



厦門理工學院
XIAMEN UNIVERSITY OF TECHNOLOGY

毕 业 设 计

中文题目 基于跨平台技术的智慧校园应用的开发与
实现

英文题目 **Development and Implementation of
Intelligent Campus Application Based on
Cross Platform Technology**

院 系:	软件工程学院
年级专业:	2013 级软件工程（移动云办公）
姓 名:	吴俊川
学 号:	1321152232
指导教师:	肖伟东
职 称:	教授

2017 年 5 月 1 日

毕业设计（论文）诚信声明书

本人郑重声明：在毕业设计（论文）工作中严格遵守学校有关规定，恪守学术规范；我所提交的毕业设计（论文）是本人在肖伟东指导教师的指导下独立研究、撰写的成果，设计（论文）中所引用他人的文字、研究成果，均已在设计（论文）中加以说明；在本人的毕业设计（论文）中未剽窃、抄袭他人的学术观点、思想和成果，未篡改实验数据。

本设计（论文）和资料若有不实之处，本人愿承担一切相关责任。

学生签名：

年 月 日

基于跨平台技术的智慧校园应用的开发与实现

【摘 要】 伴随着互联网技术与行业的快速发展，各类移动终端应用的开发需求出现了爆炸性增长，这导致了移动互联网开发者人才的紧缺。当前智能手机市场主要由 iOS、Android 这两个移动端操作系统平台瓜分。移动操作系统平台中 iOS 封闭性与 Android 开源性特性，都在不同方面导致开发速度缓慢、问题多，开发软件应用产品成本大大增加。同一款应用在不同的平台上通常拥有绝大多数共同的业务逻辑、相似的用户界面。但由于两个不同操作系统阵营的存在，导致这些代码并无法直接复用。本项目将尝试使用 Facebook 开源的 React Native 库实现 iOS 与 Android 的跨平台开发。

大学生是一个特殊的群体，他们更容易接受新技术与新应用。如今几乎每位大学生都有一部以上的移动智能设备，因此移动应用对于大学生群体而言并不陌生。在此基础上，为了服务广大学生群体，以用户体验为基准，结合校园的实际需求，根据用户需求，提出基于跨平台技术的智慧校园应用的开发与实现。

【关键词】 混合应用开发，跨平台开发，智慧校园，校园资讯

Development and Implementation of Intelligent Campus Application Based on Cross Platform Technology

【Abstract】 With the development of Internet Technology and industry, The development of various types of mobile terminal applications demands increase rapidly. Based on this situation it lead to the shortage of mobile Internet developers. The current smart phone market is dominated by iOS, Android two mobile operating system platform. Because of mobile operating system platform between iOS and Android closed-opened features, are in different ways led to the development of slow and more problems. The cost of developing software applications has increased significantly. One application on different platforms usually has the vast majority of common business logic, similar user interface. However due to the two different operating system camp, leading to codes can't direct reuse. This project will attempt to use the React Native Library which is open source code by Facebook to implement iOS and Android cross platform developing.

College students are a special group, they are more receptive to new technologies and new applications. Almost every college student has more than one mobile smart device nowadays. So college students is no stranger to mobile applications. on the basis of these situations, in order to serve the broad masses of students. Based on user experience ,combined with the actual needs of the campus, according to user needs. So I put forward development and realization of intelligent campus application based on cross platform technology.

【Keywords】 Hybrid application development 、 Cross-platform development、 Wisdom campus、 Campus information

目 录

第一章 引言	1
1.1 研究背景及意义	1
1.2 国内外研究现状	1
1.3 本文研究内容	2
1.4 论文的组织结构	2
第二章 系统相关技术	4
2.1 前端技术	4
2.1.1 React 介绍	4
2.1.2 React Native 技术概述	4
2.2 后端技术	5
2.2.1 Node.js 介绍	5
2.2.2 Express 技术概述	5
2.3 数据库技术	5
2.4 其他技术	6
2.5 本章小结	6
第三章 系统分析	7
3.1 系统调查	7
3.1.1 系统初步调查	7
3.1.2 可行性分析	7
3.1.3 系统详情调查	8
3.2 系统需求分析	8
3.3 本章小结	10
第四章 系统设计	11
4.1 系统设计目标与设计原则	11
4.2 系统总体设计	11
4.3 系统详细设计	13
4.3.1 系统模块设计	13

4.3.2 系统界面设计	14
4.3.3 组件化设计	15
4.4 代码设计	15
4.5 数据库设计	16
4.5.1 系统 ER 图	16
4.5.2 系统数据表	17
4.6 本章小结	18
第五章 系统实现与测试	19
5.1 系统环境	19
5.1.1 安装 Homebrew 及相关软件依赖	19
5.1.2 安装 React Native	19
5.1.3 部署 Express 项目	19
5.1.4 初始化 MariaDB	20
5.2 系统的关键实现	20
5.2.1 程序的 UI 框架的实现	20
5.2.2 工具类库的封装	21
5.2.3 程序调试	22
5.3 系统测试	24
5.3.1 学院新闻模块测试	24
5.3.2 图书管理模块测试	26
5.3.3 便利服务模块测试	27
5.4 导出项目	30
5.4.1 iOS 打包	30
5.4.2 Android 打包	31
5.5 本章小结	34
第六章 总结和展望	35
6.1 总结	35
6.2 展望	35
致 谢	37
参考文献	38

Contents

Chapter 1	Introduction.....	1
1.1	Research Background And Significances.....	1
1.2	Research Status at Home and Abroad	1
1.3	Research Content.....	2
1.4	The Organizational Structure of the Dissertation	2
Chapter 2	System Technologies	4
2.1	Font-end Technologies	4
2.1.1	Introduction to React	4
2.1.2	Overview of React Native Technology	4
2.2	Back-end Technologies	5
2.2.1	Introduction to Node.js	5
2.2.2	Overview of Express Technology	5
2.3	Database Technology	5
2.4	Other Technologies.....	6
2.5	Summary.....	6
Chapter 3	System Analysis.....	7
3.1	System Investigation	7
3.1.1	Preliminary System Investigation.....	7
3.1.2	Feasibility Analysis	7
3.1.3	Detailed Investigation of System.....	8
3.2	System Requirement Analysis.....	8
3.3	Summary.....	10
Chapter 4	System Design	11
4.1	Principles and Objectives of System Design.....	11
4.2	Overall System Design	11
4.3	Detailed System Design	13
4.3.1	System Module Design.....	13
4.3.2	System User Interface Design	14
4.3.3	Component Design	15
4.4	Code Design	15

4.5 Database Design.....	16
4.5.1 System ER Diagram.....	16
4.5.2 System Data Sheet	17
4.6 Summary	18
Chapter 5 System Implementation and Test	19
5.1 System Environment	19
5.1.1 Install Homebrew and Related Software Dependencies	19
5.1.2 Install React Native	19
5.1.3 Deploy Express Project.....	19
5.1.4 Initialize MariaDB	20
5.2 Key Implementation of System.....	20
5.2.1 Implementation of Program UI framework	20
5.2.2 Encapsulation of Tool Library	21
5.2.3 Program Debugging	22
5.3 System Testing	24
5.3.1 College News Module Test.....	24
5.3.2 Library Management Module Test	26
5.3.3 Convenience Service Module Test	27
5.4 Export Project	30
5.4.1 Packaging iOS Application.....	30
5.4.2 Packaging Android Application.....	31
5.5 Summary	34
Chapter 6 Conclusions and Future Work.....	35
6.1 Conclusions.....	35
6.2 Future Work	35
Acknowledgements.....	37
References.....	38

第一章 引言

1.1 研究背景及意义

伴随着移动互联网的快速发展，各类移动应用的开发需求呈现爆炸性增长，这导致了移动互联网开发者紧缺。当前智能手机市场主要由 iOS、Android 这两个移动端操作系统平台所瓜分。对于消费者而言，多了一种选择；但对于开发者而言则要面临各种适配问题的烦恼。同一款应用在不同的平台上通常拥有绝大多数共同的业务逻辑、相似的用户界面。但由于两个不同操作系统阵营的存在，导致这些代码并无法直接复用。移动操作系统平台中 iOS 封闭性与 Android 开源性特性，都在不同方面导致开发速度缓慢、问题多，开发软件应用产品成本大大增加。

软件工程中的代码复用一直都是软件开发从业工作者的追求目标。所谓的软件代码复用，就是在不改动原有代码的情况下，重复利用原有代码，从而实现在软件开发中避免重复枯燥劳动的方法论，实现软件开发成本的节省，进而有效地提高程序员的生产效率与代码的健壮性，降低代码耦合程度。在这种需求下，移动应用开发方式也逐渐丰富起来，快速、高性能、跨平台成为开发人员的热点。市面上已出现许多跨平台开发的解决方案，在某种程度上确实实现了跨平台开发，降低了开发成本。但是普遍都存在严重的性能问题。

普及高校信息化程度，建设智慧校园手机移动客户端已经成为了我国推进高等院校数字信息化系统进程中的重要组成部分。伴随着智能手机的普及化大众化，学生逐渐改变了他们获取信息的渠道与习惯，传统的媒体明显已经不能够满足学生的使用要求。

基于以上背景，为了服务广大学生群体，以用户体验为基准，结合校园的实际需求，根据用户需求，提出基于跨平台技术的智慧校园应用的开发与实现。

1.2 国内外研究现状

由于原生开发存在无法跨平台复用代码、适配麻烦等问题，所以现在普遍存在的解决方案是通过采用HTML5的解决方案，即利用HTML5跨平台的特性来实现客户端的展示与后台数据的交互。再通过WebView控件在每个平台上分别打包，所以可以做到一次开发，处处使用。这种方案很多，包括比较出名的PhoneGap与AppCan^[1]。国内流行的HTML5方案有5+Runtime搭配mui框架^[2]、或者APICloud^[3]。但是归其根本，它们都是对

原生组件的一种动画模拟，归其本质是一种 SPA（单页应用程序，Single Page Application）。这些通过浏览器技术模拟出来的组件效果远远不如原生组件，伴随着卡顿、掉帧等问题，导致用户体验差。特别是 WebView 组件在 Android 平台实现的版本实现无法统一，兼容性差，通常不能及时的释放内存，导致内存占用上升，甚至会引起应用奔溃闪退。降低开发成本、保证应用体验是移动平台开发一直追求和关注的两个核心点。在过去，移动开发中 Web 应用与原生应用之争以原生应用的胜利而告终，因为 Web UI 的性能还无法与原生控件相提并论。

移动互联网的普及化，开发应用技术的门槛逐渐降低，现在企业和学生自主创业的校园应用层出不穷，其中最为知名的是当属超级课程表与课程格子^[4]；有个别高校立项建设校园官方应用（例如清华大学、上海交通大学、北京邮电大学等知名高校）。由此可见服务移动化已经是大势所趋。

1.3 本文研究内容

本项目使用 Node.js 作为后端开发语言，采用 Express 框架，部署在阿里云 CentOS 上。前端使用了 React Native 跨平台开发技术，分别实现了厦门理工学院软件工程学院的校园应用的 Android 版本与 iOS 版本。校园新闻模块实现了校园新闻的抓取与详情页面的展示；图书管理模块实现了对图书的添加与借阅图书的管理，添加的时候只需要输入 ISBN 码即可完成对图书信息的采集；便利服务模块实现了天气查询、快递查询、宿舍电费查询、E 通卡查询、成绩查询等功能。这些功能都是通过后端去获取网页并且解析成 JSON 数据格式呈现给前台。

本系统的开发意义在于将 React Native 这项新技术尝试应用在实践中，以此来弥补原生开发上的缺陷与传统跨平台开发上的缺陷，同时解决在新的移动互联时代校园资讯聚合的难题，给校园生活增添一些便利性，方便校园师生生活，为高校信息化建设做贡献。

