****

本科毕业设计（论文）

**题目：基于百度云和NFC技术的会展导游系统**

**--移动终端设计与实现**

**学 院 软件学院**

**专 业 软件工程**

**学生姓名 叶剑宏**

**学生学号 200930636250**

**指导教师 刘艳霞**

**提交日期 2013 年 5 月20日**

摘 要

本文详细介绍了基于百度云和NFC技术的会展导游系统的实现。

由于展会中存在大量未经筛选或提炼的信息，它们过分模糊了用户的焦点，导致用户的视线被噪音所遮掩，无法快捷地获取感兴趣的信息；并且传统的宣传媒介已无法满足用户获取更大的信息量、更多种类信息的需求。针对这种情况，本系统采用了手机移动客户端的方式，解决了用户浏览过程中关于信息获取的问题。

会展导游系统采用传统的C/S架构，使用了百度提供的各种云能力。在客户端上，通过富媒体展示，解决用户获取商品详尽信息、获取商品优惠券的需求；基于百度LBS云能力，解决用户获取周边商铺信息的需求；基于百度PCS云存储，解决用户对信息的持久化存储的需求；接入了流行的社交渠道，增加了用户浏览过程中的趣味性。

本文首先将对系统的产生背景做了介绍，并对现有媒介能力不足的现状做了分析；对百度提供的各种云能力，也做了详尽的说明。而后将会讲解系统的需求，与详尽的系统设计。最后，会预估系统对用户和对展商的收益，说明该系统的使用，不仅能为用户带来更友好的浏览体验、也能为展商带去更多的收益。

**关键词**：百度云能力；NFC(近场通讯技术)；展会；需求；系统设计；移动互联网

Abstract

In this page, an android mobile application system called ExGuide, which is based on Baidu Application Engine and near field communication technology is introduced here. The page will not only show the idea of the system, but also the design and implementation of which.

The idea comes from scene of exhibition. When you attend an exhibition, such as GuangZhou Auto show , a motley variety of new cars or sexual Models will be displayed there You may be distracted by them, and can’t focus your attention .You may not get what you wants immediately and fully. A direct mail can’t satisfy your requirement. So this android mobile application is design to deal with the situation mentioned above.

Client / Server architecture is applied on this system ,the Server is deployed on Baidu Application Engine .With the cloud ability of Baidu Application Engine, the Client will provides rich media of exhibition and production for users, Baidu Location Based Service Cloud Service search restaurant near by venue for users, Baidu Personal Cloud Storage Service can save the rich media attachment on internet .On the other side, visitors should enjoy their trip by publishing what they see in exhibition through Sina Weibo with ExGuide Client .

This page will introduce the background of the idea of ExGuide first ,then analyze the current situation of advertising system , point out whose weakness. In addition , Baidu Application Engine will be introduced too. Finally , it will estimate its value that bring for exhibitor .

**Keyword**: Baidu Application Engine, Near Filed Communication ,Exhibiton, Requirement, System Design, Mobile Internet

目录

[**摘 要 I**](#_Toc356680160)

[**Abstract II**](#_Toc356680161)

[第一章 绪论 3](#_Toc356680163)

[1.1 会展经济 3](#_Toc356680164)

[1.1.1 需求背景 3](#_Toc356680165)

[1.1.2 面临的问题 3](#_Toc356680166)

[1.2 移动互联网 4](#_Toc356680167)

[1.2.1 移动互联网概述 4](#_Toc356680168)

[1.2.2 将会展驶入移动互联网快车道 5](#_Toc356680169)

[1.3 项目意义 5](#_Toc356680170)

[1.4 系统概要介绍 6](#_Toc356680171)

[第二章 基础技术介绍 7](#_Toc356680172)

[2.1 百度云开发环境 7](#_Toc356680173)

[2.1.1 百度应用引擎（BAE） 7](#_Toc356680174)

[2.1.2 百度LBS云 7](#_Toc356680175)

[2.1.3 百度个人云存储（PCS） 9](#_Toc356680176)

[2.2 安卓(Android)系统简介 9](#_Toc356680177)

[2.2.1 应用程序 10](#_Toc356680178)

[2.2.2 应用程序框架 10](#_Toc356680179)

[2.2.3 系统运行库 10](#_Toc356680180)

[2.3 近场通讯技术（NFC） 11](#_Toc356680181)

[2.4 会展导游系统架构概述和运行原理 12](#_Toc356680182)

[2.5 小结 12](#_Toc356680183)

[第三章 需求分析 14](#_Toc356680184)

[3.1 用户需求分析 14](#_Toc356680185)

[3.1.1 用例1 14](#_Toc356680186)

[3.1.2 用例2 15](#_Toc356680187)

[3.1.3 用例3 15](#_Toc356680188)

[3.1.4 用例4 16](#_Toc356680189)

[3.1.5 用例5 16](#_Toc356680190)

[3.1.6 用例6 17](#_Toc356680191)

[3.2 系统需求分析 17](#_Toc356680192)

[3.2.1 系统界面运行流畅 17](#_Toc356680193)

[3.2.2 第三方授权 17](#_Toc356680194)

[3.3 小结 17](#_Toc356680195)

[第四章 系统概要设计 18](#_Toc356680196)

[4.1 系统前端设计 18](#_Toc356680197)

[4.1.1 交互与原型设计 18](#_Toc356680198)

[4.1.2 瀑布流布局 22](#_Toc356680199)

[4.2 系统后端设计 22](#_Toc356680200)

[4.2.1 系统模块设计 22](#_Toc356680201)

[4.3 系统视图层设计 24](#_Toc356680202)

[4.3.1 展会列表界面刷新流程 24](#_Toc356680203)

[4.3.2 展会概要界面刷新流程 25](#_Toc356680204)

[4.3.3 展品详情界面刷新流程 26](#_Toc356680205)

[4.4 小结 26](#_Toc356680206)

[第五章 系统详细设计与实现 27](#_Toc356680207)

[5.1 视图层详细设计与实现 27](#_Toc356680208)

[5.1.1 展会列表首页的详细设计与实现 27](#_Toc356680209)

[5.1.2 展会详情页的详细设计与实现 28](#_Toc356680210)

[5.1.3 展品详情页的详细设计与实现 29](#_Toc356680211)

[5.2 逻辑层详细设计与实现 30](#_Toc356680212)

[5.2.1 联系人模块详细设计与实现 30](#_Toc356680213)

[5.3 关键问题及解决方案 32](#_Toc356680214)

[5.3.1 瀑布流图片的设计与实现 32](#_Toc356680215)

[5.3.2 资源加载任务的设计与实现 34](#_Toc356680216)

[5.3.3 异步加载线程的设计与实现 34](#_Toc356680217)

[5.4 小结 36](#_Toc356680218)

[第六章 总结与展望 37](#_Toc356680219)

[6.1 总结 37](#_Toc356680220)

[6.2 展望 37](#_Toc356680221)

[**参考文献 38**](#_Toc356680222)

[**致谢 39**](#_Toc356680223)

# 绪论

## 会展经济

* + 1. 需求背景

会展是一个前景广阔的朝阳产业和绿色产业,正逐步成长为第三产业中的一颗耀眼明星。

近年来，中国内地已成为全球发展最快的展览市场。数据显示，从2006年至2011年，中国展览场地面积增长了48%，而美国和德国则分别增长5%和2%。此外，中国的展览场地数量和面积在整个亚洲也分别占了57%和69%，居亚洲第一位。目前，我国的会展业取得了良好的经济效益，并已成为第三产业中最具发展潜力的行业之一，对经济的拉动作用非常明显。

有一组数据可以佐证，据商务部统计数据显示，2011年全国共举办展览6830场，展出面积8120万平方米，提供社会就业岗位1980万人次，直接产值3016亿元，比上年同期增长17.7%，拉动效应2.7 万亿元，占全国GDP的0.64%，占全国第三产业的13%。

