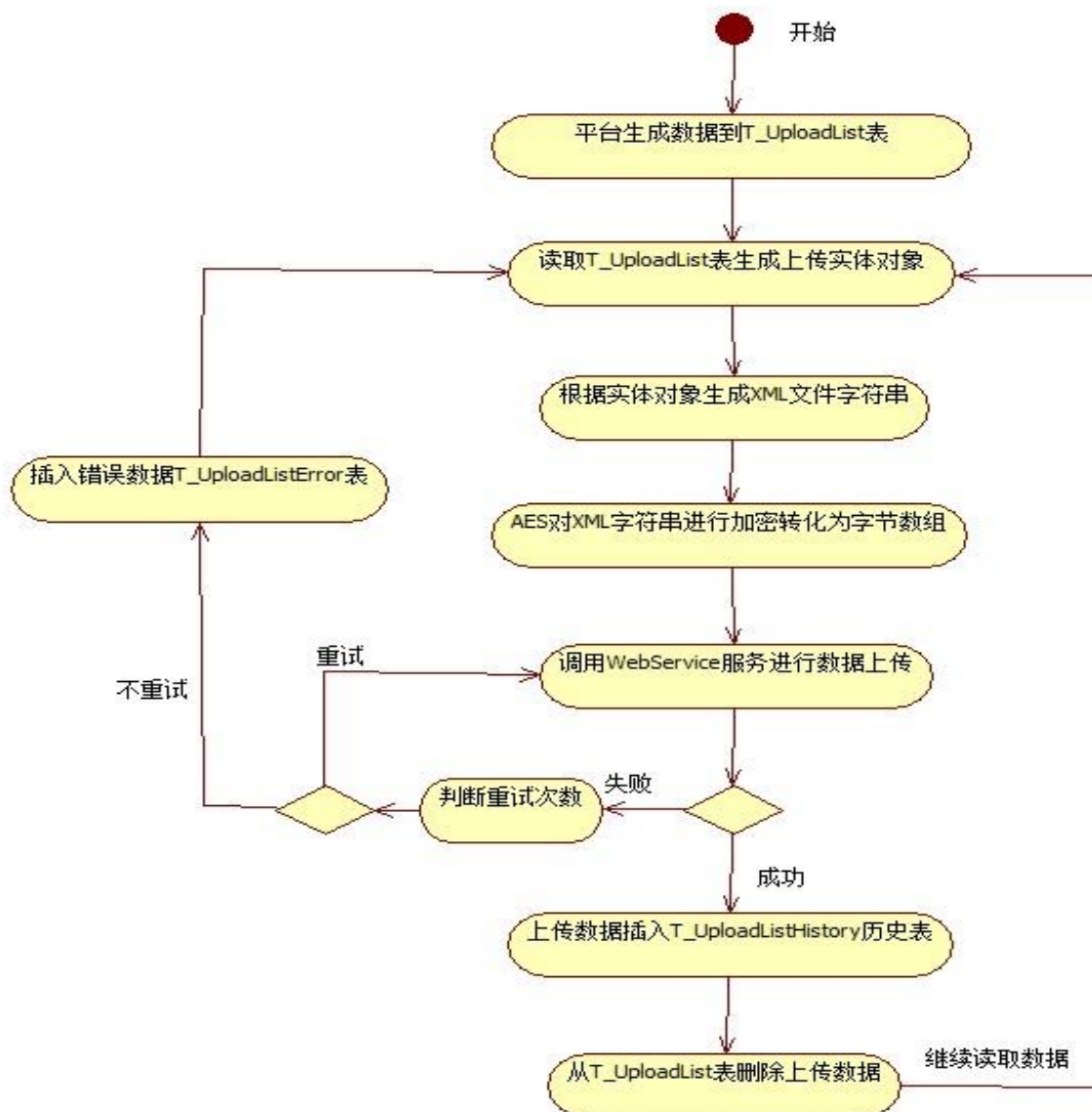


图 3.20 数据上报活动图

### 3.2.2 PC RIA Application

“PC RIA Application(PC 富客户端 Web 应用程序)”主要包括“GIS 导航”、“电耗监测”、“节能分析”、“能耗公示”、“系统维护”五个功能模块。

(1) GIS 导航功能的用例描述如下:

- a. 在地图上标示出校园的学校、校区及建筑的用能情况和概况。
- b. 从地图上可以跳转到对应的监测页面。

它的业务逻辑比较简单,是在 GIS 地图上一级一级的缩放,最后查看能耗情况。如图 3.21。

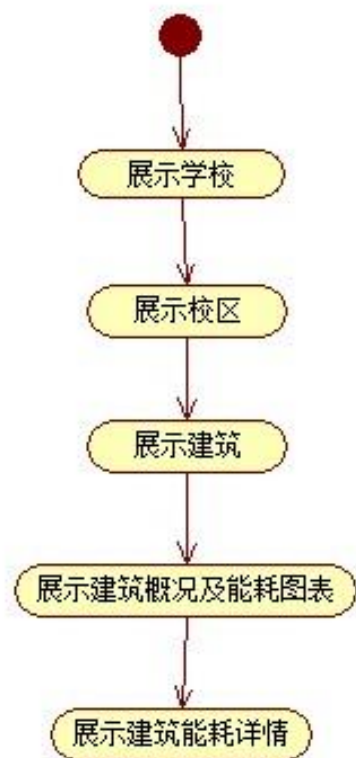


图 3.21GIS 导航活动图

(2) 电耗监测的功能主要描述如下：

- a. 查看“校园”、“校区”、“建筑”、“建筑分类”、“配电所”的电耗情况。
- b. 查看“逐时”、“逐日”、“逐月”、“逐年”的电耗情况。
- c. 查看“某个时间点”和“某个或几个时间区间”的电耗情况。
- d. 查看“空调”、“照明”、“动力”的用电情况。
- e. 选择“柱形图”、“线形图”、“面积图”、“散点图”等多种图表类型。
- f. 设置图表是“3D”或“2D”，及是否允许缩放等。

此功能的业务逻辑是选择监测对象，选择查询条件，调用接口，将数据查询出来并在页面做图表和报表展示。最后还支持导出功能。如图 3.22 所示。

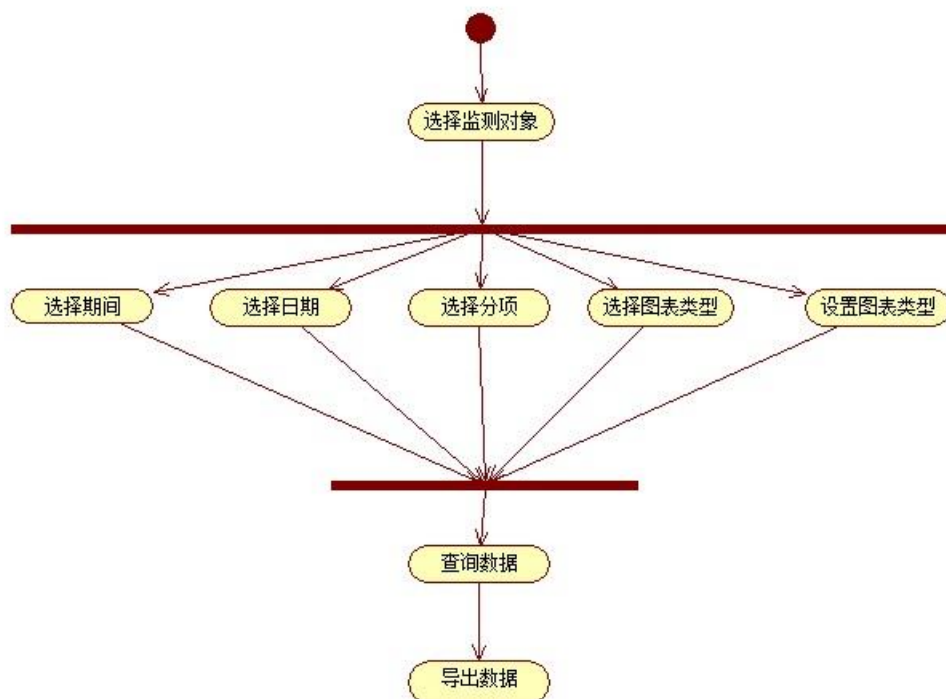


图 3.22 电耗监测活动图

(3) 节能分析功能主要是使用某些数据计算规则对数据进行对比分析。比如,“模拟数据和实测数据对比”、“昼夜对比”、“四季对比”、“不同学校对比”、“节能改造前和节能改造后对比”等对比分析。它的具体实现逻辑和电耗监测类似,只是查询条件不同,如图 3.23 所示。