1.4 论文的组织结构

本论文分为六章来阐述，首先说明了研究的背景及意义，然后对系统所用到的技术进行简单的介绍，然后根据软件工程的一般开发流程分别从系统分析、系统设计、系统实施，测试，最后总结来组织论文。

第一章：引言，主要介绍说明了本项目的研究背景与研究意义，对应内容的国内外

研究现状，说明了主要的研究内容，并且对论文的大致结构进行阐述

第二章：系统相关技术，主要介绍说明了本项目所用到的技术，分别从前端技术栈和后端技术栈、数据库技术等其他技术方面介绍。

第三章：从系统的角度对智慧校园应用进行了分析，在系统调查过程中分析了这类校园移动应用的现实意义。从经济可行性与技术可行性两方面论证开发智慧校园的可行性。对智慧校园应用大体实现的功能模块进行介绍。

第四章：系统设计，主要介绍了系统的整体架构和系统的数据结构，模块的划分及基于组件化思想的 UI 界面设计。

第五章：系统实施，介绍系统运行所需要的环境配置，及主要模块的具体实现细节说明。

第六章：总结与期望，主要描述全文的工作总结和对未来工作的展望。

第二章 系统相关技术

2.1 前端技术

2015 年到 2016 年是前端界风起云涌的黄金时期，各种前端框架层出不穷，突破了长期以来 jQuery 在前端的统治地位，出现了像 Angular、React、Vue 这样的 MV* 前端框架。JavaScript 这门从出生到现在饱受争议的语言也开始大放光彩，开始向其他领域扩散。

2.1.1 React 介绍

2013 年 Facebook 在代码托管网站 GitHub 上开源了内部项目 React。React 专注于提供清晰、简洁的视图层解决方案。React 将用户界面抽象成一个个组件，开发者可以通过组合这些组件，最终可以得到功能丰富、可交互的页面。React 引入了 JSX 语法，使组件化变得更加容易，保证了组件结构的清晰。在 React 中，为了方便 View 层的组件化，JSX 承担了构建 HTML 结构化页面的职责^[5]。JSX 是类 XML 语法的 ECMAScript 的扩展，是创建 React 组件的一个语法糖。在 Web 开发中经常需要处理各种数据的变更而引起的 UI 变化，不可避免的需要对 DOM（Document Object Model，文档对象模型）进行操作，复杂频繁的对 DOM 操作通常是产生性能瓶颈的原因之一。因此 React 引入了 Virtual DOM 机制，来提高页面渲染效率。得益于 Virtual DOM 机制，潜在提供了强大的抽象能力。在开发者的代码与实际的渲染之间加了一层抽象层，所以 React 能够渲染到浏览器以外的其他平台上^[6]。

2.1.2 React Native 技术概述

正如 2.1.1 所述，得益于 Virtual DOM 机制，React 的具有渲染到浏览器以外的其他平台上的潜在能力。如图 2.1 所示，这就是 React Native 的工作原理。2015 年发布的 React Native 是继 Facebook 公司开源的 React 之后的又一个开源项目。React Native 通过 Virtual DOM 来调用 Objective-C 的 API 去渲染 iOS 组件，调用 Java 的 API 去渲染 Android 组件^[7]。因此 React Native 不同于那些基于 Web 视图的跨平台开发方案。因为调用原生组件，所以它的性能表现明显优于其他基于 Web 视图的跨平台开发方案^[8]。本项目将使用 React Native 0.35 作为开发库，以实现 Android、iOS 的跨平台开发。

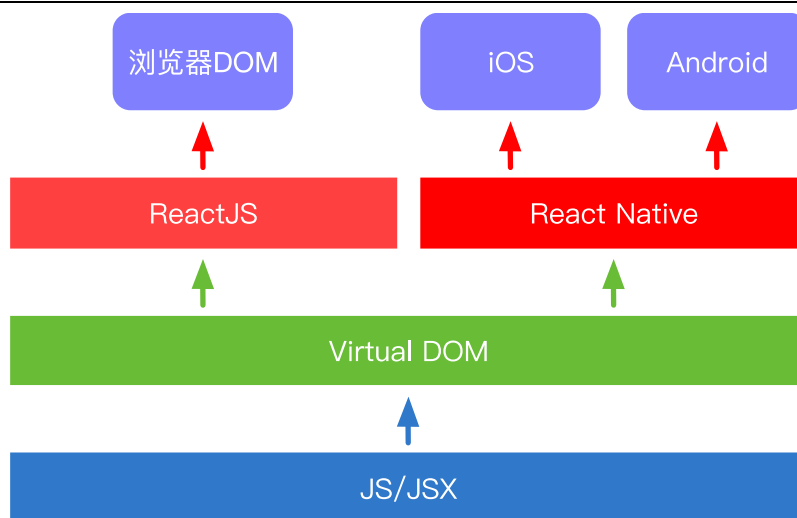


图 2.1 React 技术栈原理图

2.2 后端技术

2.2.1 Node.js 介绍

诞生于 2009 年 3 月的 Node.js 是一个开源、跨平台的、可用于服务器和网络应用的运行环境。Node.js 是一个基于谷歌 Chrome V8 引擎的 JavaScript 运行环境。它使用的是高效、轻量的事件驱动、非阻塞 I/O 模型。Node.js 包含的内置模块，使得程序可以独立 Apache HTTP Server 或 IIS 作为独立的服务器来运行。得益于 Node.js 的存在，JavaScript 可以运行在不同的地方，不再局限于浏览器中。前后端编程环境的统一可以大大降低前后端开发所需要的上下文转换的代价。本项目将采用 Node.js 6.9.4 版本作为运行环境。

2.2.2 Express 技术概述

Express 是 Node.js 基金会的一个项目，提供了基本的 Web 应用程序功能，提供各种 HTTP 程序方法和中间件，帮助开发者快速方便的创建 Web 应用程序。本项目将采用 Express 4.14.0 作为后端服务器应用程序框架。

2.3 数据库技术

MariaDB 是开源数据库 MySQL 的一个分支，主要由开源社区在维护，采用 GPL 授权许可。Oracle 公司收购了 MySQL 之后，有将 MySQL 闭源的风险，因此开源社区采用分支方式来规避这种风险。直到 MariaDB5.5 版本均兼容 MySQL 数据库。本项目将采用 MariaDB5.5 作为后端数据库。

2.4 其他技术

NPM 是 Node 的模块管理工具，是随同 Node.js 一起安装的包管理工具，能够方便地安装第三方模块，扩展已有的程序。

Homebrew 是 Mac OSX 上的软件包管理工具，可以在 Mac 运行环境中方便安装软件或者卸载软件，类似于 Linux 系统下的 apt-get。

Git 是一个开源免费的先进的分布式版本管理控制系统，分布式版本控制的每个节点都是完整仓库，因此具有小巧、快速、安全的特点。强大的分支管理可以让版本控制更为灵活。

2.5 本章小结

使用 React Native 作为智慧校园应用的客户端开发可以实现 Android、iOS 两大移动端的跨平台开发。使用这种技术开发出来的移动应用不同于其他普通的基于 Web 视图的跨平台开发。React Native 通过 JSX 来调用对应平台的原生组件，渲染效率显著优于其他基于 Web 视图的跨平台开发。使用 Node.js 作为后端服务器环境进行开发，可以减少前后端转换的上下文切换的成本，加快开发的速度。

第三章 系统分析

3.1 系统调查

3.1.1 系统初步调查

随着智能移动的时代的发展，智能手机在高校普及，人们越来越喜欢这种随时随地获取资讯的方式。现今国内高校都拥有自己的门户网站，但是这种传统的 PC 门户网站并不适合移动终端上浏览。如今有许多高校开发了自己的校园 APP，校园 APP 方便了学生，通过手机 APP 就可以完成在校生活的一些常见日常活动。甚至有创业团队、学生制作了校园 APP，比如说课程格子与超级课程表等等手机 APP。服务移动化趋势已经是大势所趋。校园应用的用户群体是特定学校的群体和教师群体，提供本地化、时效性、实用性的信息。提高校园信息化水平，优化校园信息服务质量是高校信息化建设的一个重要阵地。因此开发智慧校园应用符合当前高校信息化建设的需求。

3.1.2 可行性分析

1、经济可行性

部署校园应用 APP 服务器需要使用到云服务器，使用阿里云学生认证可以以极低的价格获取一台云服务。智慧校园应用拟采用的都是免费开源的技术，从经济的角度分析成本是极低的。开发应用的过程中可能需要用到一些第三方接口，例如天气接口与快递接口等。这些接口通常都拥有免费接口，仅调用次数有限制而已，但往往这些接口调用次数满足校园应用调用的需求，但是在项目前期使用这些接口就可以满足项目需要。所以从经济上是可行的。

2、技术可行性

使用阿里云购买的学生认证服务器，采用 Node.js 在 CentOS 系统上搭建后台，使用 Express 框架快速搭建后端应用处理移动端的请求，可以实现图书借阅管理、用户管理及各种爬虫的抓取。使用正则表达式可以完成对爬虫抓取的数据进行清洗。应用上采用 Facebook 开源免费的 React Native，通过 JSX 语法来编写应用程序代码，从而实现 Android 与 iOS 的跨平台开发。React Native 提供 Fetch API 使得网络交互更为方便，使用 AsyncStorage 组件可以实现本地数据库存取操作。因此在技术上是可行的。

3.1.3 系统详情调查

随着移动互联网的高速发展，移动应用给人们的生活学习上带来诸多便利。目前市场上实现一个移动端应用产品，客户端需要考虑两个平台，一个是苹果公司的 iOS 平台，另一个则是谷歌公司的 Android 平台。因为两个平台不同的语言特性及开放的接口不同，使得开发一个移动端应用程序的技术成本与用人成本大大增加。同一款应用在不同平台上通常拥有绝大部分共同的业务逻辑、相似的用户界面，但是两个平台间的代码迁移成本很大^[9]。

为了解决这种痛点，市面上流行的解决方案大都是基于 Web 视图的跨平台开发技术。这种采用 HTML5、CSS3、JavaScript 来实现客户端的展示和后台数据的交互，再通过 WebView 控件分别在每个平台上打包，从而实现了一次编写到处运行的效果^[10]。这种解决方案都是对原生组件的一种模拟，归其本质属于 SPA（单页应用程序，Single Page Application），这些通过浏览器技术模拟出来的组件效果远远不如原生组件，通常伴随着卡顿、掉帧等问题，导致用户体验差。

React Native 是一款用来开发真正原生、可渲染 iOS 和 Android 移动应用的 JavaScript 框架。React Native 在后台通过“桥接”的手段调用原生渲染接口，在不牺牲用户体验和应用质量的前提下，就可以实现跨平台开发。但是 React Native 推出的时间并不长，并且还没有到一个稳定的软件版本周期，Facebook 官方对其保持着一种快速迭代的开发状态中，经常会出现接口变更的情况。在使用过程中遇到的问题不如原生开发能够快速找到解决方案，这需要开发者花费较多时间去解决。React Native 的学习成本也是相对较高的，学习成本难度大。遇到一些没有现成组件的情况下，可能需要开发者动手去实现原生代码，这样的学习曲线就非常陡峭。

综上所述，考虑到智慧校园应用的受众群体不单单只局限于 iOS 或者 Android 平台上^[11]，并且智慧校园应用本身项目复杂度并不高，这类软件主要以校内资讯为主，所以十分适合混合模式移动应用软件的开发^[12]。在众多的混合模式移动应用的软件开发技术中，React Native 可以开发出媲美原生应用的一个框架，因此本项目选择使用 React Native 作为前台开发技术。