从地方的上经济会展经济数据图表，也可看到会展行业的潜力之大。

表 1‑1 2012年各地举办展会与经济效益对比

|  |  |  |
| --- | --- | --- |
| 地方 | 场次 | 备注 |
| 沈阳 | 举办展会295场 | 交易总额1901亿元 |
| 济南 | 举办展会155场 | 相关行业增收165亿元，交易额1221亿元 |
| 成都 | 举办展会268场 | 促成各类商品成交3500亿 |
| 郑州 | 举办展会158场 | 实现经济效益150亿元 |
| 南京 | 举办展会2182场（包含会议） | 相关产业拉动420亿 |
| 重庆 | 举办展会521场 | 直接收入54.3亿元，拉动消费435亿元 |

* + 1. 面临的问题

面对如此蓬勃发展的会展行业，传统营销渠道已经无法满足展商的宣传需求，它未能与高速发展的容量相配套，滞后的营销技术导致展商耗费了大量的成本，却收获不到相应的效果；另一方面，传统的营销方式，也为众多的用户带去了困扰，比如说，满天的传单和花哨的宣传让消费者苦不堪言、静态的展示无法让消费者体会到实体运动时的动感、传统方式承载的信息不足等。

传统的营销渠道有：报纸杂志、广播电视、电话、网络等。它们都有不同程度的缺点，事实上营销应该能做到消息到达及时、快速，刺激需求和市场化等几点要求。从数据上来看，到达率和转化率是一个重要的考量指标；而留存等数据，则更多地反应了产品本身的效果。

下表是对几个营销渠道的一个分析，说明传统渠道的到达率、转化率堪忧，已经无法承载高速发展的会展行业。

表 1‑2 几种传统渠道的对比

|  |  |
| --- | --- |
| 营销渠道 | 缺点 |
| 报纸杂志宣传 | 信息更新周期长，到达速度慢，受众少，成本高，转化率低 |
| 广播电视宣传 | 成本巨大，信息一旦形成，很难修改 |
| 电话营销 | 目标不准确，受众少，形式有限，不能刺激需求 |
| 网络营销 | 到达率低，操作难度大 |

所以市场需要一款产品，在解决参展商宣传需求的同时，也能为用户带去便捷、有趣的浏览体验。

## 移动互联网

* + 1. 移动互联网概述

移动互联网是指互联网的技术、平台、商业模式和应用与移动通信技术结合并实践的活动的总称。，随着宽带无线接入技术和移动终端技术的飞速发展，人们迫切希望能够随时随地乃至在移动过程中都能方便地从互联网获取信息和服务，移动互联网应运而生并迅猛发展。

移动互联网的核心是互联网，因此一般认为移动互联网是桌面互联网的补充和延伸，应用和内容仍是移动互联网的根本。虽然移动互联网与桌面互联网共享着互联网的核心理念和价值观，但移动互联网有实时性、隐私性、便携性、准确性、可定位的特点，日益丰富智能的移动装置是移动互联网的重要特征之一。

从用户需求来看，移动互联网以移动场景为主，使用上有时间碎片化、场景移动化等特点。

* + 1. 将会展驶入移动互联网快车道

移动互联网，是时下最热门的话题。数据上也反应出，网民从传统互联网向移动互联网迁徙的趋势非常明显。从2012年11月的数据反应出，移动互联网的流量已经占到了互联网总流量的13%；而2012年Q4的数据显示，用户的移动互联网在线时长，已经超过了传统的PC互联网。说明用户的习惯正发生改变。

图 1‑1 移动互联网流量已占互联网流量15%

而移动互联网的基础之一，就是移动设备，比如说智能手机。现在智能手机的发展，已经让手机不仅仅是个通信工具那么简单，随着无线网络接入技术的不断升级，智能手机与移动互联网的结合，使得移动生活越来越丰富多彩。从内容上，智能手机是一个更好的载体，能承载更多的富媒体内容。

事实上，在移动化的大潮下，会展行业也走到了十字路口。传统营销方式表现出的内容短板，在移动化的大潮下，都能在终端解决，智能终端承载的不仅仅是富媒体内容，还包括社交等娱乐因素。

所以应借力于移动互联网，将会展的营销需求也驶入移动化的快车道。

## 项目意义

本系统的意义，在于解决会展场景下参展商的宣传需求，为参展商节省了资源，使其更专注于展会本身的产品。移动端的应用，使用长连接，解决了传统方式到达率低的问题；应用展示的富媒体内容，帮助用户决策，也将大大提高转化率。另外，系统接入了社交元素，增加了用户在会展中的互动，提升了趣味性。

## 系统概要介绍

系统采用了传统的C/S架构，将服务端置于百度云应用引擎上（BAE）。使用了百度的云的托管服务、个人云存储服务（PCS）、地理位置信息服务（LBS）等等，基础服务依赖于百度云，解决了系统的基础技术支持，让系统更专注于逻辑上的功能。

客户端承载着用户接口的功能，向用户传递展会及相关展品的富媒体信息，帮助用户决策；同时提供了基于展会场景的小功能，满足用户的城需求。

# 基础技术介绍

## 百度云开发环境

2012年百度世界大会上，百度移动云事业部总经理李明远，向众多开发者公布百度的云开发环境，致力解决开发者基础服务问题，为开发者节省开支。

其中包括帮助开发者节省服务器成本的个人云存储（PCS）、帮助开发者整合地理位置服务的LBS云、解决终端适配问题的百度移动云测试中心（MTC）、安全稳定的运行环境百度应用引擎（BAE），还包括多屏幕Screen X技术、百度移动云应用生成服务（Site App）和百度移动浏览内核。

* + 1. 百度应用引擎（BAE）

百度应用引擎（BAE）是百度推出的网络应用开发平台。基于BAE架构，使开发者不需要维护任何服务器，只需要简单的上传应用程序，就可以为用户提供服务。BAE有能力将原本单机的LAMP架构，变成分布式架构。开发者可以基于BAE平台进行PHP、Java、Python应用的开发、编译、发布、调试。

同时BAE平台也提供了大量的云服务给开发者，包括fetch URL、task queue、SQL、memcache，后续会提供更多服务。在性能方面，如果开发者希望增加服务能力，可以通过申请更多执行单元的方式进行灵活的调整。

BAE服务可以让开发者在开发网络应用程序的过程中摆脱繁琐的环境、服务问题，把精力专注于业务逻辑。

本系统将服务端置于百度应用引擎上，借助于引擎的弹性计算和带宽，解决系统的网络连接问题。

* + 1. 百度LBS云

百度LBS云是服务百度地图与BAE针对LBS开发者一起推出的平台级服务。专门解决LBS开发者针对存储和高并发检索的难题。

百度LBS云服务提供海量数据的存储、检索、展示三大功能，可以解决：

* 基于位置数据的空间数据库管理和维护问题，尤其是移动开发者对于位置数据爆发式增长的海量空间数据存储压力；
* 基于位置的空间检索问题，不再依赖数据库实现周边检索这一类基于空间的位置数据计算，而是开放检索引擎彻底解决LBS应用的高并发检索瓶颈问题；
* 海量位置数据在地图上展现的问题，不再受制于Web端JS的标注能力的影响，将大数据在服务端渲染为矢量数据图层，开放图层渲染引擎，从“云”直接下发图层和矢量数据到“端”。

百度LBS云服务的到来，开创了新式的应用开发模式。传统模式，开发者需要解决基础服务和业务逻辑的问题，而在新的模式下，开发者只需要关注自身的业务逻辑，大量的地理位置存储与计算，都将通过调用百度LBS云服务来实现。