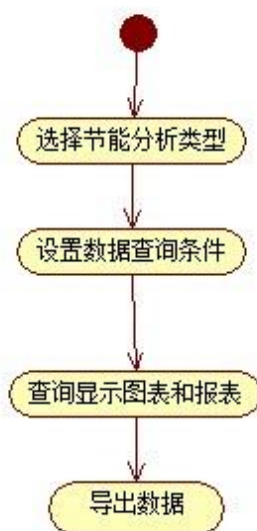


图 3.23 节能分析活动图

(4) 能耗公示功能主要是将“学校”、“校区”、“建筑”、“建筑分类”等多个维

度的数据公示给学校人群及社会公众。公示的数据以总体形式的“排序图”、“结构图”等体现。此功能的实现逻辑也与电耗监测、节能分析类似，只是最后查询数据的图表和报表表现形式不一样。如图 3.24 所示。

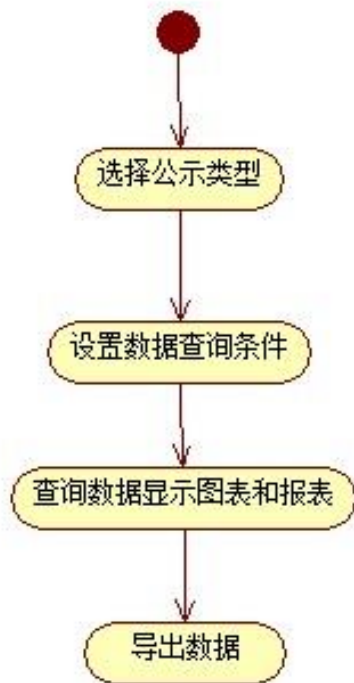


图 3.24 能耗公示活动图

（5）系统维护功能是整个监管平台的后台管理系统，涵盖了所有子系统的维护和管理。主要包括包括的功能如下：

- a. “预警信息”，统计异常信息并通知物业管理人员。
- b. “基本信息维护”，包括“学校”、“建筑”、“设备”、“网络”等基本配置信息。
- c. “关系维护”，主要维护“采集点位和设备”、“设备和建筑”、“设备和校区”、“建筑和校区”等之间的关系。
- d. “数据字典维护”，主要维护数据库的定义信息。
- e. “系统日志”，主要维护系统运行的日志信息。
- f. “用户权限信息”，主要维护系统的用户信息和每个用户的“功能、建筑”权限等信息。

3.2.3 HTML5 Mobile Website

“HTML5 Mobile Website(HTML5 移动网站)”主要包括“能耗监测”、“费用分析”、“专项监测”、“环境监测”、“日程管理”五个功能模块。它们的逻辑都比较统一，首先设置查询数据的条件，然后调用 ajax 服务，查询数据以后将数据渲染到页面上<sup>错误!未找到引用源。</sup>。如图 3.25 所示。

(1) 能耗监测功能的用例描述如下:

a. 能耗总览, 浏览建筑整体的能耗情况, 包括“本日、本周、本月、本年”等时间粒度的“能耗、电耗、能耗结构”等。

b. 能耗分析, 分析“建筑分区(南区, 北区等)”、“电耗分项(空调, 照明, 动力等)”、“学校部门(土木工程学院、建筑工程学院、软件学院等)”、“回路表具(教学北楼总进线 B、图书馆空调 A 等)”等各维度的能耗情况。

(2) 费用分析的主要功能是查看每个“区域、部门、回路”的“本日、本周、本月、本年”及“时间区间”的“峰时段、平时段、谷时段”费用。和它们每个时段“比昨日、比上周、比上月、比去年”的能耗费用增长情况。查看“峰、平、谷”时段的能耗费用结构图。

(3) 专项监测功能主要包括:

a. 空调系统监测, 查看空调系统每个机组对应时间段的“COP、温度、功率、能耗”。

b. 太阳能光伏, 查看光伏系统的发电量、辐照强度、室外温度等信息。

c. 太阳能光热, 查看光热系统的产热量。

(4) 环境监测的主要功能是查看每个环境监测点位的“温度、湿度、照度”等信息。

(5) 日程管理功能的主要功能是“制定空调机组每天的开机计划, 存储在数据库中, 然后被控制终端执行”。

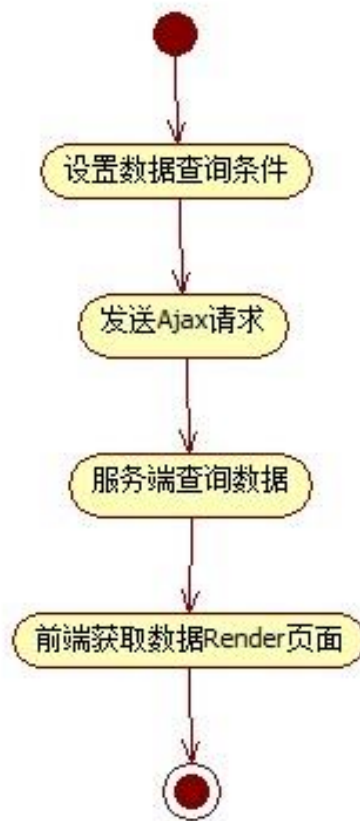


图 3.25 HTML5 Mobile Website 数据监测活动图

#### 3.2.4 Mobile Hybrid APP

“Mobile Hybrid APP (移动端混合模式 APP)”主要包括“电耗监测”、“能耗公示”、“太阳能发电监测”三个功能模块。1、首先原生 APP 启动并开启 WebView 容器页面，页面载入同时 Native 加载 “Native-HTML5” 桥接库。2、原生 APP 加载 HTML5 单页应用的静态资源，并发起数据请求，渲染出默认页面。3、用户点击 “用电监测、能耗公示、太阳能监测” 中的任意一个，页面将发送数据请求，然后根据返回的数据进行渲染。详细业务逻辑如图 3.26 所示。