3.2 系统需求分析

智慧校园应用的主要受众群体是特定学校内的学生和老师，通过分析用户需求，针对厦门理工学院软件工程学院的发展状况，智慧校园手机 APP 主要分为三个功能模块，

其中包括生活查询、新闻阅读与图书借阅。

(1) 生活查询：用户可以通过智慧校园应用实现快递的查询、当地天气预报与当前天气的查询、宿舍电费的查询，查询校园 E 通卡的余额等等日常常用的查询功能。通过对教务系统的模拟登陆，可以获取该用户的成绩。

(2) 新闻阅读：用户可以通过智慧校园应用实现对学院新闻的获取，学院新闻是实时从学院新闻列表中抓取得到的。新闻阅读包括学院新闻、通知公告、党建工作三个模块。

(3) 图书借阅：用户可以通过智慧校园应用实现图书的借阅，图书管理员可以在后台看到借阅的用户，及快速上新书籍。

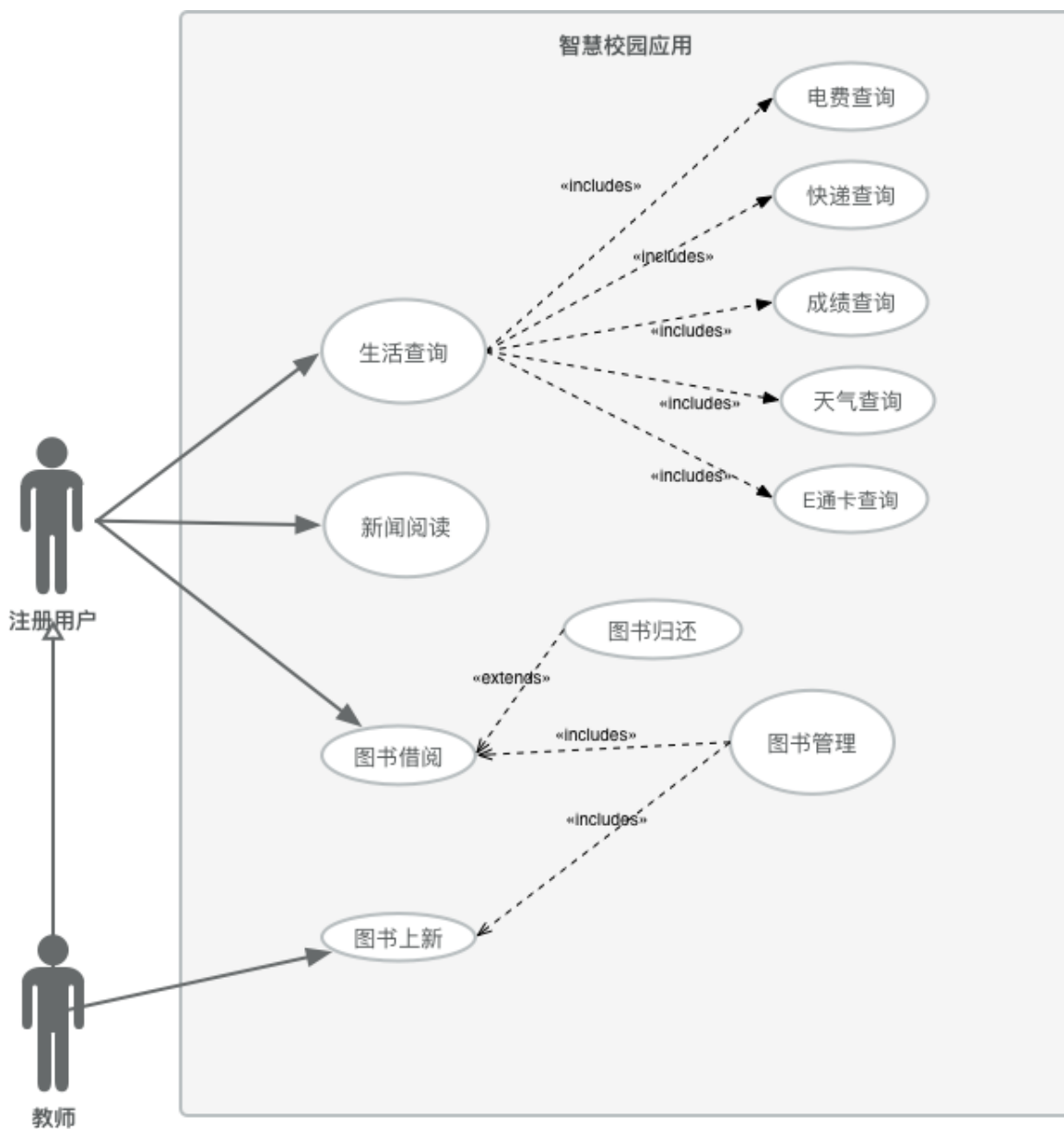


图 3.1 智慧校园应用用例图

3.3 本章小结

本章节从系统的角度对智慧校园应用进行了分析。从系统调查开始，分析了开发校园 APP 的现实意义。从经济可行性与技术可行性两个方面论证了开发智慧校园应用是现实可行的。在系统详情调查的过程中发现像智慧校园应用这本项目复杂度不高的情况下，使用混合移动应用开发是最经济、最快速的。使用 **React Native** 是其中的一种混合应用开发模式，虽然它本身有一些缺点，技术也还比较不稳定等一些缺点。但是它可以开发出媲美原生效果的移动应用，用户体验会好很多。综合本章的分析，决定使用 **React Native** 作为前台开发技术。

第四章 系统设计

4.1 系统设计目标与设计原则

通过对在校学生与老师的需求进行分析与调研，确定本系统的设计目标如下：建立满足用户需求的校园信息聚合平台，为在校师生的校园生活提供更多的便利性，随时随地获取资讯。

为了提高软件的适应性与实用性，达到上述目标，确定本系统的设计原则如下：整体保障性原则，在开发初期，先进行系统的总体规划，然后在其约束下分布开发；实用性原则，系统具有严格的业务针对性，一切从实际出发；人性化界面设计，操作简单，一般用户无需培训，使用 Sketch 在编码开发之前先设计好界面；模块化设计原则，利用前端组件化技术，使系统具备良好的伸缩性，可随时根据需要扩充模块^[13]。

4.2 系统总体设计

APP 的系统架构图如图 4.1 所示。客户端包含两个部分，一个是基于 C/S 架构的移动端；另一个则是基于 B/S 架构的 PC 端。移动端基于 React Native 实现跨平台开发。客户端通过互联网访问远程服务器，服务器分为两大部分。一个是业务/代理服务器，负责接收客户端发来的参数，进行转发数据请求或者数据库操作；另外一个资源服务器，负责响应业务/代理服务器的请求。当业务/代理服务器接收到客户端的请求时，根据请求种类进行转发。如果是业务类型的请求，则进行数据库操作，操作结束后告知客户端处理结果；如果是代理类型的请求，则转发请求，转而请求资源服务器，当获取到资源服务器的响应结果时，进行数据清洗，将响应结果过滤成 JSON 数据格式，转发给客户端^[14]。

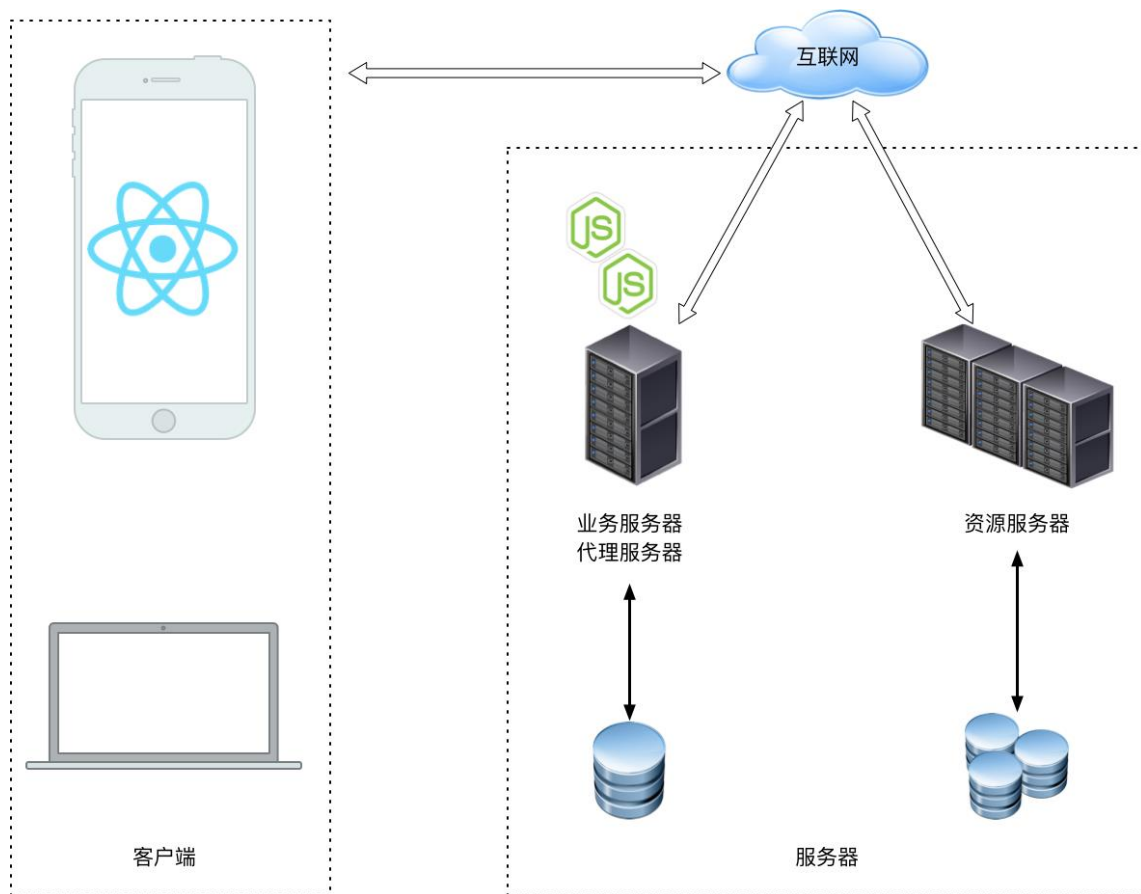


图 4.1 系统架构图

APP 的总体框架如图 4.2 所示, 登陆 APP 之后总体分为四个模块, 分别是学院新闻、图书管理、便利服务、个人中心模块。学院新闻模块实现校园官网的新闻聚合; 图书管理模块实现图书的借阅和图书信息的查看; 便利服务模块实现了天气查询、快递查询、宿舍电费查询、E 通卡查询、成绩查询等功能; 个人中心负责显示个人的信息和软件 APP 的版本信息, 提供意见反馈等操作。

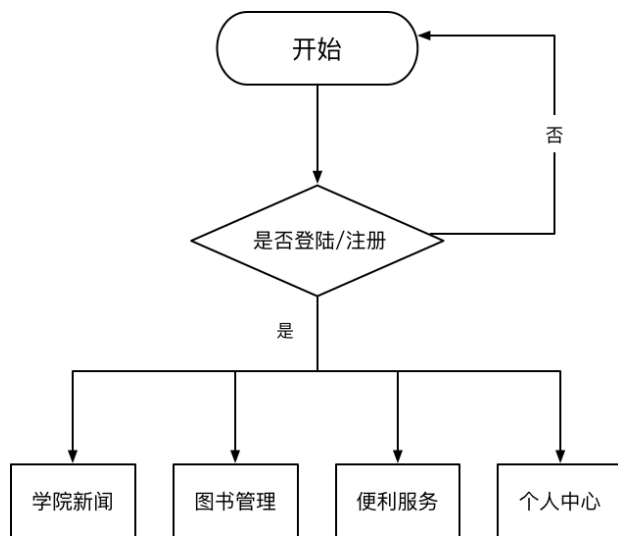


图 4.2 APP 总体流程图

4.3 系统详细设计

4.3.1 系统模块设计

学院新闻模块的流程如图 4.3 所示，该模块包含了学校新闻、党建工作、通知公告。这些模块都在学院的官网上有对应的专栏，所有的新闻都是实时从学院官网抓取的。点击具体的新闻，可以跳转到新闻详情页面查看新闻的具体内容。

