

南京邮电大学

毕业设计(论文)

题 目 基于 Android 的旅游交友系统的设计与开发

专 业 软件工程

学生姓名 宋乾坤

班级学号 B12040819

指导教师 程春玲

指导单位 南京邮电大学计算机学院

日期： 2016 年 03 月 14 日至 2016 年 06 月 17 日

毕业设计（论文）原创性声明

本人郑重声明：所提交的毕业设计（论文），是本人在导师指导下，独立进行研究工作所取得的成果。除文中已注明引用的内容外，本毕业设计（论文）不包含任何其他个人或集体已经发表或撰写过的作品成果。对本研究做出过重要贡献的个人和集体，均已在文中以明确方式标明并表示了谢意。

论文作者签名：

日期： 年 月 日

摘 要

随着社会的不断发展，人们的生活水平不断上升，越来越多的人喜欢外出旅行，并通过旅游来结交朋友，有必要设计和开发一个旅游交友系统来帮助旅游用户互相交流。目前移动智能设备已大量使用，为使旅游用户之间的交流更加便利，本课题以增强用户之间的交流体验为目标，综合考虑旅游用户的实际需求和出行使用方便，设计和实现了一款基于 Android 平台的交友系统。

本文分析了目前市场上旅游平台应用和社交平台的不足之处，阐述了课题的目的和意义。本文设计的旅游交友系统采用 MVC（Model View Controller，模型视图控制器）框架设计模式，系统包括服务器、Android 应用客户端和数据库三个部分，具有很强的可拓展性和二次开发能力。通过需求分析，将系统分为六个功能模块，分别是登录注册模块、旅游动态模块、发现用户模块、游记模块、用户信息管理模块和查看用户信息模块，然后详细说明了系统的设计、实现和测试，最后对课题进行总结并且提出了系统以后的改进方向。

关键词：社交网络服务；旅游社交；安卓；服务器；MySQL 数据库

ABSTRACT

With the continuous development of society and people's living standard rising, more and more people love to travel and make friends by traveling. It is very necessary to design and develop a tourism system for helping tourist communicating with each other. At present, mobile intelligent devices have been widely used, making communication between tourist more convenient, this topic is to enhance communication experience between tourist, comprehensively considering the actual needs of users and convenience, based on the Android platform design and implement of a social-tourism system.

This paper analyzes the current shortcomings of the tourism market of internet applications and social networking platform, elaborates the purpose and significance of the topic. The social-tourism system of the paper uses MVC (Model View Controller, Model-View-Controller) design pattern framework, including servers, Android client application and the database, with strong scalability and secondary development capacity. Through demand analysis, the system is divided into six functional modules, log-in and registration module, tourism dynamic module, finding-user module, travel module, the user information management module and viewing-user information module, then a detailed description of the design, implement and test. At last, summarizing the topic and putting forward the improved direction.

Key words: Social Networking Service; Social Tourism; Android; Server; MySQL Database

目 录

第一章 引言	1
1.1 课题的目的和意义	1
1.2 国内外研究的现状	2
1.3 论文组织结构	4
1.4 本章小结	5
第二章 交友策略及系统开发技术	6
2.1 交友策略	6
2.2 系统开发技术	7
2.2.1 Java Servlet 服务器开发	7
2.2.2 Android 软件开发	9
2.2.3 JSON 数据交换格式	12
2.3 MySQL 数据库	13
2.4 本章小结	13
第三章 需求分析和总体设计	14
3.1 需求分析	14
3.2 总体设计	17
3.3 数据库设计	19
3.4 本章小结	21
第四章 详细设计和系统实现	22
4.1 服务器访问数据库的设计和实现	22
4.2 客户端访问服务器设计和实现	22
4.3 登录注册模块的设计和实现	24
4.4 旅游动态论坛模块的设计和实现	25
4.5 发现用户模块的设计和实现	27
4.6 游记模块的设计和实现	30
4.7 用户信息管理模块的设计和实现	31
4.8 查看用户信息模块的设计和实现	33
4.9 本章小结	35
第五章 系统测试	36
5.1 测试环境	36
5.2 模块测试	36
5.3 功能测试	39
5.3 系统兼容性测试	41
5.4 系统性能测试	42
5.6 本章小结	44
结束语	45

致 谢	47
参考文献.....	48

第一章 引言

通过课题的背景说明了课题的目的和意义，对课题相关领域研究的现状进行介绍，介绍了全文的工作内容和章节结构。

1.1 课题的目的和意义

随着互联网的高速发展，人们的生活越来越离不开互联网，目前，国内外的社交平台不胜枚举，比较知名的有国外的 Facebook 和 Twitter 以及国内的微信、微博、QQ 等，但是却没有一个比较大型的面向旅游用户的社交平台。而目前市面上关于旅游的网络平台主要分为三类：旅游服务提供平台、旅游导游类平台和新兴的旅游用户社交平台。其中导游类和旅游服务平台的盈利能力比较强，发展时间比较久，所以平台数目种类比较多，比如马蜂窝、伴玩旅游智能导游平台、去哪儿、携程、同程旅游等平台。但是现代人越来越热爱自由，对于传统的跟团旅游已经厌倦，越来越趋向于自驾游或者自助游，这样传统的旅游服务平台和单纯的旅游攻略平台已经不能满足人们的出行和交友需求了，这时新兴的旅游用户交流互助平台出现了，这种平台同时具有旅游攻略和社交两种功能，注重用户之间的交流，但是由于其是一种新兴的平台，并没有过多厂商或者服务商去提供这种平台。

同时随着 3G 和 4G 网络的出现和流行，移动智能终端的网络问题得到极大的改善^[1]，而且因为旅游用户喜欢外出旅行，他们使用的互联网社交工具已经从几年前的个人电脑上的浏览器变成现在较为流行的移动智能终端应用，而 Android 系统又是移动智能终端中应用最广泛的系统之一，所以本课题所设计的社交平台是一个基于 Android 的旅游交友系统，课题着眼于用户的切实需求，为爱好旅游的安卓用户提供一个方便的交友平台，以提高用户的交友体验为主要目标来进行设计和开发。

本课题的目的在于：

- （1）针对目前旅游平台缺少用户交流的问题，为旅游用户提供一个交流的平台。
- （2）为旅游用户结交新朋友提供帮助，使他们能够更轻松找到合适的新朋友。
- （3）方便旅游用户互相邀约出行和分享出行游记。
- （4）方便旅游用户搜集景点信息和旅游经验。

本课题的意义在于：

- （1）本课题是一次针对面向旅游用户的社交平台的尝试。
- （2）课题能够切实帮助到旅游用户，为他们的出行提供方便的同时还加强了他们之间的交流。

(3) 课题提供一个用户分享自己游记的平台，有助于用户之间互相查看游记，同时游记也是用户对旅游经历的美好回忆。

1.2 国内外研究的现状

首先，目前国内外的旅游平台以市场为先导，主要是以联合酒店、餐厅、宾馆等服务的 O2O（Online To Online）平台为主，以旅游攻略介绍的平台为辅。旅游类平台的在国内比较知名的有去哪儿、同程旅游、大众点评、穷游网、面包旅行、蚂蜂窝等，国外则有 about travel、travelsur 等。

其中去哪儿和同程都是提供用户出行时的衣食住行服务，它们的重点是向用户提供酒店预订和车票机票购买的服务，但是它们几乎没有旅游攻略提供，也不能向用户提供交友功能，用户不能使用这类平台进行社交。

大众点评、穷游网和面包旅行重点是向用户提供旅游景点推荐和旅行攻略，它们的攻略是由平台进行编辑和发表的，不过这些攻略是以用户游记为基础通过信息筛选过滤得到的，这样就不能保证攻略的真实性而且太过官方。大众点评主要提供景点门票的售卖和一些简单攻略，穷游网和面包旅行则可以根据你的旅游目标帮你设计一个比较合理的旅行路线，但是他们也不是很注重用户的社交需求，只是希望用户反馈旅游信息，完善平台旅游攻略。

蚂蜂窝则是新兴的社区旅游平台，不仅注重用户的分享也重视用户的交流，它的目的是打造一个旅游交流社区。其社区功能对用户交流实现的比较好，用户可以通过社区功能进行交流和分享，其问答功能是一个非常好的用户交流思路，通过提问和回答来加强用户交流解决用户旅行问题，通过商城积分和排行榜等功能来激励用户回答问题。虽然它给用户提供了很多交流方法，但它这些功能最终还是为用户提供旅游攻略，重点还是在于旅游而不在于交友。

在国外，about travel 和 travelsur 分别是欧洲和美洲比较知名的旅游平台，它们同样是以本地区的旅游景点介绍为主，以图文并茂的方式为用户提供旅游攻略信息，同样的它们也不重视用户之间的互相交流。

总的来说目前的旅游平台大多不注重用户的社交需求，缺失用户交友功能的设计是他们共同存在的问题。

其次，就社交方面来说，SNS（Social Network Software，社交网络服务）目前基本上是指基于互联网的社交网络服务^[2]，因为互联网的信息传递速度非常快而且提供了多种交互方式，比如通过电子邮件，即时消息，语音视频实时互动和大型社区等方式。所以用户可以在互联网上非常方便快捷的与其他用户进行交流，而且交流的方式不仅仅包括文字，还可以通过图片、视频、音频等其他方式。信息通过好友间的关系无限的传输，最终形成一个庞大的网络交友平台。

在互联网发展的早期出现的各类聊天室是目前社交网络平台的雏形，也是最早的网络社交平台。而好友的概念则是源于 2003 年出现的 Friendser，Friendser 可

以通过用户资料查找自己感兴趣的好友^[3]，这种方式已经跟现在的交友网站非常类似了，但它却因为不能让用户更好地展示自己的信息导致用户不断流失，最终消失在互联网发展的浪潮中。而同期出现的微软产品 Wallop 则表现的更好一些，增加了很多不同种类的用户互动元素。

随着互联网发展，Web2.0^[4]时代来临，社交网站的开发人员和用户都已经意识到一件事情，由用户来产生信息而不是由管理者来添加一些也许很多人都不感兴趣的信息来的更为重要^[5]。而且在这段时间，互联网得到了极大的发展，各种用户都显著增加，社会进入全民互联网时代。这一变化虽然使社交网络平台得到飞速发展，但是也产生了一个很严重的问题，不是每一个用户都能提供高质量的资料。在这方面有个例子就是 Myspace，早在 2004 年，用户就可以使用 Myspace 发文章、音乐、视频等原创作品，在当时及其火爆。在 2005 年其网站的访问量甚至超越了 Google 搜索引擎的访问量，但是它没有意识到因为 Web2.0 时代的到来导致用户全民化所带来的问题，没有对用户所发送的信息进行审核和过滤，最终使得自己走向没落。

目前最大的，用户数量最多的社交网站是 Facebook，Facebook 使用了邮箱这样一种现在都极为流行的方式来验证用户信息，使用户可以通过邮箱建立真实的好友关系，而且 Facebook 首先开放了 API（Application Programming Interface，应用程序编程接口）的概念^[6]，现在不仅可以用户使用户分享主流信息还能将自己创造的应用和游戏提供给好友，吸引越来越多的用户加入其中。

而国内的社交网络因为互联网发展落后于国外，前期大多是在借鉴国外社交网络创新的基础上增加中国特有的元素，形成了具有中国特色的社交网络，例如开心网。而随着中国互联网的发展，QQ 空间和新浪微博的出现极大改善了这一情况，近几年出现的微信朋友圈创新使用了一种良好的社交技术，动态回复只有互相是好友才能看到，这极大保障了用户的隐私，满足用户在近乎透明的互联网环境的隐私要求，使得微信得到急速的发展，大量的用户涌入微信平台。

同样社交平台也存在一定的问题，无论是传统的社交网站还是现在新兴的移动社交软件几乎都是停留在面向所有用户，几乎没有去专门面向旅游用户的社交平台。而今后 SNS 可以横向的发展，提高多样性和针对性，专门的面向某一类用户，方便他们之间的交流。

随着科技的发展，3G 网络技术的成熟，互联网使用工具已经从以前的个人电脑演变成如今的个人电脑和智能移动终端，对于喜爱旅游用户来说，无论是社交需求还是旅游攻略需求都会使他们更加侧重于移动智能终端。而目前的智能终端设备主要分为两种，IOS 设备和 Android 设备，同时因为 Android 设备的开源性、封装性和高性价比性，Android 设备的用户与日俱增，Android 系统占有智能手机操作系统大部分的市场，Gartner 发布最新预测，在 2015 年 Android 系统市场占有率将达到 59%^[7]。目前各大社交网站平台和旅游平台都推出了自己 Android 应用，

而且其用户在 Android 应用上花费的时间已经远远超过他们使用网页或者电脑客户端的时间。

目前 Android 端软件开发技术分为前端和后端两部分，后端一般与传统网站后端没有太大的区别，前端应用则是使用以 Activity 为主体的程序结构^[8]，以 XML（Extensible Markup Language，可扩展标记语言）页面配置文件来进行 UI（User Interface，用户界面）设计。前后端的通信则主要分为两种，一种是 Socket 通信方式，一种是 HTTP（Hyper Text Transfer Protocol，超文本传输协议）通信方式。这两种通信方式的主要区别是，HTTP 通信使用的是“请求—响应方式”^[9]，在使用 HTTP 连接时只有在客户端请求并建立连接通道之后，需要客户端向服务器端发送请求后，服务器端才能向客户端传输返回的数据。而 Socket 通信只需要客户端和服务端建立连接之后就可以随意的进行数据传输^[10]，可以做到服务器端对消息进行主动推送，不需要客户端周期性的发送请求。

虽然目前 Android 软件的开发技术已经发展十分良好，但是目前 Android 平台开发仍存在问题：

第一，信息安全问题，Android 是一个新兴的移动操作系统，而且其还具有开源性，这两点都会导致一些安全问题。当然现在的主流 Android 应用比如：微信、支付宝等软件都对它们的软件安全性做了很多努力。

第二，流量费用问题，虽然随着 3G、4G 时代的到来，流量价格有了一定程度的下降，但是目前移动运营商的流量费用仍然很高，所以目前 Android 软件如何减少用户的流量消耗也是一个问题。

第三，操作方便问题，虽然大多数移动智能设备都有触摸屏帮助用户对软件进行操作，但是不得不承认目前移动智能终端的操作仍不如 PC 方便，所以如何提高 Android 软件关于操作的用户体验也是一个很大的问题。

1.3 论文组织结构

本课题基于 Android 系统平台，以提高旅游用户的交流和交友体验为目标，提出了旅游交友系统的设计和实现方案。

论文将从以下几个章节分别对该系统进行描述，其中各个章节的重要工作内容如下：

第一章：旅游交友系统的概述，首先介绍了本课题的背景引出课题的目的和意义，接着分别对旅游平台、社交平台和 Android 软件开发目前的现状进行了分析，然后介绍了论文的工作和组织结构。

第二章：交友策略及系统开发技术，首先对课题提出的旅游用户交友策略进行描述，然后对项目开发使用到的技术进行描述，对软件开发技术和数据库进行介绍。

第三章：需求分析和总体设计，对旅游交友系统进行了需求分析，并且通过

需求分析得到了系统的总体设计。

第四章：系统实现，根据需求分析和概要设计，对旅游交友系统的各个功能模块的具体实现进行了介绍。

第五章：系统测试，对整个系统的测试进行介绍，包括模块、功能测试、系统兼容性测试、系统性能测试，保证系统功能实现没有严重 BUG（漏洞）。

第六章：总结和展望，对课题的工作内容进行总结，对系统的新版本和发展进行展望。

1.4 本章小结

本章通过课题的背景对课题的目的和意义进行了说明，然后通过对国内外相关领域的现状进行说明，说明了现在旅游平台和社交平台都各自存在一些问题，最后介绍了本篇论文的组织结构。

