

分类号 TP391
UDC 620

学校代码 10590
密 级 公开

深圳大学硕士学位论文

面向移动应用开发的数据交互服
务框架的研究

蔡 笋

硕 士 类 别 工程硕士
领 域 名 称 软件工程
学 院（系、所） 计算机与软件学院
指 导 教 师 朱映映副教授

深圳大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文面向移动应用开发的数据交互服务框架的研发，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：

日期： 年 月 日

学位论文使用授权说明

本人完全了解深圳大学关于收集、保存、使用学位论文的规定，即：

- 按照学校要求提交学位论文的印刷本和电子版本；
- 学校有权保留学位论文的印刷本和电子版，并在校园网内提供检索与全文阅览服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存论文。

（保密论文在解密后遵守此规定）

论文作者签名：

日期： 年 月 日

导师签名：

日期： 年 月 日

摘 要

21 世纪后, 互联网企业如雨后春笋般的出现创造着形形色色的互联网神话。我国也紧追时代的步伐, 提出互联网+的概念。互联网企业出现更是呈现指数级爆发。与此同时, 如果在较短的时间内开发一个移动应用并展现其核心价值观和商业模式已经变得越来越重要并成为一个移动应用能够跟随网络潮流热点开发出相关移动应用的关键要素。

本文从以下几个方面入手, 结合理论方法与实践经验, 设计了一个适用于各种移动平台的数据交互服务的解决方案的框架, 通过实现本框架验证其可行性, 并结合应用开发实例, 阐述了本框架在现实领域中的应用能力和应用前景。

本论文首先分析总结了当前时期, 移动应用开发者在开发一个应用时最常遇到的问题, 即需要在某个话题还有热度和新鲜度的时候, 就能够结合自身独特的创意和方向, 制作相关的移动应用程序提供给相关的用户。

然后, 第二部分本论文提出一个数据交互服务的框架来解决这个问题。此框架分为两个部分, 首先是内容管理系统, 内容管理系统方便用户录入或者和导入数据, 开发者只需要利用辅助管理工具, 设计基于 XML 模型的数据业务结构, 即可自动生成内容管理系统。第二部分则是通过内容管理系统自动生成面向各个移动开发平台的开发包, 直接让移动开发者在自己的 APP 内调用相关的数据接口, 实现数据交互功能。

实现了此框架后, 本文基于数据交互服务框架实现了一个典型的 Android 新闻应用, 验证数据交互服务框架可以为开发提供快捷和便利的开发过程, 以及简易的二次开发扩展。让开发者和用户感受到只需要写少量的代码即可完成系统设计, 使得开发者可以专注于核心开发任务。

通过一系列的分析研究与实践, 本文提出的面向移动应用开发的数据交互服务框架是现实可行的解决方案之一。其实现过程简单, 易于定制和优化, 容易学习和掌握, 可以帮助开发人员提高开发效率, 并简化代码管理和维护。本开发框架具有可扩展性, 可接入几乎所有的移动平台, 实现起来相对简单, 维护成本较低, 为移动应用开发者提供了一个可行的数据交互解决方案。

关键词: 数据交互 互联网+ 移动应用开发 集成式开发

Abstract

Since 21th century, the appearance of internet enterprises that have sprung up like mushrooms has created various internet myths. China also follows the pace of the times and put forward the concept of internet+. The arising of internet enterprises has grown exponentially. Meanwhile, it becomes more and more important to develop a mobile application and display its core value and commercial mode in the shortest possible time. Mobile application is a critical element to develop relevant mobile application following internet fashion trends.

Combined with theoretical method and practical experience, a solution framework of data exchange service applied to various mobile platform data is designed in this article in terms of the following aspects. Through realizing this framework and verifying its feasibility, and combination of practical examples of application and development, this article illustrates the applied ability and promising application of this framework in realistic realms.

Analysis and conclusion are made firstly in this paper that mobile application developers have encountered the most common problems when they develop an application at present, that is, combined with their unique creation and direction, they should make some relevant mobile application program for relative users when some topic is still hot and fresh.

Secondly, a framework of data exchange service is put forward in this paper to solve this problem. It is divided into two parts. The first one is content managing system. When users input or import data, developers enable to make content managing system generate automatically only by utilizing assistant managing tool and designing business data structure based on XML model. The second one is about producing development package automatically facing to various mobile development platform through content managing system, and the developers could transfer relevant data interface in their own APP to realize the function of data exchange.

Based on this framework, a typical Android news application is realized in this article. Through development process, it is verified that this framework could provide convenience

and quickness, and simply secondary expansion for development. This application is to allow developer and reader experience the completion of system design only by writing a little code, thus users could focus on relative tasks upon development.

By a series of analysis and test, it is concluded that data exchange service framework facing to mobile application and development is one of feasible solutions in reality in this article. With simple process, easy customizing and optimizing, easy learning and mastering, this framework helps developers improve development efficiency and simplify code management and maintenance. Besides, this development framework could be expanded to switch in almost all mobile platforms, with relatively simple operation and low maintaining cost. It provides mobile application developer with a superior solution of data exchange.

Key word: Information Management System, Internet Plus, Automated Build, Application interface

目 录

摘 要	I
Abstract.....	II
第 1 章 绪论	1
1.1 本课题的研究意义	1
1.2 主要研究内容	1
1.3 论文结构	3
第 2 章 相关技术	4
2.1 数据交互服务系统的技术分析	4
2.2 移动应用开发技术分析	4
第 3 章 数据交互接口后台的设计与实现.....	6
3.1 业务数据管理系统功能需求	6
3.2 辅助工具功能需求	7
3.3 系统模块功能设计	8
3.4 数据库总体设计	9
3.5 系统部署架构	9
3.6 数据交互接口平台总体设计	10
3.7 XML 模型读取模块详细设计	11
3.8 业务模型模块详细设计	12
3.9 数据交互接口访问模块详细设计	13
3.10 XML 模型管理模块详细设计	14
3.11 XML 模型读取模块编码实现	15
3.12 业务模型编码实现	16
3.13 数据交互接口访问编码实现	19
3.14 XML 模型管理模块编码实现	20
3.15 小结	21
第 4 章 移动端数据交互组件的设计与实现.....	23
4.1 移动端数据交互组件详细设计	23

4.2 移动端数据交互组件编码实现	25
第5章 数据交互服务框架的应用	27
5.1 研究环境安装与配置	27
5.2 数据交互接口端的设计	27
5.3 远程数据数据访问端的设计	33
5.4 安卓平台新闻移动客户端的设计	35
5.5 小结	37
第6章 总结与展望	39
6.1 研究工作总结	39
6.2 后续工作展望	40
参考文献	41
致谢	42

第1章 绪论

随着互联网产业越来越发达和移动互联网的崛起，互联网产业再次迎来一次井喷。据统计，世界上在不同形式中使用互联网的人已经超过百分之八十，并且这个数字还是持续性增长的。在我国，互联网用户已经达到了 8 亿以上。随着互联网产业的现象，我国提出互联网+的概念，并且基于此概念推出“大众创业，万众创新”的观点。因此，越来越多的人加入互联网+产业的大军进行创业。

1.1 本课题的研究意义

进入 21 世纪后，互联网对现实生活产生了一系列的革命，把人们将生活带入到互联网中。在 21 世纪的前 10 年，人们已经习惯于坐在个人电脑前进行各种各样的生活，例如交友，工作和购物等^[1]。在第一波的互联网浪潮过去后，移动互联网出现了，它更是像一个巨大的浪潮般将人们的生活完全吞噬，把人们在日常生活中的种种活动和所需的带进了移动互联网中，利用网页，移动终端或者微服务等^[2]。人们的日常生活已经离不开互联网。与此同时，我国紧追着时代的步伐，与时俱进地提出互联网+的概念，互联网+是把互联网产品和人们生活结合而出现的一个新的产业，在此基础上更是提出“万众创新”的重要概念，为互联网+的创业者们提供巨大的支持^[3]。

那么，对于每一个互联网产品，其重中之重的一个首要任务就是构建其核心的信息管理服务系统，并且基于此系统，为各个终端例如网站，移动终端或其他微服务等提供数据或交互。那么，如何快速地把这套核心基础的信息管理构建出来就成为了争分夺秒的关键了。快速简单的为系统构建基础数据的数据交互接口平台将会为新兴的互联网+企业创业者带来速度上的优势，并且结合其自动数据接口服务功能，更是为其把相关的移动互联网终端紧密连接，最终快速，简洁和高效的为互联网产品提供基础服务，节省了在基础数据的数据交互接口平台和其相关的数据接口的管理时间^[4]。

1.2 主要研究内容

如何为移动应用创业者提供快速，简洁和高效的互联网产品核心的数据交互服务框架是本课题的主要关键，尤其是针对于以产品理念为驱动的移动应用和开发水平相对普通的开发者。本课题将基于面向移动开发应用的数据交互的核心问题，提出一个为数据

关系和数据业务模型的解决方案以及针对移动应用快速可以与数据业务模型和数据交互的解决方案的一整套框架。

本课题的主要设计任务分为两方面，首先本论文会提供一个可继承性并且面向二次开发的数据交互接口平台，在此系统中，用户只需要利用 UI 工具来定义基于 XML 模型相关的数据业务对象，则自动生成一个基于增删查改的数据交互接口平台，并且基于此系统开放 SOAP 协议的数据交互服务提供给相关开发的移动应用使用。相关的开发技术是基于 Apache 的 PHP 开发技术。

第二部分则是需要为移动开发者提供可以调用数据交互服务的类库，开发者通过引用相关的核心代码类库，并且添加相关的继承核心代码类库的业务数据对象（Business Data Object）和数据访问对象（Data Access Object），即可以直接访问远端的数据并且展示在移动应用中。本课题前端移动开发技术支持 Android 和 Cordova 技术，所以需要进行基于 Android 的 JAR（Java Archive, Java 归档文件）包以及基于 JavaScript 的类库的开发。

本人在此系统的中承担所有角色的任务，包括系统的设计，开发和测试等所有工作，并撰写本论文对研究内容进行阐述。

首先，在可行性分析方面，由于有众多客户的支持和长久以来为客户提供服务的基础，原来的一套从无到有来构建一套应用系统的方式已经跟不上服务态度与服务质量。开发这一套新型并且快速构建修改的系统已经成为了迫切的需求。由于有需要为客户提供业务服务和为新客户提供最快速的技术支持，本人连同老客户已经新客户加上长期的业务系统开发经验，总结业和技术难题进行可行性分析。然后根据可行性分析、客户调研以及技术分析的总结，获得需求说明书用于对整个系统的业务功能和业务规划的一个指导和说明。然后总结技术，把相关的为客户的使用的代码抽取出公用和通用的类库，用于对数据交互接口平台的支持和让其可以快速为周边的移动客户端提供服务。

在此过程中，本人将承担概要设计，详细设计，实现以及测试的一系列工作和任务。并且最终把相关的产品规划及实现过程在本文中得以体现。通过概要设计，把产品的重要轮廓描述和设计各个模块。然后通过详细设计优化产品的各个模块的功能和接口，最终通过编码来实现本论文所阐述的系统。

然后本论文将会基于本论文所开发的系统来开发一个简单新闻管理系统，用以验证实用本产品系统开发一个互联网产品的快捷性和便利性以及实用性。

1.3 论文结构

本论文首先分析和结合了国家提出互联网+概念下，移动应用开发者如何能够在市场中快速的利用自己的创意和商业模式为用户提供移动平台应用服务，对此本论文提出一个快速，高扩展性，低成本的数据交互服务框架来解决开发者的移动应用开发过程中遇到的问题，以此来节约开发中繁冗的基础数据服务器的设计时间。

第二部分本论文提出一个数据交互服务的框架来解决这个问题。并且本论文将会详细地阐述相关的使用技术和原因，然后详细地把框架的开发过程为读者进行描述。总的来说，此框架分为两个部分，首先是内容管理系统，开发者只需要利用辅助管理工具自动构建其设计的业务模型对象的内容管理系统，用户则可以直接对此设计的业务数据进行数据录入或导入。第二部分则是通过内容管理系统自动生成面向各个移动开发平台的数据接口调用开发包，直接让移动开发者在自己的 APP 内即可调用相关的数据接口来实现数据交互功能。

实现了此框架后，本论文则基于此框架实现一个典型的 Android 新闻应用，通过开发过程来验证本框架为开发提供的快捷性和便利性以及简易的二次开发扩展。让开发者和读者感受到只需要写少量的代码则已经完成了系统的设计，用户可以专心地把开发工作放在相关的核心价值上。

最后，本论文根据框架开发过程中研究的结果和应用开发过程中的体验，进行本论文课题研究成果的总结，让开发者和读者体验到实现过程简单，易于定制和优化，容易学习和掌握，帮助开发人员提高开发效率并简化代码管理和维护。同时，本论文对本框架存在的一些问题进行分析以及未来的发展情况做展望。