应用

数据逻辑

空间运算

业务数据

地理数据

网络

百度LBS服务地图、搜索、公交、驾车、地理编码

图 2 ‑1 传统LBS应用的开发模式

弊端：

开发者需要自己实现位置数据存储（空间数据库）

开发者需要自己实现空间计算（周边检索等）

开发者需要自己处理检索时效性和并发瓶颈

百度LBS服务地图、搜索、公交、驾车、地理编码

应用

数据逻辑

业务数据

空间运算

位置数据

网络

图 2‑2 使用LBS云的应用模式1

百度提供位置数据存储

百度提供空间计算能力

开发者关注应用本身的业务数据和业务逻辑

图 2- 3 使用LBS云的应用模式2

百度LBS服务地图、搜索、公交、驾车、地理编码

应用

业务逻辑

空间运算

地理数据

网络

业务数据

百度提供位置数据存储

百度提供空间计算能力

百度提供自定义扩展字段存储，让业务数据参与检索规则配置（筛选排序）

进一步方便开发者聚焦LBS应用本身的业务逻辑

本系统将使用百度LBS云能力，为系统提供定位功能，以及相关的POI(point of information)信息，解决用户查找展馆附近的餐厅需求。

* + 1. 百度个人云存储（PCS）

百度个人云存储PCS (Personal Cloud Storage)是百度2012年推出的针对个人数据的云存储服务。开发者可以利用PCS的开放接口存储用户个人数据，进而解决数据碎片化、终端多样性的问题。通过使用PCS服务，开发者无需考虑设计复杂的海量存储系统，而更专注于解决系统的业务逻辑问题。

本系统将使用百度个人云存储，解决用户需要持久化保存电子资料的需求。

## 安卓(Android)系统简介

Android是一种基于Linux的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由Google公司和开放手机联盟领导及开发。Android操作系统最初由Andy Rubin开发，主要支持手机。2005年8月由Google收购注资。2007年11月，Google与84家硬件制造商、软件开发商及电信劳动商组建开放手机聪明共同研发改良Android系统。随后Google以Apache开源许可证的授权方式，发布了Android的源。第一部Android智能手机发布于2008年10月。

Android的系统架构和其操作系统一样，采用了分层的架构。从架构图看，Android分为四个层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和Linux内核层。

* + 1. 应用程序

Android会同一系列核心应用程序包一起发布，这些应用程序包包括但不限于客户端、SMS短消息程序、日历、地图、浏览器、联系人管理程序等。因为Android是开源系统，不同厂商会基于自己的设计，对Android进行二次开发，这时核心的应用程序也将因不同的厂商而不同。而由Google发布的系统，称为原生Android系统。用户还可以在应用商店，下载Android应用。所有的应用开发都是使用JAVA语言编写的。

* + 1. 应用程序框架

因为Android的开源特性，程序开发人员均可以完全访问核心应用程序所使用的API框架。该应用程序的架构设计简化了组件的重用，使得用户可以方便的替换程序组件。

每个应用程序都可以使用一系列的服务和系统来实现自己的逻辑业务,这些服务和系统包括;

丰富而又可扩展的视图（Views)，可以用来构建应用程序，是用户接口；

内容提供器（Content Providers)使得应用程序可以访问另一个应用程序的数据（如联系人数据库)， 或者共享它们自己的数据

资源管理器（Resource Manager)提供非代码资源的访问，如本地字符串，图形，和布局文件（Layout files )。

通知管理器 （Notification Manager) 使得应用程序可以在状态栏中显示自定义的提示信息。

活动管理器（ Activity Manager) 用来管理应用程序生命周期并提供常用的导航回退功能。

* + 1. 系统运行库

Android 包含一些C/C++库，这些库能被Android系统中不同的组件使用。它们通过 Android 应用程序框架为开发者提供服务。以下是一些核心库：

* 系统 C库，它是 一个从BSD继承来的标准C系统函数库 ， 它是专门为基于Embedded linux的设备定制的。
* 媒体库 - 基于PacketVideo OpenCORE;该库支持多种常用的音频、视频格式回放和录制，同时支持静态图像文件。编码格式包括MPEG4, H.264, MP3, AAC, AMR, JPG, PNG 。
* Surface Manager - 对显示子系统的管理，并且为多个应用程序提 供了2D和3D图层的无缝融合。
* LibWebCore - 一个最新的web浏览器引擎用，支持Android浏览器和一个可嵌入的web视图。

## 近场通讯技术（NFC）

NFC是Near Field Communication缩写，即近距离无线通讯技术。由飞利浦公司和索尼公司共同开发的NFC是一种非接触式识别和互联技术，可以在移动设备、消费类电子产品、PC 和智能控件工具间进行近距离无线通信。NFC 提供了一种简单、触控式的解决方案，可以让消费者简单直观地交换信息、访问内容与服务。

它使得消费者更方便和简单地处理事务、交换数字信息和连接电子设备。NFC是一种无线电传输标准，使得无线信号可以在极短距离中传播，这个距离可能是10毫米。NFC是一种基于RFID技术的标准，它包括了通讯协议和数据传输格式。NFC设备包括RFID标签无线智能卡片。NFC技术运行在13.56MHZ的频率下，但是数据传输速率较慢，最快只能达到0.42Mb 每秒。我们需要搭载android 2.3以上版本的手机才支持NFC芯片。

近场通讯技术应用前景广泛，现在以及将会，可能在如下领域广泛应用：

* 门禁系统 ：用户通过带有NFC芯片的手机，可以代替传统的智能卡片识别系统。
* 消费者电子：用户为不同的无线智能卡片写上不同的标记，当带有NFC芯片的手机接触这些卡片时，会自动改变手机的状态。
* 医疗：用户带着感应器，则可以通过NFC传输感应器的信号
* 信息收集和交换：用户可以用NFC手机相换交换用户信息
* 支付：用户能使用NFC手机支付，不需要过多烦琐的操作，代替了现有的刷卡支 付等
* 运输：物流行业运用NFC技术清点货物，实现智能仓储系统

## 会展导游系统架构概述和运行原理

系统将服务端迁移至百度云应用引擎（BAE），其弹性的运行环境和安全架构，保证了系统运行的稳定性与安全性；使用百度云提供的各种云能力：使用百度云LBS云为客户端提供定位服务和周边商户搜索服务、使用百度个人云存储为用户保存持久化数据保存提供了可能，客户端可以通过无线局域网或手机的数据网络与云端服务器通讯、交换数据。可以说百度云能力帮助系统解决了很多基础功能问题，而系统本身专注于解决业务逻辑问题。

用户手持安装了本系统的Android终端（集成NFC芯片），打开本系统后，系统将会向服务器请求位置信息，在获取位置信息后，向用户展示所在地理位置的展会列表，用户可以查看到关于展会的富媒体信息。当用户将Android终端（集成NFC芯片）靠近芯片时，系统将为识别芯片携带的识别码，并向服务器请求识别码对应的展示的富媒体信息。

图 2 ‑4 会展导游系统使用百度云服务架构

百度BAE

百度个人云存储



百度云LBS云

数据

数据

数据

## 小结

百度云服务是百度为众多移动互联网开发者准备的基础服务，它不仅包含了应用运行的服务器环境，还将百度近年来积累的云存储、LBS（地理位置信息服务）技术等技术以云服务的形式提供给开发者，目的是帮助开发者节省了基础服务的成本，使开发者专注于业务逻辑的设计与实现，是百度构建移动互联网生态圈的重要组成一步。

而本系统因开发需要，也接入了这些云服务。服务器端将部署在百度BAE，而客户端也将使用百度提供的PCS、LBS等云服务，这些服务的应用在相关的小节已有说明。可以说，百度云服务是本系统的根基。