第二章 交友策略及系统开发技术

由于本课题着重在于为旅游用户提供社交平台，所以课题中的交友策略是重要的一部分，本章中将会介绍交友策略和本课题用到的相关技术，其中课题使用的技术又分为软件开发技术和数据库使用技术两个方面进行介绍。

2.1 交友策略

交友策略是根据人对交友的需求来提出和使用的，交友策略分为关系维持和关系拓展两种。

哈佛大学的心理学教授 Stanley Milgram 在 1967 年发现提出的度分割理论指人际网络中的任意两个个体都能都过网络中的其他个体产生关系，联系间最多的连接数量是六个^[1]，这是一个不同关系连接转换和传递的过程，在现实的社交网络中是普遍存在的，比如前段时间我的一个高中同学 A 需要一本南京理工大学自动化专业的教材，但是他在南京只认识我一个人，我又不是南京理工大学自动化专业的学生，但是我知道我有一个舍友的高中同学 B 是南京理工大学自动化专业的学生，这样我高中同学 A 就通过了 2 个个体的关系连接得到了他想要得到的教材，这种关系网络在现实中很常见。在互联网上，这种情况也普遍存在，而且因为互联网的开放和信息传递速度快的特点，这种间接的连接的需求被削弱了，目前互联网上新型的关系网络主要目标是建立一种关系网络，让人们能够轻松建立直接联系而不是需要一层层关系的传递，还是上面的例子，如果在互联网上同学 A 需要一本南京理工大学自动化专业的教材，这时他大可以不必通过我这层关系链，而是通过南京理工大学的学校论坛表明自己的需求，这样他就能通过南京理工大学的学校论坛这个社交网络平台直接认识同学 B，同学 A 就能直接得到需要的教材。

上面的例子表明其实现在六度分割理论已经不太适合飞速发展的互联网社交网络服务了，在互联网早期，人们沉迷于互联网前所未有的开放性，当时的互联网社交平台是基于关系拓展需求的，也就是这时六度分割理论仍然可以指导互联网社交平台的交友策略设计，比如早期的 BBS 论坛和聊天室，这时的交友策略中关系拓展的比重要大于关系维持。

但是随着互联网的进一步普及，QQ 等 IM（Instant Messaging，即时通讯）社交工具的诞生，使得人们对关系的管理更加方便，这样一来导致关系维持和关系拓展开始分化，人们现实中的社交关系也可以通过网络社交应用进行管理，关系维持和关系拓展开始在 IM 社交工具中并存，此时关系网络中的关系纽带仍然是多重的，而且强弱关系是并存的。社交网络服务在这个基础上继续进化，进化的方向主要分成两部分，而且他们都是做的减法进化。其中一种走向了关系维持，尤

其是现实关系的维持，例如 Facebook 和 QQ，他们减去了关系连接中的弱关系纽带，保留和加强了强关系纽带。另外一种则是走向了关系拓展，比如 Twitter 和微博，他们减去关系维持的交流能力，虽然交流能力有所下降，但是通过单向关注使得人们之间建立泛关系（关系拓展）变得十分容易。

总结起来，关系维持和关系拓展是人们对关系的两种最基本的需求，按照“邓巴数字”的理论，人们可以管理的强关系数量是有限的，但是人们也不可能放弃对关系拓展方面的需求^[12]，实际上，在人的一生中都在不停拓展关系。所以本课题的主要侧重点也是关系的拓展，使得用户可以更加轻松的与陌生人建立关系，而不是仅仅维持原有的关系。

2.2 系统开发技术

系统开发使用到的技术分为 Java Servlet 服务器、Android 软件开发和 JSON（JavaScript Object Notation）脚本语言三个方面，重点介绍课题使用到的开发手段和技术。

2.2.1 Java Servlet 服务器开发

Java Servlet 服务器是一种部署在 Web 服务器的端的 Java 程序，也可以说成是服务器端的应用程序，Servlet 即 Java 服务小程序，是使用应用程序设计接口以及相关类和方法的 Java 程序^[13]。它可以当作一种插件，嵌入到 Web 服务器中运行。Servlet 主要用于处理和客户端之间的通信，当客户端发送一个 HTTP 请求时，服务器通过调用 Servlet 来向客户端发送一个响应，Servlet 具有以下几个优点：

第一，高效。Servlet 只会运行在一个进程中，从服务器首次访问 Servlet 进程开始，一直到基于该 Servlet 的程序结束，这个过程中所有的客户端请求都会在这一个线程中进行处理和响应。即每一个请求就会建立一个进程，这样来看 Servlet 的效率明显会高出传统 CGI 许多。

第二，功能强大。Servlet 能够直接与 Web 服务器交互，能够在各个程序之间实现数据的共享，这方便了服务器快速响应。

第三，可移植性强。因为 Servlet 是采用 Java 编写的，而 Java 又是可移植的面向对象程序设计语言^[14]，所以 Servlet 可以非常方便的移植到 Apache、Microsoft IIS、JBoss 或者其他的 Web 服务器上。

第四，便捷性。Servlet 为开发人员提供了大量的工具，比如读取和设置 HTTP 响应头、处理 Cookie、自动解析 HTML 表单数据等非常实用的工具。

Servlet 在具体使用时会调用相关接口，这些接口都 Servlet 中定义好的，接口主要包含以下几种：异常接口、配置接口、实现接口、请求响应接口、会话接口、上下文接口、过滤接口等。

在 Servlet 中主要使用 `doGet` 和 `doPost` 方法对请求进行处理和响应，这两个方法都具有两个参数，它们的类型分别是 `HttpServletRequest`（后文都简称 `request`）和 `HttpServletResponse`（后文都简称 `response`），这两个参数都是 HTTP 中的一部分，其中 `request` 可以将数据从前台向后台传输，而 `response` 则可以将数据从后台向前台传输，它们都可以将 JSON 字符包装在其中，也可以简单的包含一个字符串。

Servlet 服务器接收到前台的 `request`，然后通过解析 `request` 得到数据或者请求后，很少会对这些数据进行处理，一般都是将数据交给后台的 Model 层进行数据处理或入库，请求数据一般也是交由 Model 层进行数据库的操作得到具体数据后再返回 Servlet，然后 Servlet 再将数据通过 `response` 传输给前台。Servlet 服务器的结构如图 2.1 所示：

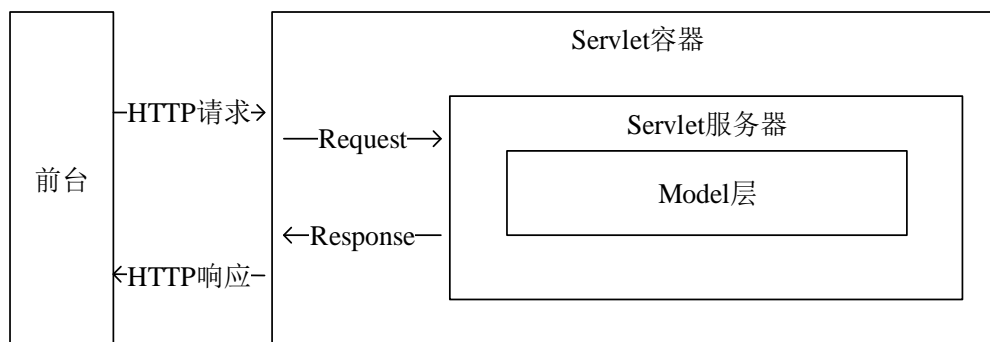


图 2.1 Servlet 服务器结构

如图 2.1 所示 Servlet 服务器包含 Servlet 程序和 Model 层，Model 层负责进行逻辑处理和与数据库的交互，而服务器一般会放在一个 Servlet 容器中，而这个容器称之为 Web 服务器，现在主流的 Web 服务器主要有：Apache、Tomcat、Jboss、IIS 等，个人 Servlet 服务器大多部署在 Tomcat 上^[15]。

Tomcat 服务器是一个免费的开源的轻量级 Web 服务器，也是一个 Servlet 容器，它是 Apache 基金会的 Jakarta 项目中的一个核心项目^[16]，由 Apache、Sun 和一些其他公司及个人共同开发完成。因为有着 Sun 公司的支持，所以 Tomcat 服务器总能跟最新版本的 Servlet 实现兼容。

Tomcat 安装目录下面包含很多文件夹，其中 `bin` 文件夹是用来存放开启关闭服务器的文件，在 Windows 平台下这些文件的格式为 `bat`；`Conf` 文件夹中存放了与服务器配置相关的文件；`temp` 文件夹用来存放服务器运行时产生的缓存文件；`work` 文件夹用来存放服务器运行时产生的工作文件；`webapps` 文件夹是实际应用中使用的最多的文件夹，web 服务器应用就是放在此文件夹中，然后通过浏览器进行访问，但是每次文件夹中的应用发生变化时都要重启 Tomcat 服务器才能使对应用的改变生效。

综上所述，Servlet 服务器就是相当于部署在 Tomcat 上面的 Java 应用，它的具体工作流程为，当 Tomcat 收到某一个具体的 Servlet 文件的请求时就会根据

web.xml 配置自动调用相应的构造函数，进行初始化，然后通过请求头信息决定调用相应的 doGet 或者 doPost 方法，在方法内通过 request 得到前台发送的数据，最后通过 response 向前台返回数据。

2.2.2 Android 软件开发

Android 是一个基于 Linux 操作系统的开源移动设备系统，由 Google 公司领导和开发，2008 年推出了 Android 的第一个版本，直到近几年 Android 已经形成非常完善的系统架构^[17]。Android 系统一共分为 4 层，从高到低分别是应用层、应用框架层、系统运行库层和 Linux 内核层，如图 2.2 所示。

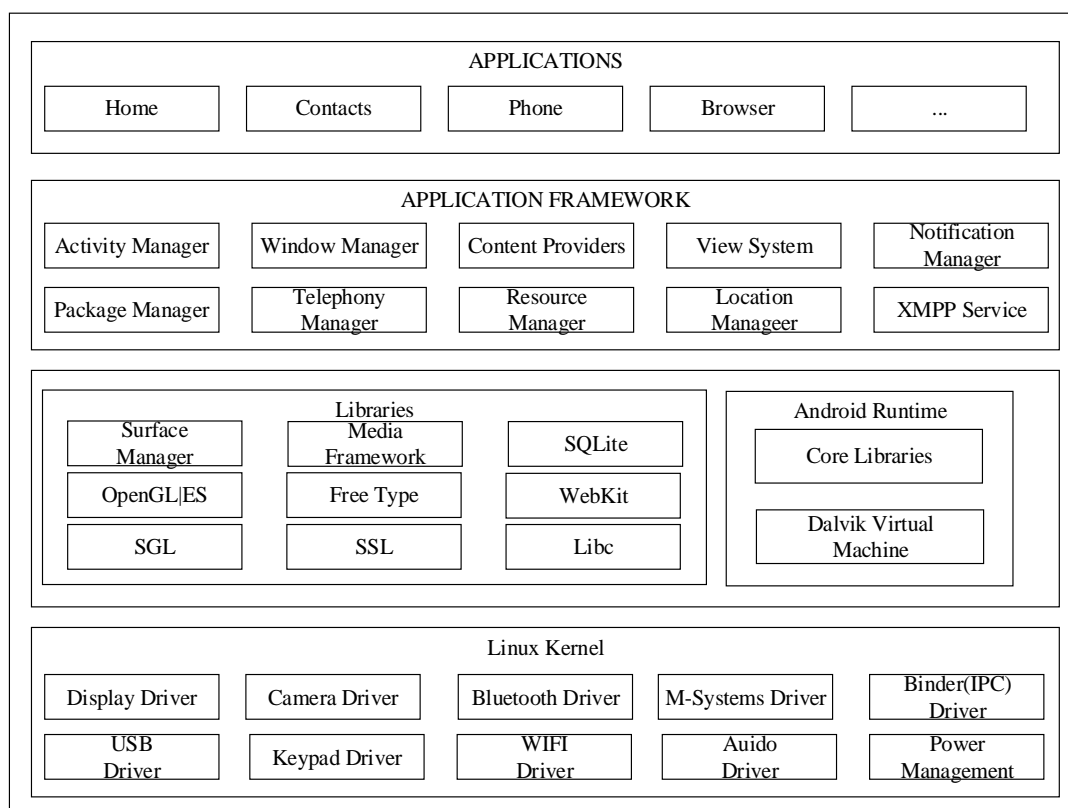


图 2.2 Android 系统架构

应用层运行在 Java 虚拟机中^[18]，是 Android 系统架构的最顶层，也是最外层，是能够直接与用户进行交互的层面。其中包括一些 Android 系统中捆绑的程序，例如电话程序、浏览器、图片浏览器、文件管理程序、日历等程序，也包括开发人员编写的应用程序，例如微信、微博、QQ 等程序。这一层主要是依靠应用框架层提供的 SDK（Software Development Kit，软件开发工具包）中的 API 来完成程序中所需要的各种功能。

应用框架层是一个开发人员可以完全访问的核心应用程序所使用的 API 框架，其简化了系统组件的重用，使用户可以很方便的替换程序组件。其中包含的 API 框架主要有：Activity Manager（活动管理器）、Window Manager（窗口管理器）、

Content Provider（内容提供者）、View System（视图系统）等，其中 Activity Manager 是比较重要的 API 之一，开发人员可以通过它管理应用程序的生命周期，并且它还提供了基本的导航和退出功能；View System 是构成一个应用程序页面最基本的组件，它主要提供了 listView、textView、button 等等页面组件。

系统运行库层是应用框架层的支柱，它在整个 Android 框架中起到承上启下的作用，它是连接内核层和应用层的纽带。包含程序库与 Android 运行库两部分，其中程序库中包含一些 C/C++库，它们通过 Android 应用程序框架为开发人员提供相应的服务，这些服务主要包含以下部分：Surface Manager（负责管理显示互动）、Media FrameWork（多媒体库）、SQLite（小型关系数据库）、OpenGL|ES（3D 绘图函数库）、FreeType（字体显示）、Webkit（内置的浏览器引擎）、SGL（2D 图形引擎）、SSL（网络安全协议）^[19]、Libc（标准 C 系统函数库）；而 Android 运行库包括 Core Libraries 和 Dalvik Virtual Machine 两部分，其中 Core Libraries 不仅包含 Java 能完成的功能 API 还包含了一些 Android 的核心 API，而 Dalvik Virtual Machine 表示 Dalvik 虚拟机，每一个 Android 应用都有一个自己专有的进程，并且不是多个应用程序运行在同一个虚拟机中，而是每个程序都对应一个独有的虚拟机实例，并且在该实例中执行应用程序^[20]。并且 Dalvik 虚拟机是一种基于寄存器的 Java 虚拟机而不是基于栈的传统虚拟机，它专门进行了内存使用的优化以便能够很好的支持多个虚拟机。