第 2 章 相关技术

介绍论文中用到的所有重要技术，对其原理及使用方法做简单介绍

2.1 数据交互服务系统的技术分析

由于 PHP 已经成为主流 Web 开发者使用的语言之一，所以本文的技术基于 PHP+APACHE 进行设计。基于此基础上，选择以继承 MVP (Model-View-Presenter) 的概念的方式来进行设计。但是因为为了尽量要把业务功能抽离，所以最终只保留用户直接控制 MVC 中的 M(模型)。作为一种新的模式，MVP 与 MVC 有着一个重大的区别：在 MVP 中 View 并不直接使用 Model，它们之间的通信是通过 Presenter (MVC 中的 Controller)来进行的，所有的交互都发生在 Presenter 内部，而在 MVC 中 View 会直接从 Model 中读取数据而不是通过 Controller。而在本框架中，用户只需要定义自己基于 XML 格式的业务模型对象则可以自动的生成一个基于增删查改的业务模型对象出现在内容管理后台中，并且直接提供以 SOAP 方式访问的 API(Application Interface)。为了更加方便用户直接设计 XML 模型对象，本系统同时基于 .net C# 开发一个数据业务模型的管理工具，用户可以在可视化的方式下直接定义其相关的业务模型。PHP 技术方面则是使用 Smarty 模板技术和本论文作者自己开发的代码框架^[5]。

在数据库方面，本系统默认提供三种数据库，以简单工厂的方式继承，所以只需要修改配置文件即可以选用自己合适的语言。

正如前文所说，本系统是具有高继承性和可二次开发的。如果用户有特殊需求，则可以基于本系统进行开发。对于有特殊数据模型业务需求的用户来说，则需要掌握 PHP 来控制数据模型在数据库存储过程中发生的业务，简单的方法就是继承通用的模型对象重写相关的增删查改的方法以拥有自己的特殊数据，同时，也会直接反映到 API 接口中。

如果开发者需要针对数据交互接口平台前端操作时定义自定义操作业务，则需要掌握 JavaScript，重写相关的接口，例如保存后调用的空方法等。同时需要掌握 Bootstrap3.0 框架来进行前端页面对象的操作。

2.2 移动应用开发技术分析

通常来说，现在的移动开发平台主要分为 3 个，Android，iOS 以及 Windows Phone。

由于 Windows Phone 平台相对闭源，现主要是以 Android 和 IOS 平分天下。从移动开发模式来说，则分为原生(Native App)，纯 Web 以及混合模式（Hybrid），以下图 1 为技术上的比较。

	Native	Html5	Hybrid
APP 特性			
图像渲染	本地API渲染	Html, Canvas, CSS	混合
性能	快	慢	慢
原生界面	原生	模仿	模仿
发布	App Store	Web	App store
本机设备访问			
照相机	支持	不支持	支持
系统通知	支持	不支持	支持
定位	支持	支持	支持
网络要求			
网络要求	支持离线	大部分依赖网络	大部分依赖网络

图 1 主流的移动开发模式对应图

由于本论文作者开发水平有限，所以暂时选用了 Android 平台的原生开发以及混合模式下通用平台的 Cordova 框架进行移动应用端的核心服务框架的技术进行开发^[6]。

在这两种框架中，本论文依然支持把数据服务接口的数据纯粹在本地手机中，Android 平台上使用 Sqlite 进行数据存储，Cordova 方面则是基于 HTML5 协议的本地数据库（Web SQL Database）。

在本系统开发的 UI 工具中，本论文同时自动生成相关的代码，比如业务数据对象代码，此对象定义了业务数据对象字段；数据访问对象代码，对数据业务对象进行本地化的增删查改以及按需要的情况下同步到服务器的数据中。尽量地减少开发者的数据业务代码的开发，使得用户可以更加专注于对用户交互已经核心价值的体现。

第3章 数据交互接口后台的设计与实现

在本课题中，本论文面向的框架使用者主要是将会在互联网+产业中创建新产品的开发者或者设计师，他们可能会创建各种移动互联网的平台、服务或者工具等产品。在本论文的设计中，面向的用户将有一定的技术背景，他们至少需要能够对进行各个移动开发平台的部分有一定的熟悉，本课题的研究成果主要是为了能够解决移动互联网产品开发者在系统开发过程中经常遇到的问题，针对这些问题为他们提供简便的方式以自动生成核心的业务数据管理系统^[7]。对于这些客户，本论文要为他们解决以下几个问题：

1)快速构建一个业务数据管理系统，其业务数据应具有简洁高雅的用户界面设计，快速的页面加载速度和方便简易的操作方式，高通用性和易拓展能够简易进行二次开发。

2)基于业务数据管理系统的自动数据生成接口，在此接口中，可以直接通过调用业务系统中相同的链接，基于不同的参数，获得相关 SOAP 协议返回的 XML 数据。

3)基于业务数据接口，本论文需要在各种终端中进行客户端的开发，例如 Android 端和 Web 端等。本课题将会基于 Android 平台提供核心的 Jar 包供用户快捷简单的获取接口数据，做异步数据交互读取的新闻系统的开发。

4)配套的辅助生产工具，例如辅助生产 XML 驱动模型，自动连接数据库进行数据库表的构建等一系列通用功能，更敏捷的辅助用户进行开发。

3.1 业务数据管理系统功能需求

为了能够服务于互联网+产品的开发者，本课题将会设计一套让开发者快速便捷就能使用的业务数据管理系统。此业务系统可以让用户只需要配置相关的 XML，用 XML 来标记其产品系统中的业务实例，然后系统中设置对应的请求页面，使请求的页面知道应该去读取哪个相关的 XML 数据业务模型。之后当用户用网页打开此配置好服务器的 PHP 页面后，则自动能够提供对该业务数据对象并提供基础的增删查改的功能^[8]。以下是功能简单说明：

1)用户定义相关的符合定义的 XML。那么什么是 XML 数据模型的定义，例如是否在列表搜索中显示某个字段或者某个字段是否可以作为搜索字段被搜索。又例如定义字段类型等，让用户在单条数据页面编辑页面输入相应的数据，例如文本，数字，图片亦

或是下拉列表或者多选框等。

2)针对用户定义的 XML, 然后通过 PHP 语言的解析, 然后在页面中显示列表页面, 编辑页面, 添加页面, 删除功能等相关页面或者操作。相关增删查改等功能则能够在 XML 定义并直接表现在相关页面中。

3)本系统应该具有高可扩展性, 应该从几个方面可让用户拓展。首先, 在前端页面显示中应该是可拓展的, 用户可以自己添加 JS 或者 CSS 从而增加 UI 上操作的通用逻辑。例如输入某个地址后, 编写代码去百度地图返回相关的位置坐标数据。其次, 在后端数据处理代码, 也就是 PHP 代码中继承的模型对象中, 可以编写相关的业务逻辑。例如用户可以根据自己的实际情况判断某个字段是否允许被录入, 或者检验数据的正确性或者其他相关的业务流程等。

4)本系统支持多种主流数据库, 包括 MSSQL, MYSQL, SYBASE, ORACLE, SQLITE 等。由于各个系统支持的语法略有分别, 所以将采用简单工厂模式的方法来生成需要的数据库以方便用户可以根据自己系统的情况选择相应的服务器。

5)本系统支持各种前端 UI 插件的拓展, 例如拓展添加日期控件 FullCalendar 等插件。也支持后端各种插件的拓展, 例如添加发送短信功能, 读取 Excel 功能等。

6)系统将会首先预定义用户管理的增删查改的定义, 以便用户学习并且可以快速进入系统的二次开发方式中。

7)系统将在增删查改的基础上, 提供数据接口交互服务, 用户可以通过该定义的 PHP 页面请求数据以 XML 的方式利用 SOAP 协议返回数据, 供各个终端业务系统返回相应的数据。同时, 用户可以自定义相关的 SQL 语句返回给终端以进行对数据交互接口数据的二次开发。

8)系统支持多语言的, 换句话说, 就是相关的文本或者长文本字段支持用户提交多语言。在系统中将以 1 对 N 的表结构映射到多语言表中。

3.2 辅助工具功能需求

为了能够让用户更少的输入代码和更快捷的使用本系统构建对应的自动化业务数据管理系统, 本课题中将会同时提供相关的辅助工具, 主要包括以下几方面:

- 1) XML 模型生成工具, 通过有用户界面的 UI 工具, 用户可以创建一个 XML 模型, 并且利用该辅助工具编辑模型的相关字段以及字段的相关功能需求等。最后生成到系统中直接使用。

- 2) 数据库表生成工具，用户可以通过导入模型的方式，根据其已经创建的字段和字段类型，自动在数据库中更新或者新增相关的表和字段，减少用户的工作量，协助用户更快速度的生成一套业务系统。
- 3) 为了能够让开发者减少代码开发工作并提高标准代码统一接口，辅助工具同时需要能够为前端数据读取组件生成相关的核心代码以及业务数据模型对象和相关的业务模型访问接口。

以下图 2 为框架功能模型。

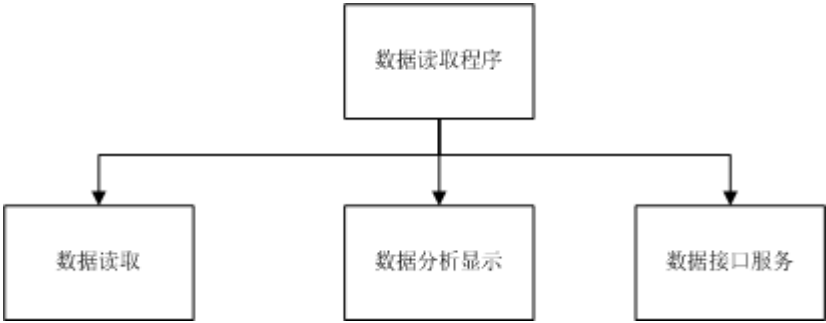


图 2 框架功能模型

3.3 系统模块功能设计

这个是整个业务系统的核心模块，其通过读取开发者定制化的 XML 业务模型，为其用户自动生成一套业务数据管理系统，并最终构建整个业务系统和为其用户提供对业务数据的增删查改功能^[9]。表 1 至表 5 列出了几个重要模块的模块以及描述：

表 1 XML 模型读取模块功能表

模块名称	XML 模型读取模块
功能描述	本模块主要通过自动读取 XML 业务模型，自动生成业务模型对象，后续系统可以通过解析业务模型对象把相关的业务操作界面展示给用户使用
接口与属性	接口需指定当前的业务对象需要读取哪个 XML。

表 2 业务模型模块功能表

模块名称	业务模型模块
功能描述	本模块主要通过加载已经生成的业务模型对象，然后根据用户提供的业务请求，展示不同的增删查改
接口与属性	用户通过传入业务模型对象以及相应的请求动作，页面自动返回相应的处理动作，至少包

	括：列表及可搜索字段，单条业务数据新增和编辑保存，批量业务数据删除，字段视图列表
--	--

表 3 数据交互接口模块功能表

模块名称	数据交互接口模块
功能描述	本模块主要通过加载已经生成的业务模型对象，或者用户自己生成的业务数据，自动生成 SOAP 协议的 XML 文件流返回到页面中让不同业务中断读取
接口与属性	用户通过传入业务模型对象以及相应的请求动作以及必须的请求参数，开发者可以返回增删查改形式的基本 XML 数据，也可以根据用自己组织业务数据，自动返回用户特殊的业务数据请求

表 4 XML 模型管理功能表

模块名称	XML 模型管理模块
功能描述	提供一个 UI 工具，是的用户可以快速根据自己的需求新增或者编辑 XML 模型，并自动更新数据库的表对象
接口与属性	指定 XML 文件名字名，并且添加模型相应的业务字段

表 5 移动终端数据交互模块功能表

模块名称	移动终端数据交互模块
功能描述	主要功能是指向相应的业务数据对象接口，然后返回有内容的业务数据对象或者返回调用接口的结果（例如保存数据，返回成功）
接口与属性	指定需要请求的 API，设计业务数据对象参数，则自动返回业务数据对象让用户在不同的移动终端上显示和操作数据

3.4 数据库总体设计

本课题设计的系统不涉及数据库设计，因为主要是通过模型驱动的方式为用户把他们实际的数据存入数据库。但是必须遵循本系统定义的数据设计。如果用管理工具辅助生成数据库表，则自动生成基础表结构的内容以及用户自定义的字段。

3.5 系统部署架构

由于本系统基于 PHP 开发，所以首选部署于 Linux 操作系统的 Apache 服务器中。必

须开通的模块包括 MBSTRING, SOAP, SimpleXML, XMLREADER 等模块。

由于本系统支持多系统, 所以只要 APACHA 中开通相应的数据库模块即可。

推荐部署架构如图 3:

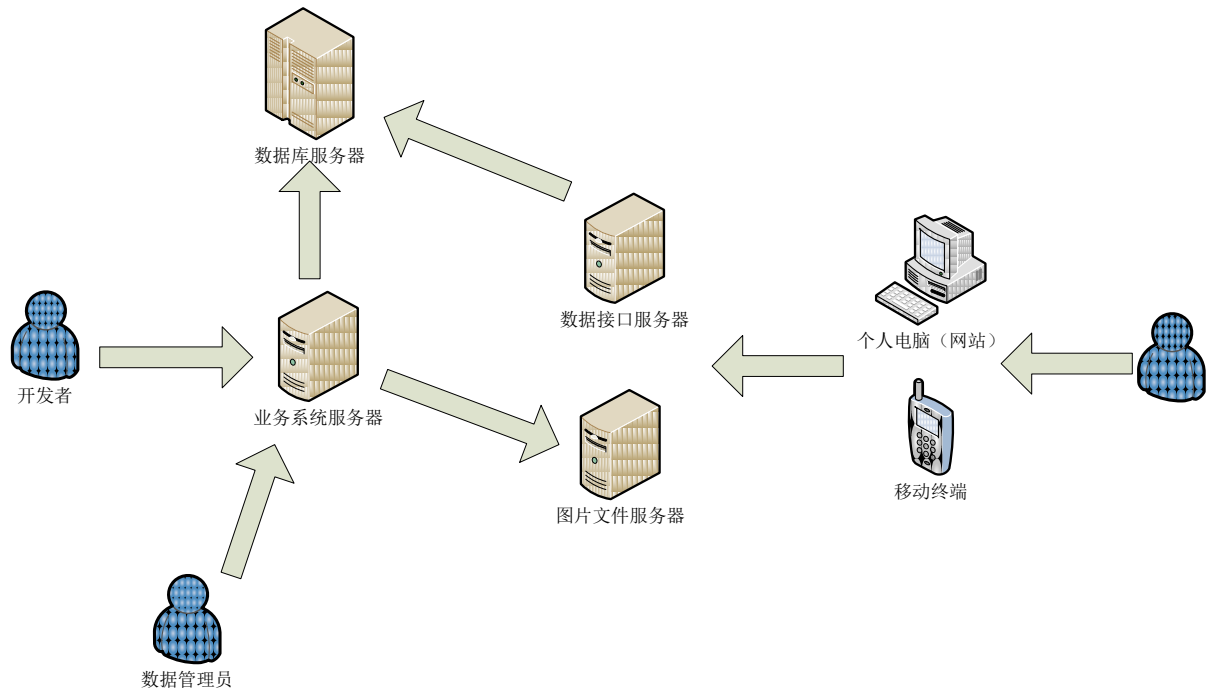


图 3 系统环境部署架构模型

3.6 数据交互接口平台总体设计

这个模块主要用于读取用户已经生成的 XML 并将其转化为相关的业务模型对象, 便于以后针对不同页面进行请求以展示不同的结果。在系统的实现中, 本论文首先遵循 MVP 的设计模型。众所周知, M 为模型 (Model), V 为视图 (View), P 为控制器 (Presenter)。作为一种新的模式, MVP 与 MVC 有着一个重大的区别: 在 MVP 中 View 并不直接使用 Model, 它们之间的通信是通过 Presenter (MVC 中的 Controller) 来进行的, 所有的交互都发生在 Presenter 内部, 而在 MVC 中 View 会直接从 Model 中读取数据而不是通过 Controller。本论文这个数据交互接口平台在业务管理模块端则一样是基于 MVP 模型的设计规范, 首先本论文通过用户设计一个业务模型对象, 此业务模型对象通过读取用户 XML 文件中对 XML 模型的定义, 最终生成模型, 然后通过控制器加载相关的业务模型对象到业务视图中展现到用户面前并操作进行增删查改^[10]。

全局设计图 4 如下:

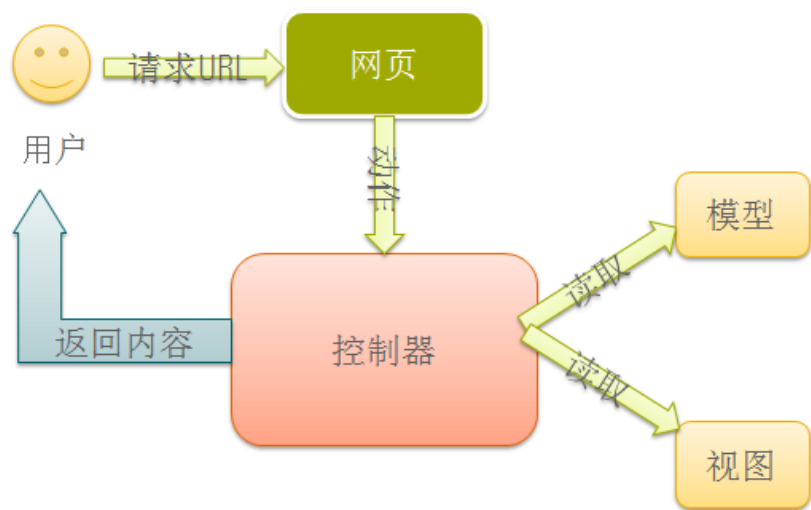


图 4 内容管理系统模型流程

3.7 XML 模型读取模块详细设计

首先必须是系统能够读取 XML。在本系统中，将会设计一个类，类专门读取 XML 文件，然后把其数据加载到相关的属性上，最终生成业务模型对象。通过读取此类，本论文可以获悉相关的数据源（通常为表或者视图）以及相关的字段类型和字段属性，最终通过处理，本论文这个业务模型对象可以被业务系统生成器使用，最终生成相关的业务。本论文尽可能通过查看 XML 模型和修改 XML 模型则可以了解和修改网页上不同的动作请求生成不同的页面字段。类设计如图 5：

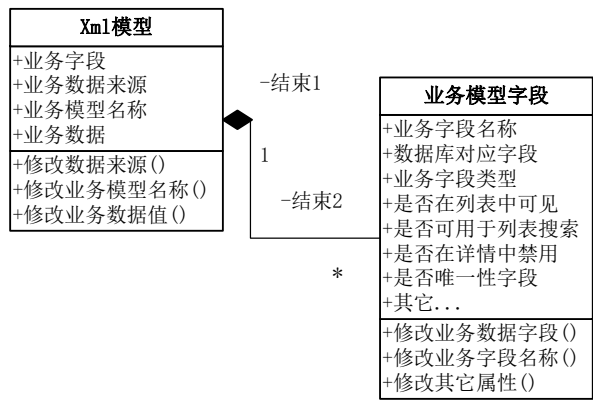


图 5 XML 模型接口设计

同时，某个业务字段可能本身就是要去读取业务数据或者根据特殊业务需求动态修改显示的数据，显示的字段或者其他业务需求等。在此，本论文可以添加一个接口，让用户直接修改业务模型对象及其相关数据。用户可以实现编辑接口的相关方法然后根据业务需求实时修改业务模型对象的内容和其相关的搜索值。接口设计如图 6。

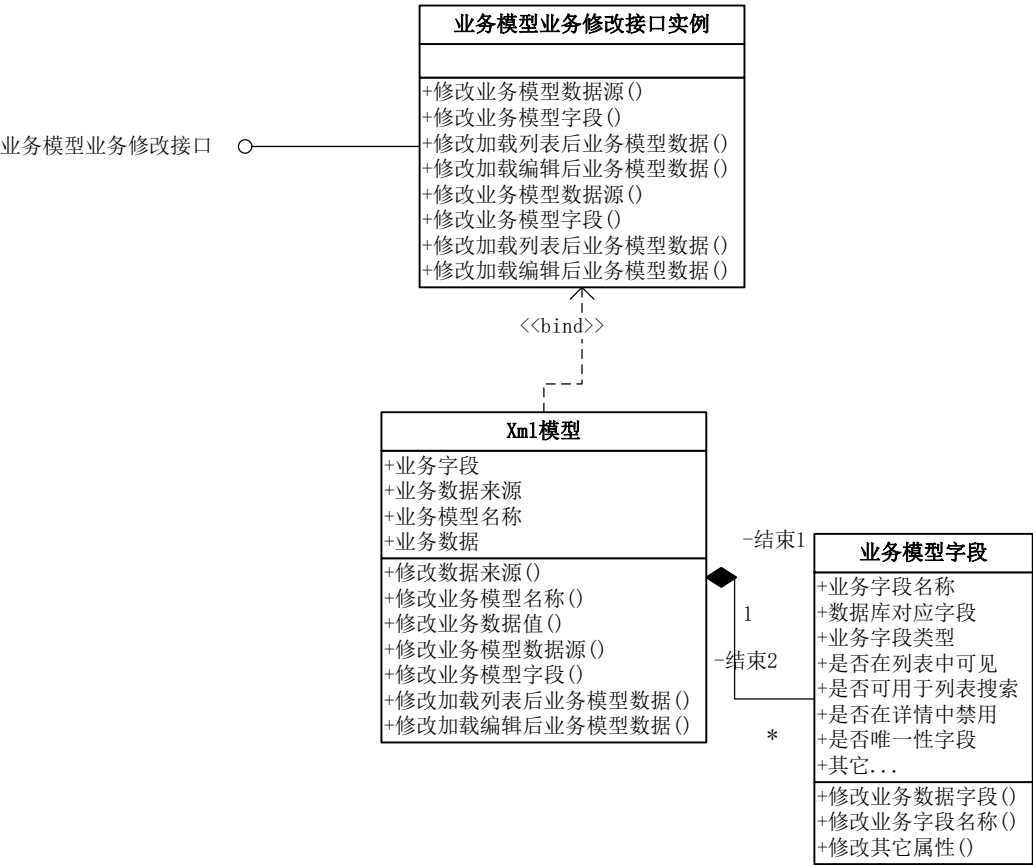


图 6 业务模型接口设计

3.8 业务模型模块详细设计

业务模型模块主要是为 MVP 设计模型的控制器和视图部分，简单的来说，用户通过调用页面并提交相关的动作请求，例如增、删、查和改等动作给控制器，然后控制器则主要负责控制需要如何加载业务模型对象到哪个相关的请求视图里去，然后把获得的页面源代码返回给用户使得用户能够进行增删查改的操作。根据这个动作，本论文可以设计出控制器类如图 7：

控制器对象
+业务模型对象 +请求的动作 +数据库对象 +业务模型对象修改接口
+获得列表视图() +获得编辑界面视图() +获得新增界面视图() +获得搜索数据表格视图() +获得表格字段数据结果视图() +显示列表界面() +显示新增界面() +显示编辑界面() +显示搜索结果表格() +响应删除方法() +显示表格字段表格()

图 7 控制器对象接口设计

在此对象中，主要是获得传入 action 参数，然后加载相应的视图模型，最后业务模型对象加载到对应的视图模型上，实现给对象，简单流程图如图 8：

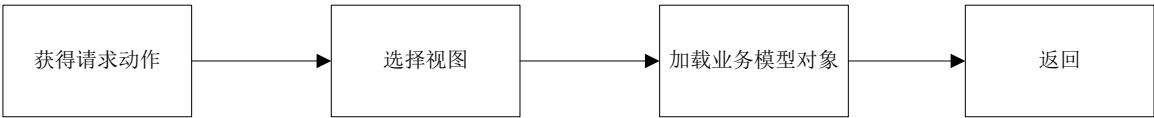


图 8 业务模型加载流程

此控制器对象的方法可以方便被重载，以方便用户进行二次开发。

3.9 数据交互接口访问模块详细设计

业务模型数据接口访问模块主要功能是把用户的输入，把读取视图显示 UI 的方式转换为直接返回 XML 页面数据给用户，以便于开发着进行其它终端例如 PC 端，PC Web 端或者安卓端等 APP 或其他应用进行数据的使用。此对象继承于控制器对象，但是主要有显示列表搜索数据，显示单条数据详情，响应保存方法，响应删除方法，另外就是响应自定义用户请求^[11]。

其请求过程相对数据交互接口平台模块更加简单，主要是获得用户的请求动作，加载业务模型对象，请求数据源数据然后返回结果。通常以特定的 XML 格式返回，以便于相关终端能够以响应的 XML 格式进行读取。接口控制器接口设计如图 9。

业务模型接口控制器对象
+业务模型对象 +用户请求 +数据库访问对象
+返回列表搜索数据() +返回单条详情数据() +更新数据() +插入数据() +删除数据() +用户自定义请求()

图 9 业务模型控制器接口设计

3.10 XML 模型管理模块详细设计

本系统中为了辅助开发者更加快速的生成数据交互接口平台，需要为用户提供一个用户管理 XML 模型的管理工具，通过使用此工具，加大用户管理的效率。

首先是设置工作空间，由于本系统设计的 XML 模型为视图的 XML 文件对象，所以讲所有的 XML 文件都集中存放于系统的固定目录。那么本系统首先设定工作空间，则可以自动加载工作空间下面有效的所有 XML 文件并且转换为 XML 模型。

当本论文得到所有 XML 模型之后，本论文就可以为开发者提供增查改的功能了。同样的，本论文需要为用户提供搜索页面，用于对当前的所有 Xml 模型进行搜索，然后可以对搜索的数据进行编辑修改或者新增相关对象。

同时，本论文需要可以通过加载相关的 XML 模型，直接自动的生成数据库表对象。通过自动生成数据库对象，本论文可以更加快速的开发和生成数据交互接口平台。首先本论文读取 XML 模型，首先生成基础表字段，主要是 id, created_date, created_user, updated_date, updated_user，如果是多语言的业务模型对象，则同时会生成多语言表的相关字段，例如 OID 和 Lang 字段。然后再根据用户定义的字段，生成相关的字段，例如用户定义字段类型为文本，则插入或者修改某个字段为 varchar 字段，如果设置为日期，则使用 Date 字段等等。