# 需求分析

## 用户需求分析



图 3 ‑1 会展导游系统用户需求

本节主要介绍用户需求。从用户观展前、观展中、观展后几个场景，梳理分析用户的需求点。

* + 1. 用例1

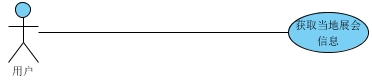


图 3 ‑2 作为用户，我希望获得当地展会的信息

用户希望获得所在城市的展会信息。

当用户有参展的兴趣时，希望通过定位功能查找当地展馆举办的展会信息，并希望系统提供展会的相关内容作为决策的依据。比如说车展，用户可以看到最新或者是热门的车型，又或者是车模的照片，吸引用户决策去参加这个展会。

* + 1. 用例2

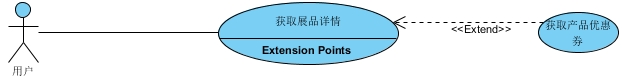


图 3 ‑3 作为用户，我希望能获取展品的详情

用户希望在参展的过程中，能获得更多的展品的介绍，必要时能获取展品优惠券。

目前参展商的宣传方式仍以传单和现场讲解为主，大屏滚动播出宣传视频也作为一种宣传手段使用。可是这些方式存在天生的缺陷，传单所承载的内容不够详尽，而现场的讲解或宣传视频，让晚到的用户无法知晓到全部内容。

所以，用户希望可以获得更多的展品介绍，无论是文字、图片、视频或者是其他形式的附件。当用户看到某些展品的介绍后，如果产生了购买的兴趣，希望能够获得展品的优惠券，以便用户在购买时享受到优惠。

* + 1. 用例3

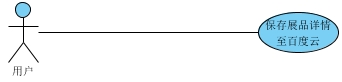


图 3‑‑4 作为用户，我希望能将展品的电子介绍保存至百度云

用户希望将感兴趣的展品的电子介绍保存至百度云。

展品的详情介绍，包含了文字介绍，但都是概要式的说明，详情中还包含了更详尽内容的富文本附件，用户对展品感兴趣时，可以将附件保存至百度云，长久的保存起来，不易丢失。

这种保存附件的形式，解放了用户。用户不需要再担心在展会看中某一心仪的展品后，回家后就忘记其名称；或者是不再需要在逛展会的时候，拿着五花八门的传单介绍。

* + 1. 用例4



图 3‑ 5 作为用户，我希望能在展馆周边找到餐厅

用户在观展其间，希望能在饭点在展馆周边找到吃饭的地方。

展馆建筑面积比较大，一般坐落在城市的近郊，四周较为荒凉，而且因为展馆更注重自身的建设，所以周边的餐饮等配套设施，有所不足，比如说餐厅的地点距展台较远，或者较偏僻，位于地下层或者是在附近某栋建筑较多。而对于异地游客，陌生的环境对于他们而言，更难找到目的地。

所以，用户希望能够在饭点，在展馆周边找到餐厅。餐厅应能在几个维度作分类查找，比如说距离、品味、价格等。

* + 1. 用例5



图 3 ‑6 作为用户，我希望交换联系人

用户在观展其间，会认识不同的人。比如说销售代表会认识客户，用户在对某件展品产生购买意向时，也会认识某些销售代表；摄影师对在展会其间认识了一些影友；用户无意见也可能搭讪某位车模。当他们需要相互留联系方式时，希望能够快捷方便的交换联系方式信息，因为他们可能在赶车时才想起留联系方式，要求交换的方便更快捷。

* + 1. 用例6

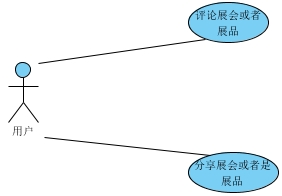


图 3 ‑7 作为用户，我希望能与朋友分享参展时的乐趣

用户在看到某件有趣的产品或者是遇见某件趣事时，希望能通过微博与朋友们分享。

## 系统需求分析

本节主要介绍系统需求，涉及第三方服务授权和系统性能等需求。

* + 1. 系统界面运行流畅

系统向用户展示展会、展品的概况，需加载大量的图片，系统使用资源较多，应使用缓存、多线程技术绘制用户界面，保证界面的流畅性。

* + 1. 第三方授权

系统引入的第三方服务，均需获得用户授权能可使用，引入的第三方服务包括百度云个人云存储（PCS）、新浪微博开放服务。百度云LBS云服务，需要应用申请权限用户无需授权。

## 小结

系统的需求分析，充分基于用户在参展的前、中、后的活动进行，是一次合理的分析。它包括参展前的决策，系统需提供富媒体信息供用户参考，帮助决策是否参展；参展中，使用近场识别技术方便用户识别展品、使用百度个人云存储服务持久化保存展品介绍。此外，还挖掘出用户在参展中有查找附近餐饮的需求和交换联系人信息的需求。

# 系统概要设计

本章主要介绍会展导游系统的详细设计，包括了前端的交互与界面原型，后端的模块设计、数据库设计、多线程时序、第三方服务的调用和与服务器的交换过程。

## 系统前端设计

* + 1. 交互与原型设计

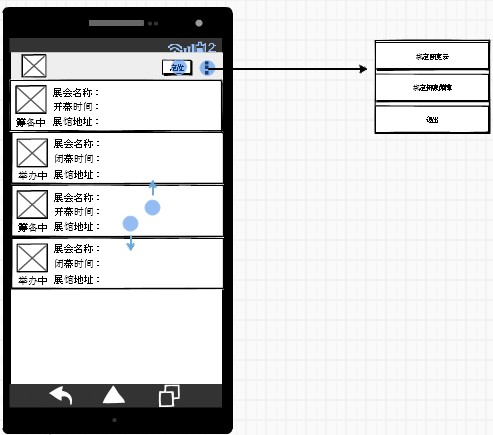


图 4 ‑1 一级页面交互设计图

该组交互包含了系统运行的首级页面、首级页面隐藏菜单的交互。

在首页中，会向用户展示系统定位的城市，并快速展示所在城市的展会举办情况。采用列表形式展示用户所在地的展会举办情况，而对于第一屏展示的展会概况信息，相信用户关注的焦点在于：展会是否已开幕、展会的名称是什么、展会的地点是哪里等。而列表可上下滑动，方便用户浏览更多信息。

交互上，采用android4.0以上版本标准交互组件。Action Bar隐藏的菜单将授权放在一级菜单中，原因是使用户更快速的进行授权。

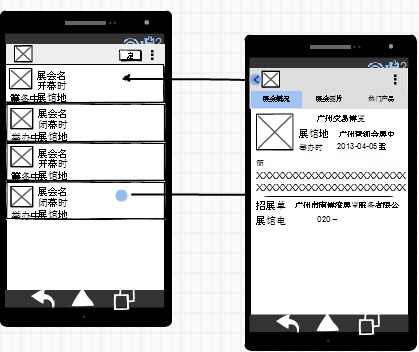


图 4 ‑2 二级页面交互设计图1

该组交互包括首级页面和二级展会详情页面的跳转交互。

当用户在单击任一展会项，系统将会跳转至二级展会详情页面。详情页面采用可滑动视图，将展会概况、展会图片和热门产品放入滑动视图。

该组交互采用android 4.0标准交互方案，在二级页面，单击返回键，即可返回首页，系统将销毁二级页面。

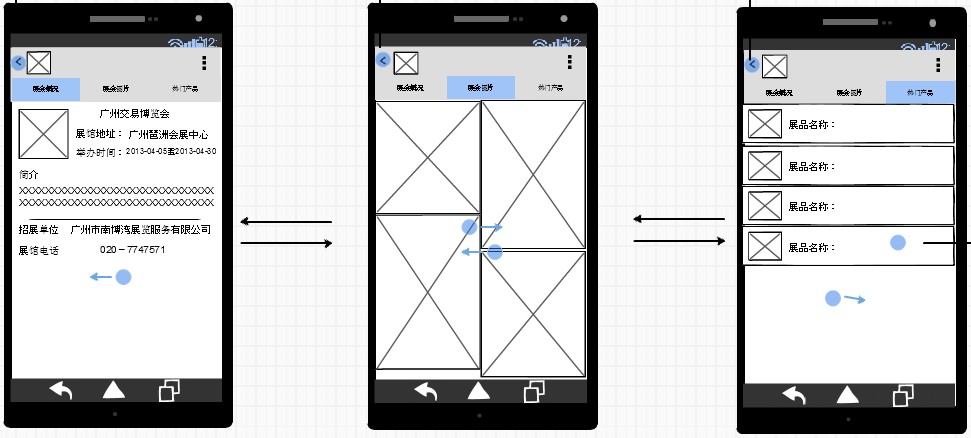


图 4 ‑3 二级页面交互设计图2

该组交互包含展会详情二级页面和该页面内的操作交互。

二级页面下采用滑动视图，将展会概况、展会图片、热门产品放入滑动视图中。展会概况为默认视图，向左滑动至展会图片，展会图片向左滑动至热门产品、向右滑动至展会概况，热门产品向右滑动至展会图片。