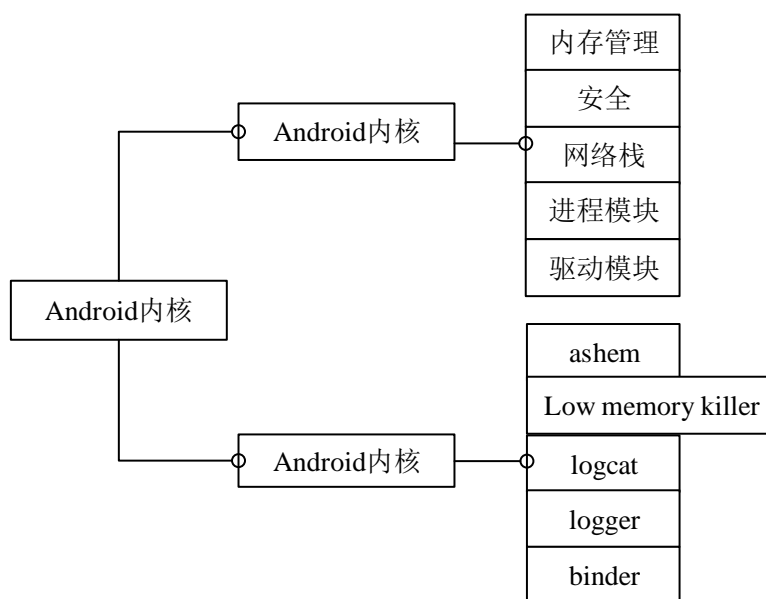


图 2.3 Android 内核层架构

Linux 内核层是 Android 架构的最底层，也是硬件和软件栈的抽象层，它依赖于 linux2.6 内核提供的系统服务。其中包含了系统运行所需要的驱动程序，比如网络栈、驱动模块、内存管理等模块^[21]。而 Android 内核又不仅仅是 linux 内核，还对其进行了一些扩展，具体结构如图 2.3 所示。

在实际的开发中只是调用上文框架中部分功能使用,但是所有的 Android 应用程序都是由 Android 的四大组件中的一个或者几个构成,它们分别是: Activity (活动)、Service (服务)、Content Provider (内容提供者)、Broadcast Receiver (广播接收器)。

Activity 是所有应用程序的根本,所有的程序都运行在 Activity 中,它是所有开发者使用最频繁也是最基本的模块之一。在四大组件中,Activity 是其中唯一可见的,它会直接展现给用户,每个 Android 应用一般都包含多个 Activity,一个 Activity 一般都是代表一个页面,上面会显示相应的控件,主要用来与用户完成交互,所有的 Activity 都需要在应用配置文件 AndroidManifest.xml 上进行注册。每个 Activity 都包含 3 对 6 个基本回调方法,分别是: onCreate 和 onDestroy、onStart 和 onStop、onResume 和 onPause,它们主要用于控制 Activity 的状态变化,这些状态变化就是 Activity 的生命周期,生命周期流程如图 2.4 所示。

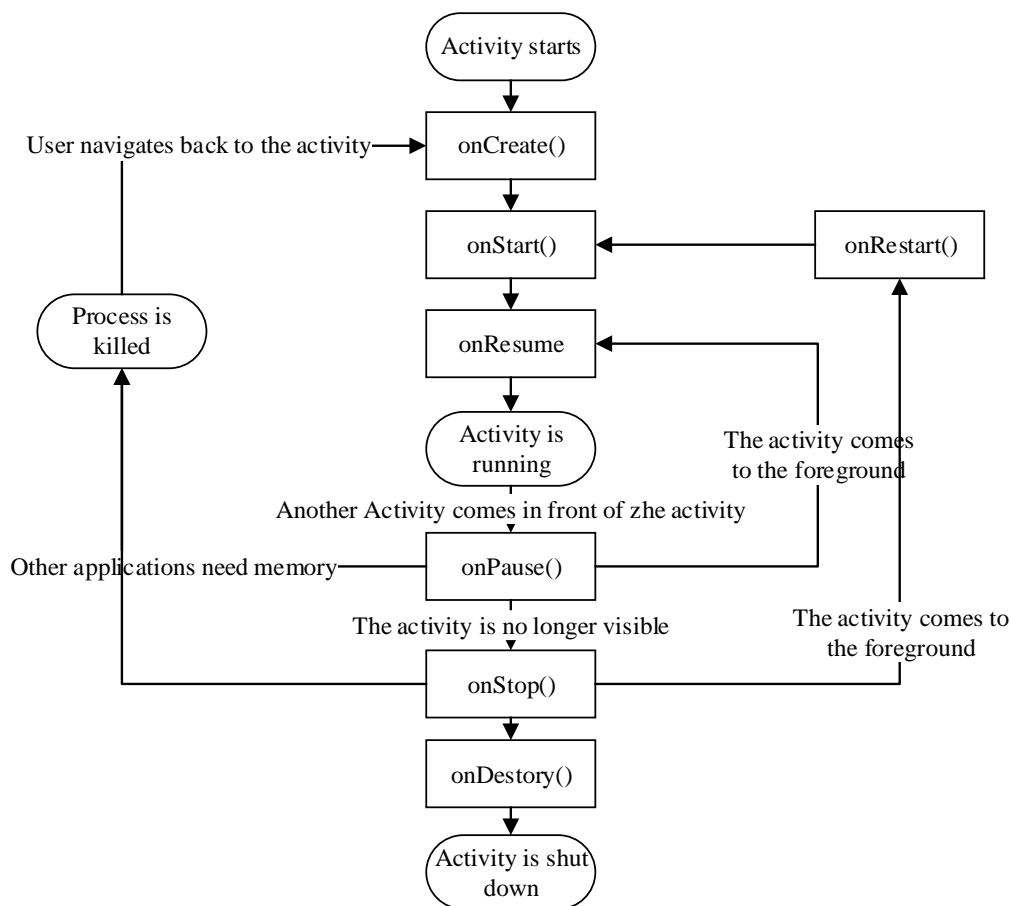


图 2.4 Activity 生命周期

一个应用的 Activity 往往不止一个,它们之间一般通过 Intent 来实现互相跳转,并可以通过 Intent 在跳转的过程中传递一些数据,跳转功能一般是通过响应页面中的按钮来激发。

Service 是一个后台运行的组件,它没有用户可见的界面,不能与用户进行交

互，它最典型的应用情况就是当用户打开一个音乐播放软件，然后切换到其他应用后，音乐仍能够正常播放。**Service** 的生命周期比较简单，回调方法也只有三个，分别是：**onCreate**、**onStart** 和 **onDestroy**。当服务第一次被启动时会先调用 **onCreate** 然后调用 **onStart**，当服务已经启动过然后再次启动时则直接调用 **onStart**，当要关闭服务的时候会调用 **onDestroy** 方法，如果用户没有点击关闭服务的按钮而直接退出的时候，服务就会一直在后台运行。

Content Provider 是 **Android** 系统提供的数据库访问组件，**Android** 系统对应用数据的保护是非常严密的，应用本身所有的数据库和文件都是不能被其他应用直接访问的，而 **Content Provider** 的作用就是提供了不同应用程序之间的数据共享方法。

Broadcast Receiver 是一种全局的监听器，它可以实现不同组件之间的通信并对应用和系统发出的通知进行处理的一种组件。它的作用就是当受到一个事件发生的通知后，对程序进行处理，比如当系统突然断网时，系统就会发送一个广播消息给所有应用的 **Broadcast Receiver**，当它接收到这个广播后就会知道现在处于断网状态，并开启应用对断网的应用措施。广播之间的信息是通过也是 **Intent** 来传递的，而且一般都是隐式调用的 **Intent**。

而在实际的 **Android** 应用开发的过程中可能不会完全用到上面四种组件，但是一定会用到其中一种或者几种，而且对于一个优秀的开发者来说，一定会积极使用这四种构件而不是自己去重新编写这些构件已经实现的功能。

在 **Android** 开发的过程中除了上述四种组件以外还有一种特别常用的工具就是 **View**（视图），所有的用户界面都由 **View** 和 **ViewGroup** 及其派生类对象组合而成^[22]，视图在文件上具体表现为 **XML** 布局文件。它是所有 **UI** 相关组件的基类，一个视图会占据屏幕上的一部分区域，负责对这片区域进行属性设置，页面渲染，事件处理。而在实际开发的过程中一般都是使用 **View** 或者 **ViewGroup** 基类派生出的子类，比如 **Button**（按钮）、**TextView**（文本框）、**ListView**（列表视图）等来组成一个页面，这些子类都可以通过 **XML** 布局文件或者 **Java** 代码来控制它们的具体行为。

总的来说，**Android** 应用开发就是通过调用应用框架层和系统运行库层提供的 **API**，使用四大组件和 **View** 工具设计出的一个具有相应功能的应用程序，最终打包成 **apk**（**AndroidPackage**，**Android** 软件安装包）文件安装在 **Android** 系统上，以供用户使用。

2.2.3 JSON 数据交换格式

JSON 是一种小量级的数据交换格式^[23]，它采用了一种独立于语言的文本格式，这使得它使用起来极为方便。它是 **JavaScript** 的一个子集语言，是目前 **web** 开发中使用最为方便和广泛的数据交换语言，它不仅易于机器的解析和生成，而且人也能非常容易的对它进行阅读和书写。

相比于传统的数据传输格式 XML 文件来说，JSON 具有很多优点：第一，它的量级比较轻；第二，开发人员能够更加容易的读写；第三，服务器和客户端能够更加轻松的序列化和解析；第四，支持多种开发语言，方便移植。

JSON 数据在传输的过程中传递的数据是类似于 Map 集合的 key-value 对应的格式，而且可以存放多对键值对，在实际使用的过程中如果需要传递多个键值对则需要先调用 JSON 的 array 方法使文本序列化为一个类似于数组的结构，然后通过 object 方法再将数组中的每一个元素序列化为一个对象，然后依次调用 key 和 value 方法来将数据的键和值一对一对的放入对象中，然后调用 endObject 方法结束对象的序列化，最后调用 endArray 方法结束数组的序列化，这样就完成对一个 JSON 文本的序列化。相比较于序列化，JSON 的解析就方便许多，首先通过 optJSONObject 方法将文本解析成多个 JSONObject 对象，然后通过调用 JSONObject 对象的 getString、getInt 等方法，方法的具体名称根据需要得到的数据的格式不同而不同，方法的参数是字符串格式的 key 值，然后就能得到相应的 key 对应的 value 值。

2.3 MySQL 数据库

MySQL 是 AB 公司开发一种开源的关系型数据库管理系统，因为其开源和免费的特点，MySQL 多用于个人和小型企业的数据库。虽然 MySQL 使用 C 和 C++ 语言开发，但是它提供了针对不同语言的 API，而且实现了多线程，它具有功能强、使用简便、管理方便、运行速度快、安全可靠性强等优点^[24]。

在 Java 中使用 JDBC 连接操作 MySQL 数据库需要先通过 Class.forName 加载 MySQL 驱动 com.mysql.jdbc.Driver，然后通过 DriverManager.getConnection 方法传入数据库地址，用户名密码等参数得到数据库连接，通过得到的连接就可以再 Java 程序中对数据库进行操作了。

2.4 本章小结

本章主要介绍了课题的交友策略研究和开发系统所使用到的开发技术，交友策略主要通过交友关系的维持和拓展两部分来介绍，目前社交网络中关系维持和关系拓展已经分离，大多数用户通过不同的社交应用来满足这两个方面的需求，通过一个应用来得到关系拓展，通过另一个应用来维持得到的关系。

本课题用到的开发技术主要为 Servlet 服务器的开发技术、Android 应用开发技术、JSON 脚本语言使用技术和 MySQL 数据库使用技术，本章着重讲了 Servlet 服务器的开发技术和 Android 应用开发技术。

第三章 需求分析和总体设计

一个系统的开发必然需要合理的需求分析，本章通过用户对课题开发的系统的社交需求入手，得到系统的功能，然后分别对系统的各个功能进行需求分析，通过需求分析得到系统的总体设计和处理流程，最后说明了系统的数据库设计。

3.1 需求分析

课题面向的用户是喜欢旅游的用户，这类用户大多喜欢结交新朋友，所以对他们来说关系拓展是比关系维持更重要的交友需求，他们不需要特别强的关系连接，只需要能够互相关注或者添加好友，相互之间存在一条关系链即可，两个陌生人之间进行关系拓展可以通过在旅游动态社区同一个帖子的回复，建立起简单的关系，然后通过邮箱查找添加关注建立确定的关注和粉丝关系，这是一种弱关系类型；而当用户在旅游动态页面募集其他用户一同去旅游或者是用户在一个景点游玩时通过附近的人添加关注互相联系一起游玩景点，两人见面前或者见面时可以互相交换联系方式，这样既加强了关系维持又方便两人在旅游时和旅游后的实时交流，这样的关系也是一种弱的关系类型，但是由于两人有过现实中的见面并且一起游玩，这种关系又比上面一种稍强一些。这两种关系对于本课题这样一个注重关系拓展的社交平台是比较合适的。

总结来说，本课题选择的交友策略主要侧重关系拓展，关系维持被极大的削弱，用户可以通过旅游动态论坛将自己的关系链充分拓展，结交更多志趣相同的朋友，由于课题项目中的关系都是弱关系，没有强的关系维持需求，所以不会出现“邓巴数字”理论中的不能维持关系的问题，而且课题项目充分满足了用户对于关系拓展方面的需求。

根据上文交友策略中需要实现的功能，将课题分成以下几个功能：用户登录注册功能、旅游动态论坛功能、游记功能、发现用户功能、用户信息管理功能。要完成以上各个功能需要在 Android 系统前台开发一个 Android 应用，用来提供用户交互功能，前台应用不仅需要能够向服务器发送请求，还需要能够接受从服务器返回的数据；系统还需要在后台搭建一个服务器，用来完成前台请求，操作数据库，实现逻辑处理，返回前台所需要的数据；系统还需要一个数据库用来存放相关信息和数据。

根据以上分析得到系统用例图如图 3.1 所示。

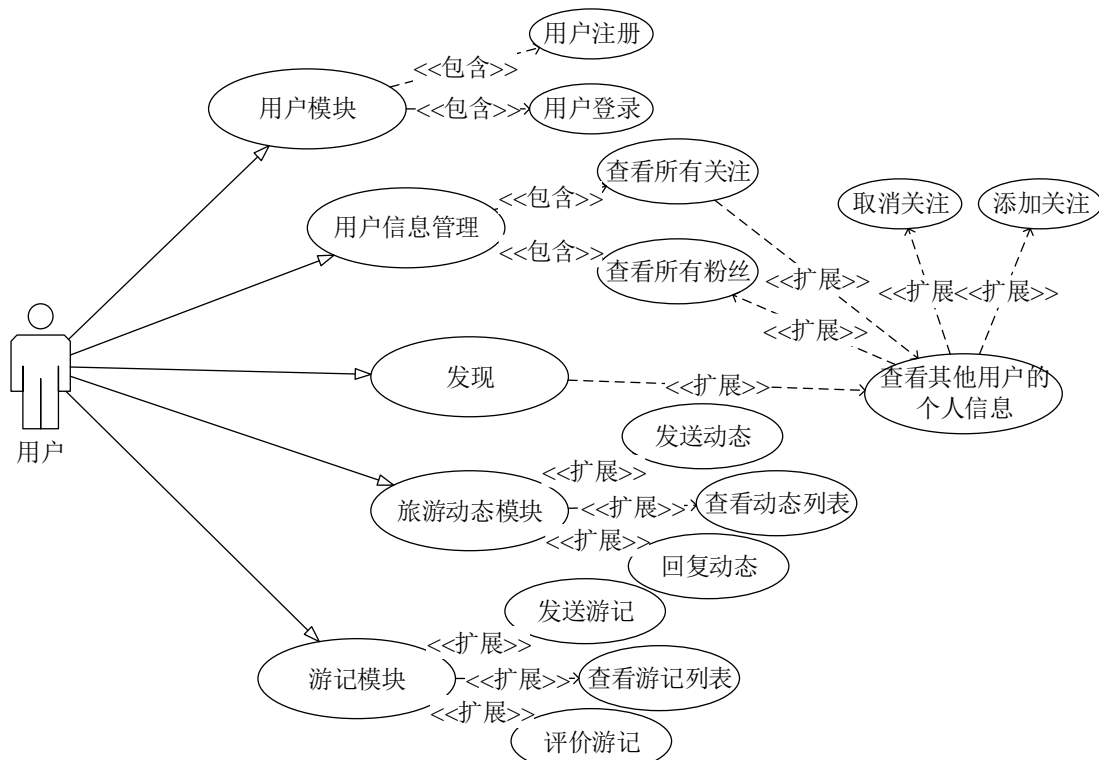


图 3.1 系统用例图

系统的功能需求为：

(1) 用户登录注册功能需求

实现用户注册，登录等基本功能，其中用户注册需要使用邮箱进行注册，注册完毕后使用邮箱密码进行登录，注册时需要提供的信息包括：邮箱、密码、用户名、所在地、性别、生日。该功能的用例图如图 3.2 所示。