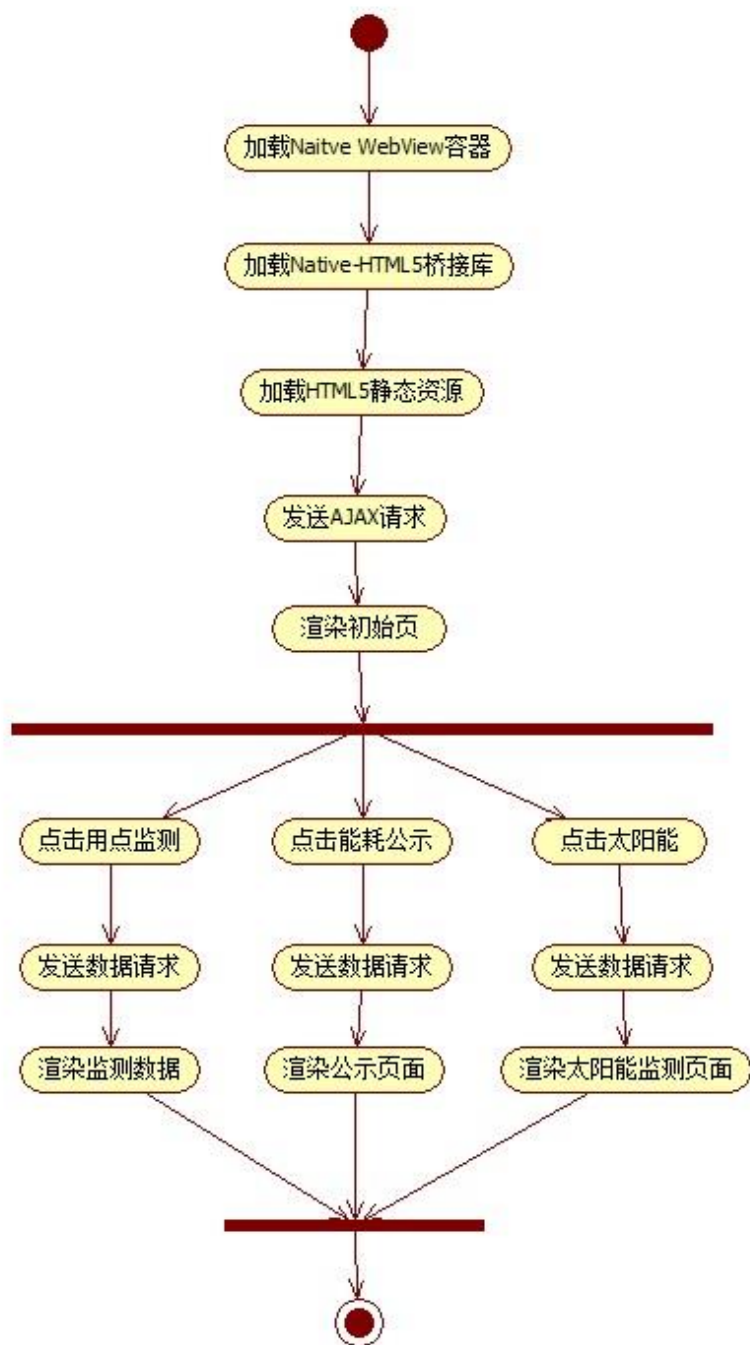


图 3.26 Mobile Hybrid APP 数据活动图

### 3.3 数据库设计

本平台的数据库规模比较大,包括几百张表,几千个数据采集点和大量并发处理作业。一个稳定和高性能的数据库特别重要,直接影响各个子系统查询和处理数据的效率。

3.3.1 数据建模

E-R 图,实体-联系模型（简称 E-R 模型）,被广泛作为数据库建模的工具<sup>错误!未找到引用源。</sup>。

在节能监管平台系统中，采用数据实体属性图和 E-R 图对系统中的数据进行抽象，建立实体关系模型。整个平台系统使用了 300 多张数据表，在这里我们对每个子系统使用到的核心数据对象进行建模并给出数据实体属性图和 E-R 图。“学校、校区、建筑、建筑分类”信息是系统的基本配置信息，一个学校包含若干校区，每个校区又包含很多建筑，而且建筑还可以分为不同的种类。每个建筑又包含很多监测设备，比如“网关机、电表、水表、温度计、流量计、风速仪”等。每个设备又具有很多监测点位信息和自身的配置信息。另外，还有很多专项的监测系统，比如建筑独有的“太阳能监测系统、地源热泵空调系统”等。由于数据实体对象比较多，在这里我们给出不同数据对象的典型代表。

建筑是节能监管平台分析的主要对象，它包括配置信息和数据信息，以及它和表具设备之间的关系。其中数据信息又包括“时、日、月、年”的电耗量，以及“总量、空调、照明、动力”等不同分项的电耗量。这些信息的存储和处理需要一系列数据表来支撑。与建筑有关的数据实体属性图如图 3.27、图 3.28、图 3.29、图 3.30、图 3.31 所示。