展会概况简要介绍了与展会相关的信息，包含logo、展会名称、展馆地址、举办时间、简介，扩展信息包含了招展单位和展馆电话。

该组交互采用android4.0标准交互组件，左上角返回键可返回一级视图，左右滑动是系统定义的标准手势。

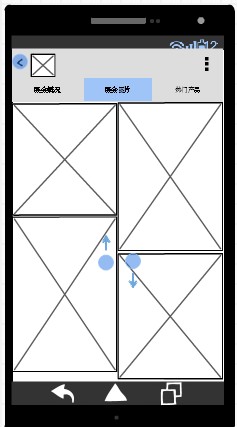


图 4‑4 瀑布流视图交互设计图

该组交互包含二级展会详情页滑动视图下，展会图片的操作交互。

图片的展示方式采用时下流行的瀑布流展示，特点是交互友好，单张图片显示得更大，便于用户仔细查看，缺点是一屏显示的图片数量有限。向下拉动至底后，系统将会自动加载更多图片。关于瀑布流交互，将在稍后小节详细介绍。

该组交互采用android4.0标准交互组件，左上角返回键可返回一级视图，上下滑动是系统定义的标准手势。

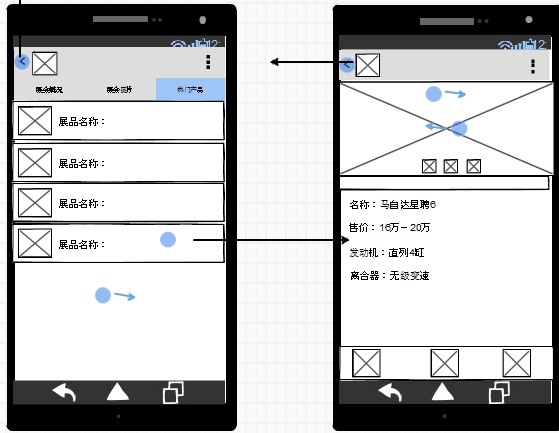


图 4‑5 三级页面交互设计图

该组交互包含展会详情二级页面与展品详情三级页面的跳转交互。

在二级页面的滑动视图中，在热门产品视图下，通过单击任一展品项，可跳转至展品详情三级页面。三级页面上方采用滑动视图展示展品的海报图片，下方区域包含展品的文字介绍。页面下端平均摆放三个按钮，左起分别为“赞”（对展品评分时使用）、分享给好友、更多。在更多的下级列表中，提供观看宣传视频的入口。

展品详情的三级页面，还可以通过系统自带的近场通讯芯片，识别展品后，激活该页面。

该组交互采用android4.0标准交互组件，左上角返回键可返回一级视图，上下滑动是系统定义的标准手势。

* + 1. 瀑布流布局

瀑布流，又称瀑布流式布局。是目前比较流行的一种网站页面布局，视觉表现为参差不齐的多栏布局，随着页面滚动条向下滚动，这种布局还会不断加载数据块并附加至当前尾部。最早采用此布局的网站是Pinterest，逐渐在国内流行开来。国内大多数清新站基本为这类风格。

瀑布流有非常多的优点：

布局简单，没有特别的难点。不必明确知道数据块高度，当数据块中有图片时，就不需要指定图片高度。

但缺点也非常明显：

1、列数固定，扩展不易，当浏览器窗口大小变化时，只能固定的x列，如果要添加一列，很难调整数据块的排列。

2、滚动加载更多数据时，还要指定插入到第几列中，不够方便。

3、数据块排列是从上到下排列到一定高度后，再把剩余元素依次添加到下一列，这个本质上就不一样了。

## 系统后端设计

* + 1. 系统模块设计

图 4 ‑6系统架构设计

视图层

实体层

activity

adapter

百度云BAE

系统后台

百度LBS云、PCS

view

业务数据

控制层

多线程

数据库

服务

本系统采用MVC架构，将实体、控制与视图相剥离。这样子做的好处是，系统架构清晰，模块划分细化，各模块耦合度大大降低，便于系统的开发与维护。

下面将概述MVC架构在系统中的使用。

视图层，是用户接口，是用户与系统交互的重要层。它包括三个模块，活动（activity）组件、视图(view)组件和适配器(adapter)等。

* 活动（activity）组件模块：是视图的容器，是捕捉用户与系统交互的模块。
* 视图（view）组件模块：是数据展示的容器，负责将后台数据展示给用户，是活动组件的内容。
* 适配器（adapter）模块：是数据的容器，负责将后台数据整理，活动组件所展示的数据，由适配器提供。

控件层，是逻辑模块，负责与用户的交互逻辑和系统与服务器的交互。它包括三个模块的内容，包括多线程管理、数据库管理、接口等。

* 多线程模块：系统实现需要大量的资源消耗，为了保证用户界面流畅和系统执行效率，资源的加载与渲染采用了异步加载的方式。该模块负责与后台交互，获取相关数据并封装。
* 数据库模块：该模块的设计，是为了持久化保存系统数据，包括保存了第三方授权的acess\_token信息、联系人信息等。
* 服务模块：系统使用了第三方提供的服务，包括百度云服务和新浪服务开放平台等，该模块是在第三方提供的服务基础上再进行的逻辑封装。

实体层，是数据的容器模块。系统运行时维护了大量的业务数据，为更有效的管理业务数据而设计的层。

* 业务数据：它包括数据库存储的第三方授权的access\_token、联系人、实体类等。

## 系统视图层设计

* + 1. 展会列表界面刷新流程

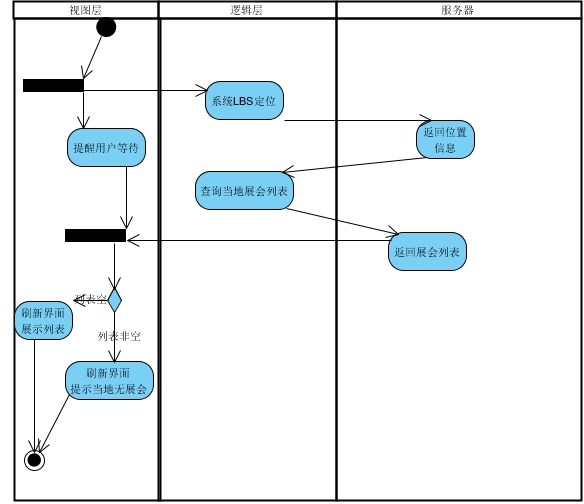


图 4‑7 展会列表界面刷新流程

系统运行初，将首先定位用户的位置，这过程将使用百度LBS云服务。云服务器将返回位置信息，之后系统以位置为参数，向后台查询当地展会列表。如果列表这为空，则提示用户当地无展会，如果有则向用户显示展会列表。