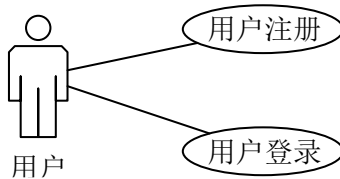


图 3.2 登录注册功能用例图

(2) 旅游动态论坛功能需求

实现旅游动态的发布，查看，对旅游动态进行回复。在这个功能中会显示所有用户的旅游动态，可以在查看动态时进行刷新。该功能的用例图如图 3.3 所示。

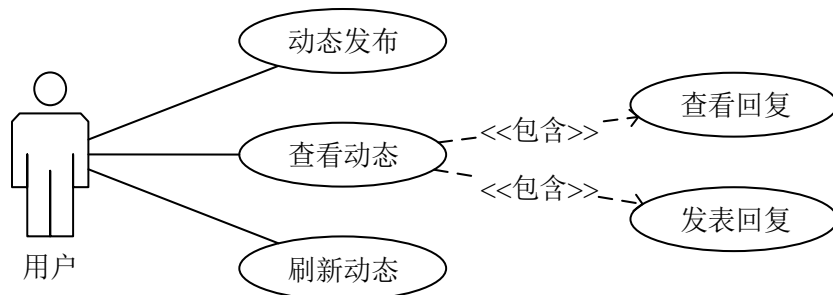


图 3.3 旅游动态功能用例图

（3）发现用户功能需求

通过附近的人或者邮箱查找其他用户，并进入查看用户信息功能，完成相应的操作，完成关系拓展的功能。该功能的用例图如图 3.4 所示。

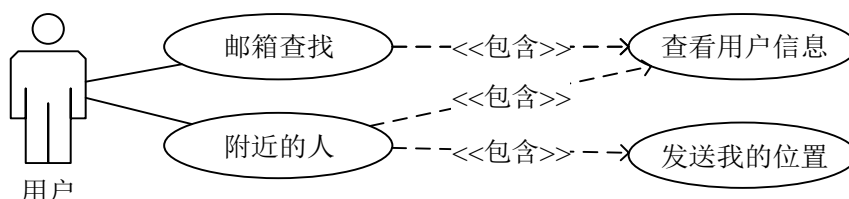


图 3.4 发现用户功能用例图

（4）游记功能需求

用户可以在这个功能中查看自己以及自己关注的用户发布的游记标题列表，可以进一步查看游记的具体内容并对其进行评论。该功能的用例图如图 3.5 所示。

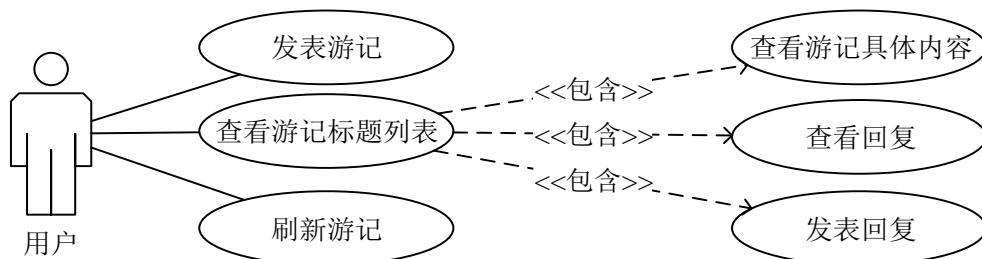


图 3.5 游记功能用例图

（5）用户信息管理功能需求

用户可以在这个功能中查看自己注册时填写的用户信息，修改登录密码，查看自己发布的所有动态和游记，查看自己所有的关注和粉丝。该功能用例图如图 3.6 所示。

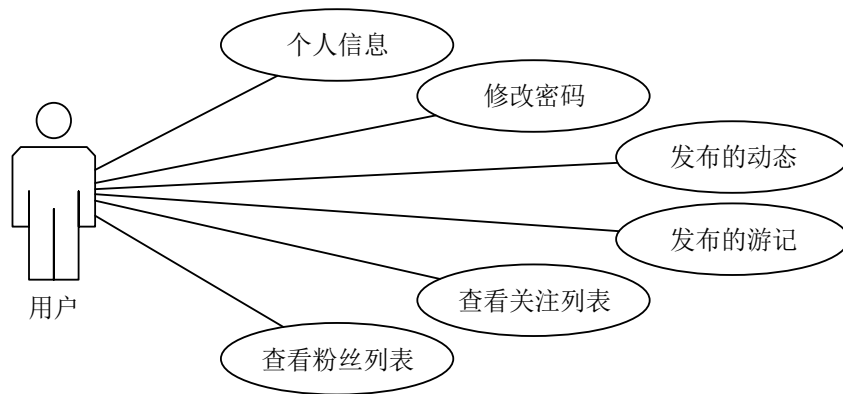


图 3.6 用户信息管理功能用例图

（6）查看用户信息功能需求

用户通过发现用户或者查看粉丝和关注列表可以查看其他用户的信息和其发表的所有动态和游记，并对其进行添加关注或者取消关注操作。该功能用例图如图 3.7 所示。

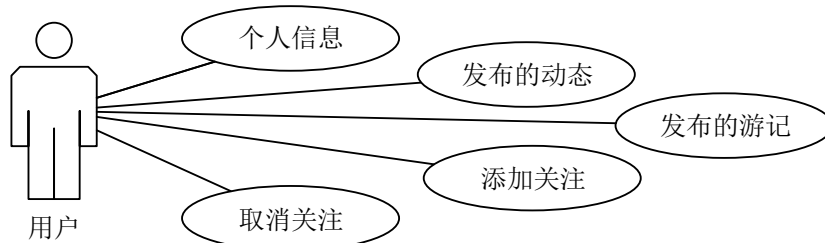


图 3.7 查看用户信息功能用例图

系统非功能的需求为：

- （1）性能需求，保证用户所有操作的响应时间不超过 5s，系统运行过程不出现卡死崩溃等情况；
- （2）代码可读性，保证代码符合相应的代码规范和包含良好的注释说明；
- （3）较好的用户体验，保证交互界面的友好，提高系统的可操作性。

3.2 总体设计

课题最终目标是在 Android 提供的 API 的基础上通过自行搭建的服务器和数据库完成一个旅游交友系统，满足用户旅游交友的目的。

系统由一个 Android 手机端应用以及一个 Web 服务器和 MySQL 数据库构成，在需求分析过程中，课题一共获得 6 个核心功能，分别是：用户登录注册功能、旅游动态论坛功能、发现用户功能、游记功能、用户信息管理功能、查看用户信息功能。所以系统的总体架构如图 3.8 所示。

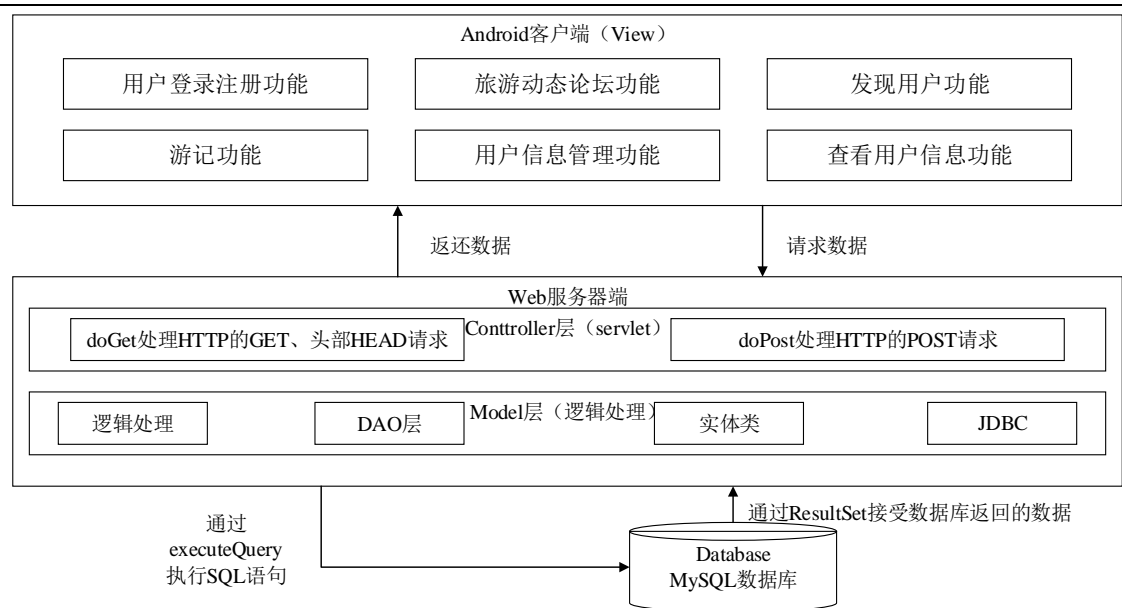


图 3.8 系统总体架构

整个系统采用 MVC 模式框架进行设计和实现，具有良好的可维护性和可扩展性^[25]。MVC 框架基于分层的思想，把整个系统分为三层，分别是：View（视图）层、Model（模型）层和 Controller（控制）层。本系统中，Android 客户端是系统的视图层，模型层和控制层都放在 Web 服务器中。

视图层主要使用 Android 的 Activity 组件，并且使用 XML 布局文件对页面进行设计和布局。Activity 在创建时会加载相应的 XML 布局文件生成用户界面，并且会对一些事件进行监听，比如焦点变化、按钮点击、触摸屏幕等，通过各种事件响应提高用户体验。

控制层和模型层都属于服务器，对于服务器的总体设计如图 3.9 所示。

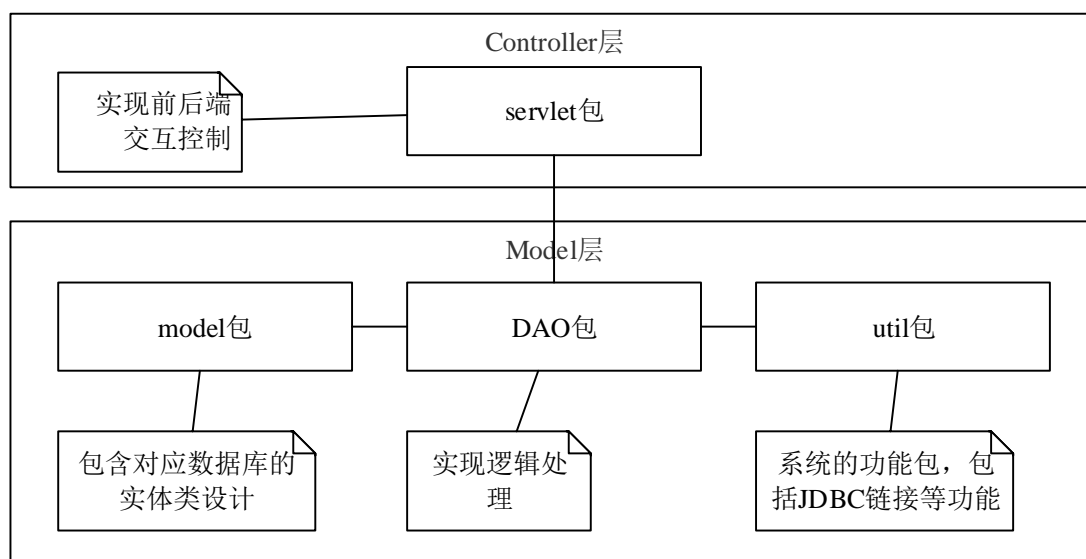


图 3.9 服务器总体设计

控制层主要包含逻辑控制部分，是由系统中所有的 **Servlet** 类组成，主要负责协调控制视图层和模型层，是两层之间的一个桥梁。得到客户端传来的数据和请求后调用相应的模型层代码，把前台传来的数据传到模型层，然后把模型层传过来的数据进行 **HTTP** 封装^[26]，根据业务需求通过 **HTTP** 协议返回给视图层。

模型层包含了具体业务的处理部分和数据库操作部分，主要工作内容是将控制层传来的数据进行入库或者查询等操作，然后将从数据库得到数据进行 **JSON** 序列化，然后返回给控制层。模型层是功能的具体实现部分，主要的逻辑实现代码都会存放在模型层，与数据库的交互也放在模型层。

根据以上设计得到系统的处理流程：

- （1）**Android** 客户端与用户进行人机交互，得到用户的请求等；
- （2）客户端通过简单的处理分析，通过 **HTTP** 请求向后台 **Web** 服务器进行数据请求；
- （3）后台 **Web** 服务器控制层 **Servlet** 类通过 **doGet** 或 **doPost** 方法得到前台请求并进行简单的处理，然后将请求传到 **Model** 层，**Model** 层对请求进行处理并调用 **JDBC** 链接数据库并获取数据并进行 **JSON** 序列化；
- （4）后台 **Web** 服务器将所得到的信息进行处理并返回给客户端；
- （5）客户端得到数据并进行 **JSON** 解析，然后将用户所需信息显示在页面上，用户通过页面获取相关信息。

3.3 数据库设计

数据库用来存放系统数据，数据库名称为 **android**，主要分为七个表，分别是：**user** 表，用来存放用户信息的表；**friend** 表，用来存放用户关系的表；**message** 表，用来存放旅游动态的表；**revert** 表，用来存放旅游动态回复的表；**travel** 表，用来存放游记的表；**comment** 表，用来存放游记评论的表；**gps** 表，用来存放用户地点信息的表；各个表的具体说明如表 3.1 到 3.7 所示。

表 3.1 user 表

列名	类型	约束	键	描述
User_id	Int(8)	Not null	主键	自增
User_name	Varchar (40)	Not null		用户昵称
Password	Varchar(20)	Not null		密码
Email	Varchar(128)	Unique		用户邮箱
Address	Varchar(100)	Not null		所在地
Sex	Varchar(10)	Check option 男 女		性别
Birthday	Date	Not null		生日

表 3.2 friend 表

列名	类型	约束	键	描述
My_id	Int(8)	Not null	主键/外键	参照 users 表
Friend_id	Int(8)	Not null	主键/外键	参照 users 表

表 3.3 message 表

列名	类型	约束	键	描述
Message_id	Int(8)	Not null	主键	自增
Owner_id	Int(8)	Not null	外键	发布者的 id
Create_time	Timestamp	Not null		发布时间, 随系统时间自动更改
Message_text	Varchar(512)	Not null		发布的文字内容