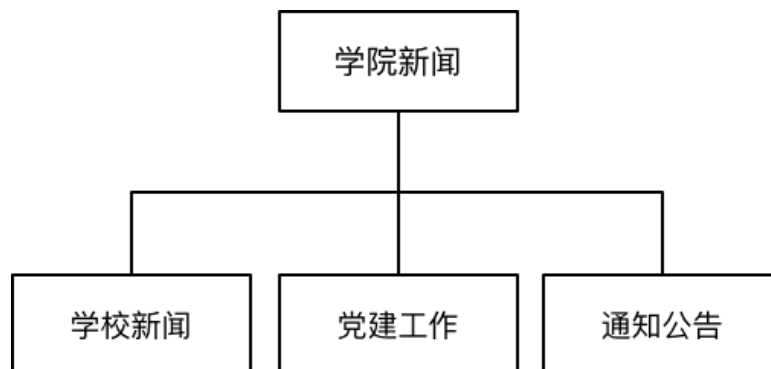


图 4.3 学院新闻模块

图书管理模块的流程如图 4.4 所示，该模块包含了图书上新、借阅图书、归还图书、查看图书信息等功能。这个模块分为两个平台来实现，在移动端上只有借阅的与查看图书信息的功能。在 PC 端上有图书上新与归还图书的功能。

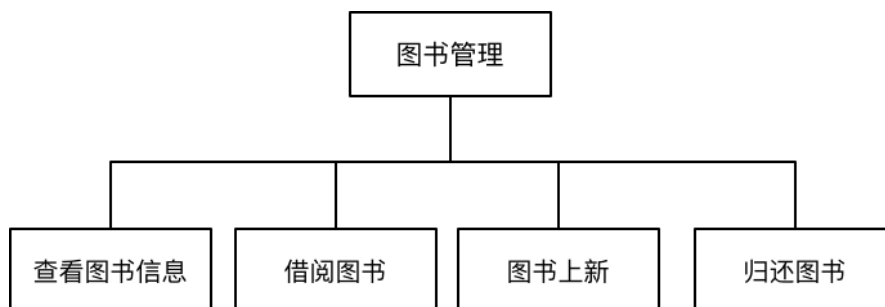


图 4.4 图书管理模块

便利服务模块的流程如图 4.5 所示，该模块包含了 E 通卡查询、当地天气状况及天气预报查询、快递查询、方正教务系统成绩查询等功能。

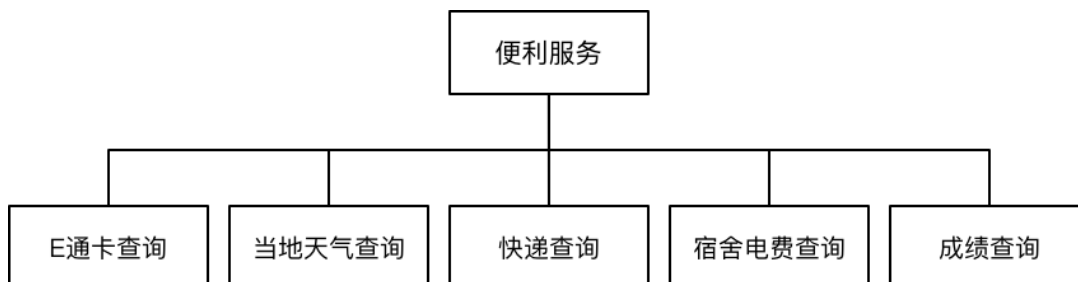


图 4.5 便利服务模块

4.3.2 系统界面设计

根据对用户需求的理解与系统的大体设计的基础上,使用 Sketch 勾勒了 iOS 下智慧校园应用界面的大体外观。Android 界面会在 iOS 基础上做平台上的规范适配。例如 iOS 平台的设计规范是提倡开发者使用右滑后退,将功能模块放在底部导航栏上;而 Android 平台的设计规范是提倡开发者使用 back 键实现后退操作,将功能模块放在侧滑栏中(在 Android 的官方组件称之为抽屉)^[15]。

根据智慧校园的主题与功能,设计了这样的图标(如图 4.6):



图 4.6 厦理助手 Icon

学院新闻模块界面设计,顶部导航栏可以切换学院新闻、通知公告、党建工作,支持左滑右滑进行切换。



图 4.7 学院新闻界面

4.3.3 组件化设计

为了提高代码的可复用性与可维护性，在设计界面的同时通常也进行组件的拆分与分解。如图 4.7 学院新闻界面的设计稿中，我是这样一步一步完成组件化的（如图 4.8）。



图 4.8 组件化过程

首先先找出界面中可能需要重复利用的部分，进行拆解；然后对各个组件进行分别的设计；最后像堆积木一样将其组装起来。拆分的粒度适中，不宜太细。如果拆分得太细，后期编码会很繁琐；拆分粒度太粗的话，则会导致组件不够灵活，影响后期修改的灵活性。

举个例子，就像学院新闻模块一样，首页至少包含了三个组件，分别是顶部导航栏与底部导航栏，中部是新闻列表页面。新闻列表页面包含两种新闻项的组件，分别是右侧有图片的组件与纯文本标题的组件。

4.4 代码设计

根据系统模块的划分及 UI 界面设计，遵循软件工程代码设计过程的六大基本原则。设计了如图的所示的代码组织结构图。首先 APP 启动的时候 React Native 会判断当前的运行环境是 iOS 还是 Android 平台，对应调用不同的 index 文件。对不同平台进行差异性处理。根据单一职责原则，按照业务模块的划分分别划分出 News、Service、Library、About 四大组件。除了进行平台差异性处理的组件不能 iOS、Android 之间共用以外，其他的组件都能够复用，满足代码复用原则。

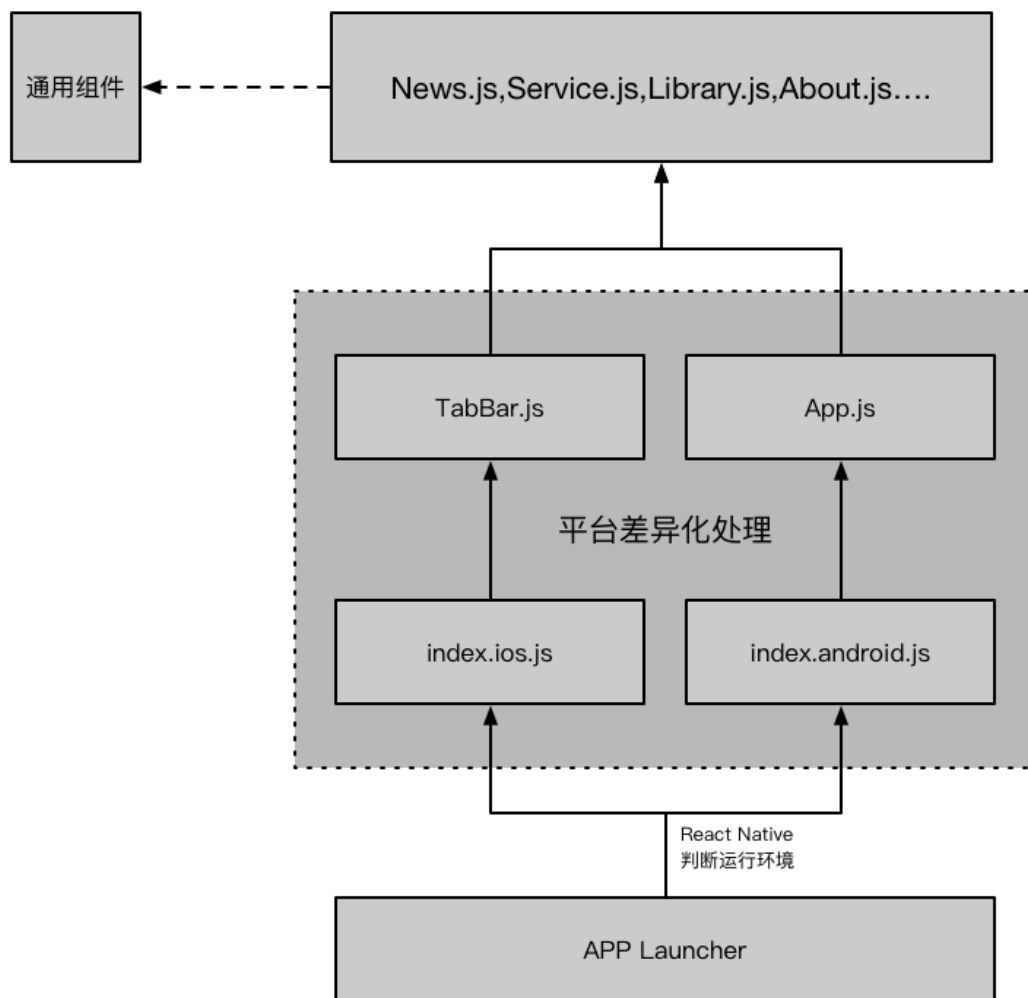


图 4.9 代码设计

4.5 数据库设计

4.5.1 系统 ER 图

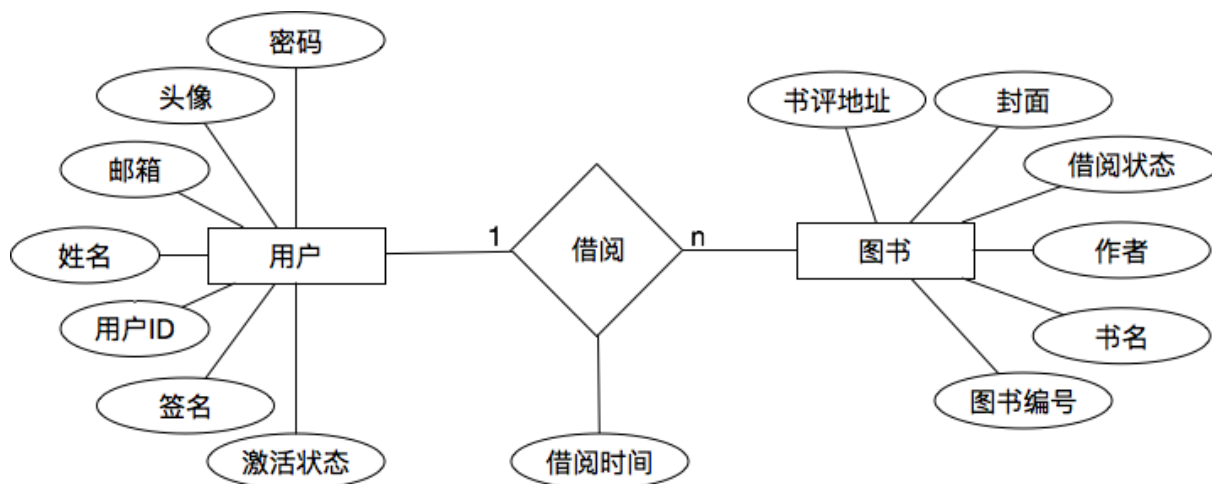


图 4.10 系统实体关系图

智慧校园应用只有在图书管理系统模块中才有用到数据库，根据对图书管理的业务流程分析，可以获得用户、图书这两个实体。如图 4.10 所示，用户、图书两个实体通过借阅产生联系，用户与图书的借阅关系是一对多的关系。也就是一名用户可以借阅多本图书。