最后，本论文将根据用户定义的业务模型对象，自动生成第三方移动终端平台开发产品的代码，例如业务数据对象的相关字段以及其 getter 和 setter 访问器。生成数据库访问对象，自动生成用户需要保存到第三方移动终端平台的本地数据库对象的增删查改方法。生成的业务代码方法用户直接服务到相关的项目中。

图 10 为基本的功能设计：

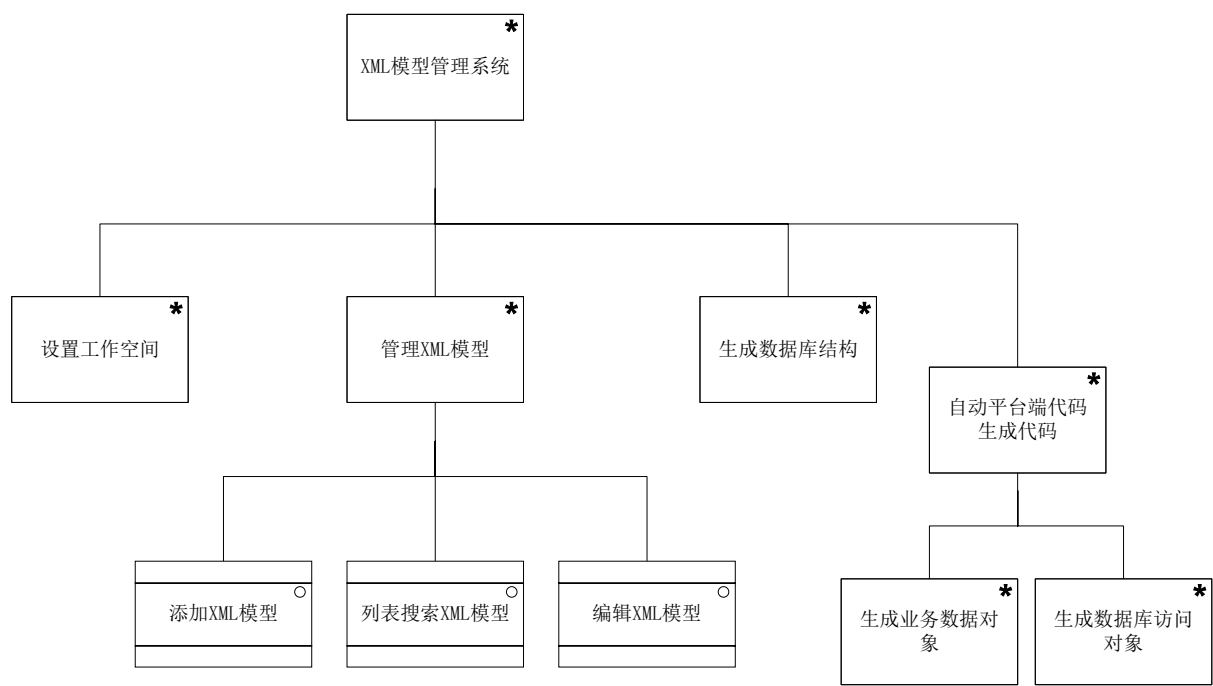


图 10 后台数据管理系统功能模型

3. 11 XML 模型读取模块编码实现

这个模块主要用于读取用户已经生成的 XML 并将其转化为相关的业务模型对象，便于以后针对不同页面进行请求以展示不同的结果。首先本论文读取用户表示的 XML 模型名称读取 model 目录下面的 XML 模型文件，然后检查其是否符合相关定义的格式，接着本论文首先读取器数据来源字段和业务数据名称字段，再来读取相关的业务字段，最后读取定义的业务模型功能。此业务模型对象提供完整的数据格式对应数据库表，并且提供接口让开发者可以再次根据自己的实际业务进行二次开发，预留相关的接口。相关的二次开发接口分为两方面，主要是前端界面和 PHP 数据业务处理后端代码的开发。用户可能会根据自己需求在前端界面工作时对用户操作进行约束或者引导。也可能在后端用户数据保存时进行数据计算的约束等。业务模型数据接口访问模块主要功能是把用户的输入，从读取视图的方式，转换为不读取视图直接返回 XML 页面数据给用户，以便于开发着进行第三方平台例如安卓、苹果、微软等 APP 应用进行数据，此对象继承于控制器对象，但是主要有显示列表搜索数据，显示单条数据详情，响应保存方法，响应删除方法。同时，基于此模型，同样地返回数据访问接口的数据，增删查改能够得到的数据是相同的^[12]。程序流程图如图 11：

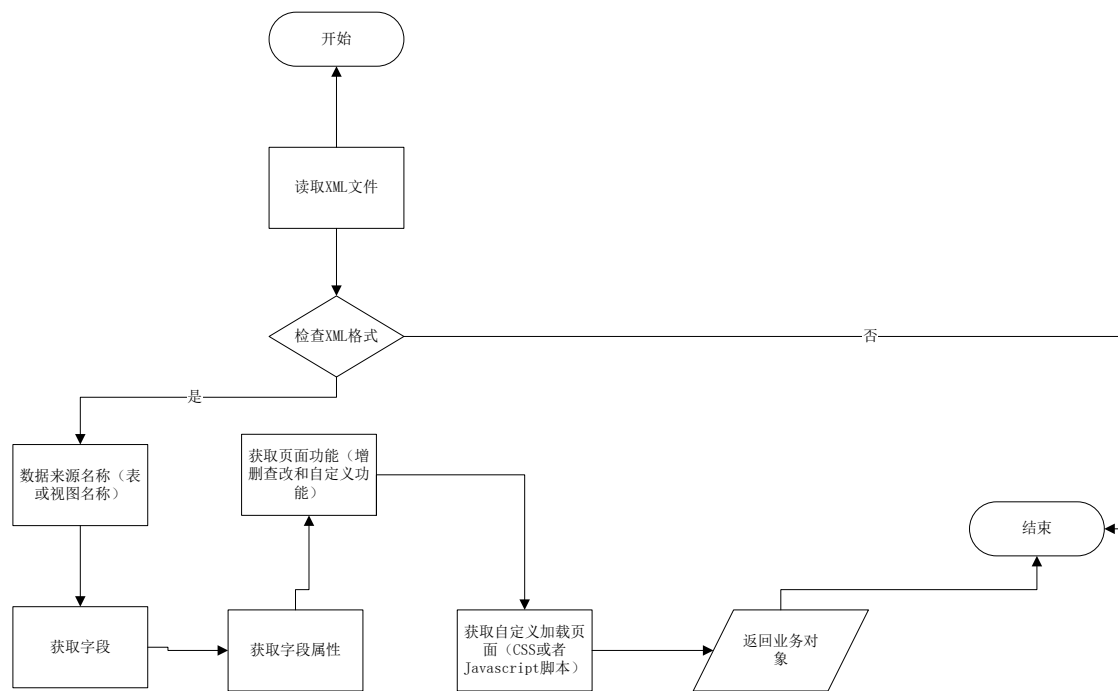


图 11 业务系统模块运行流程

3.12 业务模型编码实现

本模块的编码实现中，主要本论文在控制器中基本流程代码的实现，以流程图方式简单概括，其相关的接口保留方法则不在本节中展现。读者如果感兴趣可以自定阅读相关的源代码。

首先本论文接受用户请求的动作\$action 参数，判断其是否常规的定义，例如“”，默认为显示列表界面，“search”搜索数据返回到列表页面中，“grid”搜索数据返回到表格字段中，“new”显示新增业务对象的业务数据新增界面，“edit”，加载相关数据并显示业务数据编辑界面，“delete”删除用户数据或者其它进入用户自定义方法的业务逻辑中。然后加载这个控制器相关的业务模型对象，如果需要加载数据，则根据是请求生成相应的数据对象，然后把业务模型对象和数据对象一起加载到相应的视图对象中，利用视图中本身签入的业务逻辑代码返回相应的html源代码，返回到用户界面中。以流程图方式简单概括，其相关的接口保留方法则不在本节中展现^[14]。

首先是整体的程序流程如图 12。

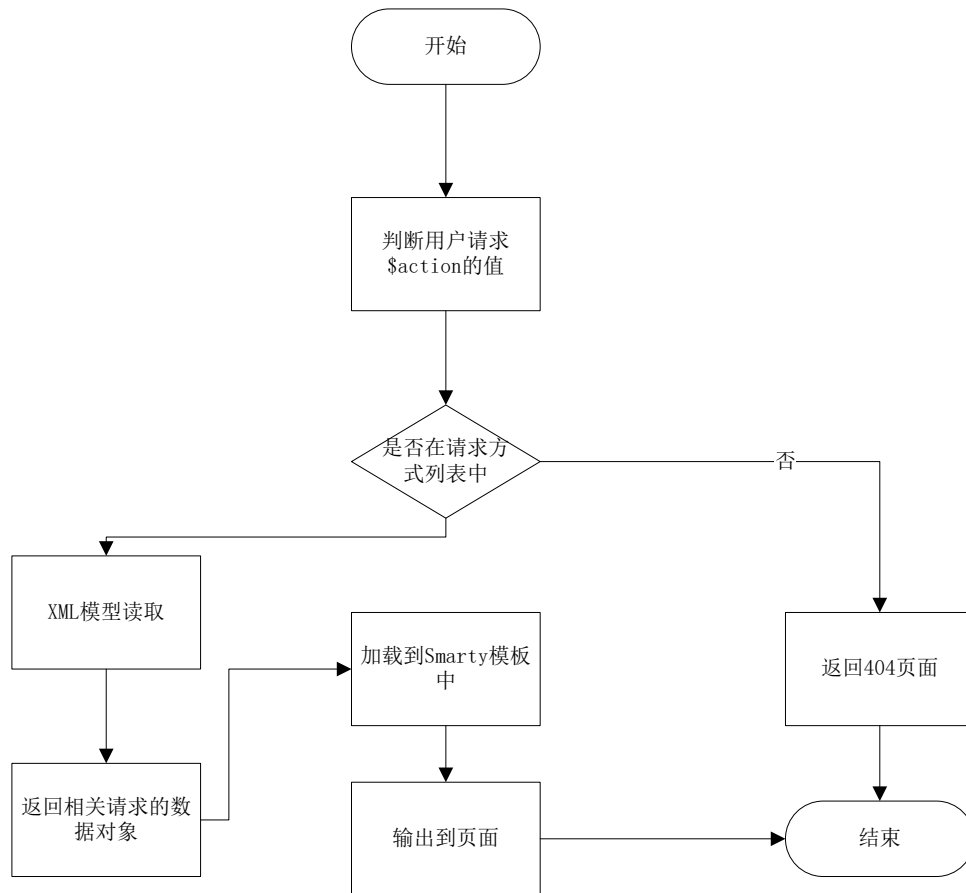


图 12 用户请求到页面输出程序流程

接下来本论文详细的分解响应列表显示界面的程序流程，首先本论文获得业务模型对象，然后判断其相关的外键字段，在本系统中，本论文需要定义集中外键字段，例如下拉列表的数据源可能是另一个表的数据而不是预定于 XML 模型的预定义简单选择字段。多选组合字段也同样可能来源于其它表的数据。所以，首先本论文需要加载相关的外键字段的数据，然后在把当前业务模型对象传入业务模型修改接口实例进行自业务的业务逻辑修改加工，然后则开始把业务模型输出到列表视图中。

在列表视图中，搜索本论文读取业务模型对象加载业务模型管理功能，包括自定义功能和预定义增删改功能，然后加载业务模型定义了 search 为 1 的字段显示到界面中，最后完成其他相关默认的 JS 脚本的编写。例如在这个页面中，本论文需要响应用户点击搜索按钮后的代码，点击删除后的代码，点击新增或者编辑后的页面 JavaScript 代码。如果用户定义了自定义的通用功能和页面相关业务逻辑，本论文则需要用户自定义一个 js 脚本文件设置到业务模型对象中，然后开发者就可以通过书写相关的 JS 代码来结合自己业务逻辑的流程实现更加丰富多彩的业务管理需求。程序流程图如图 13。