表 3.4 revert 表

列名	类型	约束	键	描述
Revert_id	Int(8)	Not null	主键	自增
Message_id	Int(8)	Not null	外键	参照 message 表
Revert_text	Varchar (521)	Not null		回复的内容
Owner_id	Int(8)	Not null	外键	参照 users 表
Create_date	Timestamp	Not null		回复的时间, 随系统时间自动更改

表 3.5 travel 表

列名	类型	约束	键	描述
Travel_id	Int(8)	Not null	主键	自增
Owner_id	Int(8)	Not null	外键	参照 user 表
Title	Varchar (128)	Not null	外键	游记标题
Time	Timestamp	Not null		发表的时间, 随系统时间自动更改
Text	Varchar (1024)	Not null	外键	游记内容

表 3.6 comment 表

列名	类型	约束	键	描述
Comment_id	Int(8)	Not null	主键	自增
Travel_id	Int(8)	Not null	外键	参照 travel 表
Text	Varchar (521)	Not null		评论的内容
Owner_id	Int(8)	Not null	外键	参照 users 表
Time	Timestamp	Not null		评论的时间

表 3.7 gps 表

列名	类型	约束	键	描述
User_id	Int(8)	Not null	主键/外键	参照 user 表
Latitude	Double	Not null		维度
Longitude	Double	Not null		经度

上面这七个表构成系统的整个数据库，课题使用 MySQL 来实现数据库，Navicat for MySQL 界面工具来管理数据库。

3.4 本章小结

本章通过对用户社交需求的分析得到系统的六个功能点，并通过用例图说明了这六个功能点的具体需求，还对系统的非功能需求进行说明。然后根据需求分析对系统进行一个总体设计，整个系统基于 MVC 框架进行设计，并且分别简要描述了 View、Model 和 Controller 三层的主要工作内容，最后对系统的数据库设计进行了说明。

第四章 详细设计和系统实现

本章首先介绍了服务器访问数据库和客户端访问服务器的设计和实现，然后对系统的各个功能模块的设计和实现进行说明。

4.1 服务器访问数据库的设计和实现

Java Servlet 服务器访问数据库一般使用 JDBC（Java Data Base Connectivity，java 数据库连接）技术实现^[27]，系统设计将数据库连接封装在 JDBCUtil.java 类中，这个类存在于 Model 层的 util 包中，这个类提供了获取数据库连接的接口，用户可以根据自己的需求和数据库连接进行与数据库相关的操作。

JDBCUtil 类的构造函数中直接就会得到数据库连接，其中 getConnection 方法的各个参数分别表示：数据库 android 位于服务器本机并向端口 3306 提供数据而且连接数据库后数据传输格式为 utf-8，数据库用户名为 root，密码为 123456。如果其他类需要用到数据库连接则可以直接通过调用 JDBCUtil.connect 来得到数据库连接，获取数据库连接的核心代码如下：

```
private String url = "jdbc:mysql://localhost:3306/android?useUnicode=true&
                    characterEncoding=utf8";

public Connection connect = null;
public Statement statement = null;
public ResultSet rs = null;
public JDBCUtil() {
    try {
        Class.forName("com.mysql.jdbc.Driver");// 加载 JDBC 驱动
        connect = DriverManager.getConnection(url, "root", "123456");
        statement = connect.createStatement();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

4.2 客户端访问服务器设计和实现

因为从 Android3.0.6 以后的版本中要求 Android 应用程序中所有涉及到 Intent 网络服务的操作都需要采用异步任务的方式进行，所以在客户端程序设计的过程

中，所有对服务器的访问部分也采用异步的方式来实现，即在不影响 UI 线程（主线程）的情况下定义一个子线程，将对服务器访问相关的操作放到子线程上去执行，然后通过调用继承了 `Handler` 类的自定义子类来实现线程间的异步通信，`Handler` 运行在主程序中，它可以与子线程通过 `Message` 对象来进行数据传递，这样就可以在子线程发送数据之后在主线程对数据进行显示。

Android 程序设计中，所有对于服务器的访问的功能都在 `tools` 包的 `Tools` 类中实现，该类规定了一系列的用户自定义方法来实现程序对服务器请求的不同实现。其中与服务器连接的方法是固定的。首先通过 `BasicNameValuePair` 对象来存放需要传递到后台的数据，然后再定义一个 `list` 集合包含所有用户定义的 `BasicNameValuePair` 对象，然后定义一个 `HTTPEntity` 来封装 `list` 集合，然后定义一个 `HTTPPost` 对象并将要请求的 `url` 地址传入这个对象，然后使用 `DefaultHttpClient` 类的 `execute` 方法发送 HTTP POST 请求，并返回 `HttpResponse` 对象，然后通过调用 `HttpResponse` 对象的 `getStatusCode` 来得到 HTTP 状态码，如果状态码不等于 400 则表示服务器访问成功，否则访问失败。

关于服务器访问相关的操作全都要放在 `Thread` 类的 `run` 方法中，这样当系统运行时就会创建一个子线程来完成这些操作，通过 `EntityUtils.toString(httpResponse.getEntity()).trim()` 可以得到服务器放回的数据的 `String` 字符串格式，然后可以通过 JSON 解析得到服务器返回具体数据。比如，对于登录注册功能，服务器返回的数据可能只是一个很简答的表示是否成功的键值对，那么 JSON 解析就比较简单，但是对于旅游动态功能来说，服务器传回来的数据就比较复杂，首先是一个 `JSONArray` 对象，它是一个 `JSONObject` 对象的数组，其中包含许多 `object` 对象，每一个对象中又是类似于 `map` 集合的多对键值对存放的数据，这样的返回数据解析起来就比较复杂，所以具体实现的过程中需要根据实际情况进行不同的处理，客户端访问服务器的核心代码如下：

```
try {
    String URL = "http://10.0.2.2:8080/Android/servlet/MessageServlet";
    HttpPost request = new HttpPost(URL);
    List<NameValuePair> params = new ArrayList<NameValuePair>();
    params.add(new BasicNameValuePair("u_id", uId));
    params.add(new BasicNameValuePair("operate", "1"));
    request.setEntity(new UrlEncodedFormEntity(params, HTTP.UTF_8));
    HttpResponse httpResponse = new DefaultHttpClient().execute(request);
    if (httpResponse.getStatusLine().getStatusCode() != 404) {
        back = EntityUtils.toString(httpResponse.getEntity()).trim();
    }
} catch (Exception e) {
```

```
e.printStackTrace();
}
```

4.3 登录注册模块的设计和实现

登录注册模块的设计如下，用户第一次打开应用时首先打开登录页面，登录页面提供两个输入框给用户输入登录邮箱和密码，然后提供一个登录按钮和一个注册按钮，登录按钮可以提交输入的信息并登录，注册按钮可以跳转到注册页面。注册页面提供一系列用户注册是需要输入的用户信息输入框，包括注册邮箱、密码、确认密码、用户名、住址、性别、生日，还会提供两个按钮，分别是注册和登录，注册按钮可以提交输入的信息并注册，登录按钮可以跳转到登录页面。当用户登录或者注册后会直接进入系统，登录注册模块设计的流程如图 4.3 所示。

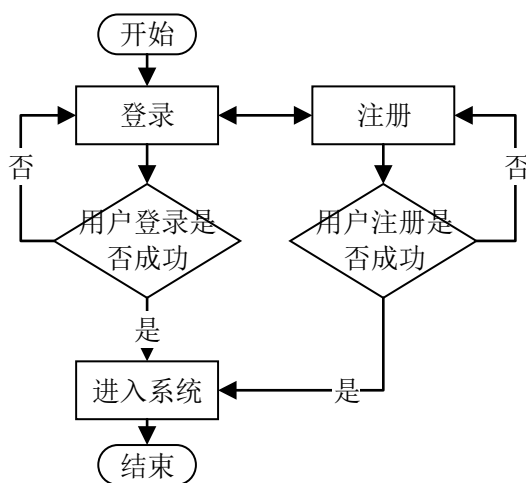


图 4.3 登录注册模块流程

在该模块的具体实现时，页面布局 XML 文件中输入框使用 EditText 组件来实现，按钮使用 Button 组件来实现，整体的布局都采用 LinearLayout 布局，在 Android 项目的 Java 文件中在 onCreate 方法中使用 setContentView 方法来在页面创建时加载页面布局文件，然后实现显示相应的页面。通过调用按钮的 setOnClickListener 方法来对按钮添加点击监听，该方法的参数为 OnClickListener 对象，重写该对象的 onClick 方法来调用 Tools 类中的登录方法，完成与服务器的交互，得到返回数据通过 Handler 类的 sendMessage 方法将得到的数据发送给主线程，然后主线程中通过重新 Handler 类的 handleMessage 方法判断服务器是否允许用户登录或者注册，然后进行相关的跳转处理或者错误提示。

该模块中页面跳转实现核心代码如下：

```
Intent it = new Intent(ActivityLogin.this, ActivityMain.class);
it.putExtra("u_id", userEmail.getText().toString());
ActivityLogin.this.startActivity(it);
```

服务器中判断用户是否允许登录的核心代码如下：

```
pStatement = connection
    .prepareStatement("select * from user where Email=? and Password=?");
pStatement.setString(1, user.getEmail());
pStatement.setString(2, user.getPassword());
```

服务器中判断用户是否允许注册的核心代码如下：

```
pStatement = connection
    .prepareStatement("insert into User (User_name , Password , Email ,
        Address , Sex , Brithday) values(?,?,?,?);");
pStatement.setString(1, user.getUserName());
pStatement.setString(2, user.getPassword());
pStatement.setString(3, user.getEmail());
pStatement.setString(4, user.getAddress());
pStatement.setString(5, user.getSex());
pStatement.setString(6, user.getBrithday());
```

系统登录和注册时前端向后端请求的接口和参数如表4.1所示。

表4.1 登录注册模块接口及参数表

访问类型	访问接口	参数
登录	http://10.0.2.2:8080/Android/servlet/LoginServlet	email password
注册	http://10.0.2.2:8080/Android/servlet/RegistServlet	email password userName address sex birthday

后端返回的数据比较简单，只会返回true或者false。

4.4 旅游动态论坛模块的设计和实现

旅游动态论坛模块的设计如下，当用户登录或者注册后进入系统，显示的首页就是旅游动态论坛，用户可以在论坛查看和刷新所有用户发布的旅游动态，自己也能发布旅游动态，同时点击某一条旅游动态可以查看该旅游动态的详情和回复，然后对该旅游动态做出回复，还需要提供三个底端按钮用来跳转到用户信息管理模块，游记模块和发现用户模块，旅游动态论坛模块设计的流程如图 4.4 所示。

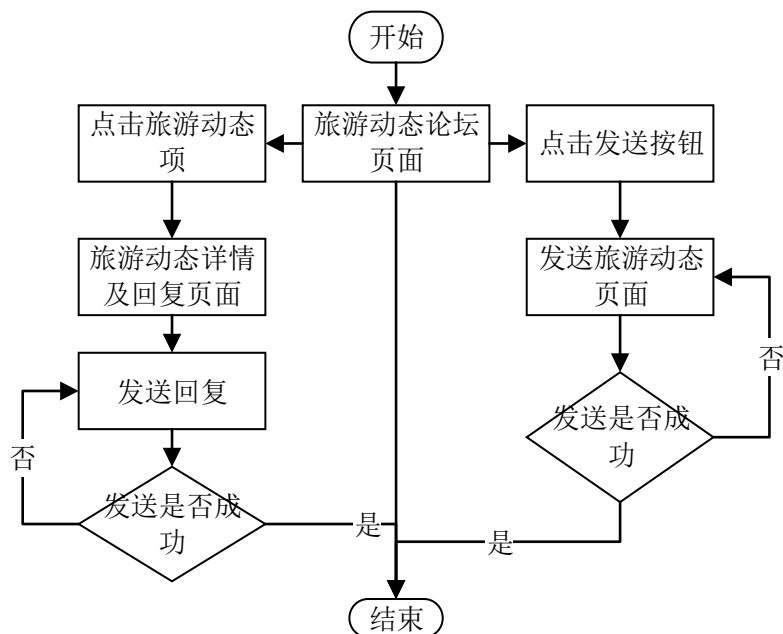


图 4.4 旅游动态模块流程

该模块具体实现时可以在页面配置 XML 文件中使用 `ListView` 组件来实现旅游动态的列表显示，使用此组件时需要定义一个新的页面配置文件，并在其中说明列表中的每一项的显示布局是怎样的，并且要在每一项中添加点击事件监听和跳转响应，发送页面和回复则使用 `EditText` 组件和 `Button` 组件实现。同样的这个模块中所有与服务器交互的部分全都放在 `Tools` 类中，并且需要创建一个子线程来执行这些操作。

在列表中的每一项中加载需要的信息并添加点击事件监听和跳转响应的核心代码如下：

```

simpleAdapter = new SimpleAdapter(ActivityMain.this, lists,
    R.layout.item_message, new String[] { "owner_name", "time",
    "text" }, new int[] { R.id.owner_name, R.id.time, R.id.text });
listview.setAdapter(simpleAdapter);
listview.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View
        view , int position, long id)
    }
}
  
```

系统获取动态列表、发表动态、获取动态回复和发表回复时前端向后端请求的接口和参数如表4.2所示。

表4.2 旅游动态模块接口及参数表

访问类型	访问接口	参数
获取动态列表	http://10.0.2.2:8080/Android/servlet/MessageServlet	u_id
发表动态	http://10.0.2.2:8080/Android/servlet/sendMessage	u_id content
获取动态回复	http://10.0.2.2:8080/Android/servlet/RevertServlet	m_id
发表回复	http://10.0.2.2:8080/Android/servlet/sendRevert	u_id m_id content

服务器向数据库获取所有用户的动态的代码比较简单，核心代码如下：

```
pStatement = connection.prepareStatement("select b.* from (select
message.message_id,message.owner_id,user.user_name,message.create_time,me
ssage.message_text from user,message where message.owner_id = user.user_id)
b order by b.create_time desc;");
```

服务器向客户端返回动态列表数据的打包方法代码如下：

```
jsonString.object().key("message_id")
.value(rSet.getInt("message_id")).key("owner_id")
.value(rSet.getString("owner_id"))
.key("owner_name")
.value(rSet.getString("user_name"))
.key("create_time")
.value(rSet.getTimestamp("create_time"))
.key("message_text")
.value(rSet.getString("message_text")).endObject();
```

客户端通过请求动态列表得到的数据格式如下：

```
[{"message_id":23,"owner_id":1,"owner_name":"宋一",
"create_time":"2016-04-25 17:28:18.0","message_text":"周日去黄山玩，一起去的评论报名参加"},{...}...]
```

客户端通过发送动态或者回复得到的数据是true或false。

客户端通过请求回复列表得到数据格式如下：