4.5.2 系统数据表

通过对实体关系图的分析，我们分析得到了三张表，分别是：用户信息表、图书信息表、借阅信息表。下面分别详细介绍这三张表的表结构。

表 4.1 用户信息表（users）

字段名	说明	类型	长度	是否为空	主键
id	用户的系统 ID	Int	11	否	是
name	名称	Varchar	45	否	否
email	邮箱	Varchar	100	否	否
password	密码	Varchar	100	否	否
avatar	头像	Varchar	100	是	否
signature	个性签名	Varchar	100	是	否
active	激活状态	Int	4	是	否

用户登录后返回用户个人信息的字典数据，保存到本地存储中，便于后期操作标识。如表 4.1 所示，id 是用户的系统 ID，唯一标识一个用户，name 是用户的姓名，email 是邮箱地址，password 是登陆密码，avatar 是头像的存储地址，signature 是用户的个性签名，active 是用户的激活状态。在数据库中命名为 users 表。

表 4.2 图书信息表结构（book）

字段名	说明	类型	长度	是否为空	主键
id	图书的系统 ID	Int	11	否	是
name	图书名称	Varchar	45	否	否
url	书评地址	Varchar	45	否	否
author	作者	Varchar	100	否	否
image	封面图片路径	Varchar	45	是	否
statusId	借阅状态 id	Int	4	是	否
statusDesc	借阅状态	Varchar	45	是	否
borrower	借阅者	Varchar	45	是	否

如表 4.2 所示，id 是图书记录的系统 ID，唯一标识一个图书记录。name 是图书名称，url 是豆瓣的书评地址，author 是作者，image 是封面图片路径地址，statusId 是借阅状态代码编号，statusDesc 是借阅状态的详细解释，borrower 是借阅者的 id。在数据库中命名为 books 表。当后端收到添加图书的请求时，根据 ISBN 码去抓取图书信息，并将 name、url、author、image 写入这张表中。当借阅图书时，将借阅者的用户 id 写入 borrower 字段中，并更新 statusId 与 statusDesc 状态；当归还图书时，则将 statusId、statusDesc 复原，borrower 清空。

表 4.3 借阅记录表结构

字段名	说明	类型	长度	是否为空	主键
id	借阅记录的系统 ID	Int	11	否	是
bookId	图书 id	Int	11	是	否
userId	借阅者 id	Int	11	是	否
statusId	借阅状态 id	Int	4	是	否
date	借阅时间	Varchar	45	是	否

如表 4.3 所示。id 是借阅记录的系统 ID，是系统自动产生的，用来标识唯一的借阅记录；bookId 是借阅的图书系统 id，与 books 表中的 id 关联；userId 是借阅者的用户 id，与 users 表中的 id 关联。这表主要用来存储与图书相关的操作记录。

4.6 本章小结

本章节从系统设计的角度对智慧校园应用进行了分析，确定了系统设计目标应以满足在校师生的日常生活需求为标准依据，以提高软件适应性与实用性为设计原则；基于对用户需求的收集与理解的基础上，从系统整体结构的角度确定了系统架构图和 APP 的总体框架，接着对 APP 的每个模块进行细化设计；对数据库结构进行设计；运用组件化思想对 UI 界面进行初步设计，基本勾勒出应用的大体框架，为后面的编码实现工作打下坚实的基础。

第五章 系统实现与测试

5.1 系统环境

5.1.1 安装 Homebrew 及相关软件依赖

Homebrew 是 MacOS 上的一个套件管理工具，使用它可以方便的安装运行环境所需要的软件依赖包。系统环境安装基于这个工具来进行。因此第一步就是安装 Homebrew 套件管理器。打开终端，输入命令（如图 5.1）：

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

图 5.1 安装 Homebrew

安装完 Homebrew 后，可以一键部署 Git、Node.js、watchman（Facebook 用于提供监视文件系统变更的工具，便于提高开发效率实现实时刷新）、MariaDB。打开终端，输入命令：

```
brew install node & brew install git && brew install watchman && brew install mariaDB
```

即可完成安装系统环境。

5.1.2 安装 React Native

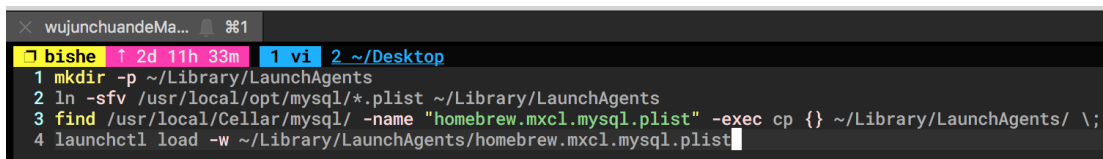
利用 Node 的包管理器 NPM 全局安装 React Native CLI。打开终端输入命令：npm install -g react-native-cli。使用 Facebook 提供的 React Native 脚手架可以快速初始化一个 React Native 项目。打开终端输入命令 react-native init ReactNative_intelliSchool。然后输入 cd ReactNative_intelliSchool。执行 react-native run-ios 即可运行 iOS 版本。如果是要运行 Android 版本，则执行 react-native run-android。

5.1.3 部署 Express 项目

打开终端命令行，执行 npm init -y 即可完成初始化项目。执行 npm install express -s 即可完成安装 Express 库的支持。同理在开发过程中遇到需要依赖的第三方库，可以执行 npm install someLibrary -s。带上参数-s 是为了将依赖记录在 package.json 文件中，别人使用项目是只需要执行 npm install，系统就会根据 package.json 文件内容对应去下载依赖包。

5.1.4 初始化 MariaDB

MariaDB 向下兼容 MySQL，因此它的命令行也是使用 MySQL 作为调用关键词。在 Windows 上安装 MySQL 较为容易，在 Mac 上需要手动修改配置文件实现配置。当使用 Homebrew 完成对 MySQL 的安装之后，在终端上执行 `mysql_secure_installation` 进入初始化向导，完成对 MySQL 的账号密码配置。如果能让 MySQL 开机自动启动，需要在终端内输入（如图 5.2）：



```
wujunchuandeMa... %1
bishe 1 2d 11h 33m 1 vi 2 ~/Desktop
1 mkdir -p ~/Library/LaunchAgents
2 ln -sfv /usr/local/opt/mysql/*.plist ~/Library/LaunchAgents
3 find /usr/local/Cellar/mysql/ -name "homebrew.mxcl.mysql.plist" -exec cp {} ~/Library/LaunchAgents/ \;
4 launchctl load -w ~/Library/LaunchAgents/homebrew.mxcl.mysql.plist
```

图 5.2 初始化 MariaDB

随后执行 `mysql.server start`，启动 MariaDB 服务。

5.2 系统的关键实现

5.2.1 程序的 UI 框架的实现

程序的 UI 框架分别有两种实现，一种是 iOS 平台上的，一种是 Android 平台上的，都使用了 React Native 的开源组件。iOS 平台上的 UI 框架是基于 TabBariOS 的实现；Android 平台上的 UI 框架是基于 DrawerLayoutAndroid 实现的。基于 React Native 的开源组件 TabBariOS、DrawerLayoutAndroid 分别封装了两个组件。在启动应用时判断，分别加载这两个组件，实现不同平台不同界面。效果如图 5.3 与图 5.4 所示。

就像单页应用一样，移动端应用通常因为屏幕小，所以经常需要实现多个界面的切换。React Native 有两个组件可以实现场景的切换，分别是 Navigator 和 NavigatoriOS。其中 Navigator 是适配 Android 和 iOS，而 NavigatoriOS 则封装了 UIKit 的导航功能，可以使用左滑功能回到上一级页面。因此在 iOS 平台上采用 NavigatoriOS，在 Android 平台上采用 Navigator。



图 5.3 iOS 界面



图 5.4 Android 界面

5.2.2 工具类库的封装

在开发的过程中经常需要读取屏幕尺寸信息、和网络请求业务逻辑。对此本应用封装了一个 Utils.js 文件，用于封装常用的开发方法。

命名为 getScreenParam 的函数，通过调用 React Native 暴露的 PixelRatio 接口读取设备屏幕像素密度；调用 React Native 暴露的 Dimensions 接口读取设备屏幕的尺寸。

使用 Fetch 标准库可以方便的发起网络请求，对此本应用封装了 post 方法和 get 方法，对应与 HTTP 协议中的 POST 与 GET。如图 5.5 的 post 方法中，先对网络请求进行了参数的配置与读取。随后调用 Fetch 标准库发起请求，在成功的回调函数中调用 response.json() 对其数据进行 JSON 格式化。

```
/**
 * 对post方法的封装,底层为fetch方法
 * @param url
 * @param data
 * @param callback
 */
static post(url, data, callback,errorCallback=()=>{}) {
  const fetchOptions = {
    method: 'POST',
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
  };
  fetch('http://api.caogfw.cn:10017/api'+url, fetchOptions)
    .then((response) => {
      return response.json();
    })
    .then((responseData) => {
      callback(responseData);
    })
    .catch((error)=>{
      errorCallback(error);
      console.log('[Fetch Error] + error.message);
    });
}
```

图 5.5 封装 Fetch 库

5.2.3 程序调试

使用 React Native 的专用工具可以帮助我们更加轻松地完成工作。在 iOS 模拟器上通过快捷键 Command+Control+Z 来触发,也可以通过按下 Android 的硬件按钮或摇晃设备来触发。(如图 5.6)

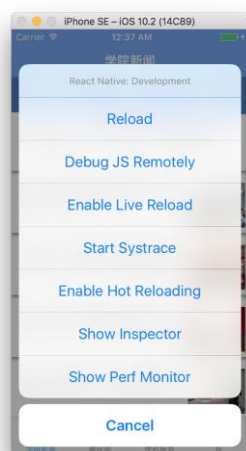


图 5.6 显示调试工具

在菜单中选择 **Debug in Chrome** 可以启动前端开发工程师们所熟悉的 **Chrome** 开发者工具。在这里你可以看到所有信息输出，观察发生了什么。也可以通过打断点来调试程序。（如图 5.7）

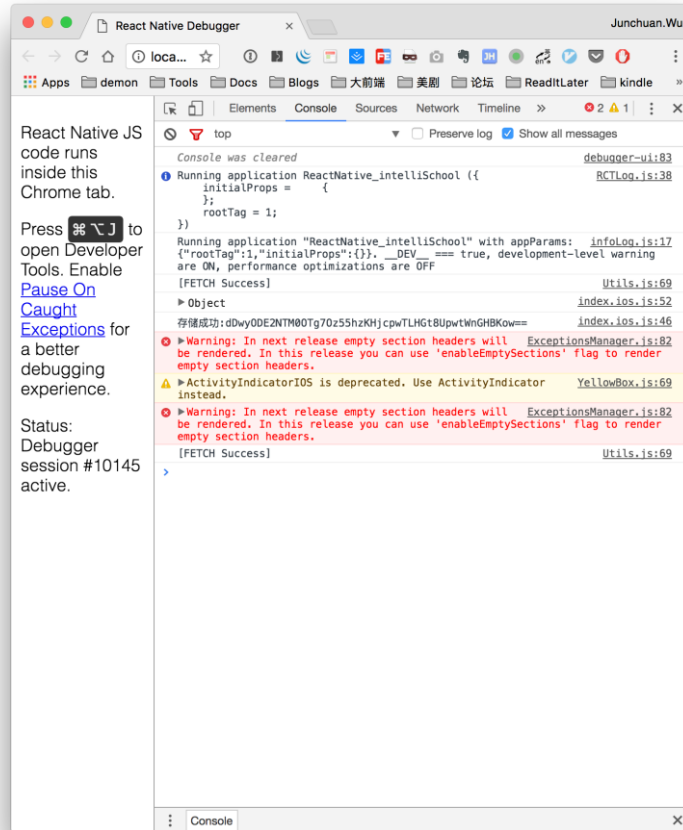


图 5.7 在 Chrome 中调试

使用 **Enable Live Reload** 可以实现修改代码后界面实时更新。除此之外在开发应用界面的时候，还可以使用内建的审查元素功能。它支持查看样式信息，提供便捷的挖掘组件层次的功能（如图 5.8）。通过 **Network** 功能还可以查看网络请求信息（如图 5.9）。通过使用内置的开发工具可以帮助我们更好更快地对 **React Native** 项目进行开发，提高开发效率，降低前端开发工程师跨域开发的不适感。