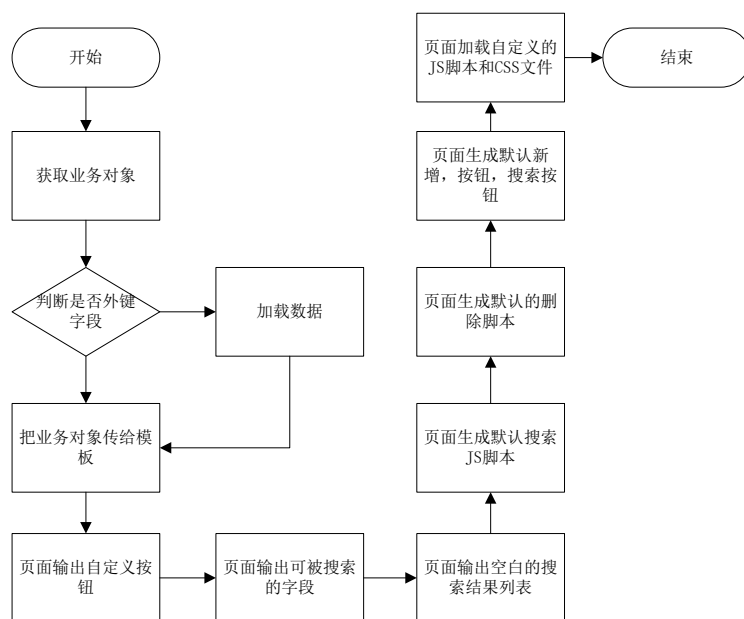


图 13 前端页面加载流程

响应用户点击搜索按钮的程序流程图如图 14。

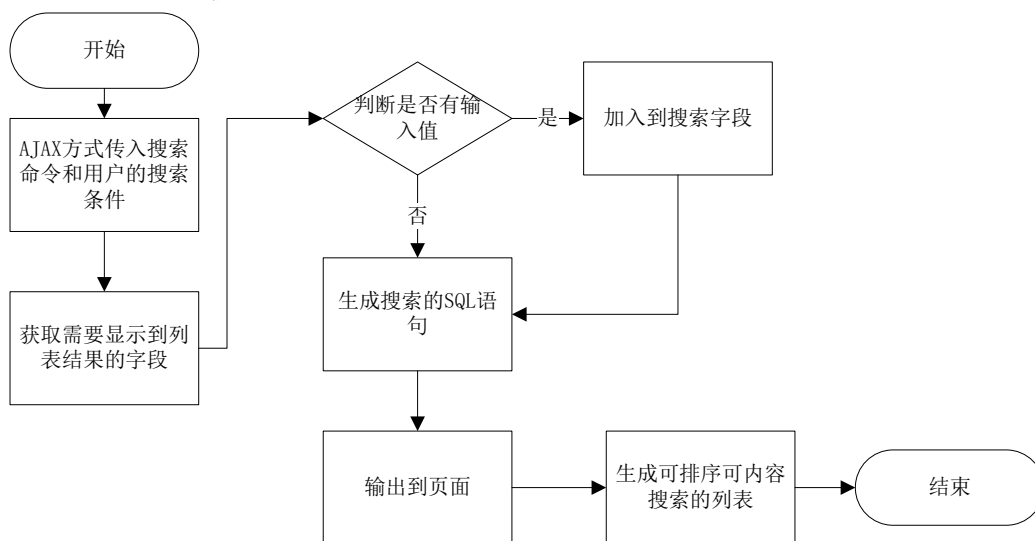


图 14 用户搜索事件响应流程

同样的，响应新增和编辑的整体流程方法与加载列表数据类似，唯一不同则是当用户显示了编辑页面之后，用户点击保存后，本论文则生成 JSON 数据，提交到相关的后台响应控制器的 save 方法，插入或者更新数据，并返回是否成功的结果给用户。开发者也可以继承控制器对象，来自定义业务逻辑的约束或者数据保存方式的修改。以下提供流程图，程序流程图如图 15。

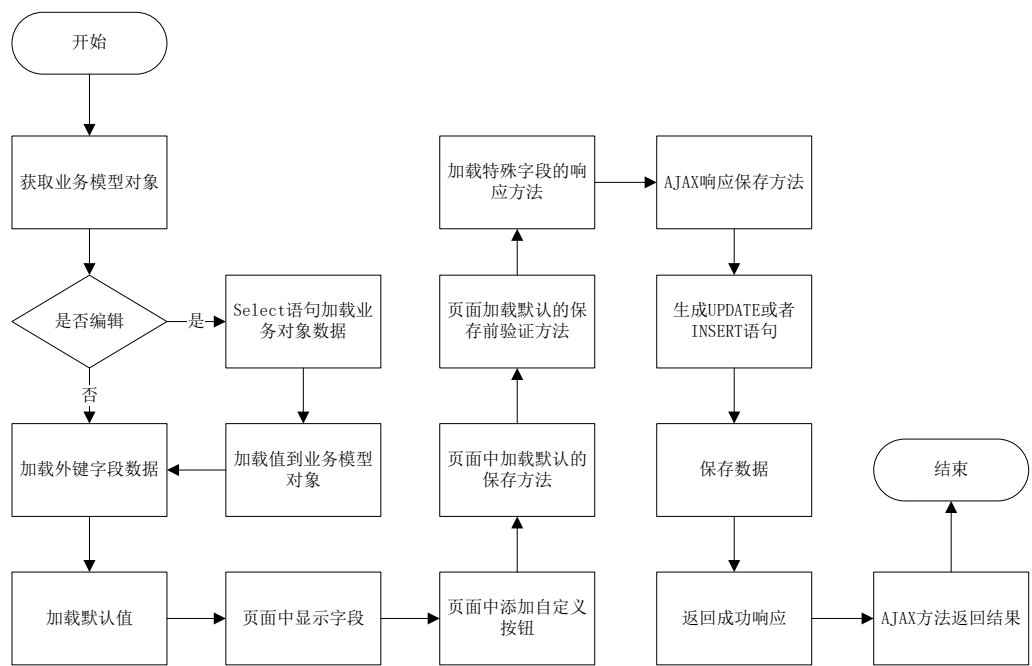


图 15 响应用户数据新增保存流程

响应删除的程序流程图如图 16。

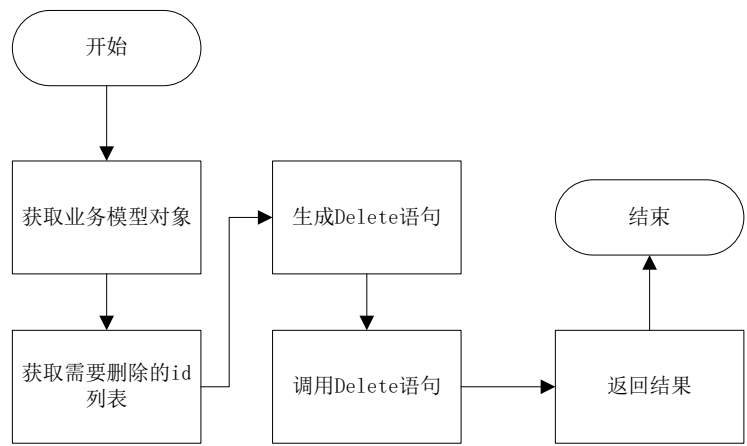


图 16 响应用户数据删除流程

3.13 数据交互接口访问编码实现

当用户通过移动终端通过 URL 利用 GET 或者 POST 方式请求业务模型数据接口事，本论文有以下流程，首先依然是解析请求类型：

“”默认为请求列表数据，请求列表数据与在数据交互接口平台中点击搜索按钮类似，其不同结果为不需要在最终加载到视图中，而是直接把在数据源中获得的数据直接转换为 XML 数据格式，返回到页面中，返回给请求方。

“one”则会获取某一条业务数据的详细信息，根据业务模型的定义，本论文会显示相应的所有字段，其同时要求数据的 id。

“save”则是请求新增或者编辑的业务相应，如果没有附带 id，则默认认为是请求新增，最终将会在数据库中插入一条数据。当然，本论文在此会判断业务的准确性。返回结果为固定的格式，一行加三个字段，分别是 id, result, return。Id 则是操作标识码，通常 0 位成功，其他均为失败，result 为返回结果，通常作为对 id 的解析。Return 则是返回结果，在 save 的请求中，通常返回其相关的业务数据 id。

“delete”则为删除单条或者多条数据，其结果依然返回结果类型的固定格式。

其他请求，用户可以读取 url 请求中的其他类型，自己编写 SQL 语句，最后直接以二维固定格式的方式返回 XML 数据中。程序流程图如图 17。

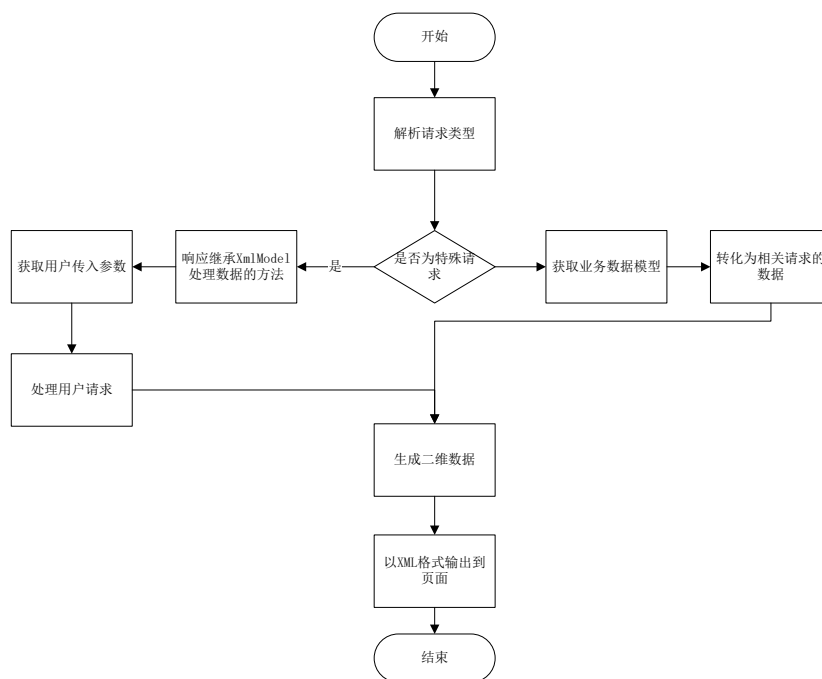


图 17 数据应用程序接口业务流程图

3.14 XML 模型管理模块编码实现

本系统中将会提供 XML 模型管理模块，方便用户用可视化的界面最直接的方法创建

相关的业务对象模型，使得快速完成业务模型的设计，同时方便开发者进行管理。程序流程图如图 18。

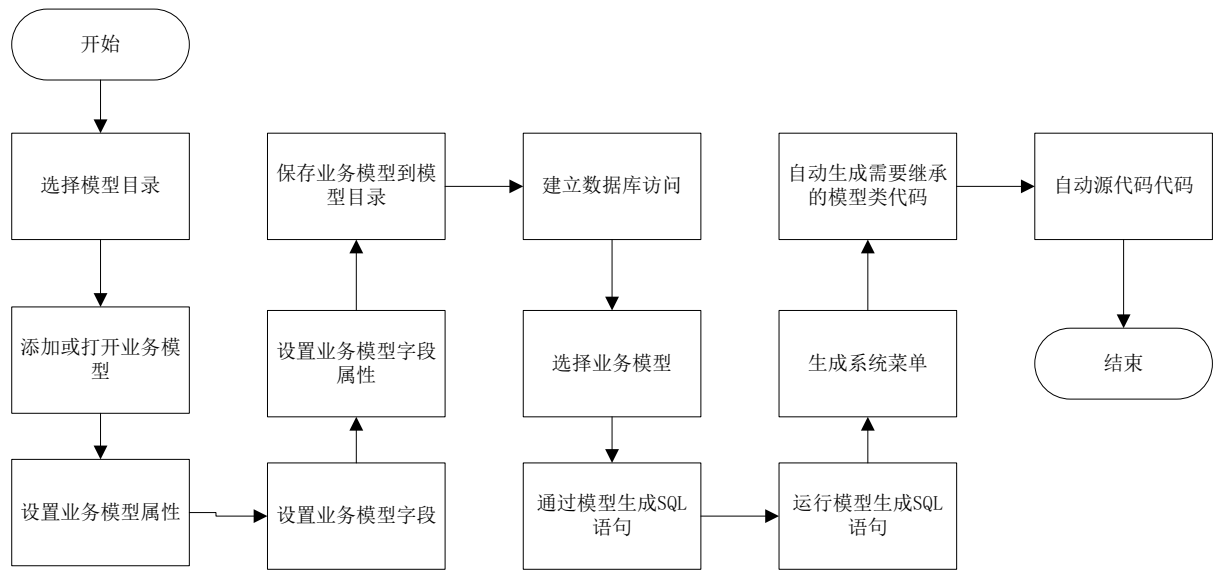


图 18 系统自动架构程序流程

3.15 小结

以上的过程就是设计整个数据交互接口后台管理系统的过程，总的来说，用户通过定义一个丰富内容的 XML 模型，然后本论文代码中的 XML 模型读取模块则自动根据 XML 模型的定义自动来生成相应的前端界面，前端界面数据，后台数据操作等功能。用户只需要根据自己的实际业务模型来定义 XML 模型的结构，就可以自动生成一个数据管理后台和数据交互接口。数据管理后台直接服务于数据管理人员录入数据和管理数据。而根据录入的数据生成数据交互的数据交互接口数据供移动端进行数据交互读取。

整个过程的主要设计就是为了减少用户在反反复复的后台数据管理系统中进行重复编码的工作。如读者所知，正常的任何一个不同形式的应用，实际上都是对输入输出的数据进行交互，除非读者开发的是一个纯单机应用，否则，一般都会花时间和代价开发一个后台数据管理系统供开发者和管理员对用户数据进行管理。本系统的第一部分设计则是希望在最大程度上免去开发者在这方面的重复性工作，开发者只需要定义数据业务模型生成 XML 模型，则可以自动生成整个后台业务数据管理系统^[15]。