```
[{"owner_name":"宋一","time":"2016-04-26 09:09:14.0","text":"参加的也可以发邮件告诉我，邮件地址见我的个人信息"},{...}...]
```

4.5 发现用户模块的设计和实现

用户通过主页旅游动态论坛页面的按钮跳转到发现用户模块，该模块的设计

如下，用户可以通过输入框输入想要查找用户的注册邮箱然后找到该用户，或者是通过附近的人按钮获取附近一定范围内的其他用户，并通过列表显示出来，然后点击一个用户列表项进入查看用户信息模块，同时用户也可以只发送自己的位置，让其他用户来寻找自己。发现用户模块设计的流程如图 4.5 所示。

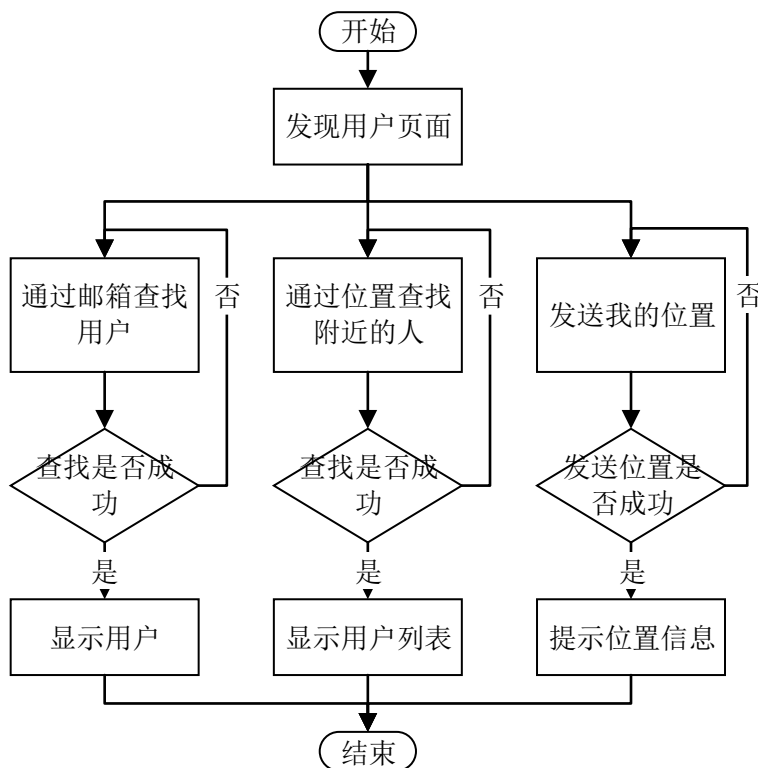


图 4.5 发现用户模块流程

该模块的具体实现时列表，输入框和按钮的实现都与上述模块实现类似，而获取用户当前位置信息是本模块实现的难点，首先需要定义一个 `LocationManager` 对象来加载 Android 系统的位置驱动，然后定义获取位置的参数 `Criteria` 对象，然后将参数传入 `LocationManager` 对象的 `getBestProvider` 方法得到一个最好的位置参数字符串 `provider`，然后注册一个 1 分钟为周期的监听函数，周期性的监听位置的变化，最后将 `provider` 传入 `LocationManager` 对象的 `getLastKnownLocation` 方法得到用户最后的定位位置，这样能得到用户的当前位置信息的 `Location` 对象。

获取用户位置的核心代码如下：

```

LocationManager locationManager = (LocationManager) getSystemService(
    Context.LOCATION_SERVICE);

// 设置 Criteria 的信息
Criteria criteria = new Criteria();
// 经度要求
criteria.setAccuracy(Criteria.ACCURACY_FINE);
criteria.setAltitudeRequired(false);
  
```

```

criteria.setBearingRequired(false);
criteria.setCostAllowed(false);
criteria.setPowerRequirement(Criteria.POWER_LOW);
// 取得效果最好的 criteria
String provider = locationManager.getBestProvider(criteria, true);
// 注册一个周期性的更新，1min 更新一次
locationManager.requestLocationUpdates(provider, 0, 1000 * 60,
    new getGpsLocationListner());
// 得到坐标相关的信息
Location location = locationManager.getLastKnownLocation(provider);

```

获取用户位置信息后附近的人功能通过经纬度范围来实现，在服务器中将位置信息在当前用户位置经纬度一定范围内的用户全部返回给客户端即可，服务器端获取附近的用户的核心代码如下：

```

double latdistance = rSet.getDouble("latitude") - myLat;
double londistance = rSet.getDouble("longitude") - myLon;
if (latdistance < 1 && latdistance > -1 && londistance < 1 && londistance > -1)

```

系统邮箱查找用户、发送我的位置和查找附近的人时前端向后端请求的接口和参数如表4.3所示。

表4.3 发现用户模块接口及参数表

访问类型	访问接口	参数
通过邮箱查找	http://10.0.2.2:8080/Android/servlet/UserServlet	u_id
发送我的位置	http://10.0.2.2:8080/Android/servlet/sendGPS	u_id latString longString
查找附近的人	http://10.0.2.2:8080/Android/servlet/searchNearServlet	u_id

服务器向客户端返回附近人列表数据的打包方法代码如下：

```

jsonString.object().key("getUser_id")
    .value(getUser.getUserId()).key("getUser_name")
    .value(getUser.getUserName()).key("email")
    .value(getUser.getEmail()).key("sex")
    .value(getUser.getSex()).key("city")
    .value(getUser.getAddress()).endObject();

```

客户端通过邮箱查找用户得到的数据格式如下：

```

[{"user_id":1,"user_name":"宋一","email":"sy@163.com","City":"南京","sex":"男","Bday":"1996-01-15"}]

```

客户端通过发送我的位置得到的数据是true或false。

客户端通过查找附近的人得到数据格式如下：

```
[{"getUser_id":4,"getUser_name":" 许 二 ","email":"xe@163.com","sex":" 男", "city":"南京"},{...}...]
```

4.6 游记模块的设计和实现

用户通过主页旅游动态论坛页面的按钮跳转到游记模块，游记模块的设计大致与旅游动态论坛相似，主要的区别是用户发送的游记只能被自己和自己的粉丝看到，游记模块设计的流程如图 4.7 所示。

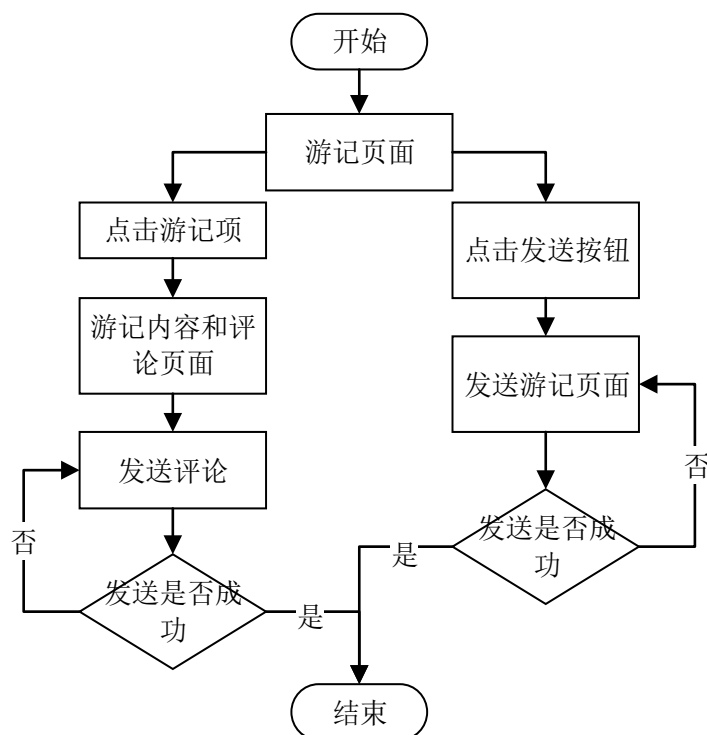


图 4.7 游记模块流程

游记模块的实现难点主要在于如何使游记只能被作者和作者粉丝看到，处理的核心在加载游记时就只在数据库查找用户自己发布的游记和关注的人发布的游记，所以重点在于查找数据库时的 SQL 语句，具体实现的语句如下：

```
select b.* from (
    select travel.travel_id , user.user_name , travel.time , travel.title , travel.text
    from user , travel
    where travel.owner_id in(select friend_id from friend where my_id=?) and
    travel.owner_id = user.user_id
    union select travel.travel_id,user.user_name,travel.time,travel.title,travel.text
    from user,travel
    where travel.owner_id = user.user_id and travel.owner_id=?
```

) b order by b.time desc;

系统获取游记列表、发表游记、获取游记评论和发表评论时前端向后端请求的接口和参数如表4.4所示。

表4.4 游记模块接口及参数表

访问类型	访问接口	参数
获取游记列表	http://10.0.2.2:8080/Android/servlet/TravelServlet	u_id
发表游记	http://10.0.2.2:8080/Android/servlet/sendTravel	u_id title content
获取游记评论	http://10.0.2.2:8080/Android/servlet/CommentServlet	t_id
发表评论	http://10.0.2.2:8080/Android/servlet/sendComment	u_id t_id content

服务器向客户端返回游记列表数据的打包方法代码如下：

```
jsonString.object().key("travel_id")
    .value(rSet.getInt("travel_id")).key("owner_name")
    .value(rSet.getString("user_name"))
    .key("create_time")
    .value(rSet.getTimestamp("time"))
    .key("travel_title").value(rSet.getString("title"))
    .key("travel_text").value(rSet.getString("text"))
    .endObject();
```

客户端通过请求游记列表得到的数据格式如下：

```
[{"travel_id":4,"owner_name":"宋一","create_time":"2016-05-07
16:01:28.0","travel_title":"鼎山","travel_text":"今天没去爬了一次鼎山，从大三之后
就没去过，今天再去别有一番滋味。"},{...}...]
```

客户端通过发送游记或者评论得到的数据是true或false。

客户端通过请求评论列表得到数据格式如下：

```
[{"owner_name":"许二","time":"2016-05-07 12:01:08.0","text":"真的吗？下次有
时间我也去玩玩"},{...}...]
```

4.7 用户信息管理模块的设计和实现

用户通过主页旅游动态论坛页面的按钮跳转到用户信息管理模块，用户信息管理模块的设计如下，在查看自己发布的旅游动态和游记列表上的实现与上述两个模块的设计类似，在查看自己的粉丝和关注的实现显示的用户列表功能与发现

用户模块的实现类似，本模块的特有的设计是修改密码，粉丝和关注用户的判断，用户信息管理模块设计的流程如图 4.8 所示。

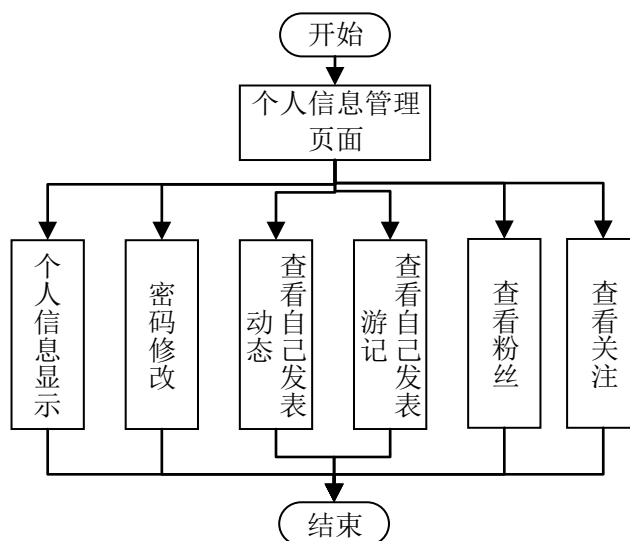


图 4.8 用户信息管理模块流程

用户信息管理模块显示的用户信息是通过页面配置 XML 文件的 TextView 来实现的，它可以在系统的 Java 文件中进行显示操作。而用户粉丝和关注的判断主要在查询数据库时实现，查找用户所有的粉丝的 SQL 语句：select * from friend where My_id=？，查找用户所有关注的 SQL 语句：select * from friend where Friend_id=？。

修改用户密码在服务器中实现的核心代码如下：

```
pStatement = connection
```

```
.prepareStatement("update user set Password=? where User_id=?");
```

```
pStatement.setString(1, nPass);
```

```
pStatement.setInt(2, user.getUserId());
```

系统获取当前用户信息、修改密码、查看自己发布的动态、查看自己发布的游记、获取粉丝列表和关注列表时前端向后端请求的接口和参数如表4.5所示。

表4.5 用户信息管理模块接口及参数表

访问类型	访问接口	参数
用户个人信息	http://10.0.2.2:8080/Android/servlet/UserServlet	u_id
修改密码	http://10.0.2.2:8080/Android/servlet/ChangePassServlet	u_id newPass
我的动态列表	http://10.0.2.2:8080/Android/servlet/OneMessageServlet	u_id
我的游记列表	http://10.0.2.2:8080/Android/servlet/OneTravelServlet	u_id
根据type获取 相应用户列表	http://10.0.2.2:8080/Android/servlet/GetUserList	u_id type

服务器向客户端返回用户列表数据的打包方法代码如下：

```
jsonString.object().key("getUser_id")
    .value(getUser.getId()).key("getUser_name")
    .value(getUser.getUserName()).key("email")
    .value(getUser.getEmail()).key("sex")
    .value(getUser.getSex()).key("city")
    .value(getUser.getAddress()).endObject();
```

客户端通过请求个人动态列表和个人游记列表得到的数据格式与旅游动态模块和游记模块中得到数据格式一致。

客户端通过修改密码得到的数据是true或false。

客户端通过请求关注或者粉丝列表得到数据格式如下：

```
[{"getUser_id":4,"getUser_name":" 许 二 ","email":"xe@163.com","sex":" 男",
"city":"南京"},{...}...]
```

4.8 查看用户信息模块的设计和实现

用户通过点击发现用户模块或者用户信息管理模块中的用户列表中的某一项就可以跳转到查看用户信息模块，在这个模块中可以查看到用户的个人信息和发布过的旅游动态和游记，然后还能在查看用户信息是对该用户进行添加关注和取消关注的操作，提供相应的跳转按钮，用户信息管理模块设计的流程如图 4.9 所示。

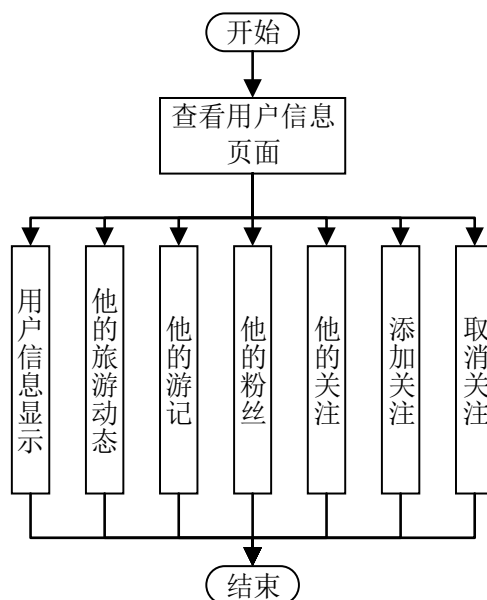


图 4.9 查看用户信息模块流程

首先在实现点击某一项进入查看用户信息模块的跳转实现与旅游动态的跳转实现类似，都是通过 ListView 来实现，显示用户信息则与用户信息管理模块一样使用 TextView 来实现，发布过的动态和游记的实现也与用户信息管理模块类似。

点击用户列表跳转到查看用户信息模块的核心代码如下：

```
simpleAdapterNear = new SimpleAdapter(ActivityNear.this, NUList,
    R.layout.item_nearuser, new String[] { "user_name",
    "user_email" }, new int[] { R.id.nuser_name,
    R.id.nuser_email });
near_userList.setAdapter(simpleAdapterNear);
near_userList.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id)
    }
}
```

添加和删除关注在服务器中实现的核心代码如下：

