|  |  |
| --- | --- |
| 学号: | 1306010517 |
| 年级: | 13级 |



**本科毕业设计**

**基于百度地图和Bmob后端云存储的租车平台的设计实现**

|  |  |
| --- | --- |
| **专 业** | 计算机13级 |
| **姓 名** | 严德美 |
| **指导教师** | 张鹏程 |
| **评 阅 人** |  |

**2017年5月**

**中国 南京**

**BACHELOR'S DEGREE THESIS**

**OF HOHAI UNIVERSITY**

**Design and Implementation of**

**bike rental platform based on Baidu map and Bmob cloud storage**

College ：Hohai University

Subject ：Computer science and technology

Name ： Yan Demei

Directed by ：Zhang Pengcheng Associate Professor

NANJING CHINA

摘 要

近几年，特别是近两年来，共享单车如雨后春笋一般在全国各地不断涌现，以摩拜单车为代表的共享单车正在服务着通过互联网连接的我们，伴随着共享单车的盛行，人们的出行越来越离不开共享单车，这种共享经济极大地给人们带来了便利。同时我发现大家的自行车多数时候都处于闲置状态，所以产生一个想法，开发一个租车平台将大家闲置的单车出租出去，更好的利用资源。在这篇论文中，主要是基于百度地图、Bmob云存储、安卓平台的租车平台的设计与实现，百度地图提供了基础地图、实时定位、室内定位、导航、POI检索、路线规划等一系列有关地图和地理信息方面的API,其中定位的方式多样化，采用GPS、WIFI基站、IP混合定位模式。Bmob后端云存储提供了实时数据与文件存储功能，轻松实现应用“云与端”的数据连通，提供可视化的云端数据表设计界面，轻松建库建表。支持10种数据类型存储：如字符串、整型、数组等。Bmob通过提供的API接口和多语言的SDK，为平台提供快速上传、安全可靠的数据存储服务。租车平台的出发点是对大学生闲时单车的利用，本着诚信原则，拥有单车的学生可以使用租车平台出租自己的单车，有租车需求的学生可以在租车平台上进行租车，骑行结束返还单车，进行计费服务。

关键词：百度地图；Bmob云存储；共享单车

**ABSTRACT**

In recent years,especially in the past two years, the sharing of bicycles has been sprung up everywhere in all parts of the country. The shared bicycles ,represented by Mobike,are serving us via the Internet,along with the prevalence of shared bicycles. People travel more and more inseparable from sharing the bike,this shared economy has greatly brought convenience to people. At the same time I found that most of the bicycles are in idle state, so produce an idea, develop a bicycles rental platform so that people can rent out their own idle bicycles, better use of resources. In the thesis, mainly about the design and implementation of bike rental platform based on baidu map, Bmob cloud storage,Android platform. Baidu map provides a basic map, real-time positioning, indoor positioning,navigation,POI search,route planning and a series of related Maps and geographic information on the API,where the positioning of the way diversified, using GPS,WIFI base station, IP mixed positioning mode. Bmob back-end cloud storage provides real-time data and file storage capabilities,easy to implement the application of “cloud and terminal” of data connectivity, to provide a visual cloud data table design interface, easy to build the table. Supports 10 types of data storage: such as string, integer, array, and so on. Bmob provides fast upload, secure and reliable data storage services for the platform through API interfaces and multilingual SDKs. The starting point of the bike rental platform is the use of leisure time cycling, in line with the principle of good faith, students has bikes can rent out their bikes use the bike rental platform. Students need hire a bike can hire a bike in the bike rental platform. End of ride back to the bike, billing services.

**Key words:** baidu map；Bmob cloud storage；shared bike

第1章 绪论

1.1 研究背景及研究意义

1.1.1 研究背景

随着智能手机这一计算机平台的流行，越来越多的人拥有手机，移动互联网普及速度越来越快，普及面越来越广，将移动端的应用和线下的单车通过互联网连接起来的这种B2C的模式很受欢迎。共享单车最早开始于北京大学的校园内，随着科技的进步，经济的发展，人们的生活节奏越来越快，交通的方式也越来越多样化，即使交通方式多样，也很难直接到达目的地，需要共享单车这一代步工具，有效解决了用户离目的还有1公里左右打车太奢侈、长途跋涉再步行太痛苦的窘境，而且能够响应国家大力提倡的绿色低碳出行的号召，与此同时骑车还能锻炼身体，强身健体，欣赏骑行的风景，享受生活，给用户带来极大的便利，提高用户体验。随着共享单车的流行越来越暴露出一个问题，就是自己有车不骑去骑共享单车，甚至长时间停在停车场无人问津，被厚厚的灰尘覆盖。