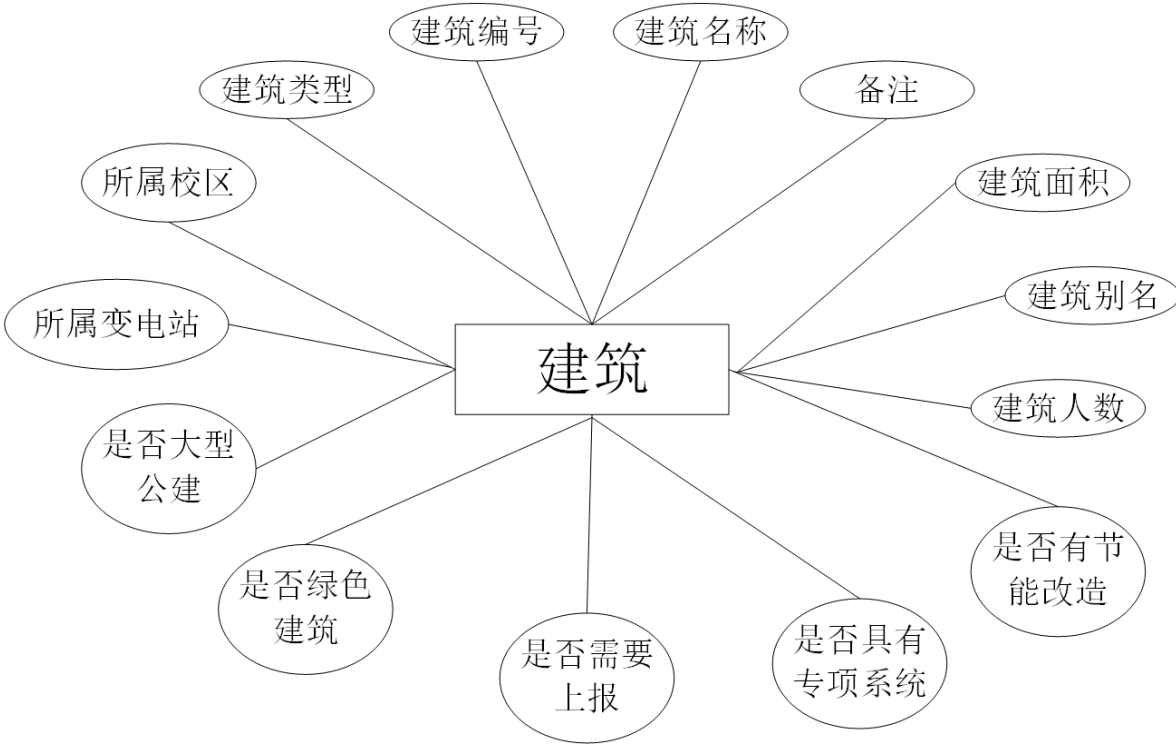


图 3.27 建筑基础信息数据实体属性图

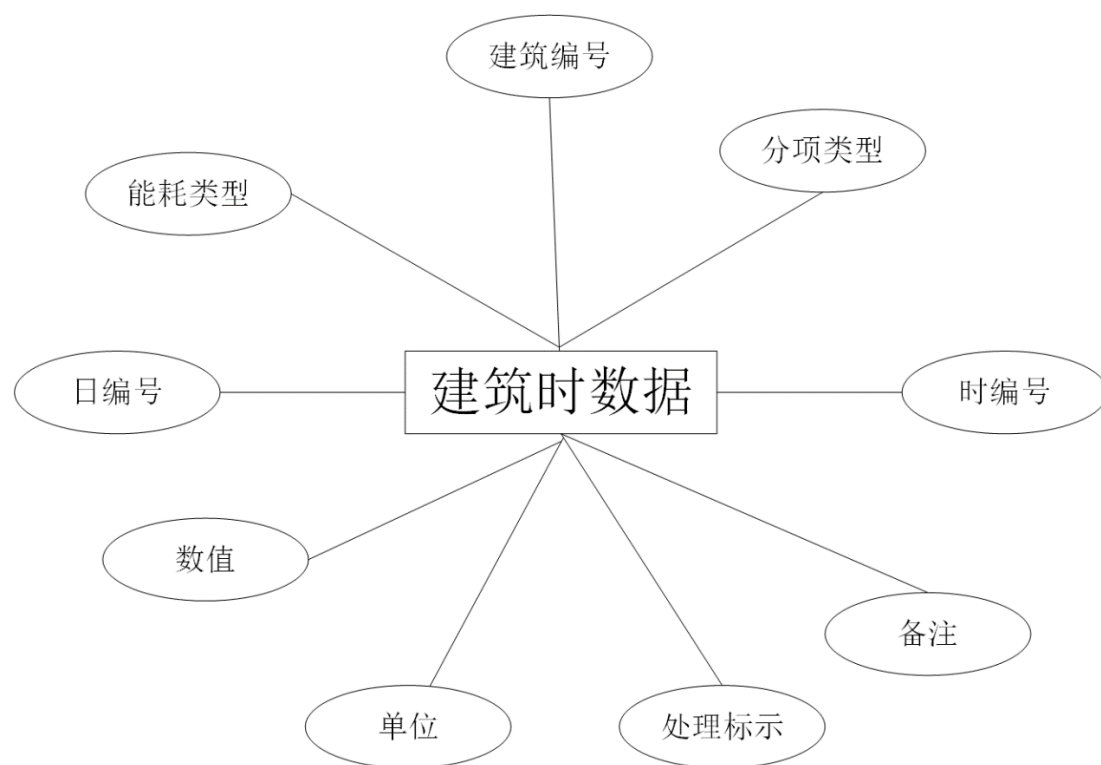


图 3.28 建筑时数据数据实体属性图

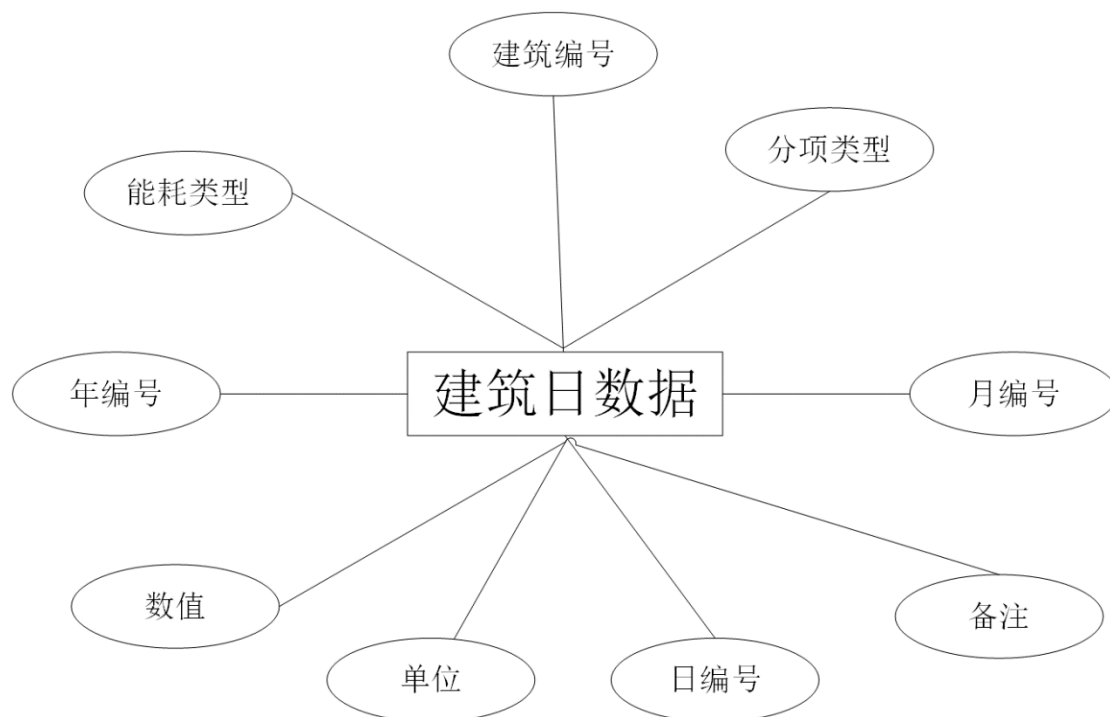


图 3.29 建筑日数据数据实体属性图

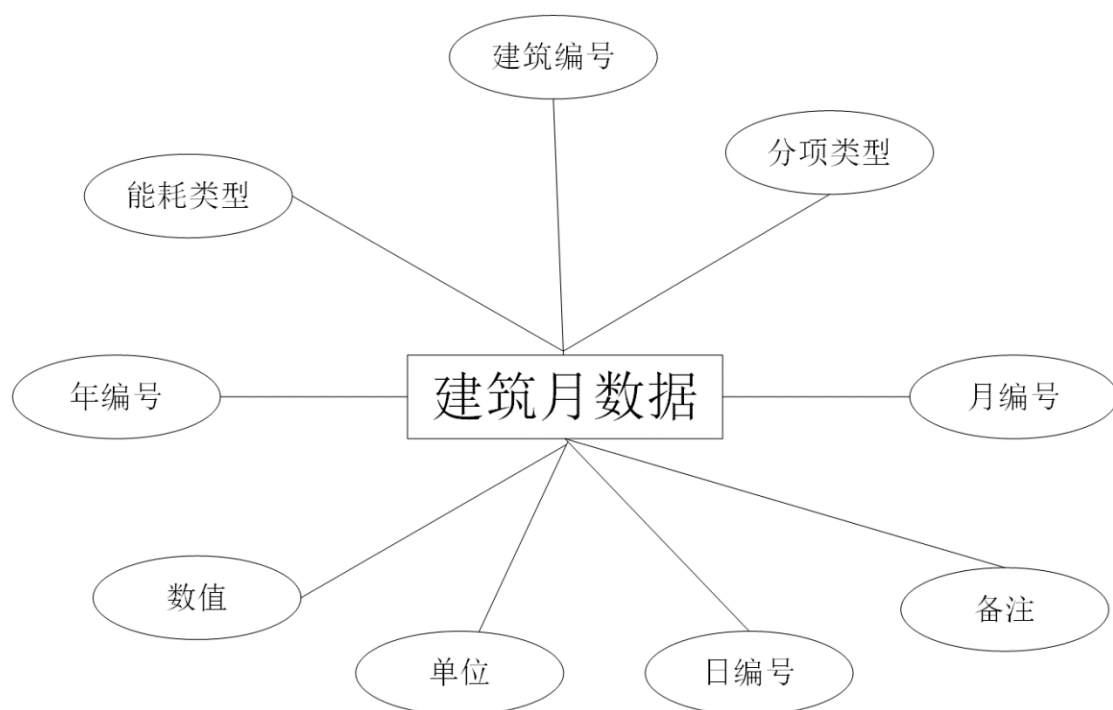


图 3.30 建筑月数据数据实体属性图

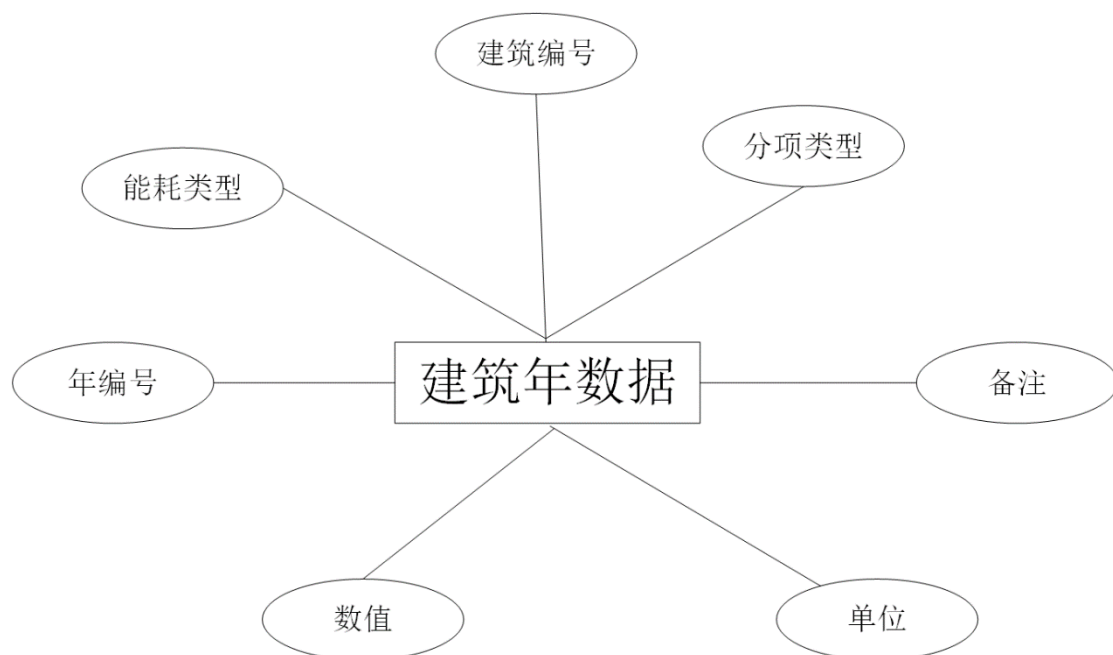


图 3.31 建筑年数据数据实体属性图

每个建筑拥有很多监测设备，比如“电表、水表、温度计”等。每个设备有若干个数据点位。比如一个电表设备就具有“电量、电功率、电流、电压”等多个采集点位。设备和数据点位对应的数据实体属性图如图 3.32、图 3.33、图 3.34、图 3.35 所示。