```
pStatement = connection
    .prepareStatement("insert into friend(My_id,Friend_id) values(?,?)");
pStatement.setInt(1, user.getUserId());
pStatement.setInt(2, otherUser.getUserId());
pStatement = connection
    .prepareStatement("delete from friend where My_id=? and Friend_id=?");
pStatement.setInt(1, user.getUserId());
pStatement.setInt(2, otherUser.getUserId());
```

系统获取用户信息、查看他人发布的动态、查看他人发布的游记、获取他人粉丝列表和关注列表、添加关注和取消关注时前端向后端请求的接口和参数如表 4.6 所示。

表4.6 登录注册接口及参数表

访问类型	访问接口	参数
用户信息	http://10.0.2.2:8080/Android/servlet/UserServlet	u_id
我的动态列表	http://10.0.2.2:8080/Android/servlet/OneMessageServlet	u_id
我的游记列表	http://10.0.2.2:8080/Android/servlet/OneTravelServlet	u_id
根据type获取相应用户列表	http://10.0.2.2:8080/Android/servlet/GetUserList	u_id type
添加关注	http://10.0.2.2:8080/Android/servlet/AddFollowServlet	u_id otherUser_id
取消关注	http://10.0.2.2:8080/Android/servlet/RemoveFollowServlet	u_id otherUser_id

服务器向客户端返回用户列表数据的打包方法代码如下：

```
jsonString.object().key("getUser_id")  
    .value(getUser.getId()).key("getUser_name")  
    .value(getUser.getUserName()).key("email")  
    .value(getUser.getEmail()).key("sex")  
    .value(getUser.getSex()).key("city")  
    .value(getUser.getAddress()).endObject();
```

客户端通过请求个人动态列表和个人游记列表得到的数据格式与旅游动态模块和游记模块中得到数据格式一致。

客户端通过添加关注和取消关注得到的数据是true或false。

客户端通过请求关注或者粉丝列表得到数据格式如下：