同时，因为现在移动互联网的流行，大部分的移动互联网应用都会需要跟服务器进行数据交互以进行业务流程的推进和完成。本系统则是把后台管理系统生成的 XML 模型

进行二次利用，直接生成一个可以被移动端调用的数据交互接口。开发者同样不需要进行任何代码编写就已经可以通过调用数据交互接口来完成移动应用端对后台进行数据较。当然，根据用户的自己的需求，可以定义特殊自定义相关请求接口，更加人性化和合理化方便开发者提供给移动应用的服务^[16]。

因此，本论文开发一个易于定制和优化，容易学习和掌握，可以帮助开发人员提高开发效率并简化代码管理和维护。本开发框架具有可扩展性，可接入几乎所有的移动平台，实现起来相对简单，维护成本较低，为移动应用开发者提供了一个优秀的数据交互解决方案。

第 4 章 移动端数据交互组件的设计与实现

在本课题中，本论文将为各种主流的数据终端提供数据交互的核心代码。相关的代码包括对远程数据接口的数据读取，对远程数据接口进行数据提交，将远程数据接口的数据转化为相应的业务数据对象，以及提供相应的数据访问对象接口可以实时对本地数据进行增删查改，使有选择性的让本地用户数据与服务器上定义的业务数据保持同步，最终使用户的想法和业务成功展现在移动终端用户面前。本课题中的提供的代码应该是高拓展高继承性高集成性的，让用户只需定义对应数据请求接口和数据返回对象，就立即可以得到业务数据对象。目的是尽可能的节省用户的编码，使其可以集中精力在自己相关的业务代码的编写上^[17]。

本课题讲主要针对安卓客户端和进行 JAR 代码继承包的编写和基于 Cordova 的 JS 包，满足现有的 APP 主流开发平台。

4.1 移动端数据交互组件详细设计

移动终端实际上是最終的目标开发程序，当今互联网产品的开发，离不开开发移动互联网端的产品提供给用户相对于网页以外的体验，目的就是为快速为移动终端平台或者网站平台提供业务数据的支持以及提供一个数据访问接口方便开发者开发移动终端业务模块。开发一个可以被用户使用的 JAR 包，用户可以把这个 JAR 包添加到开发项目中，然后继承根据业务模型自动生成的业务对象代码。例如新闻的业务数据对象，数据访问对象和网络交互对象等，重载其指定的相关的抽象方法，来指定移动终端的用户开发模块应该去读取数据访问接口中的相关数据^[18]。

本论文来详细描述相关的抽象类的几个重要业务对象。首先本论文会定义一个抽象业务数据对象，Business Object 对象。此对象主要用户装载业务数据，以方便开发者在开发过程中对业务数据对象的处理和传输，解析 SOAP 传送的单行数据到当前对象的相关业务数据字段，和解析数据表游标的数据到当前对象的相关业务字段，定义如图 19：

业务数据对象BO
+id
+解析XML()
+解析游标()

图 19 业务数据对象接口设计

首先本论文会定义一个抽象的数据访问对象，Data Access Object 对象。此对象主要提供对某个业务数据的增删查改功能。然后本论文需要定义一个远程数据接口访问器对象，此对象主要有几个抽象方法等待用户实现，首先需要定义相关调用的数据访问接口的超链接，只有定义了访问相关的超链接，本论文才可以准确的去响应的 URL 中获取 SOAP 协议的 XML 对象返回到本地。然后利用 BO 对象的解析 XML 功能，将其加载为一个 BO 对象，然后根据需要，本论文需要设定其是否需要更新到本地数据库，此处也是一个抽象方法，用户需要返回是否需要调用本地。最后则是加载数据或者写入本地数据库成功后，本论文需要定义一个抽象方法让用户实现是否需要调用回调函数让用户立即去处理用户相关数据。对象定义如图 20：

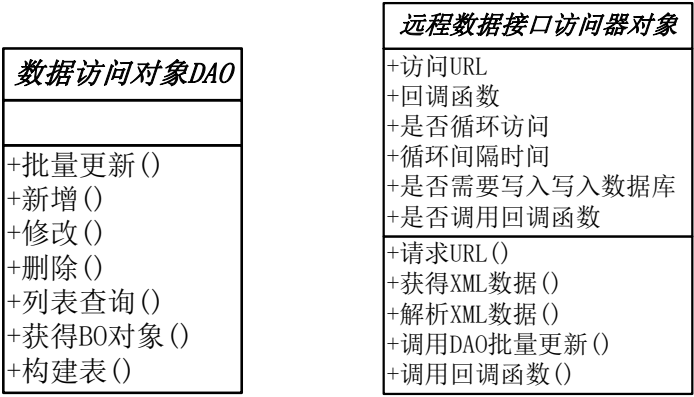


图 20 数据访问对象接口设计

最后，本论文设计一个简单流程用以描述从请求到结束返回的过程如图 21：

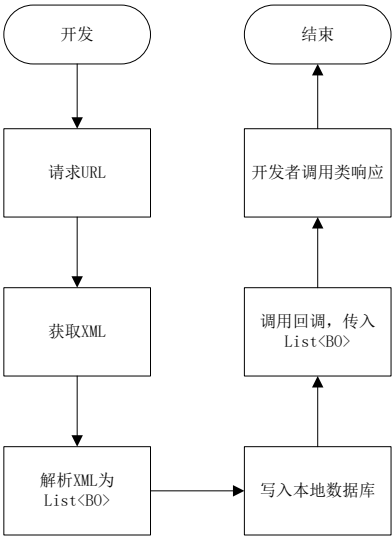


图 21 远程数据加载流程

4.2 移动端数据交互组件编码实现

在此模块中，本论文将针对安卓平台设计一个 JAR 包方便用户直接调用生成的 API 进行交互。分为两部分，首先本论文主要定义两个基础对象，BO 对象的 AbstractObjs 对象，其首先有一个 ID 字段，还有需要实现的两个方法，分别是解析 XML 行数据，本论文在 WebLoad 对象会首先解析为一个 List<Dictionary<string,string>>对象，其数据就是 XML 数据中的一行一行数据对应一个 Dictionary，然后将指点对象传入到 BO 对象中，让其继承的对象把相关的字段数据解析到相关的字段属性上。用这个方法完成把远程数据访问接口的数据读取到本地。这时，在相应的运行时代码中，本论文已经获得了远程的服务器数据交互接口返回的数据，以新闻列表数据为例，这是，本论文通过继承了 WebLoad 抽象对象的 NewsWebLoad 对象进行远程数据的访问，在回调中本论文就获取了继承了 AbstractObjs 抽象对象的 NewsObj 对象，此对象以及拥有了相关的数据。当然，这里提到的 NewsWebLoad 和 NewsObj 都是自动生成的代码直接可以被用户调用已进行调用并直接显示到用户自定义的移动终端应用界面中^[20]。

最终，根据其相关的 DAO 类中，本论文把改数据对象更新到客户端本地的数据库中
进行离线数据访问。二次开发远程数据访问流程图如图 22 和图 23。

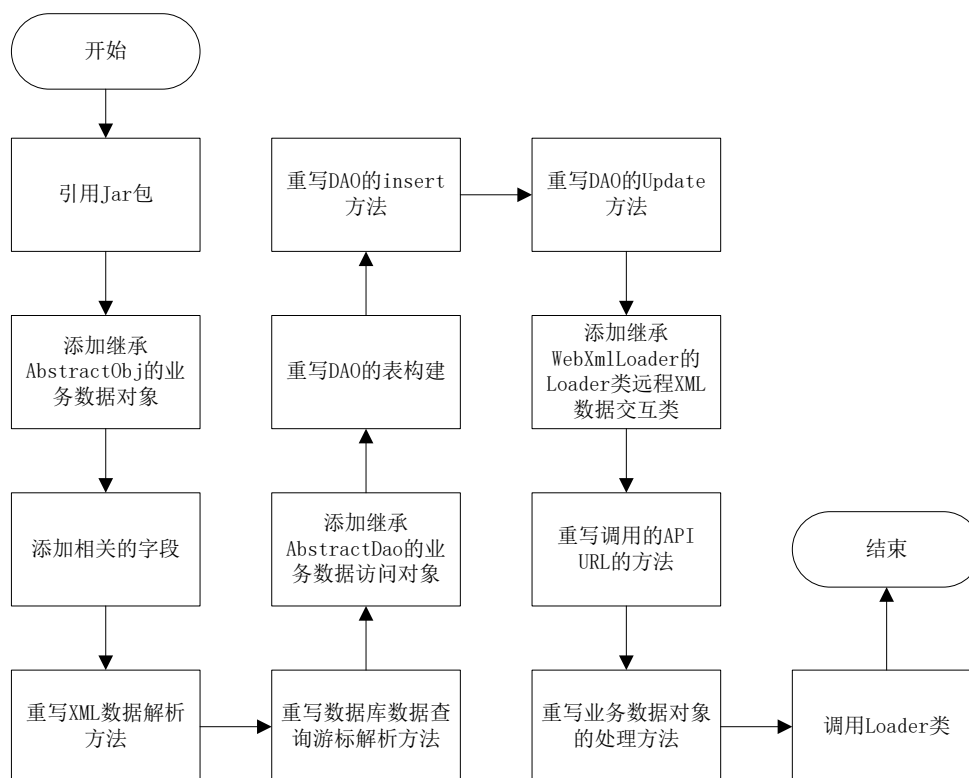


图 22 二次开发远程数据访问流程

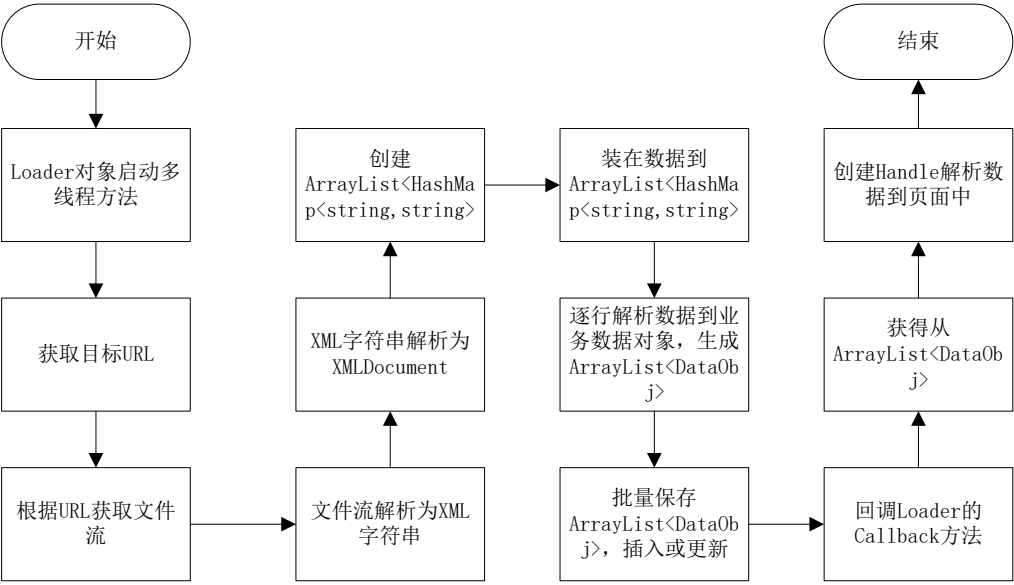


图 23 远程数据接口数据返回解析流程

第 5 章 数据交互服务框架的应用

面向移动应用开发的数据交互服务框架已经设计完成，接下来本论文则需要验证本论文所开发的系统如何应用到实际的开发过程中，以证明利用此系统开发互联网产品的快捷性和实用性。

在本章中，本论文将设计开发一个安卓平台的新闻移动客户端，充分利用本平台来开发的提供的所有工具来暂时开发的代码。并且在此过程中，本论文将会同时验证本数据交互接口平台是否符合设计目的和完成所有计划的需求。

5.1 研究环境安装与配置

首先，本数据交互接口平台的内容管理系统和数据访问接口部分都是部署在 Apache+PHP 的环境中。数据库则不作要求，只要用户能够连上数据库即可。所以第一步请先配置到 APACHA+PHP+目标数据库环境，本论文中选择使用 MYSQL 环境。

第二步，把源代码中的 CMS 文件夹所有的文件部署到对应的 APACHA 源代码文件夹中，然后就是修改 \CMS\Config\config.php 中的配置，例如修改部署路径，上传文件共享目录，数据库配置等。

第三步，把源代码中的 API 文件夹所有的文件部署到对应的 APACHA 源代码文件夹中，修改 API\Config\config.php 中的配置，主要是修改当前工作路径和数据库链接配置。