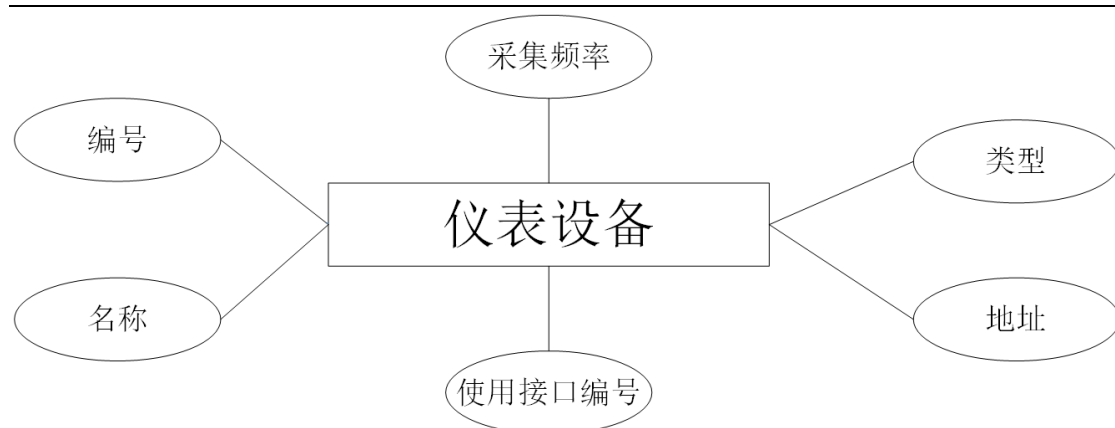


图 3.32 仪表设备基础信息数据实体属性图

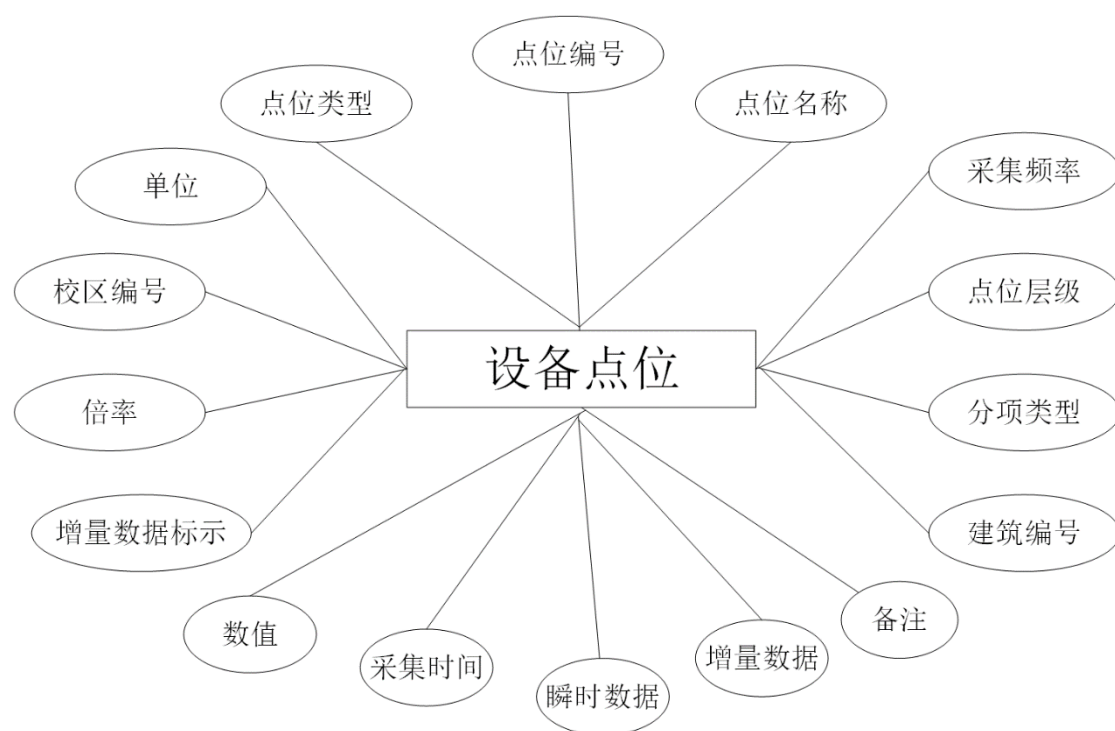


图 3.33 设备点位基础信息数据实体属性图

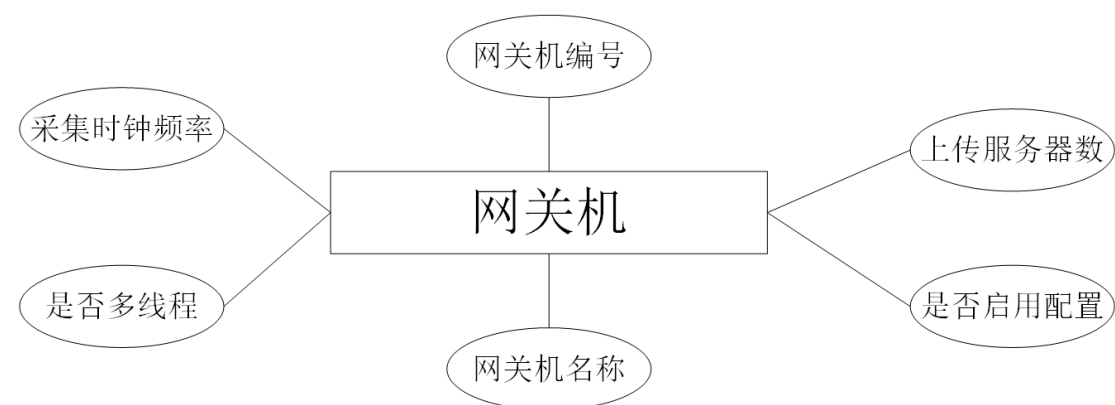


图 3.34 网关机基础信息数据实体属性图

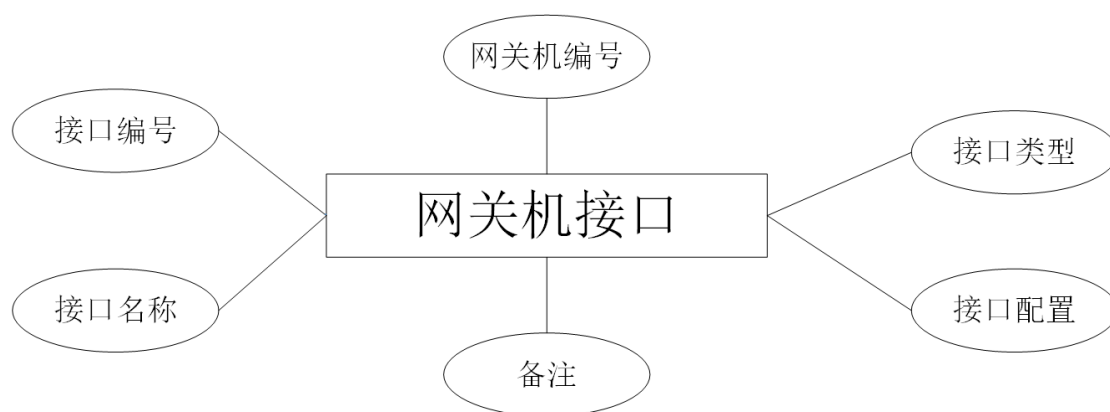


图 3.35 网关机接口基础信息数据实体属性图

每个点位也有一系列的配置信息和数据，包括网关机中使用的“点位缓存数据”、“网关机采集缓存数据”、“网关机采集缓存时间”、“点位增量数据”等。详细数据实体属性图如图 3.36、图 3.37、图 3.38、图 3.39 所示。