* + 1. 展会概要界面刷新流程

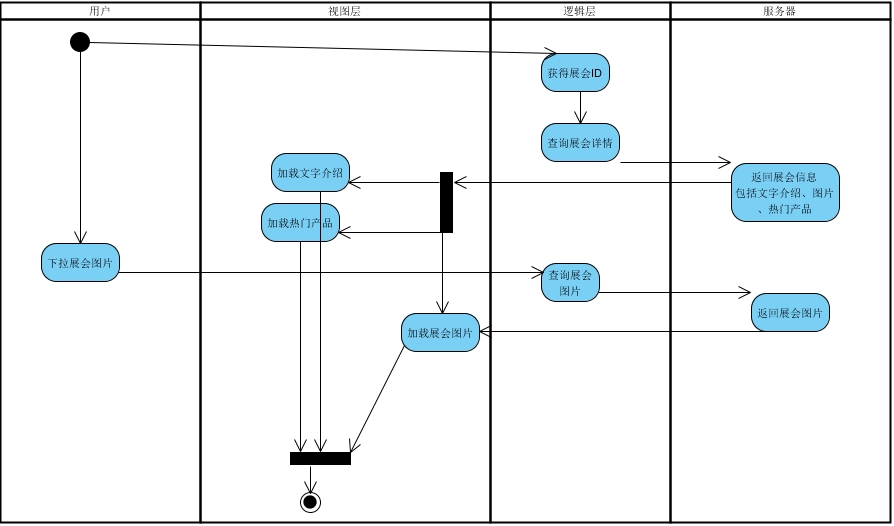


图 4 ‑8 展会概要界面刷新流程

当系统从一级页面向二级跳转时，系统将首先获得用户所选择的展会ID，之后与服务器交互，以ID为参数查询展会的概要。服务器返回的信息由逻辑层封装之后，再返回视图层将数据展示。

当用户不断下拉瀑布流时，展会图片也会不断加载，这个过程需要系统再次向服务器请求图片数据。

* + 1. 展品详情界面刷新流程

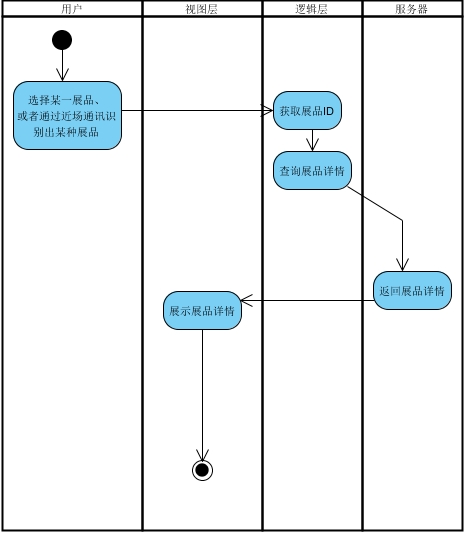


图 4 ‑9 展品详情界面刷新流程

系统首先识别展品的ID ，再以ID为参数向服务器查询展品详情信息，并向用户展示。

## 小结

系统的前端设计采用基于用户体验的交互设计原则，交互组件使用Android4.0以上版本标准组件。后端采用MVC架构，将逻辑与视图相剥离，降低了系统的耦合度。系统模块细分，除掉系统的粒度，便于系统的开发。

# 系统详细设计与实现

本章节主要说明各系统模块的详细设计，包括相关类的依赖关系及消息传递的时序等。

## 5.1 视图层详细设计与实现

本节将会分小节说明视图层中各类的职责，并简述其设计，最后将说明该类与逻辑层交互的时序逻辑。

* + 1. 展会列表首页的详细设计与实现

展会列表首页（一级页面），指用户打开系统后看到的第一屏。

其主要职责示向用户展示所在城市的展会列表。其展示的内容与交互参照第四章相关小节。

类名：HomeActivity

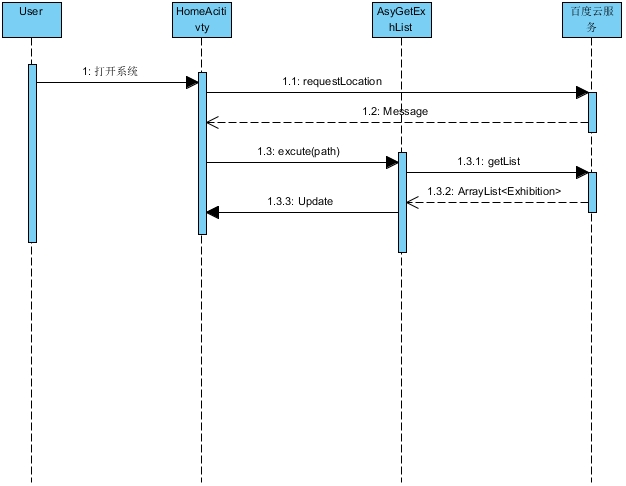


图 5 ‑1 展会列表首页（一级页面） 刷新时序

展会列表首页(一级页面)刷新交互时序如上图所示：

1、 用户打开系统

2、 系统调用RequestLocation方法，同步向百度LBS云服务器请求用户所在地理位置信息

3、 百度云服务器返回信息并封装成Message返回给系统

4、 通过服务器返回值，系统启动多线程，向服务器同步请求所在地展会列表

5、 服务器返回展会列表信息，逻辑层将信息封成ArrayList<Exhibition>实体类列表，再调用Update，更新首页的UI（user interface）

* + 1. 展会详情页的详细设计与实现

展会详情页指的是用户在展会列表页，选中某一展会后，查看到的对应展会的详情页。

其主要职能是向用户展示展会详情信息，包括文字介绍、图片介绍、热门产品介绍等。相应的交互与视觉设计，参照第四章相关小节。

类名：ExhActivity

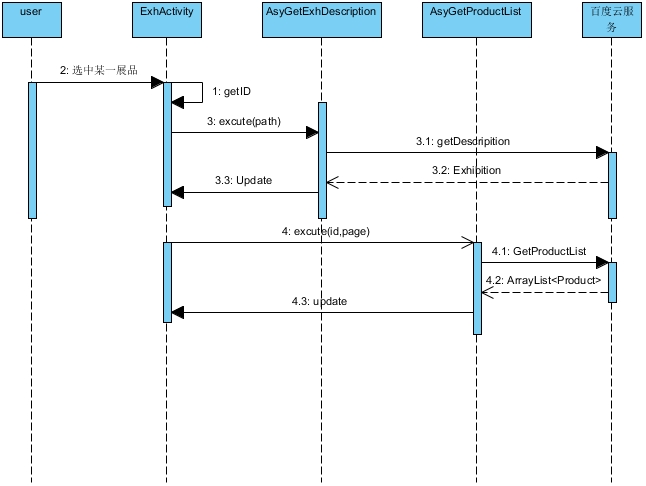


图 5 ‑2 展会详情页（二级页面）刷新时序

展会详情页（二级页面）刷新交互时序如上图所示:

1、 用户在展会列表页选中某一展会，系统跳转至展会详情页

2、 系统调用GetID，获取用户选中的展会ID

3、 系统启用多线程，向服务器同步请求返回展会详情

4、 服务器返回详情信息，逻辑层将信息封装为Exhibition实体类，调用Update刷新界面

5、 系统启用多线程，异步向服务器请求展会热门展品信息

6、 服务器返回热门展品信息，逻辑层将信息封装成ArrayList<Product>实体类列表，并调用Update刷新界面

* + 1. 展品详情页的详细设计与实现

展品详情页指的是用户展会热门产品视图下，选中某一展品后，查看到的对应展品的详情页，或者是用户将集成有NFC芯片的手机（安装了本系统）靠近展会中摆放的识别卡片后，系统识别了卡片携带的展品识别信息后，打开的展品详情页。