第四步，导入 \DBBackup\init.sql 到数据库中，主要是关于用户管理模块基础数据。

最后，配置 android 平台开发环境，并且引入 Android\helpfooter.jar 包。

至此，本论文可以开始进行新闻客户端的开发了。

5.2 数据交互接口端的设计

如上设计和实现，本论文只需要利用 XML 模型管理系统生成 XML 模型和相关的访问菜单，就可以完成新闻管理端的配置，首先本论文进入 \ModelCenter\Startup.exe，然后进入，选择工作目录，自动会显示所以在工作空间下面的所有 XML 模型对象，然后本论文新增一个 news 模型，名字为新闻，数据来源为 tb_news。然后为其添加需要的字段，本论文在此添加标题，导读，内容，发布日期，状态等。并且添加新闻，删除功能。保存后，本论文可以在 CMS\Model 文件夹下面多了一个 news.xml 文件，他就是本论文前

文所述的业务模型，本论文通过工具生成直接生成 Xml 的。主要的模块在为如下的几部分：

主要节点如下：

```
<root>
  <name>模块标题</name>
  <tablename>数据库表名称</tablename>
  <nolist>0</nolist> //是否没有列表页面，仅有编辑页面，0 为不是，可以省略
  <searchcondition> r_main.status<';D' </searchcondition> 默认搜索条件，此例子中，Status 为 D 的是在数据库中已经删除的数据。
  <fields>相关字段</fields>
  <options>相关操作，例如新增，导入，自定义按钮的功能</options>
  <javascripts>加载自定义或者拓展的 JS 文件</javascripts>
  <csss>加载自定义的 CSS 文件</css>
</root>
```

然后本论文再看一下一些比较有代表性的字段：

比如新闻标题字段：

```
<field>
  <name>标题</name>
  <key>title</key> //数据库代表的字段
  <type>text</type> //字段类型，有 text,number,date 等
  <displayinlist>1</displayinlist> //是否在列表中作为搜索结果显示
  <gotodetail>1</gotodetail> //是否能够点击该字段跳转到详情页
  <search>1</search> //是否作为搜索字段
  <notnull>1</notnull> //是否非空
  <unique>1</unique> //是否唯一
</field>
<field>
```

当然，这是最基本的字段，本论文还是定制有相关的 type 为 select 的下拉字段，为 grid 的引用内容页面，为“content”的 Wordpress 字段，为“fkey”的外键字段，为“flist”的多选列表或者外表 1 对 N 的关联字段等，此处略。

接下来本论文点击生成数据表按钮，其就会立刻在数据库中添加一个表，并完成其表结构的设计。图 24 为表接口设计。

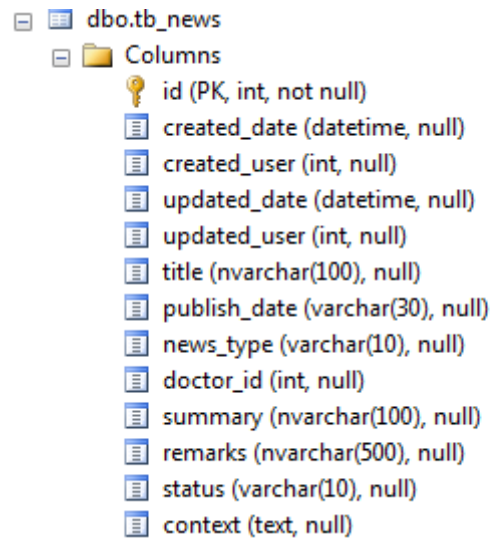


图 24 新闻业务数据库表

然后本论文需要打开编辑系统菜单的选项，添加一个网页和生成网页，然后在 CMS 系统中就会生成一个 CMS 文件夹中生成一个 news.php，并且里面已经有基本可调用的代码，代码如下：

```
<?php
require '../include/common.inc.php';
include ROOT.'/include/init.inc.php';

$action=$_REQUEST["action"];
$model=new XmlModel("news","news.php");
$smarty->assign("MyModule","website");
$model->DefaultShow($smarty,$dbmgr,$action,"news",$_REQUEST);

?>
```

如果开发者需要编写相关的业务逻辑代码，则可以继承相关的控制器对象或者业务模型对象修改接口实例，对数据进行修改，在本例子中，本论文添加一个特殊的业务逻辑，设置发布日期必须为当天。本论文继承 XmlModel 对象，然后命名为 NewsXmlModel 对象，重载其 Save 方法。代码如下图 25：

```
<?php
class NewsXmlModel extends XmlModel{
    public function __construct($pagename){
        parent::__construct("banner",$pagename);
    }

    public function insert($request){
        //设置字段默认值,发布日期为当天
        setFieldValue("publish_date",Date('Y-m-d'));
        return parent::Save($dbMgr,$request,$sysuser);
    }

    public function update($request){
        //设置此字段不会被更新
        setFieldNoUpdate("publish_date");
        return parent::Save($dbMgr,$request,$sysuser);
    }
}
```

图 25 NewsXmlModel 插入和更新代码重写

接下来,本论文就可以验证是否已经完成新闻模块的管理的开发了。首先本论文使用浏览器进入 CMS\index.php,使用默认登录名登录系统。登录界面如图 26。



图 26 登录页面

然后见到主界面，是关于此系统的介绍。本论文找到新闻管理的菜单，点击进入图 27 界面：

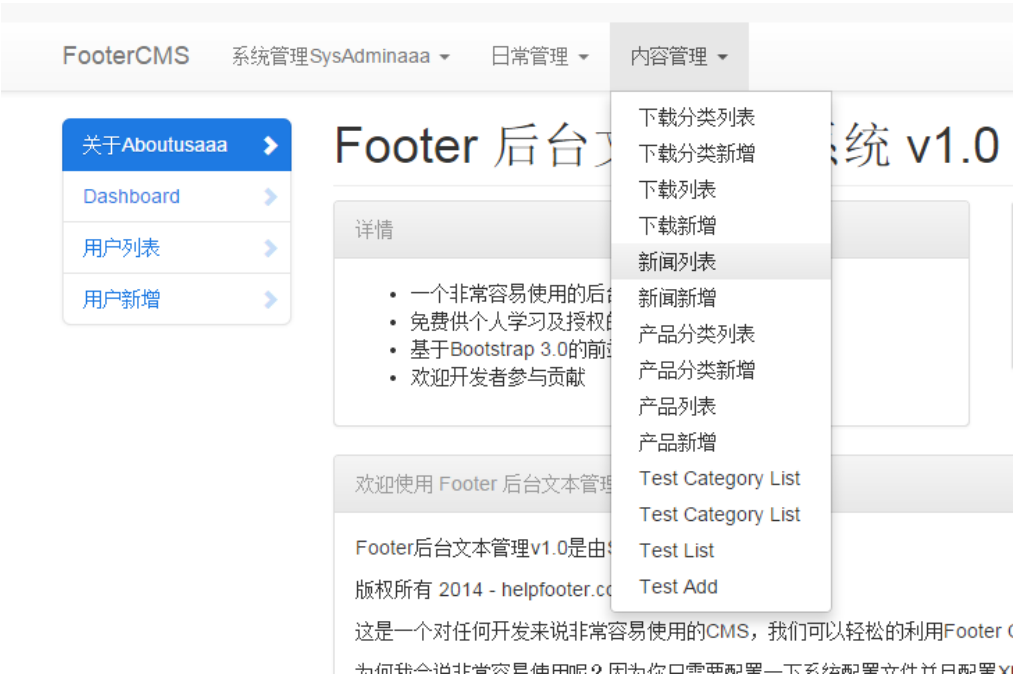


图 27 登录后控制面板

然后本论文点击新增添加一条新闻，输入内容，点击下方的保存按钮，如图 28。



图 28 新增界面

首先进入新闻列表，界面如下，发现左上角新增和删除按钮的功能已经添加如图 29：

新闻 - 列表

删除

新增

搜索条件

标题

发布日期

起始日期

-

结束日期

导读

状态

--请选择--

搜索

搜索结果

	标题	发布日期	状态
--	----	------	----

图 29 新闻列表

点击搜索可以看到起通过 AJAX 局部刷新搜索结果栏，显示现有的新闻数据如图 30：

搜索结果

10 records per page

Search:

	标题	发布日期	状态
<input type="checkbox"/>	标题1, 中国万岁	2015-09-05	启用
<input type="checkbox"/>	标题2, 祖国万岁	2015-09-18	启用
<input type="checkbox"/>	新闻标题3	2015-10-15	启用

Showing 1 to 3 of 3 entries

← Previous

1

Next →

图 30 新闻列表搜索结果

点击“新闻标题 3”进入修改界面对数据进行修改，保存后在列表搜索界面中本论文可以验证数据已经被修改。

接下来验证删除功能，在搜索结果的列表中勾选需要删除的数据，然后点击右上方删除按钮，点击确认删除，则不会在搜索到相关数据。删除界面如图 31。

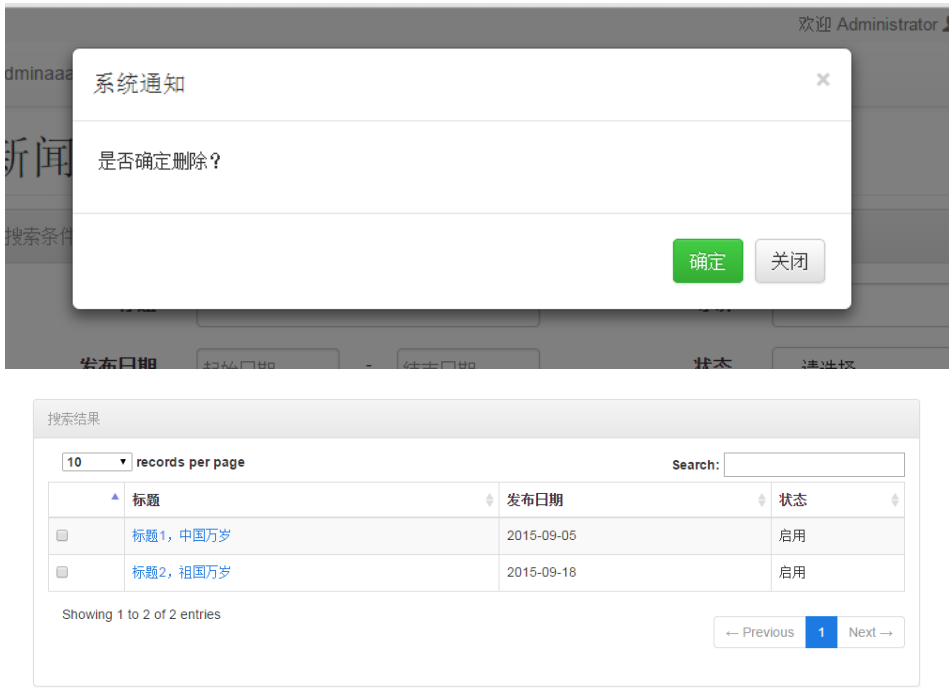


图 31 新闻列表

至此，本论文已经验证，数据交互接口平台已经可以对新闻业务模块内容进行增删查改的实现，而本论文添加的代码仅仅只是特殊需求对发布日期的强制写死录入。基本上，本论文通过 XML 模型管理工具已经完成了所以基本代码的生成，而在这个过程中，本论文并没有书写任何代码就完成一个业务模块的管理。

5.3 远程数据数据访问端的设计

本论文已经可以基本上完成数据访问接口端的开发，当本论文调用 news.php 页面时，本论文获得一个 XML 格式的网页结果返回如图 32。

```
<table>
  <row>
    <id>2</id>
    <title>新闻2标题例子</title>
    <published_date>2015-09-24</published_date>
    <status>启用</status>
  </row>
  <row>
    <id>1</id>
    <title>新闻1标题例子</title>
    <published_date>2015-09-18</published_date>
    <status>启用</status>
  </row>
</table>
```

图 32 Xml 格式数据接口列表字段返回

当本论文需要获取某一条数据具体的所有数据时，本论文调用的 url 为 news.php?action=get&id=1，然后获得返回结果如图 33。

```

▼<table>
  ▼<row>
    <id>2</id>
    <title>新闻2标题例子</title>
    <summary>新闻2导读例子</summary>
    <published_date>2015-09-24</published_date>
    <content><h1>新闻2内容例子</h1></content>
    <viewcount>0</viewcount>
    <status>A</status>
    <created_date>2015-09-18 05:05:15</created_date>
    <created_user>1</created_user>
    <updated_date>2015-09-18 05:05:15</updated_date>
    <updated_user>1</updated_user>
  </row>
</table>

```

图 33 Xml 格式的数据接口详情字段返回

在此基础上，本论文添加一个自定义需求，当用户在第三方客户端访问页面后，本论文需要 news 表中的访问数量字段。于是本论文基础修改 NewsXmlModel 代码如图 34：

```

<?php

class NewsXmlModel extends XmlModel{

    public function __construct($pagename){
        parent::__construct("banner",$pagename);
    }

    public MyAPIAction($action,$request){
        if($action==""){
            $id=$request["id"];
            $sql="update tb_news set readcount=readcount+1 where id=".$id;
            $this->dbmgr->query($sql);
            return outResult(0,"success",$id);
        }
        parent::MyAPIAction($action,$request);
    }

}

?>

```

图 34 NewsXmlModel 二次开发代码

当客户端调用 url news.php?action=read&id=1 之后，本论文则动态修改数据库中新闻访问数量的字段。提供开发者更简单直接建立一个机遇 SOAP 协议返回 XML 数据