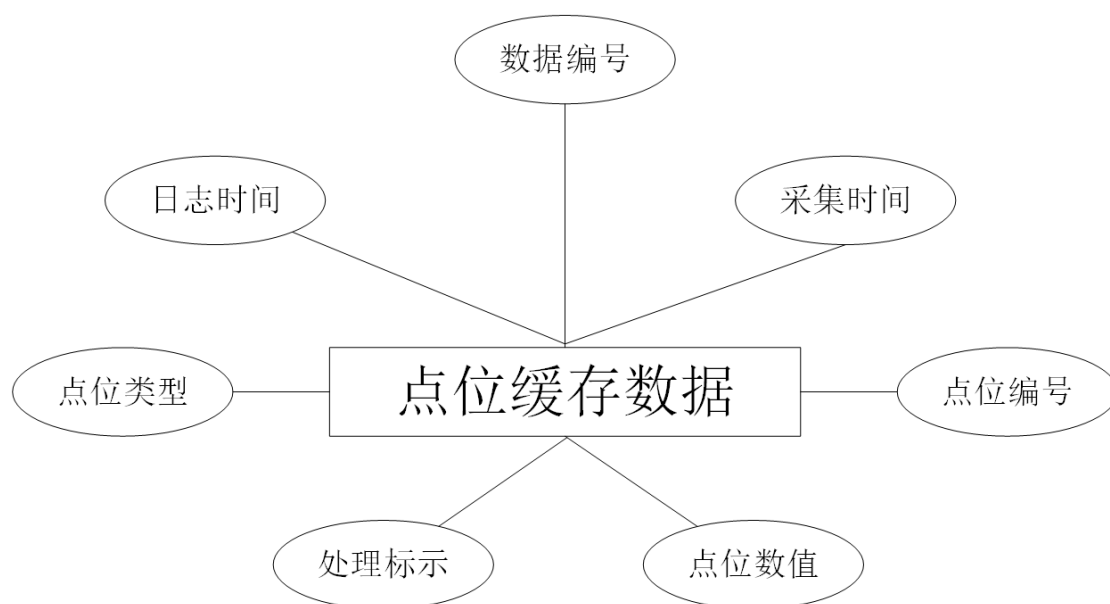


图 3.36 点位缓存数据数据实体属性图

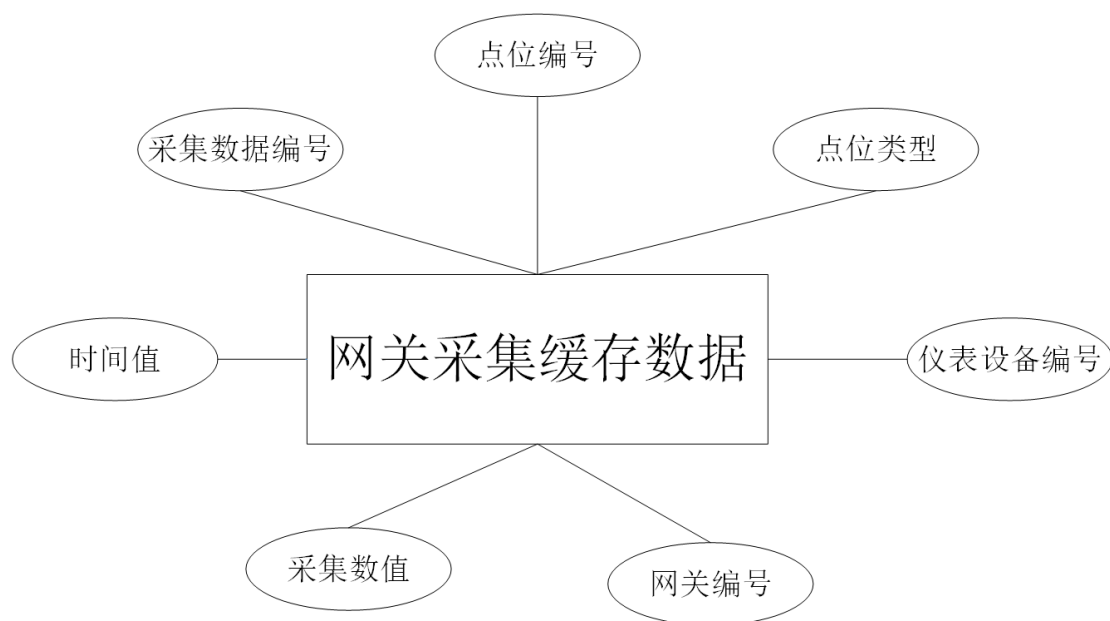


图 3.37 网关机采集缓存数据数据实体属性图

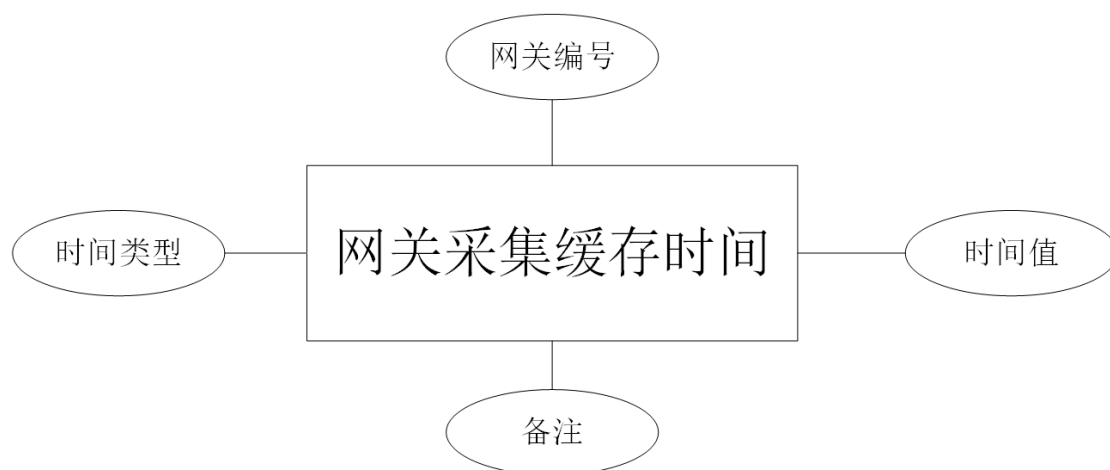


图 3.38 网关机采集缓存时间数据实体属性图

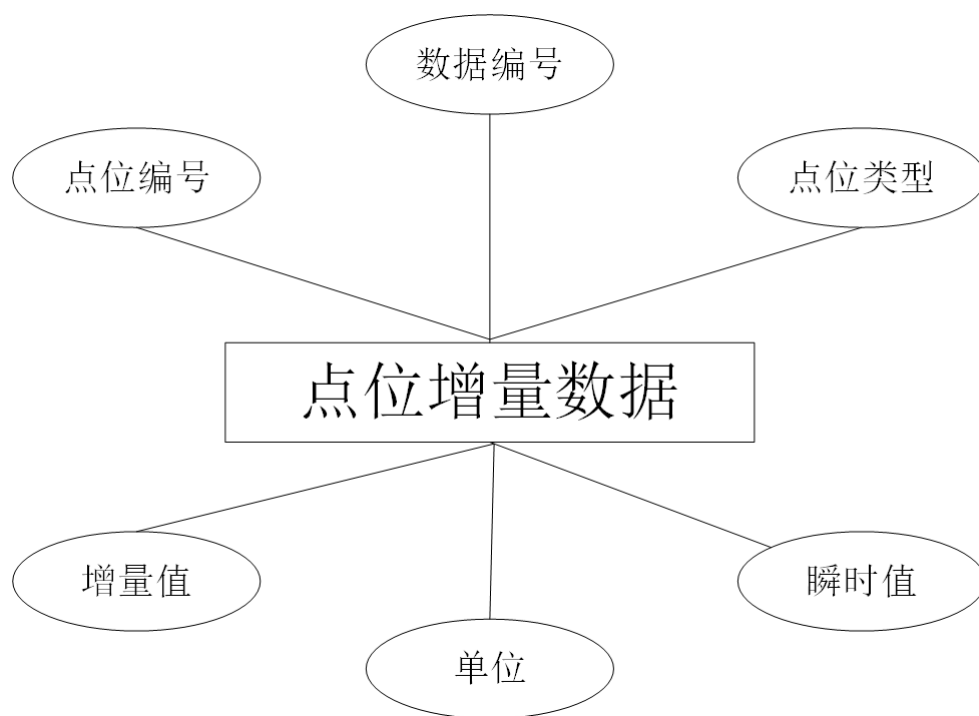


图 3.39 点位增量数据数据实体属性图

建筑和点位之间的关系影响着数据统计计算的结果。准确无误的建筑点位关系是平台统计分析的保障。一旦对应关系发生错误，那么有关的统计数据也就会发生偏差。而且点位之间也存在层级关系，总量点位下可以加挂分项点位。详细数据实体属性图如图 3.40、图 3.41 所示。

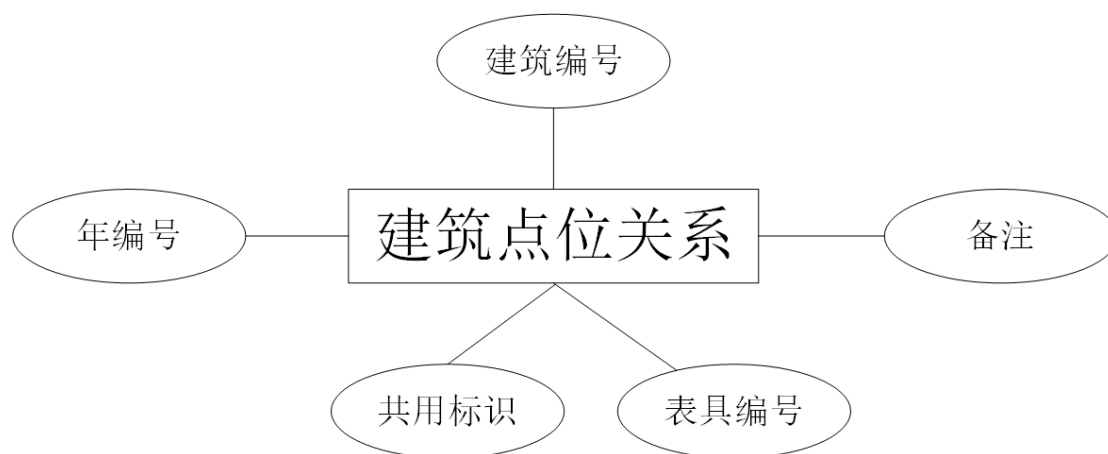


图 3.40 建筑点位关系数据实体属性图