图 5.8 审查元素



图 5.9 查看网络请求信息

5.3 系统测试

5.3.1 学院新闻模块测试

测试校园新闻、通知公告、党建工作显示是否正常，点击对应的新闻标题是否跳到对应的新闻页面。

首先我们打开应用查看校园新闻栏目的新闻列表（图 5.10），对照厦门理工学院软件工程学院的校园官网里的学院新闻板块的新闻列表（图 5.11），可以发现两者是一样的。



图 5.10 App 里的新闻列表



图 5.11 校园官网的新闻列表

为论文结构简洁，本文就不累述类似的测试。依次通过测试通知公告与党建工作的新闻列表与校园官网的新闻列表作对比，可以得出新闻列表显示正常。

随机点击一项新闻《喜报！我院学子荣获 2016 年首届大学生软件测试大赛全国二等奖》可以跳转对应的新闻详情页面（如图 5.12）。说明新闻列表项与具体新闻的详情页面对应正确。



图 5.12 新闻详情页

5.3.2 图书管理模块测试

测试图书馆图书是否能够添加，是否能够跳转到正确的豆瓣书评页面，是否能够正常的借阅与归还。

假设图书馆新进一系列图书，其中有一本书的 ISBN 号是 9787111162209。进入图书管理首页/kit/library/点击添加图书，如图 5.13 所示，输入 ISBN 码及相关字段：

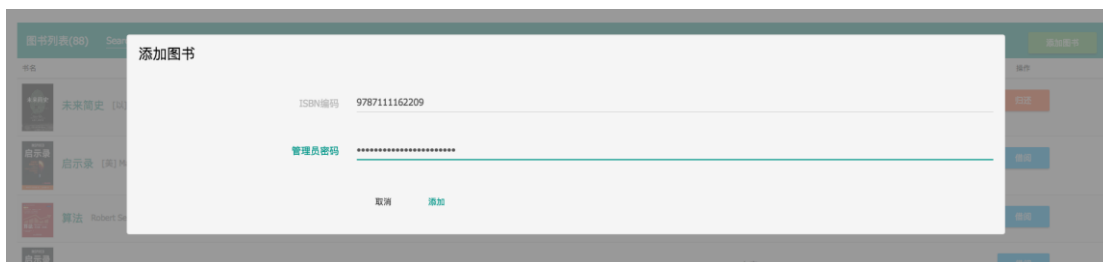


图 5.13 添加图书

如图 5.14 打开移动端的智慧校园应用，可以查看到已经出现刚才添加的图书《Java 编程思想》。如图 5.15 点击可以跳转到豆瓣书评，查看图书信息。



图 5.14 图书列表



图 5.15 图书信息

5.3.3 便利服务模块测试

测试便利服务中的天气预报、电费查询、快递查询、E 通卡查询是否能够正常工作。我们通过抓包的形式来验证参数是否正确传递，收到的 JSON 数据是否正确，显示是否正确。

如图 5.16 所示，当点击天气预报的选项时，能够正确获取到当地的最新天气预报。并且返回的 JSON 数据与显示的结果一致。测试通过。

如图 5.17 所示，点击查询宿舍电费，输入宿舍号 E510。宿舍号有被正确传递，并且返回的 JSON 数据与显示的一致。测试通过。

如图 5.18 所示，点击快递查询并输入快递单号 1901777930484。快递单号有被正确传递并且正确匹配快递公司，返回 JSON 数据，并且在界面上正确显示。测试通过。

如图 5.19 所示，点击 E 通卡查询并输入 E 通卡编号 0125150190。E 通卡编号有被正确地传递，正确地返回 JSON 数据，并且在界面上正确显示。测试通过。

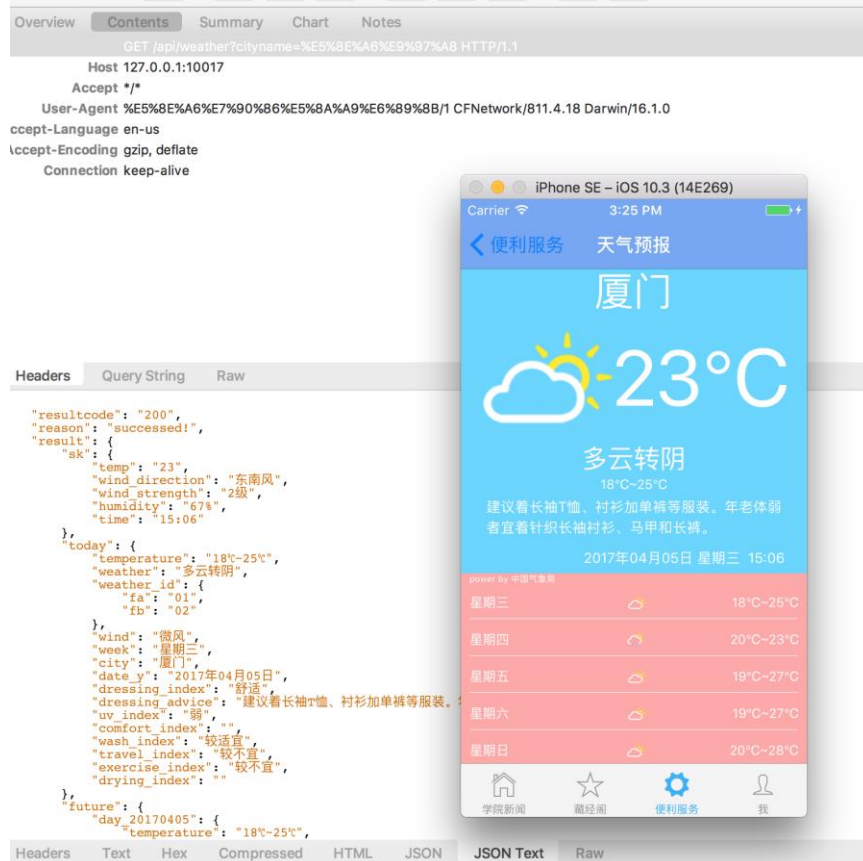


图 5.16 天气预报

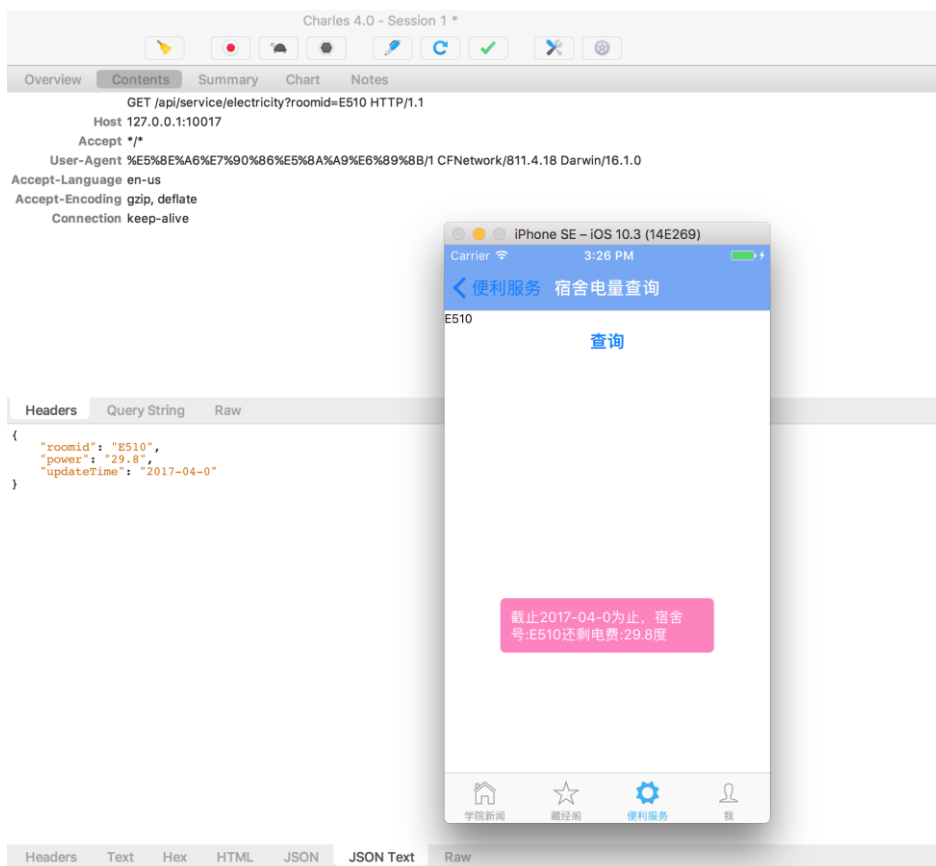


图 5.17 电费查询

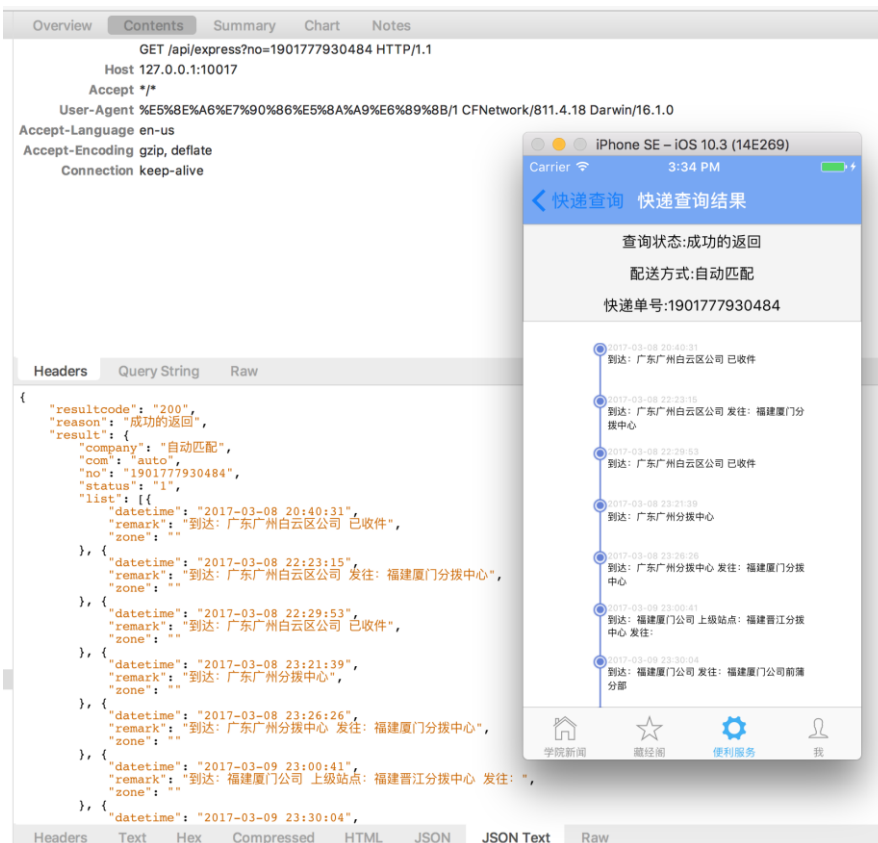


图 5.18 快递查询



图 5.19 校园卡查询

5.4 导出项目

5.4.1 iOS 打包

通过查看命令参数可得打包的命令如图 5.20:

```
wujunchuan /Users/wujunchuan/Project/code/ReactNative_intelliSchool master
react-native bundle --entry-file index.ios.js --platform ios --dev false --bundle-out
put ./ios/bundle/index.ios.jsbundle --assets-dest ./ios/bundleReact
```