的设计框架。

到这里为止，本论文已经可以看出，本论文并不需要为数据访问接口端书写太多的代码，本论文依然是想信息管理系统一样自动生成业务数据以成功生成系统，并且如果用户需要加入自定义的业务时，修改接口也是非常简单灵活的。

5.4 安卓平台新闻移动客户端的设计

开发一个简单的新闻移动客户端，我只需要在项目中添加两个 activity，一个是显示新闻列表，一个是显示新闻详情。本例子中，本论文为了减少客户存储过多的数据到本地数据库中，本论文将不会把新闻的详细内容加载到客户端中，只会简单的字段，例如新闻标题，新闻导读，发布日期等建短信息。

为此，本论文需要首先继承 AbstractObj 类，然后添加访问器和实现其定义的抽象方法。代码如下：

```
public abstract class AbstractObj {
    public int id=0;
    public int getId() {
        return id;
    }
    public void setId(int val) {
        this.id=val;
    }

    //该方法用于解析从本地数据库读取数据时的方法
    public abstract void parseCursor(Cursor cursor);
    //该方法用于解析从数据访问接口访问到某一行使加载数据的接口
    public abstract void parseXmlDataTable(HashMap<String, String>
1stRowValue);
}
```

相关代码已经自动生成

然后本论文需要添加一个继承与 AbstractDao 的 NewsDao，新闻数据访问类，并实现其相关的抽象方法。对于本论文要做的则是实现相关的构造函数方法，不过系统中也

已经自动生成该代码，用户可以直接使用 NewDao.java 类，代码如下：

```
public class NewsDao extends AbstractDao {

    public NewsDao(Context ctx) {
        super(ctx, "tb_news");//指定表名
    }

    //自定义查询的接口方法
    public ArrayList<AbstractObj> getNewsList() {
        return super.getList(" status='A' order by publish_date, id desc ");
    }

    @Override
    public void insertObj(AbstractObj abobj) { //代码自动生成}

    @Override
    public void updateObj(AbstractObj abobj) { //代码自动生成}

    @Override
    AbstractObj newRealObj() { return new NewsObj(); }
}
```

有了这两个对象，本论文的新闻数据就可以保存在内存中和本地数据库中。

然后本论文在定义一个继承 WebLoader 的 NewsListWebLoader 对象，通过这个代码本论文就可以自动去获取 URL 对象并将其保存到本地数据库中：

```
Public class NewsListLoader extends WebXMLLoader{

    @Override

    Public String getCallUrl() {

        Return "HOST\news.php"

    }

}
```

本论文在 NewsListActivity 中，可以如此调用此 NewsListLoader 对象：

```
Private void InitDate() {
    NewsListLoader loader = new NewsListLoader(this);
    loader.setCallBack(this);
    loader.start();
}

Public void WebLoaderCallBack(ArrayList<AbstractObj> lstObjs) {
    //用户根据返回的数据内容显示用户数据，此处显示新闻列表
    ShowNewsList(lstObjs);
}
```

最终，本论文可以在安卓客户端中看到新闻列表和详情如图 35 界面：

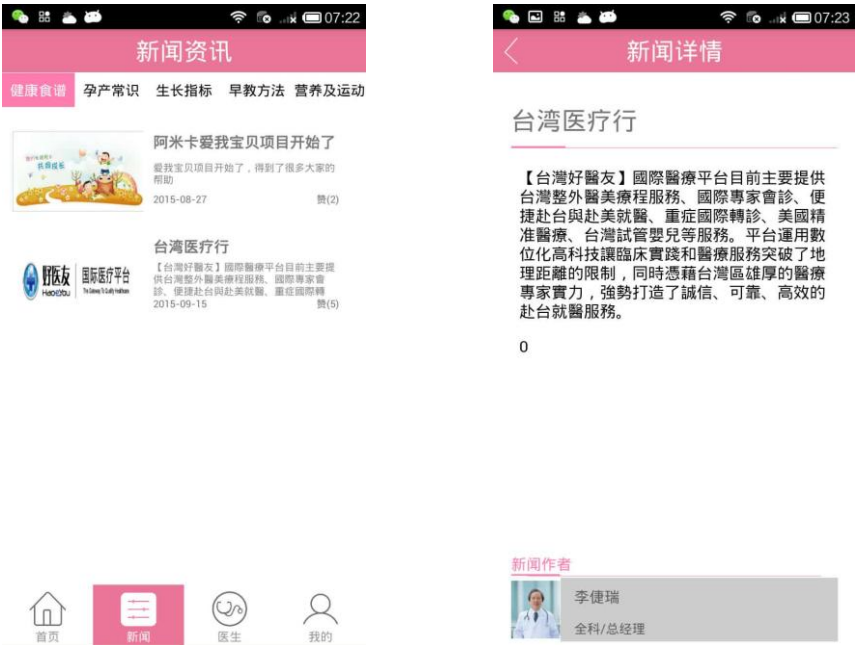


图 35 新闻列表和详情

至此，本论文的安卓平台的新闻移动客户端已经开发完成，开发者可以快速的添加几个类，就可以自动和远程数据访问接口中的数据交互。可见，本论文除了对安卓界面的代码进行构建以外，本论文并不需要添加太多的代码就已经完成了新闻模块代码的构建。

5.5 小结

从以上的安卓平台新闻系统的设计过程可见，本论文已经不需要为书写太多的代码

就已经可以完成整个系统开发，本论文主要要做的事情是利用面向对象的设计思想把相关的对象和模型定义好，然后本论文利用 XML 模型管理系统来服务生成数据交互接口平台端以及远端数据访问接口端。然后在移动客户端对业务模型对象进行数据对象，数据访问对象和数据交互对象的定义，就可以快速完成移动客户端的设计，中间不需要过多的关注系统业务逻辑。

第 6 章 总结与展望

6.1 研究工作总结

本论文提出的面向移动应用开发的数据交互服务框架具有开发快，可继承性高，配置化等特点，它能够帮助开发者为互联网+的产品快速提供核心业务数据管理系统，并且提供了相关的移动终端数据交互方位 Jar 代码包。通过这样的方法，开发者可以在最短时间内基于业务系统来完成业务数据系统的设计，可以较快实现开发者的设计思想。

本框架能解决大多数移动应用的数据接口访问问题，实现过程简单，易于定制和优化，容易理解和学习掌握，可帮助开发人员提高开发效率，并简化代码管理和维护。本开发框架具有可拓展性，可应用于跨平台的移动开发中，实现起来相对简单，维护成本较低，是一个可行的面向移动应用开发的数据交互服务框架。

实现本框架所用到的前后端技术，都是本论文书写阶段的流行而且成熟的应用。PHP 已经成为 Web 开发中最常用的技术之一，据统计也是开发人数最多的语言之一，本研究使用 PHP 就是面向 PHP 的学习成本低，开源且部署费用低等优点。所以对于使用本框架的开发者来说，要实现和优化本框架，只需要掌握主流开发技能则可，容易理解和掌握框架开发的核心思想。开发者可以根据自身项目需要，对框架本身做更多的优化与定制，从而提高应用程序开发的效率。

对于框架的使用者来说，只需要基础的移动应用开发技术即可，不需要掌握困难的经验，因为平台的框架和代码都已经解决了开发过程中的大部分问题。例如在后端数据管理系统中的繁冗的增删查改的代码的开发，在相关的增删查改的过程中，又涉及某些字段需要各种需求例如唯一性等，本研究均在后台提供的服务中给予解决，用户只需要集中精力关注自己的业务对象就可以，大大地降低了开发门槛，提高了开发效率。而在移动应用的部分，解决了网络，数据传输，数据解析等问题，用户直接调用远程数据加载接口或者本地数据库访问接口即可以直接获取数据直接显示，根据自己实际的需求选择是否要存到数据库中，或者是否必须要 WIFI 状态下可用等，同样也大大提高了开发效率并减少了代码量。

6.2 后续工作展望

本课题依然有较多的地方需要改善，例如安全性问题，因为本系统兼容多数据库，所以难以做到兼顾这个系统防止用户注入的问题。因为本课题设计的系统主要基于追求性能的快和开发速度的快和开放性上面，所以减少了用户请求的校验，这对于数据系统的安全性来说是一个需要解决的问题^[23]。

在后续的工作中，本系统将会加入私有云的概念并成立为一个平台，根据用户的申请自动创建一个服务器和相关数据交互服务系统，用户可以直接在私有云中添加自己的业务驱动模型即可以在相关开发的移动终端产品中访问相关私有云自动化移动业务管理系统和其数据访问接口，使其成为更加高效、免于部署的自动化业务管理系统。

参 考 文 献

- [1] Abel Avram. Adobe Is Distributing Cordova under the PhoneGapBrand.
<http://www.infoq.com/news/2012/03/Adobe-Cordova-PhoneGap>[N], 2014
- [2] Reference implementation for building RESTful Web services[N]
<http://jsr311.dev.java.net/nonav/releases/1.1/index.html>, 2013
- [3] Meier R. Professional Android 4 Application Development[R], 2014, 5-24
- [4] Imielingski T, Badrinath B R. Mobile wireless computing: challenges in data management[R].
Communications of the ACM, 2014:18-46
- [5] Feng Y, Fen G J. Incremental Updating Algorithms for Mining Association Rules[R]. Journal of
Software, 2012:20-34
- [6] G H Forman, J Zahorjan. The Challenges of Mobile Computing[R]. IEEE Computer, 1994:1-6
- [7] Zina Pozen. WMI and .NET. MSDN Magazine, 2002:12-16
- [8] 闫晓亮. 基于 MVC 设计模式的轻量级 PHP 开发框架的研究与实现[D]. 长春工业大学, 2016:4-48
- [9] 汪盛. 基于轻量级框架的企业应用快速开发平台的设计与实现[D]. 电子科技大学, 2012:2-26
- [10] 所修文. 轻量级 JAVA EE 框架的研究和实现[D]. 天津大学, 2012:4-12
- [11] 杨迪. 基于 Android 平台的美拓暴风系统的设计与实现[D]. 哈尔滨理工大学, 2015:2-58
- [12] 邹煜. 企业级移动应用平台建设与安全保障体系探析[J]. 网络空间安全, 2016:95-98
- [13] 王荣海. 基于 HybridApp 技术的企业移动应用系统构建研究[J]. 软件工程, 2016:35-38
- [14] 廖成玉. 企业级移动应用平台的选择与建设[J]. 现代电视技术, 2015:64-88
- [15] 王壮志. 关于软件过程模型构建策略的分析和思考[J]. 内蒙古民族大学学报(自然科学版),
2014:72-74
- [16] 王璐. Web 模式下基于 XMPP 的即时通信系统的设计与实现[D]. 北京邮电大学, 2014:6-38
- [17] 赵哲. 面向开放平台的移动应用构建与推送引擎的设计与实现[D]. 北京邮电大学, 2013:5-46
- [18] 高嘉泽, 高强, 吴国全, 魏峻, 黄涛. 面向移动应用的后端服务平台[J]. 计算机系统应
用, 2014:16-17
- [19] 陈皞. 移动应用系统的设计与实现[D]. 北京邮电大学, 2012:18-43
- [20] 王欣. 跨平台移动应用研究与实现[D]. 北京邮电大学, 2013:5-41
- [21] 刘海岩, 梁建龙. 基于中间件的分布式系统开发过程的研究[J], 《计算机应用研究》, 2014:21-23
- [22] 设计模式-可复用面向对象软件的基础[M]. 背景. 机械工业出版社, 2012:2-89
- [23] 朱华旻. AJAX 在 WEB 开发中的应用研究[D]. 哈尔滨. 哈尔滨工程大学, 2013:15-46
- [24] W3C. HTML5 differences from HTML4[N]. www.w3.org/TR/html5-diff

致 谢

时光飞逝，在深圳大学共度软件工程硕士的日子也已经到了尾声。回想起这几年的工作与学习的生活，各种困难与快乐，仿佛还历历在目。即将临别之际，在论文进入尾声，以此致谢，表达我的感激之情。

首先，感谢我的指导老师朱映映副教授。在论文的撰写过程中，朱老师以其严谨的治学态度，诚恳育人的心，给予了我许多的帮助和指导。朱老师对本文所提的宝贵意见，敦促我不断精益求精，孜孜以求，使此篇论文得以更加严谨和完备，成为一篇合格的学术论文。朱老师不论在治学上还是为人处事上，都是我学习的榜样。

其次，感谢学园的各位老师和领导，感谢他们的帮助和鼓励，在上课的过程中，他们的课程给我流下了深刻的印象，我在其中获益良多，帮助我在工作中能够以更深刻技术积累，使得工作更进一步。

最后，在共度软件工程硕士的过程中，我结实了许多志同道合的同学们。他们来自五湖四海，从事不同的专业工作。在于同学们的交流中，我获取了别处获取不到的实战经验，从他们身上，学到了很多书本上学不到的知识和经验。我非常珍惜这段同学友情，他们是我人生难得的财富。走出校园，友情还将延续，人生也因而更加丰富灿烂。