其主要职能是向用户展示展品详情信息，包括文字介绍、图片介绍、视频介绍等。相应的交互与视觉设计，参照第四章相关小节

类名：ProductActivity

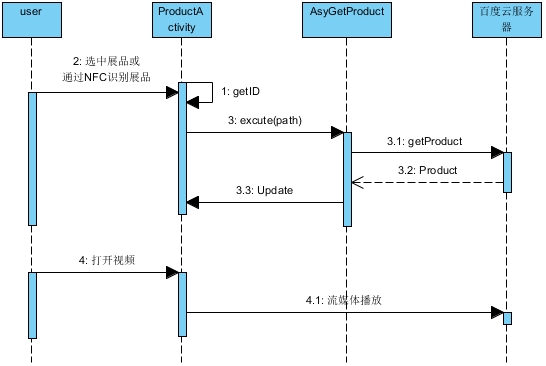


图 5 ‑‑3展品详情页（三级页面）刷新时序

展品详情页（三级页面）刷新交互时序如上图所示：

1、系统调用GetID，获得展品的ID信息

2、系统启用多线程，向服务器同步请求展品详情信息，逻辑层将信息封装成Product实体类，并调用Update刷新页面

1. 用户打开视频，系统打开自带视频播放器向服务器请求资源播放

## 5.2 逻辑层详细设计与实现

本节将会简要说明逻辑层各类的职责，并简述其设计。

* + 1. 联系人模块详细设计与实现

本小节主要说明联系人模块的详细设计与实现，包括数据库表的设计，与联系人模块使用流程。

表 5‑1联系人数据库表

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 举例 |
| Id | Int | 1 |
| Name | Varchar | 叶XXX |
| Phone | Int | 18500067965 |
| Home\_num | Int | 010-56799446 |
| Company\_add | Varchar | 北京市海淀区西二旗上地东路鹏寰国际大厦 |
| Company\_nam | Varcahr | 百度在线网络技术（北京）有限公司 |
| QQ | int | 307971172 |
| Image | Varchar | /storage/exguid/image/xxx.jpg |
| Rec\_Tag | Varchar | 百度产品经理 |

类名:ContactManager

该类的职责是，联系人管理。功能包括管理用户本人的信息和使用终端交换的联系人信息，这些信息支持插、删、改、查的操作。系统交换联系人逻辑模块也由其负责。

下图说明的是用户使用系统录入联系人信息流程：

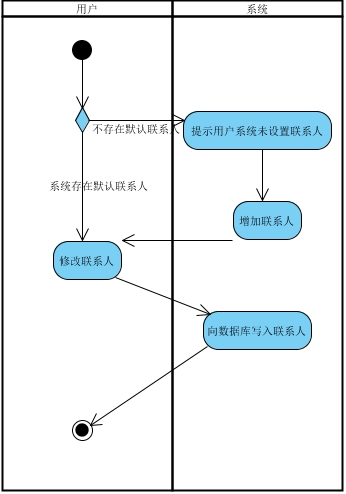


图 5 ‑4 系统录入联系人信息流程

系统将在用户进入联系人管理界面后，检查系统是否存在默认联系人，如果不存在将提示用户输入。

下图说明用户使用系统交换联系人信息流程

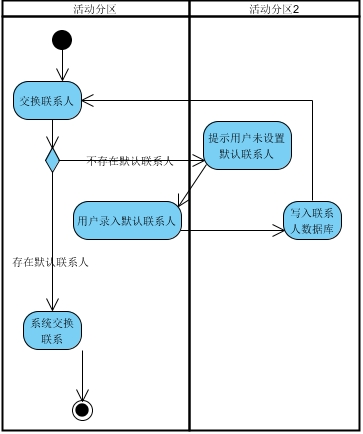


图 5 ‑ 5 系统交换联系人信息流程

## 关键问题及解决方案

本节将主要介绍系统的设计过程中发现的问题，及其解决方案。包括瀑布流实现与资源加载任务的设计等。

* + 1. 瀑布流图片的设计与实现

瀑布流是近几年流行起来的图片展示方式，相关介绍在第四章相关小节有所提及。但在移动终端上实现瀑布流存在几个缺陷，主要是移动终端设备的原因导致的。因为移动终端本身处理能力有限，内存小，移动浏览器解析能力弱，更残酷的是，因为移动终端显示屏小的原因，导致瀑布流在移动端不仅实现难度加大，交互的效果也会大打折扣。

但得益于Android系统架构的优良设计，在实现瀑布流效果上，系统可以不需要使用HTML语言实现，而使用Android系统本身提供的应用框架即可解决。

瀑布流的布局实现，采用动态加载Layout(布局)的方式解决。而图片资源和其他耗时资源的加载带来的巨大的系统开销将在下一个小节中提及解决方案。本小节着重解决瀑布流的布局问题

瀑布流布局分为三个部份，分别是：外层容器、内层容器和内部元素。

外层容器（outercontainer）：指最外层布局，它承载着整个瀑布流布局，通过向外层容器增加内层容器，可以增加瀑布流的展示列数；从实现上来讲，即是一个layout文件中里唯一的LinearLayout；

内层容器（innercontainer）：指内层布局，它是外层容器内的元素，承载着瀑布流中的某一列瀑布；从实现上来讲，它是向outercontainer添加的LinearLayout，有多少列瀑布就应该有多少个innercontainer.，即是向外层布局添加多个LinearLayout；

内部元素：指图片布局，它是内层容器的元素，一张图片对应一个内部元素的对象。从实现上来讲，可以是一个ImageView，也可以是包含ImageView的复杂布局。需要向innercontainer添加内部元素，才能在用户界面展示。内部元素是动态向内层容量添加的。

将瀑布流布局这样划分的好处是：

易于管理，通过容器对象即方便的向界面中添加元素；

易于扩展，通过全局常量，修改内部容器数量，从而增加或者是减少瀑布流的列数；

易于实现，该实现使用的是Android提供的标准组件，实现的难度大大降低。

下图表达的是三者的依赖关系与系统的实现;



图 5 ‑6 瀑布流容器设计

红色线框内即是外层容器，即是Layout中的LinearLayout；

蓝色线框内即是内层容器，即是向外层容器中添加的LinearLayout；

绿色线框内即是内部元素，即是向内层容器中添加的ImageView.

下图说明的是瀑布流图片的加载流程

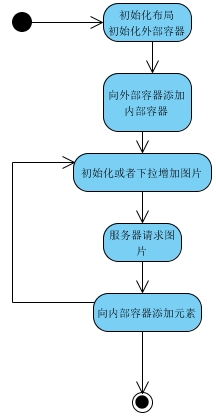


图 5 ‑ 7 瀑布流图片的加载流程

* + 1. 资源加载任务的设计与实现

因为系统将使用多种类型的资源，比如说文字、图片等。它们的加载将为系统带来巨大的开销，而使用异步线程解决耗时资源的加载问题，是一个必然的选择。但如果为不同类型的任务设计不同的异步线程会使得系统结构看起来复杂，所以设计中使用了任务（Task）的概念。设计任务接口，将逻辑操作与UI更新操作封装起来，不同的资源加载将实现任务接口，封装成不同任务。

异步线程将维持任务的队列，线程取出任务后，将任务逐出队列并执行任务。任务执行结束后调用相应接口，完成任务。

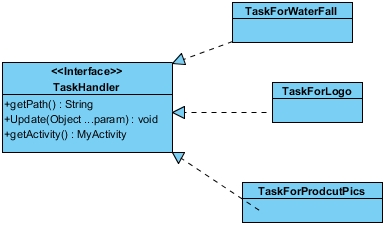


图 5‑8资源加载任务接口中与实现关系

TaskHandler接口，实现该接口必须实现下列接口：

getPath()函数，获得任务与后台交互的URL；

Update(object…param)函数，更新用户界面；

getActivity()函数，获得创建任务的父活动。

TaskForWaterFall、TaskForLogo、TaskForProductPics是基于TaskHandler接口，处理不同耗时资源加载任务的实体类。