图 5.20 打包参数

为了方便使用，我们将其写入 npm script 中，这样我们就可以通过 npm run bundle-ios 直接进行打包。

```
7 {
6   "name": "ReactNative_intelliSchool",
5   "version": "0.0.1",
4   "private": true,
3   "scripts": {
2     "start": "node node_modules/react-native/local-cli/cli.js start",
    "test": "jest",
    "bundle-ios": "react-native bundle --entry-file index.ios.js --platform ios --dev false --bundle-output ./ios/bundle/index.ios.jsbundle --assets-dest ./ios/bundle",
    "dev": "react-native run-ios --simulator= iPhone SE"
  },
1  "dependencies": {
2    "react": "15.3.2",
3    "react-native": "0.35.0",
4    "react-native-button": "~1.7.1",
5    "react-native-gifted-listview": "0.0.15",
6    "react-native-loading-spinner-overlay": "~0.4.1",
7    "react-native-material-design": "~0.3.7",
8    "react-native-modalbox": "~1.3.9",
9    "react-native-scrollable-tab-view": "^0.6.0",
10   "react-native-swiper": "^1.5.1",
11   "react-native-vector-icons": "^2.1.0"
12 },
13 "jest": {
14   "preset": "jest-react-native"
15 },
16 "devDependencies": {
17   "babel-jest": "16.0.0",
18   "babel-preset-react-native": "1.9.0",
19   "jest": "16.0.1",
20   "jest-react-native": "16.0.0",
21   "react-test-renderer": "15.3.2",
22   "whatwg-fetch": "1.0.0"
23 }
24 }
25 }
26 }
```

图 5.21 npm 脚本

在进行打包之前先确保./iOS/bundle 文件夹是否有存在。打包结果如图 5.22 所示。如果没有错误发生的话，就可以在./iOS/bundle/下查看到生成的离线资源文件。

如图 5.23，将 Xcode 构建配置修改为 Release 模式，将禁用掉开发者工具菜单并且从本地读取 Bundle。这样就完成了项目的 iOS 打包工作，生成的文件在 Products 下。

```
wujunchuan /Users/wujunchuan/Project/code/ReactNative_intelliSchool develop npm run bundle-ios
> ReactNative_intelliSchool@0.0.1 bundle-ios /Users/wujunchuan/Project/code/ReactNative_intelliSchool
> react-native bundle --entry-file index.ios.js --platform ios --dev false --bundle-output ./ios/bundle/index.ios.jsbundle --assets-dest ./ios/bundle

Unable to parse cache file. Will clear and continue.
[4/3/2017, 10:48:42 PM] <START> Building Dependency Graph
[4/3/2017, 10:48:42 PM] <START> Crawling File System
[4/3/2017, 10:48:42 PM] <START> Finding dependencies
[4/3/2017, 10:48:45 PM] <END> Crawling File System (3612ms)
[4/3/2017, 10:48:45 PM] <START> Building in-memory fs for JavaScript
[4/3/2017, 10:48:46 PM] <END> Building in-memory fs for JavaScript (267ms)
[4/3/2017, 10:48:46 PM] <START> Building in-memory fs for Assets
[4/3/2017, 10:48:46 PM] <END> Building in-memory fs for Assets (196ms)
[4/3/2017, 10:48:46 PM] <START> Building Haste Map
[4/3/2017, 10:48:46 PM] <START> Building (deprecated) Asset Map
[4/3/2017, 10:48:46 PM] <END> Building (deprecated) Asset Map (90ms)
[4/3/2017, 10:48:46 PM] <END> Building Haste Map (331ms)
[4/3/2017, 10:48:46 PM] <END> Building Dependency Graph (4438ms)
transformed 608/608 (100%)
[4/3/2017, 10:48:56 PM] <END> Finding dependencies (13803ms)
bundle: start
bundle: finish
bundle: Writing bundle output to: ./ios/bundle/index.ios.jsbundle
bundle: Copying 16 asset files
bundle: Done writing bundle output
bundle: Done copying assets
```

图 5.22 打包结果

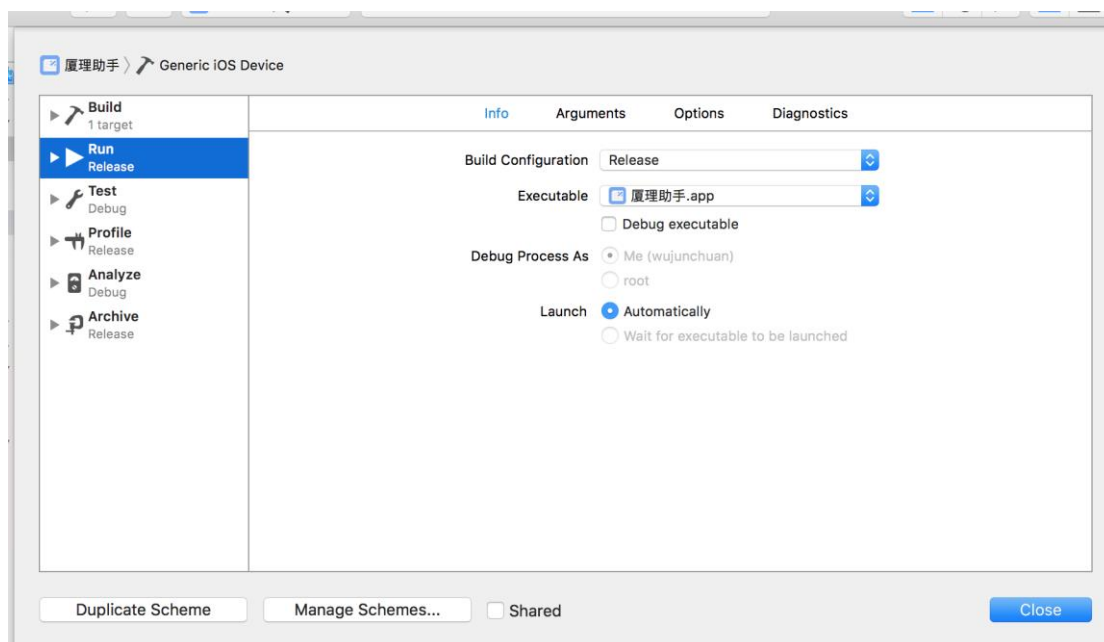


图 5.23 Xcode 构建界面

5.4.2 Android 打包

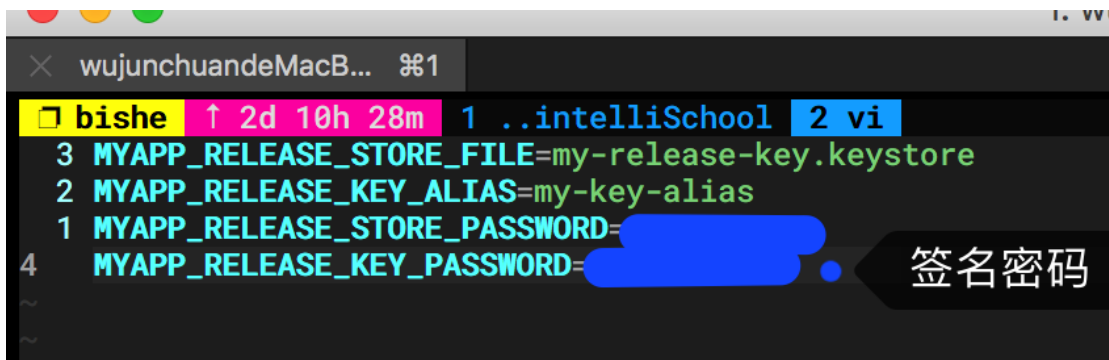
Android 在安装应用程序之前需要所有的应用都进行数字证书的签名认证，因此，如果想要发布 APK 包之前，需要使用 keytool 进行签名。如图 5.24，根据提示输入名字、组织、国家等相关信息，生成 my-release-key.keystore 的签名文件。


```
wujunchuan /Users/wujunchuan/Project/code/ReactNative_intelliSchool/ios develop
keytool -genkey -v -keystore my-release-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: John Trump
What is the name of your organizational unit?
[Unknown]: UED
What is the name of your organization?
[Unknown]: meetyou
What is the name of your City or Locality?
[Unknown]: Xiamen
What is the name of your State or Province?
[Unknown]: Fujian
What is the two-letter country code for this unit?
[Unknown]: zh
Is CN=John Trump, OU=UED, O=meetyou, L=Xiamen, ST=Fujian, C=zh correct?
[no]:
What is your first and last name?
[John Trump]: John Trump
What is the name of your organizational unit?
[UED]: UED
What is the name of your organization?
[meetyou]: meetyou
What is the name of your City or Locality?
[Xiamen]: Xiamen
What is the name of your State or Province?
[Fujian]: Fujian
What is the two-letter country code for this unit?
[zh]: zh
Is CN=John Trump, OU=UED, O=meetyou, L=Xiamen, ST=Fujian, C=zh correct?
[no]: y

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=John Trump, OU=UED, O=meetyou, L=Xiamen, ST=Fujian, C=zh
Enter key password for <my-key-alias>
(RETURN if same as keystore password):
Re-enter new password:
They don't match. Try again
Enter key password for <my-key-alias>
(RETURN if same as keystore password):
Re-enter new password:
[Storing my-release-key.keystore]
```

图 5.24 生成签名文件

将 my-release-key.keystore 文件放在项目目录下,编辑 ~/.gradle/gradle.properties 文件,如图 5.25 所示,配置签名密码。如图 5.26 进入项目目录下,编辑 android/app/build.gradle 文件,添加签名信息。最后在终端输入 `cd android && ./gradlew assembleRelease` 运行 gradle 来生成已签名的 apk 文件,如图 5.27,生成的 apk 文件为 app-release.apk。



```
wujunchuandeMacB... 881
1 bishe 2 2d 10h 28m 1 ..intelliSchool 2 vi
3 MYAPP_RELEASE_STORE_FILE=my-release-key.keystore
2 MYAPP_RELEASE_KEY_ALIAS=my-key-alias
1 MYAPP_RELEASE_STORE_PASSWORD=[redacted]
4 MYAPP_RELEASE_KEY_PASSWORD=[redacted] 签名密码
```