```
[{"getUser_id":4,"getUser_name":" 许 二 ","email":"xe@163.com","sex":" 男  
","city":"南京"},{...}...]
```

4.9 本章小结

本章主要说明了系统的设计和实现，首先介绍了系统实现中的后端与数据以及前后端的连通实现，然后对系统设计中客户端的每一个功能模块通过具体分析，画出模块设计流程图，然后介绍了系统实现过程中的核心功能代码和逻辑，列出了每个模块前后端交互的接口和参数，并且说明了服务器返回给客户端的各种数据的格式。

第五章 系统测试

一个应用程序开发的是否成功不仅取决于系统的设计和编码质量高低，能否通过测试也十分重要。本章介绍对系统的测试，包括模块测试、功能测试、系统兼容性测试和系统性能测试。

5.1 测试环境

模块测试和功能测试都是使用 Android SDK 自带的安卓模拟器进行测试。

系统兼容性测试和系统性能测试需要大量 Android 系统设备，由于没有足够数量的设备进行测试，所以本课题使用百度公司的智能云计算服务平台“百度开放云”提供的移动 App 测试功能，对课题开发的旅游交友系统进行系统兼容性测试和系统性能测试。具体测试方法是注册百度账号，实名完善个人信息，然后平台会提供一次免费进行测试的机会，通过平台提供的接口上传系统应用的 apk 安装文件，系统会自动测试该应用的兼容性和性能，系统总共进行了 47 次测试，将课题开发的应用在不同机型不同版本的 Android 系统设备上进行了测试。

课题的系统的测试分为四种测试：模块测试、功能测试、兼容性测试和性能测试。

5.2 模块测试

对于登录注册模块的测试主要是分别输入正确和错误的用户名和密码后能否登录成功，输入错误或者为空的注册信息能够注册成功，具体的测试用例和结果如表 5.1 所示。

表 5.1 登录注册模块测试用例表

输入或动作	期望的输出或相应	实际结果
输入正确的账号密码	登录成功	符合预期
输入错误的账号密码	登录失败	符合预期
输入已经注册过的邮箱	注册失败	符合预期
两次密码输入不一致	不允许注册	符合预期
输入错误的性别或生日	不允许注册	符合预期
输入正确的注册信息	注册成功	符合预期

对于旅游动态模块的测试主要是能否正确的加载和刷新数据，能否正常发布动态，能否正常查看动态详情和回复以及能否正常发表回复。具体的测试用例和结果如表 5.2 所示。

表 5.2 旅游动态论坛模块测试用例表

输入或动作	期望的输出或相应	实际结果
无	正常加载旅游动态	符合预期
点击刷新按钮	刷新成功	符合预期
点击发送按钮	跳转到发送动态页面	符合预期
输入正确的动态	发送动态成功	符合预期
输入错误的动态	发送动态失败	符合预期
点击动态列表某一项	正确跳转到相应的页面	符合预期
输入正确的回复	发送回复成功	符合预期
输入错误的回复	发送回复失败	符合预期

对于发现用户模块测试主要是能否正确的通过输入的用户邮箱查找到用户，能够正确的发送自己的位置，能否查找到附近的人以及点击用户列表某一项可以正确跳转到查看用户信息模块。具体的测试用例和测试结果如表 5.3 所示。

表 5.3 发现用户模块测试用例表

输入或动作	期望的输出或相应	实际结果
输入正确的邮箱	输出相应的用户	符合预期
输入错误的邮箱	不输出任何用户	符合预期
点击发送我的位置	发送成功	符合预期
点击查找附近的人	输出附近用户列表	符合预期
点击用户列表某一项	正确跳转到用户信息页面	符合预期

对于游记模块的测试与旅游动态论坛测试类似，主要是能否正确的加载和刷新数据，能否正常发布游记，能否正常查看游记详情和评论以及能否正常发表评论。具体的测试用例和测试结果如表 5.4 所示。

表 5.4 游记模块测试用例表

输入或动作	期望的输出或相应	实际结果
无	正常加载游记	符合预期
点击刷新按钮	刷新成功	符合预期
点击发送按钮	跳转到发送游记页面	符合预期
输入正确的游记标题和内容	发送游记成功	符合预期
输入错误的游记标题和内容	发送游记失败	符合预期
点击游记列表某一项	正确跳转到相应页面	符合预期
输入正确的评论格式	发送评论成功	符合预期
输入错误的评论格式	发送评论失败	符合预期

对于用户信息管理模块的测试比较复杂，首先是能否正确的显示用户个人信息，然后测试修改密码功能能否正常使用，查看我的动态，游记，粉丝和关注按钮能否跳转到相应的页面并正确的加载列表项。具体的测试用例和测试结果如表 5.5 所示。

表 5.5 游记模块测试用例表

输入或动作	期望的输出或相应	实际结果
无	输出正确的用户信息	符合预期
点击修改密码	正确跳转到修改密码页面	符合预期
输入新的密码	修改密码成功	符合预期
输入的两次新密码不一致	不允许修改密码	符合预期
点击我的动态按钮	正确跳转到动态列表页面并 正确加载相应动态数据	符合预期
点击我的游记按钮	正确跳转到游记页面并正确 加载相应游记数据	符合预期
点击我的粉丝按钮	正确跳转到粉丝页面并正确 加载粉丝列表数据	符合预期
点击我的关注按钮	正确跳转到关注页面并正确 加载关注列表数据	符合预期

对于查看用户信息模块的测试主要是能否正确加载用户信息和他的动态，游记，粉丝和关注按钮能否正常跳转和加载数据，以及各种情况的关注和取消关注能够正确响应。具体的测试用例和测试结果如表 5.6 所示。

表 5.6 游记模块测试用例表

输入或动作	期望的输出或相应	实际结果
无	输出正确的用户信息	符合预期
点击他的动态按钮	正确跳转到动态列表页面并 正确加载相应动态数据	符合预期
点击他的游记按钮	正确跳转到游记页面并正确 加载相应游记数据	符合预期
点击他的粉丝按钮	正确跳转到粉丝页面并正确 加载粉丝列表数据	符合预期
点击他的关注按钮	正确跳转到关注页面并正确 加载关注列表数据	符合预期
点击添加关注按钮	正确的关注用户	符合预期
点击取消关注按钮	正确的取消关注	符合预期

5.3 功能测试

功能测试的结果如下图 5.1 到 5.6 所示。



图 5.1 登录注册功能测试结果

图 5.1 说明系统的登录注册功能和页面是正确实现的，通过输入或者点击测试证明登录和注册功能和页面控件都能正常使用。



图 5.2 旅游动态功能测试结果

图 5.2 说明系统中旅游动态列表和旅游动态详情是能够正确加载的，通过输入和点击测试证明功能实现正确，页面控件可以正常使用。

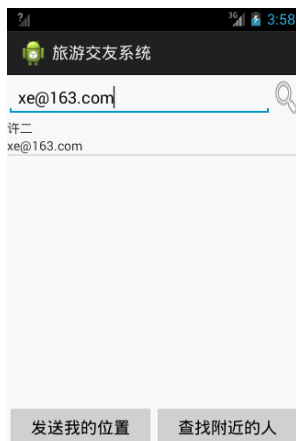


图 5.3 发现用户功能测试结果

图 5.3 通过输入和点击测试说明系统中通过邮箱查找用户，发送我的位置和查找附近的人这三个功能和页面控件都能正常使用。



图 5.4 游记功能测试结果

图 5.4 说明系统中驴友圈游记列表和游记详情是能够正确加载的，通过输入和点击测试说明游记相关功能和页面控件都能正常使用。



图 5.5 用户信息管理功能测试结果

图 5.5 说明系统中用户信息管理中的各个功能和其页面都是被正确实现的，点击测试表明页面上的各个按钮都能正常使用。



图 5.6 查看用户信息功能测试结果

图 5.6 说明系统中查看用户信息中的各个功能及其页面都是被正确实现的，点击测试表明页面上的各个按钮都能正常使用。

5.4 系统兼容性测试

系统兼容性测试的总体测试结果如图 5.7 所示。可以看出系统通过了全部 47 次验证，说明本系统能够兼容目前市场上绝大多数 Android 设备。

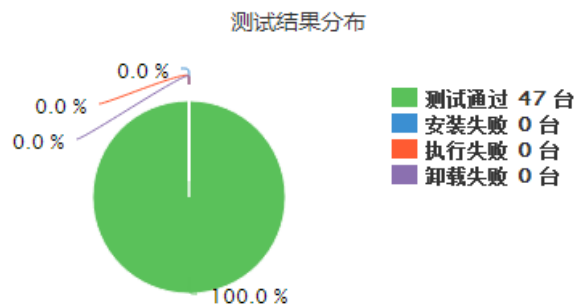


图 5.7 系统兼容性测试结果

系统对于不同品牌机型的兼容性测试结果如图 5.8 所示。可以看出对于目前市场上用户较多的几种品牌，本系统都能够完美兼容。

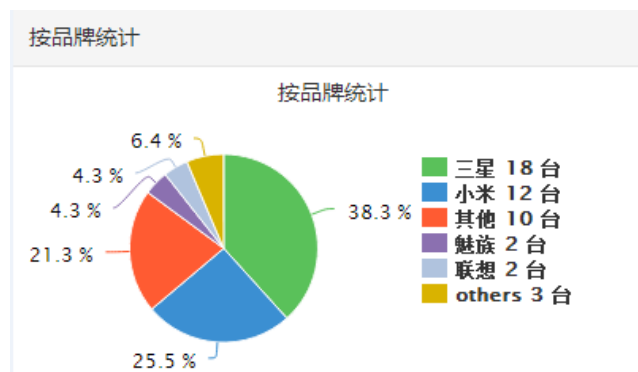


图 5.8 不同品牌机型兼容性测试结果

系统对于不同 Android 版本的机型的兼容性测试结果如图 5.9 所示。可以看出系统对于目前主流的 Android 版本都能够很好地兼容。

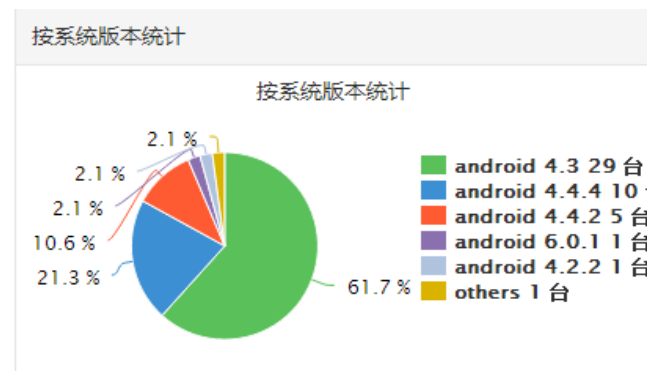


图 5.9 不同 Android 版本兼容性测试结果

系统对于不同分辨率设备的兼容性测试结果如图 5.10 所示。可以看出系统对于一些常用分辨率的兼容性很好，可以给绝大多数用户提供正常的页面和用户体验。

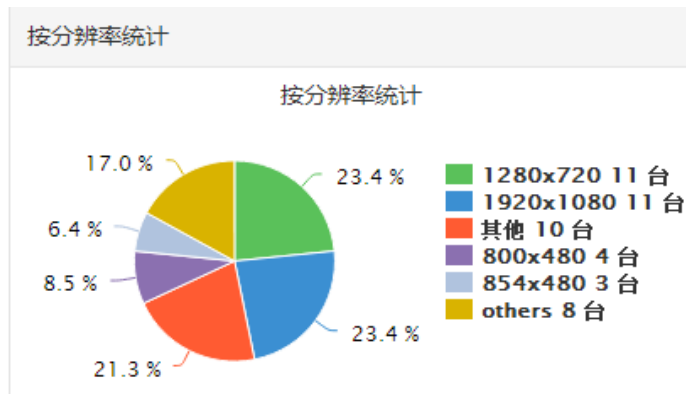


图 5.10 不同分辨率兼容性测试结果

根据上面的测试结果，可以说明本系统对于目前主流的不同型号不同系统版本不同分辨率的 Android 设备都能很好的兼容。

5.5 系统性能测试

对于系统性能的测试主要分为五个方面，cpu 占用测试、内存占用测试、启动耗时测试、流量耗用测试、电量耗用测试。测试结果分别如图 5.11 到 5.15 所示。

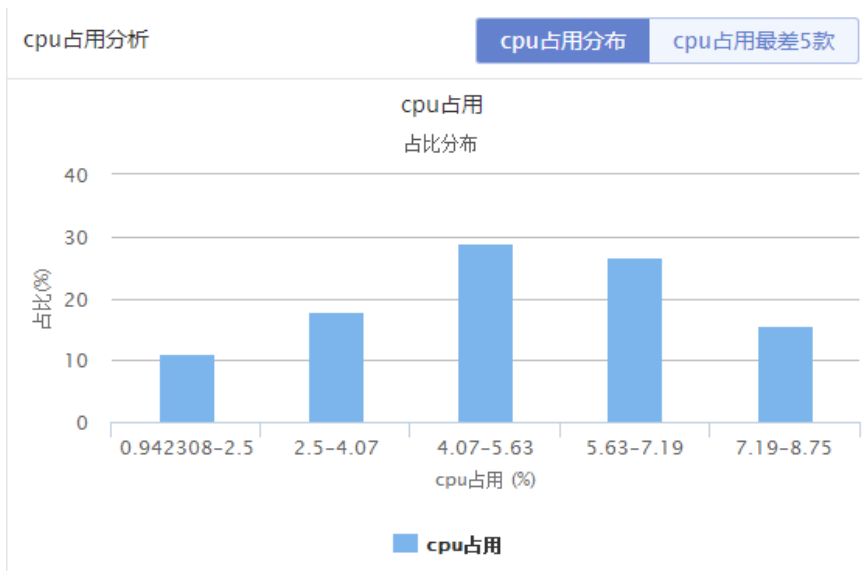


图 5.11 cpu 占用测试分析

从图 5.11 可以看出系统的 cpu 占用虽然分布层次较多，但是最高的 cpu 占用也没有超过 8.75%，说明系统的 cpu 占用较低，对设备性能要求较低。

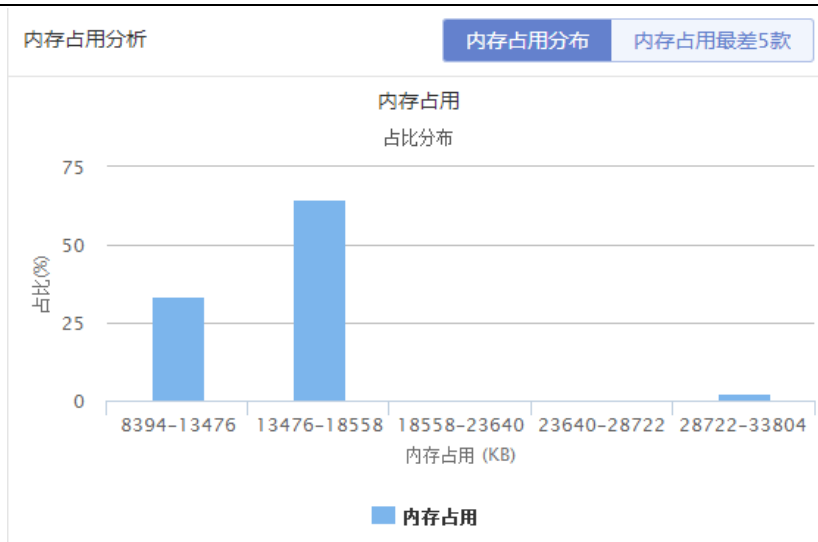


图 5.12 内存占用测试分析

从图 5.12 可以看出系统的内存占用主要分为两个区间，其中 8394KB-13476KB 区间大概有 30%，13476KB-18558KB 区间大概有 65%，说明系统对于运行内存的要求只有 20MB，目前用户使用的设备几乎都能满足这个要求。

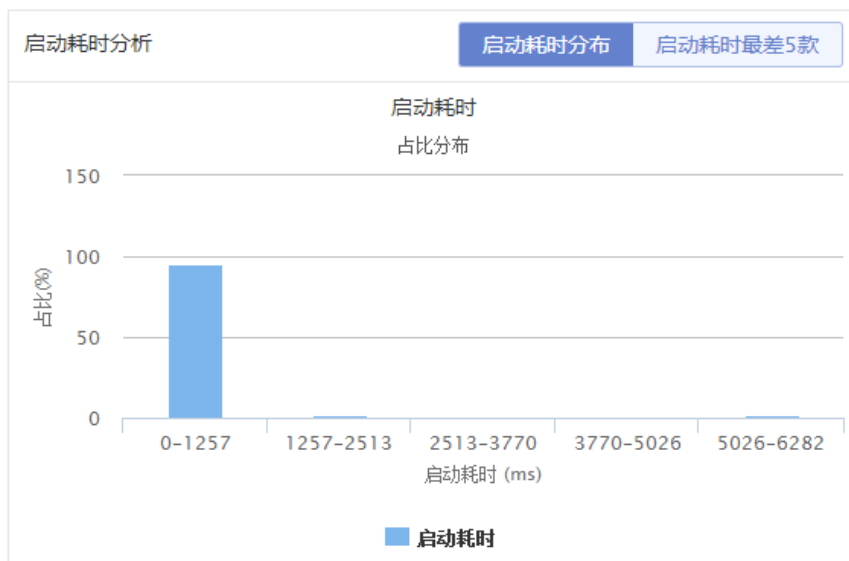


图 5.13 启动耗时测试分析

从图 5.13 可以看出系统启动时间在 0ms-1257ms 区间占比高达 96%，说明系统正常启动的耗时一般都低于 1257ms，启动时间较短能够满足用户的需求，用户体验也比较良好。

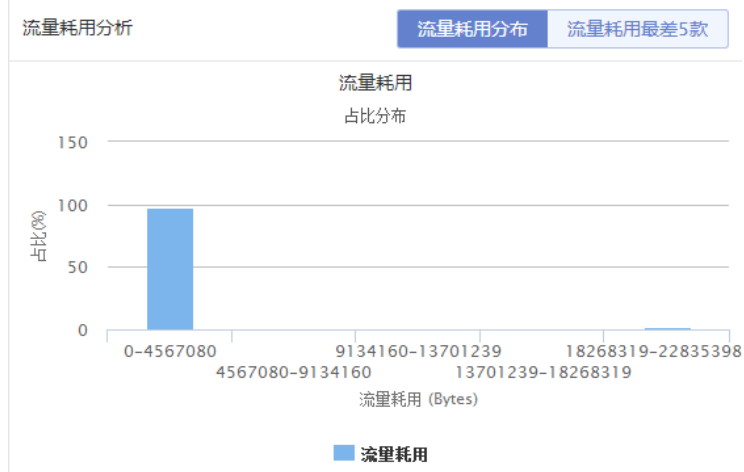


图 5.14 流量耗用测试分析

从图 5.14 可以看出系统正常使用的流量耗用分布在最低的一档有 98%，说明正常使用本系统程序耗用的流量较低，不会使用户产生高额的流量费用。

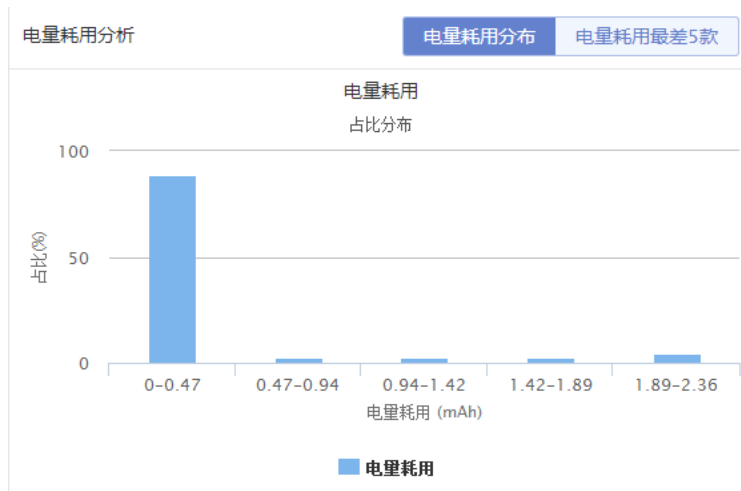


图 5.15 电量耗用测试分析

从图 5.15 可以看出系统使用的电量大多分布在 0mAh-0.47mAh 之间，说明系统耗电量不高，不会在用户使用时消耗大量电量，提高用户使用时间。

通过以上系统性能测试，说明系统总体的性能比较好，可以提供给不同用户正常使用。

5.6 本章小结

本章分别对系统的模块测试，功能测试，系统兼容性测试和系统性能测试进行说明，详细写出了测试的结果，根据测试结果对系统出现的问题进行解决，对系统功能的不足加以改进，保证系统能够正常使用。

结束语

基于目前旅游平台没有用户交流与和社交平台不能针对性面对旅游用户这两个问题，本课题完成了基于 Android 的旅游交友系统的设计和开发。论文第一章讲述了旅游交友系统的课题背景，通过对现在交友网络服务的发展与不足，说明了旅游交友系统在当今互联网时代具有的开创性意义。通过对国内外社交网络服务系统的研究和分析，目前无论是传统计算机社交网络平台还是新兴移动社交平台都是面向全部用户的，但是相信随着互联网的继续发展，人们兴趣越来越广泛，传统面向所有用户的平台将不能帮人们找到志同道合的朋友，人们将越来越倾向与使用针对性面对同一类用户的社交平台，说明了专注于面向旅游用户的旅游交友系统在未来是有很大大发展前景的。

然后对课题项目设计所需要的交友策略进行了研究，同时对项目实际开发所用户的技术手段进行了学习。其中技术手段包括 Java Servlet 服务器开发、Tomcat 服务器使用、MySQL 数据库使用、HTTP 协议使用、JSON 数据格式的序列化和解析、Android 系统框架、Android 组件开发技术，这些技术手段的学习是最终能完成整个系统开发的基础。

接下来对系统需求进行了分析，研究了用户群体的交友策略，喜欢旅游的用户在交友需求方面关系拓展需求大于关系维持需求，同时站在用户的角度上对旅游交友系统提出了功能需求和非功能需求，按照功能不同将业务需求分为六个模块，分别是登录注册、旅游动态论坛、发现用户、游记、用户信息管理、查看用户信息，并根据需求给出用例图。然后根据课题选用的 MVC 框架的分层思想，对课题系统进行了架构设计。将整个系统分为视图层、控制层和模型层三个层次，得到了系统处理流程。

在上面的基础上，对系统的每个功能模块进行了详细设计，说明了每个模块所需要的具体功能，介绍了每个模块的核心代码，然后对旅游交友系统的客户端和服务端进行同步开发，最终完成了系统的编码实现。

最后通过对系统的各种测试，改善系统可能出现的 BUG，提高系统的运行速度和用户体验。

本人在一个将近学期的时间里设计和实现了课题中的旅游交友系统的客户端、服务器和数据库。回想这几个月的自己的学习情况，平心而论仍然存在很多不足之处。因此，在以后的工作学习中更要不断强化自己的设计和编码能力，端正自己的态度，才能不断的提高自身价值。同样在这几个月的毕业设计时间里我也学习到了很多的知识，增强了自身的能力，在做毕业设计之前我从没完成过一个前后端同时独立开发的系统级软件，这次毕业设计是对我的一次挑战也是对我能力的一次升华，使我对系统前后端交互，接口定义规范等有了更加深刻的认识，同时我以前没有学习过 Android 应用的开发，通过这次毕业设计我对 Android 开发

具有一定的了解和应用，相信这对我以后的学习和工作都会有很大的帮助。

虽然当前整个系统已经开发完成，但是系统还是存在一些不足和问题，比如由于没有消息推送服务器，所以用户之间不能实时交流，因为服务器负载有限，目前仍然不能实现动态和游记中添加图片内容。如果有机会我将会在以后的版本中将以上实用的功能加入到系统中。

相信随着时代的不断前进，人们交友范围将会越来越广泛，使用旅游交友系统的用户也会不端的增加，同时随着人们生活水平的上升，喜欢旅游的用户也会越来越多，这样本系统就能不断的帮助用户找到他们期望的朋友。并且系统也会为所有用户提供越来越好的用户体验，并且在互联网浪潮中找到自己的商业模式和盈利方式，成为旅游用户社交的首选平台。

致 谢

经过为期四个月的毕业设计时间，本课题在程春玲老师的细心指导和严格要求下顺利完成，程老师是我大学四年中遇到的最为严谨的老师，从此次毕业设计的开题到最后的答辩，每次我出现问题老师总是能够指出我的问题，给我提供指导和帮助，在此我向程老师致以最真诚的谢意和最崇高的敬意。

我还要感谢大学期间教导过我的所有老师，正是他们的谆谆教诲，才让我能够掌握足够的知识完成此次毕业设计。

我还要感谢所有为本课题提供帮助的参考文献和技术博客的作者，正是他们的文献或者博客为我提供了技术支持，使我能够正确的完成课题的系统实现，同时我还要感谢为系统提供兼容性测试和性能测试的“百度开放云”平台。

参考文献

- [1] Liu Jennifer J. - N, Chlamtac Imrich. Mobile Ad Hoc Networking with a View of 4G Wireless: Imperatives and Challenges[M]. Wiley-IEEE Press, 2009:1-45.
- [2] 能向群. SNS:网络人际传播的现实化回归[J]. 河北大学学报:哲学社会科学版, 2006, 31(2):130-131.
- [3] 刘贵. 基于 Android 系统的校园社交平台的设计与实现[D].吉林: 吉林大学,2015.
- [4] Shi S. The use of Web2.0 style technologies among Chinese civil society organizations[J]. Telematics & Informatics, 2013, 30(4):346-358.
- [5] 林永兴, 李晓蕾. 基于 Android 的校园助手手机应用程序的研究与设计[J]. 硅谷, 2012, 06(3):60-60.
- [6] 夏帮贵. 基于 Android 平台的移动学习系统[J]. 西华大学学报:自然科学版, 2011, 30(5):81-84.
- [7] 刘雪雯. 中关村在线消息[EB/OL]. [2016-06-02]. <http://pad.zol.com.cn/503/5032232.html>.
- [8] 郭宏志. Android 应用开发详解[M]. 北京: 电子工业出版社, 2010:11-20.
- [9] 祝瑞, 车敏. 基于 HTTP 协议的服务器程序分析[J]. 现代电子技术, 2012, 35(4):117-119.
- [10] 孙剑, 董超, 夏继媛,等. 基于 Android 的 Socket(TCP/IP)在 LED 灯光控制系统中的应用[J]. 深圳职业技术学院学报, 2012, 11(5):19-23.
- [11] Phani Raj Tadimety. Six Degrees of Separation[J]. Buyouts, 2007, 3(12):964-965.
- [12] 王洪伟, 任豪. 社交网络的有效信息量研究:基于节点强度的分析[J]. 北京工业大学学报, 2015, 03(5):693-701.
- [13] 吴晨清, 荣震华. 用 JSP/Servlet 技术构建 Web 应用[J]. 计算机工程, 2001, 12(1):170-172.
- [14] 陈丹丹. Java 学习手册[M]. 北京: 电子工业出版社, 2011:60-72.
- [15] 赵东. 利用 Apache 和 Tomcat 构建支持 Servlet 的动态网站[J]. 金融电子化, 2000, 17(12):47-49.
- [16] 杨伟斌. tomcat 安装配置在高校 web 课堂中的应用研究[J]. 速读（下旬）, 2015, 07(9):309-309.
- [17] 杨敏. 互联网电视机顶盒客户端系统的设计与实现[D]. 北京: 北京交通大学, 2014.
- [18] Google. Android 开发者 API 指南 [EB/OL]. [2016-06-02]. <http://developer.android.com/develop/index.html>.
- [19] Onwuzurike L, De Cristofaro E. Danger is my middle name: experimenting with SSL vulnerabilities in Android apps[C]. Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks. New York: ACM, 2015:15-15.
- [20] Kang B, Kang B J, Kim J, et al. Android malware classification method: Dalvik bytecode frequency analysis[C]. Proceedings of the 2013 Research in Adaptive and Convergent Systems. New York: ACM, 2013: 349-350.
- [21] Janakiram D, Gunnam A, Suneetha N, et al. Object-oriented wrappers for the Linux kernel[J]. Software Practice & Experience, 2008, 38(13):1411-1427.
- [22] 罗雷. Android 系统应用开发实战详解[M]. 北京: 人民邮电出版社, 2014:49-62.
- [23] Rodrigues C, Afonso J, Tomé P. Mobile Application Webservice Performance Analysis: Restful Services with JSON and XML[M]. Springer Berlin Heidelberg, 2011:162-169.
- [24] 兰旭辉, 熊家军, 邓刚. 基于 MySQL 的应用程序设计[J]. 计算机工程与设计, 2004, 25(3):442-443.

- [25] Bigland-Ritchie B, Johansson R S, Lippold O C J, et al. Contractile speed and EMG changes during fatigue of sustained MVC[J]. Journal of Neurophysiology, 1983, 50(1):313-24.
- [26] 张淮中. 实现基于 HTTP 的 servlet 与 applet 的通信[J]. 计算机工程与设计, 2000, 21(6):16-19.
- [27] Pradhan S S, Huang W V. Virtual manufacturing information system using Java TM, and JDBC TM[J]. Computers & Industrial Engineering, 1998, 35(1-2):255-258.