* + 1. 异步加载线程的设计与实现

本系统的设计中，为了满足用户获取更多展会或者是展品信息的需求，系统将向用户展示大量的富媒体信息，其中包括了展示大量的图片。图片的加载是非常消耗资源的，单线程阻塞式加载对于用户体验的损伤是非常大的。

解决方案是使用多线程异步加载，但是考虑到瀑布流的实现需要更多的资源，所以将采用两个异步图片加载线程的方式，对于临界资源采用加锁的方式，同步两个线程。

图 5 ‑9 异步加载图片线程

Put Task

Critical Resource

LoaderImageTask1

Task

LoaderImageTask2

Remove Task

System

Task

需要异步加载的图片，系统会将其封装成Task，将Task放入临界资源。LoaderImageTask的两个实例，会不断轮询将临界资源取出，并执行，之后被执行的任务会从临界资源中移走。系统中存在两个LoaderImageTask实例，所以可能使系统处于不安全状态。

为了避免使系统处于不安全状态，引用临界区调度原则，如下：

1、如果有若干进程要求进入空闲的临界区，一次仅允许一个进程进入。

2、任何时候，处于临界区内的进程不可多于一个。如已有进程进入自己的临界区，则其它所有试图进入临界区的进程必须等待。

3、进入临界区的进程要在有限时间内退出，以便其它进程能及时进入自己的临界区。

4、如果进程不能进入自己的临界区，则应让出CPU，避免进程出现“忙等”现象。

根据临界区调度原则，设计中采用了加锁的方法，避免系统处于不安全状态。

当线程进入临界区时，打开锁，使得其他线程无法进入临界区。线程在临界区执行获取Task和从临界资源Remove Task的工作。在线程退出临界区前，关闭锁，使得其他线程可以进入临界区。

这样的设计不仅有效地避免的不安全状态的出现，得益于这种设计，线程的实例可以根据系统的负荷添加，使得执行效率大大提高。

## 小结

为保证系统运行流畅，设计中多处使用了多线程异步处理的办法。多线程的运用虽然解决了系统的运行效率问题，但仍然可能带入不安全状态。所以在设计过程中，对多线程问题作了严谨的设计，包括其线程运行时序问题。

图片的展示使用了当下流行的瀑布流形式，但是加载图片资源会带来的巨大的资源开销并影响系统的运行效率，针对这种情况，也采用了多线程和任务的方式解决。可这种方式会带来死锁问题，为了避免死锁的产生，系统也采用了相应的设计解决。这些都在本章的各个小节有所讲解。

# 总结与展望

## 总结

本文在观察国家宏观经济运行的基础上，抓住了经济转型的关键节点上，最有生命力的经济增长点之一 ―― 会展经济，以此为切入点，设计并实现了这一系统。

会展行业是朝阳产业、绿色产业，在相当多的一线二线城市，已经是支柱产业。但通过研究当前会展行业的现状，发现它在新媒体宣传方面存在严重滞后的问题。论文结合时下IT（Internet Technology）界最热门的两大话题，云计算和移动互联网，运用相关的知识理论和先进经验，以图解决这些问题。

本文在认真分析了用户需求和参展会的商业需求的基础上，对在系统的功能设计、交互设计做了大量的工作。包括使用时下流行的交互方式，和精力设计的系统模块，保证了系统高效的运行。依托百度提供的强大云能力，和借力移动互联网的高速发展，成功实现了该系统。

## 展望

本系统的实现，虽然为解决展会宣传能力滞后于展会发展的问题提供了一种可能性。但就目前本系统的实现和设计，仍存在着较大的不足，也意味着系统还有很大的提长空间。总的概括起来，包含以下几个方面：

1、用户需求和商业需求覆盖不足，只着重解决了用户的问题，但无法体会系统对于会展行业发展的商业价值。对于系统本身而言，如果能获得大量用户是有益的，但更大的意义在于转化这些用户消费，这是系统在解决了基础问题后，急需思考的问题。

2、系统的交互上，虽然尽力做到以用户的体验为出发点，但限于个人的能力，设计得不尽完善，应在用户体验上下更大的功能，以期获得更多用户的肯定。

3、系统的虽然使用了百度的云能力，但事实上，百度能提供的云能力还有很多。比如说Channel（管道，长连接），这些能力的使用能更大幅度的提高系统的可用性，无论是从用户出发还是商业目的出发，这都是有价值的，应该更多地使用。

4、系统架构和实现，限于个人水平，仍有相当大的提升空间，希望以后可以就系统运行效率、资源消耗等问题上着手，提升系统的质量。

参考文献

[1] Craing Larman著 李洋等译. UML和模式应用[M]. 原书第3版. 北京:机械工业出版, 2006.

[2] 李刚. 疯狂Android讲义[M]. 第1版. 北京:电子工业出版社, 2011.

[3] 杨丰盛. Android应用开发揭秘[M]. 第1版. 北京:2010, 机械工业出版社.

[4] Jesse James Garrett 著. 用户体验要素[M]. 原书第2版. 北京:机械工业出版社, 2011.

[5] 张波. O2O:移动互联网时代的商业革命[M]. 第1版. 北京:机械工业出版社, 2013.

[6] Ferraro R, Murat Aktihanoglu 著, 李丽译. LBS应用开发[M]. 第1版. 北京:人民邮电出版社, 2012.

[7] Juhani Lehtimaki 著. Smashing Android UI [M]. 第2版. John Wiley & Sons Inc, 2012.

[8] Jason Ostrander 著. Android UI Fundamentals: Develop & Design[M]. 第2版. Peachpit Press Publications, 2012.

[11] 百度. 百度个人云存储(PCS)[EB/OL]. [2013-5-14]. http://developer.baidu.com/wiki/index.php?title=docs/pcs.

[12] 百度. 百度应用引擎Baidu App Engine[EB/OL]. [2013-5-14]. http://developer.baidu.com/bae.

[13] 百度. 百度地图API[EB/OL]. [2013-5-15]. <http://developer.baidu.com/map/>.

[14] 马丽米克尔. 互联网趋势报告[R]. 斯坦福大学:KPCB, 2012-12-3.

[15] 邵维忠 杨芙清 著. 面向对象的系统分析[M]. 第2版. 北京:清华大学出版社, 2006.

[16] 邵维忠 杨芙清等. 面向对象的系统设计[M]. 第2版. 北京:清华大学出版社, 2006.

[17] Clifford A. Shaffer 著. 数据结构与算法分析[M]. 第2版. 北京:电子工业出版社, 2009.

致谢

本论文工作是在刘艳霞老师的悉心指导下完成的，非常感谢刘艳霞老师给予我这么大的自由，允许我在自己的设想下自由发挥。这种自由的学习与创作，使我保持了研究热情和攻克难关的信心，是我日日夜夜努力的源泉和动力。完全自主的选题，充分发挥了我的主观能动性，使我的能力进一步提高。再次感谢刘艳霞老师对我的信任和支持。

同时也非常感谢与我合作的李灏同学，和你一起开展项目是一段很愉快的经历。不仅让我在技术上有所提升，也让我认识到相互合作的意义和过程。另外，你在工作上表现出来的专业态度和坚忍不拔的精神更深深地影响了我，你身上所表现的优秀品质让我学习到了很多东西，

此外，对所有指导、帮助过我的老师和同学们都表示一致的感谢和敬意，你们专业的技术能力和助人为乐的做人态度都将令我永生难忘。

特别要感谢的，是在我求学生涯当中，一直坚定不移站在我身后，给予我支持和鼓励的父母，正因为你们的无私付出和无微不至的关怀，使得我能排除干扰专心学习、能不断地追求自己的梦想，正因为你们的爱，才有我今天的成绩。感恩父母，你们对我的痛爱将是我一生的财富。

最后，感谢所有帮助过我和关心过我的人，谢谢你们！