图 5.25 配置签名密码

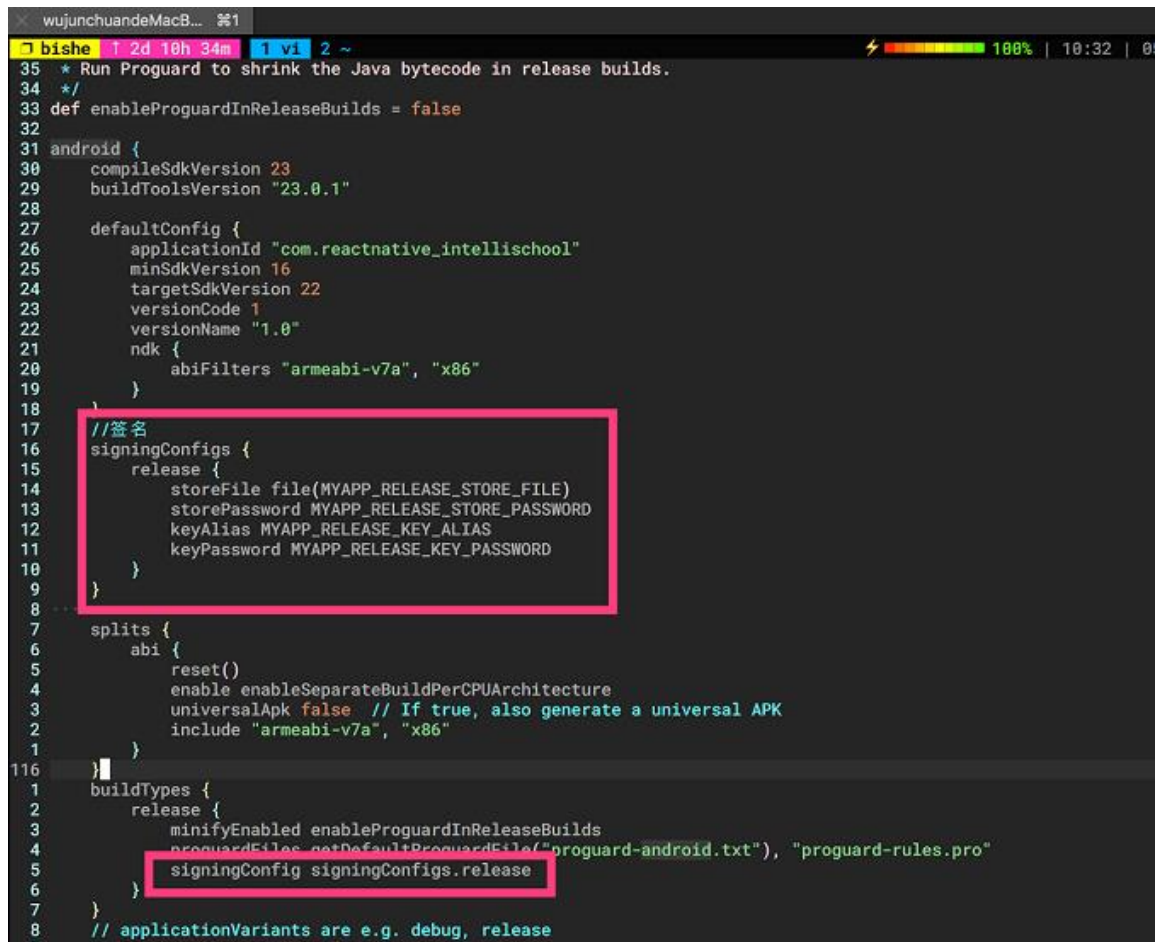


图 5.26 添加签名信息

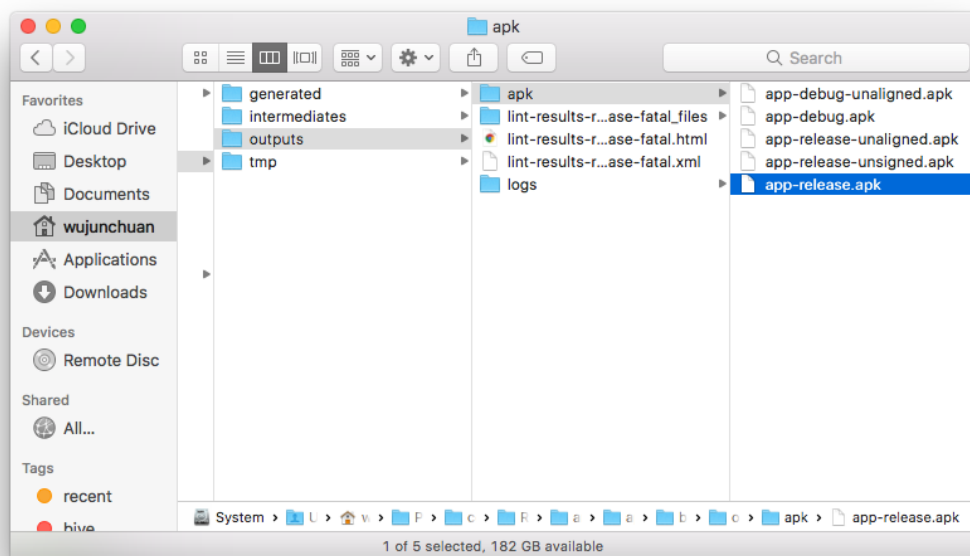


图 5.27 生成已签名的安装包

5.5 本章小结

本章节从环境的搭建开始介绍，配置了智慧校园应用的前后端环境依赖；介绍了智慧校园应用的 UI 框架关键代码的实现与工具库的封装；使用 **React Native** 进行开发调试与 Web 平台上开发调试 **HTML** 代码有相似的体验。大多数熟悉前端开发的工程师有许多熟悉的工具可以使用；对系统的各个模块进行了测试；介绍了 **React Native** 如何使用打包工具将其导出最总产成品(.ipa 文件与.apk 文件)。

第六章 总结和展望

6.1 总结

本论文针对传统校园信息门户网站的不足提出了移动端手机应用智慧校园的解决方案,使得通过手机可以随时随地获取校园资讯。通过手机应用实现了一些校园内常见的生活服务,使在校师生都能够在移动智能时代获得生活上的便利。为现代高校信息建设做出一份贡献。本应用主要实现了校园新闻模块、图书管理模块、便利服务模块。其中校园新闻模块实现了实时的校园新闻抓取与详情页面的展示;图书管理模块实现了图书的添加与借阅的管理,添加时只需要输入 ISBN 码即可完成对图书信息的采集;便利服务模块实现了天气查询、快递查询、宿舍电费查询、E 通卡查询、成绩查询等功能。

本应用采用了 Facebook 开源的全新的跨平台开发方案——React Native,从技术层面上解决了以往传统跨平台开发上的不足。通过这项技术,实现了 iOS 与 Android 的跨平台开发。运用了前端开发技术中的组件化思想,使得 iOS 与 Android 在代码复用率上达到 90% 以上。从软件工程的角度上降低了模块与模块之间的耦合性,提高模块内部的内聚性。虽然在设计的过程中是可以达到完全的代码复用的,但是我认为 iOS 与 Android 平台的设计哲学是不一样的,从用户体验的角度来讲操作习惯也应该要有所区别。因此分离出了部分组件以适配不同平台。虽然 React Native 在跨平台开发上具有性能高、组件化方便、代码可复用性强的优势,但是我们也应该清楚的认识到的 React Native 仍然是一项新兴的技术,目前版本号仍然维持在小版本号上,并且更新迭代的速度非常快。在项目快速更新的情况下文档还有一定的提升空间。在 React Native 的开发过程中遇到的问题不如原生开发那样能够快速找到解决方案,需要开发者花费更多的时间与精力去解决。并且 React Native 的学习门槛相对较高,学习成本较大,在遇到一些没有现成组件的情况下,可能需要开发者自己动手去实现原生代码,这样的学习曲线非常陡峭。在项目开始前的技术选型中,我本可以选择传统的 HTML5 来实现跨平台开发的,毕竟我在这方面有商用项目经验的积累。但是我清楚使用 HTML5 来实现跨平台开发只是对原生组件的一种模拟,通常伴随着卡顿、掉帧等缺点,是损害用户体验为代价的。

6.2 展望

校园应用是一个服务涉及范围很广的应用,显然当前的功能实现还是比较单一的。

基本上都是实现信息聚合、工具服务的功能。在未来的开发中，应该更多的实现社区功能，建立类似论坛一样的校园社区；继续提高现有服务的水平，例如因教务系统的限制而导致的教务成绩查询只能获取当前学期的成绩，还有改进空间。同时 **React Native** 的强大之处并不仅仅局限于跨平台开发，它在热更新领域同样具有优势。更新服务器可以下发更新包到基于 **React Native** 的手机应用，实现静默更新。在未来的开发中，应该注意实现这项功能，这可以极大地缩短软件更新周期，实现快速迭代的过程。对于商业公司而言，最主要的是可以绕过苹果的应用审核周期，发现 **bug** 能够及时修复。

跨平台开发、编写高可复用的代码一直是程序员的追求。**Web** 是未来，**Native** 是当下，而 **React Native** 处在未来与当下之间。**React Native** 就像 **Web** 开发侵入移动开发的一个中间过渡产品，但是在 **Web** 战胜原生应用之前，**React Native** 仍然值得我们去学习。

致 谢

时光荏苒，岁月匆匆，转眼间就到了毕业的时候了。大学的生活很快就要过去，在这四年的学习生活中我收获了许多，而这些成绩的取得是和一直关心帮助我的人分不开的。四年的大学生活就快走入尾声，我们的校园生活就要划上句号，心中是无尽的难舍与眷恋。从这里走出，对我的人生来说，将是踏上一个新的征程，要把所学的知识应用到实际工作中去。

在论文完成之际，我要特别感谢我的指导老师对我的热情关怀和悉心教导。从毕业设计任务书下发、文献的收集、文献综述与外文文献翻译再到现在论文的成稿一路披荆斩棘，离不开我的指导导师不厌其烦地指导。在论文的撰写过程中老师们给予我很大的帮助，帮助解决了不少的难点，使得论文能够及时完成，这里一并表示真诚的感谢。在编写项目代码的过程中我付出同时也收获着，我从一开始对 **React Native** 这项技术的完全不了解到能够独立编写应用程序，都要感谢导师给了我的帮助。

还要感谢实习单位美柚 **UED** 部门的各位同事，是他们让我认识到设计的重要性，并教会了我一些设计工具，我才能够在编码前先做好设计，避免了来回返工的麻烦。在企业实习的过程中，学习到了很多新的知识和技术，丰富了我的知识面。

最后我还要感谢微云工作室的各位同学，怀恋当初一起做项目的激情与快乐。祝他们身体健康，万事如意。

参考文献

- [1] 严新巧. 基于移动开发现状探讨 React Native[J]. 电脑知识与技术: 学术交流, 2016 (11Z): 76-77.
- [2] 顾学海, 胡牧, 蒋厚明, 等. 基于 HTML5 的混合移动应用开发[J]. 计算机系统应用, 2016, 25(5): 237-240.
- [3] 陈旭山, 顾剑柳, 游小荣, 等. 基于 APICloud 的常州纺院 APP 平台设计与实现[J]. 无线互联科技, 2016 (5): 42-43.
- [4] 谢计红, 李小庆. 高校校园 APP 框架构建研究[J]. 武汉交通职业学院学报, 2015, 02: 20-22.
- [5] 陈屹. 深入 React 技术栈[M]. 人民邮电出版社, 2016.
- [6] Eisenman B. Learning React Native: Building Native Mobile Apps with JavaScript[M]. " O'Reilly Media, Inc.", 2015.
- [7] Witte D, von Weitershausen P. React Native for Android: How we built the first cross-platform React Native app[J]. Facebook Code, 2015, 14.
- [8] Occhino T. React Native: Bringing modern web techniques to mobile[J]. Facebook code, 2015, 26.
- [9] 海鹏娇. 基于用户体验的校园 APP 设计研究[D]. 西安美术学院, 2015.
- [10] 王建波. 论移动应用编程的新方向[J]. 科技展望, 2016, 26(33).
- [11] 姜翠. 混合开发模式在移动校园 APP 中的应用研究[J]. 无线互联科技, 2016 (1): 17-19
- [12] 高晶. 校园业务中混合模式移动应用软件架构的实现[J]. 中国信息技术教育, 2015 (15): 119-121.
- [13] 戴翔宇. Web 前端工程组件化的分析与改进[D]. 吉林大学, 2016.
- [14] 邓森泉, 杨海波. Promise 方式实现 Node.js 应用的实践[J]. 计算机系统应用, 2017 (4): 218-223.
- [15] 胡伟峰, 辛向阳. 智能手机 iOS&Android 系统功能交互行为对比研究[J]. 装饰, 2016 (4): 82-83.