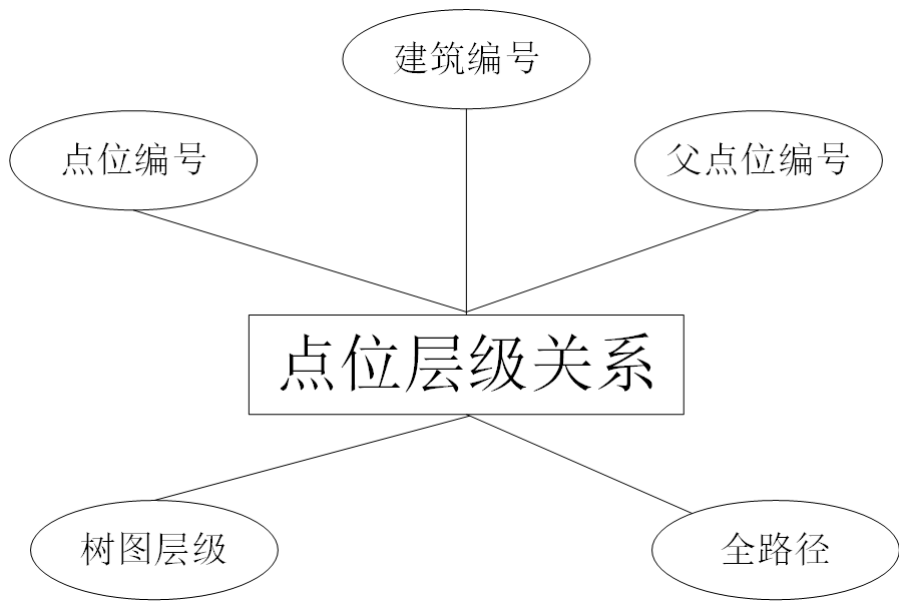


图 3.41 点位层级关系数据实体属性图

一所高校有若干校区，每个校区都有自己的基本配置信息，如图 3.42 所示。校区与建筑之间存在关系，通过建筑数据可以得出整个校区的能耗信息。

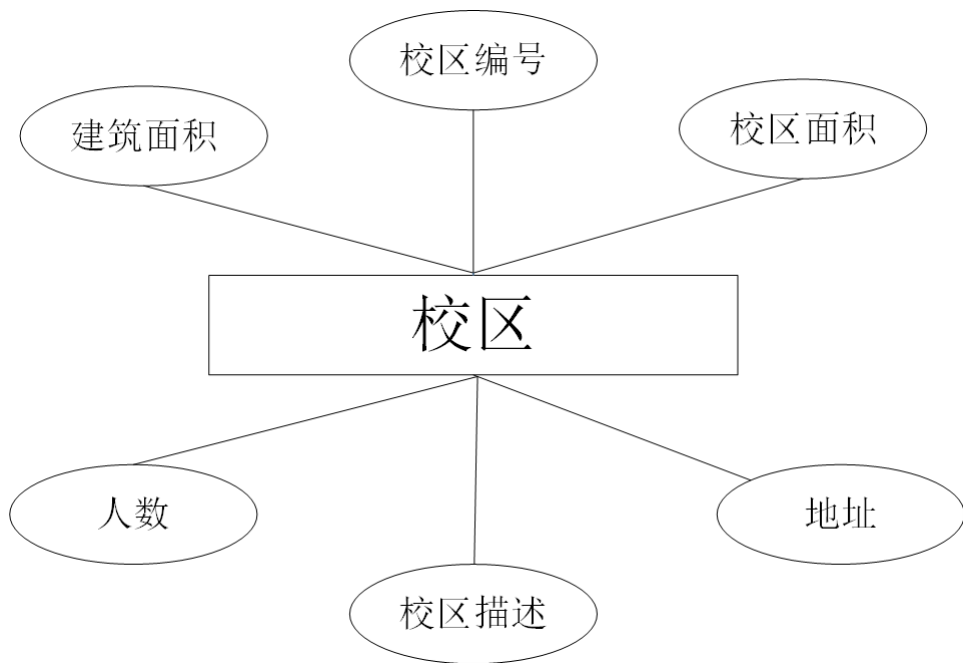


图 3.42 校区基础信息数据实体属性图

数据实体属性图介绍了各个数据实体具体的内容信息，而数据实体之间又存在非常复杂的对应关系。结合前面所有的数据实体对象，分析它们之间的属性关联，进一步进行建模，设计出对应的数据实体对象 E-R 图，如图 3.43。



### 3.3.2 数据预处理

为了满足高并发和快速效应客户端查询的要求,数据库采用“分类存储”、“分层汇总”、“提前计算”的思想,将数据分类处理成“原始数据”、“过程状态数据”、“结果数据”三类进行存储,以形成数据集市保证数据的响应速度。详细业务逻辑如图 3.44 所示。

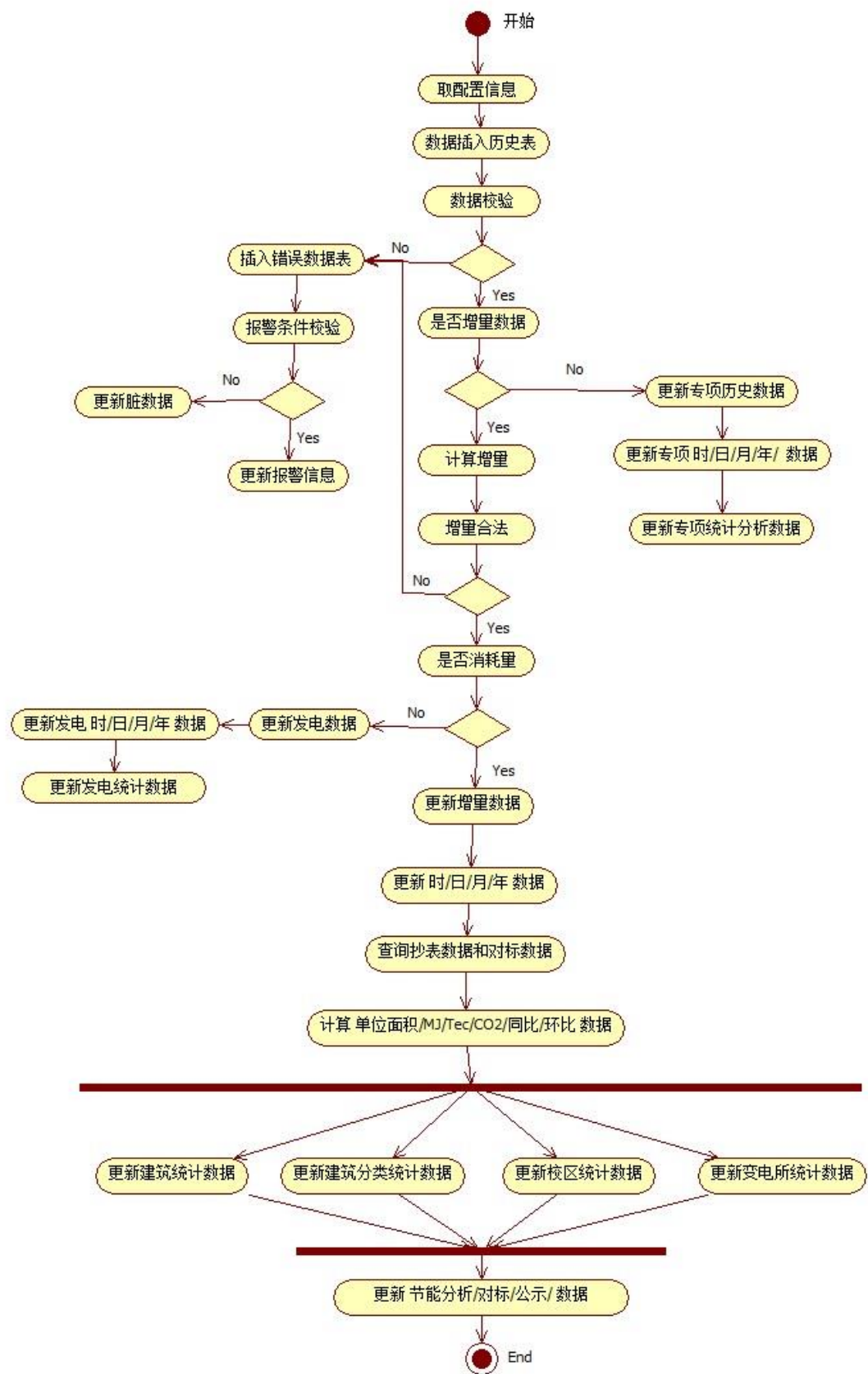


图 3.44 数据汇总处理活动图

3.3.3 数据表

数据表是关系数据库存储数据的主要对象，根据 E-R 图我们可以建立一系列的数据表。其中最具代表性的包括“建筑表”、“校区表”、“监测点位表”、“采集数据存储表”等。

建筑，是节能监管平台的主要监测对象。建筑对应的数据表如表 3.1 所示。

表 3.1 建筑信息数据表

字段名	数据类型	长度（B）	约束	默认值	描述
BuildingNo	char	20	PRIMARY		建筑编号
BuildingName	nvarchar	100	NULL		建筑名称
BuildingFunction	char	1	NULL		建筑类型
TotalArea	decimal	16	NULL		总面积
TotalPersons	bigint	8	NULL		总人数
RegionNo	nvarchar	100	NULL		校区编号
UploadFlag	int	4	NULL	0	上传标示
BigBldg	int	4	NULL	0	大型建筑标示
GreenBuilding	int	4	NULL	0	绿色建筑标示
Retrofit	int	4	NULL	0	节能改造标示
RNE	int	4	NULL	0	可再生能源标示

校园包含很多校区，每个校区也是重要的监测对象。校区信息数据表如表 3.2 所示。

表 3.2 校区信息数据表

字段名	数据类型	长度（B）	约束	默认值	描述
RegionNo	char	10	PRIMARY		校区编号
RegionName	nvarchar	100	NOT NULL		校区名称
RegionArea	decimal	16	NULL		校区面积
BuildingArea	decimal	16	NULL		建筑面积
PersonNum	bigint	8	NULL		人数

监测点位是获取监测信息的基本对象，它是系统基础信息的来源。监测点位信息数据表如表 3.3 所示。

表 3.3 监测点位信息数据表

字段名	数据类型	长度（B）	约束	默认值	描述
MeterNo	char	20	Primary		表具编号
MeterName	nvarchar	200	NULL		表具名称
MeterType	char	4	NULL		表具类型
Unit	char	10	NULL		单位
MaxQty	bigint	8	NULL		最大值
BaseQty	bigint	8	NULL		基础值
DAFrequence	bigint	8	NULL		采集频率
meterCategory	varchar	20	NULL		表具目录
meterUseType	varchar	20	NULL		表具分项类型
buildingNo	varchar	30	NULL		建筑编号
usePlace	varchar	30	NULL		使用地点
Rate	decimal	16	NULL		倍率
collect	int	4	NOT NULL		是否进行汇总计算
RTQty	decimal	16	NULL		当前采集数值
RTQty01	decimal	16	NULL		当前采集数值 2
RTime	datetime	8	NULL		当前采集时间
MStatus	char	2	NULL		表具状态
MAXIncQty	int	4	NULL		表具最大增量值
ELQty	decimal	16	NULL		上一次采集数值

存储数据信息的数据表有很多，这里举例建筑逐时分项数据表，它用来存储建筑每一个小时对应的分项能耗信息。具体的建筑逐时分项数据表如表 3.4 所示。

表 3.4 建筑逐时分项数据表

字段名	数据类型	长度（B）	约束	默认值	描述
-----	------	-------	----	-----	----

BuildingNo	char	20	Primary		建筑编号
UseType	char	4	NOT NULL		分项类型
LineType	char	4	NULL		回路类型
DADay	char	8	NOT NULL		日编号
DAHour	char	2	NOT NULL		时编号
SumQty	decimal	16	NULL		数值
Unit	char	4	NULL		单位
TCEQty	decimal	16	NULL		TCE 数值
CO2Qty	decimal	16	NULL		CO2 数值
MJQty	decimal	16	NULL		MJ 数值
Note	nvarchar	1000	NULL		备注
Lable	int	4	NULL		处理标示

数据库中还包含其它很多信息表,比如“设备信息表”、“表具关系信息表”、“原始采集数据表”、“专项系统监测数据表”、“数据处理中间数据表”、“数据增量表”、“监测信息逐年信息表”、“监测信息逐月信息表”、“监测信息逐日信息表”、“用户信息表”、“用户角色信息表”、“用户权限信息表”等。

### 3.4 本章小结

本章首先针对系统规模大及技术复杂度高的特点,对平台进行了整体分层的架构设计。进而又对“Mobile Hybrid APP、HTML5 Mobile Website、PC RIA Web Application、数据采集控制系统”四个主要的子系统进行了详细的架构设计。最后又对这四个子系统及数据库进行了详细设计。