现在的共享单车都是提供服务的公司投放的，使用者是用户，所以共享单车的商业模式是B2C的模式，而暂时还没有C2C的商业模式，使用者无法出租自己的自行车。在校园内不难发现停车场内的自行车大多时候处于闲置状态，而且拥有自行车的同学由于满足好奇心宁愿去骑共享单车，体验一下共享经济带来的用户体验。

1.1.2 研究意义

共享单车的好处就在于不管你在何地，只要你所处的地图上有共享单车，你就能骑行，也不管你骑行的目的地是哪里，都能够停车，而且不用考虑停车之后的事情，因为停车之后的单车就不属于你了，就又回到了地图上，这样节省了自己管理单车的时间，免去了单车迁移的烦恼。共享单车能够有效疏解地铁站、火车站、飞机场、公交车站、居民住宅区等地方的人流量，共享单车也符合可持续发展和和谐社会的发展理念。2016年北京市日均浓度未达标天数达到168天，占全年45.9%的时间，给城市空气质量带来巨大的威胁；2016年，济南、哈尔滨、北京等主要一二线城市的拥堵指数均高达2左右，致使城市出行效率大大降低；随着环境污染、城市拥堵问题的日益加重，政府、资本以及用户对出行环境的改善的需求也愈发强烈，共享单车以其绿色环保、轻便高效的出行模式赢得了社会的好感度和政府的鼓励支持。

对于坐车上班的人来说，他们最头疼的是从站台到公司或者居住地到站台需要步行，共享单车的出现正好帮他们解决了问题，能够有效的节省时间并且锻炼了身体，对于在校大学生来说，校区非常大，常常在图书馆和宿舍、宿舍和教学楼、教学楼和图书馆之间步行，有些时候上课甚至会迟到，自己买辆新的自行车，没过几个月就被偷了，既浪费钱又影响学习的情绪，得不偿失。对于普通人员来说，你去几公里外的地方，如果没有自行车只能步行，既浪费时间又浪费体力，特别是一个人来到一个公交和地铁不到达的陌生的地方，就更需要共享单车的服务了。到现在为止，市场上已经出现了多达接近30个的共享单车应用。在我身边就有很多同学手机上安装了多家共享单车的应用，上班下班，上学放学都骑共享单车已经成为了一种时尚。

因此，共享单车能够提供便利，优质的服务。本租车平台在共享单车的基础上，对单车的对象和服务进行了拓展，使得在租车平台上注册的任何人都可以出租自己的自行车，也可以在平台上进行单车的租用。大多时候，大家拥有的自行车都处于闲置的状态，都停在停车棚内，在别人需要骑车时，看见停车棚内停满了车辆，但是就是没有一辆单车能够供自己骑行，在租车平台上将自己的自行车出租出去不仅能赚得一份收入，同时还会给其他人带来便利。

此外，现在智能手机成为了大家生活中不可或缺的一部分，给平台的使用带来了便捷性，借助于百度地图，租车平台能够提供实时的路况和位置信息，对于出行导航更方便。

1.2 研究现状

现在有很多关于共享单车的应用，见得比较多的有：摩拜单车，ofo小黄车，Hello Bike,小蓝单车。共享单车更好地解决了用户“最后一公里”出行的痛点，成为市场新热点，同时国家和地方政府对共享单车持正面积极态度并出台了相关的政策，带动了共享单车的快速发展。随着国内经济持续上升，智能手机的渗透率不断增长，也给共享单车应用的推广带来了便捷性。2016年GDP增长至74.4万亿元，整体智能手机保有量稳步上升，2016年智能手机渗透率已达到75.7%。在经济快速发展的同时，空气污染严重，雾霾频发，城市交通拥堵，这一系列出行问题亟待解决，而共享单车正好可以缓解这些问题。然而随之共享单车在迅速被推广之时，也出现了很多问题。共享单车采取的是随时随地租用，随时随地还车的方式，在共享单车为人们提供便利之时也出现了很多素质低下的人破坏单车和违规停放的情况。媒体报道比较多的就是ofo小黄车被上私锁，小黄车被当做自家自行车来使用，在寻车过程中我也发现好多摩拜单车被市民推到了屋内，共享单车私用。共享单车租用结束后随意停放，甚至停在路中间，严重影响交通，还有人甚至刮花单车上的自行车编号，导致人们无法再租用。在上下班的高峰期，共享单车的应用经常性出现登录不了，无法连接，地图刷新不了，无法进行租车等问题，共享单车想要实现真正的全民共享，满足人们低碳出行的需求，就必须优化后台管理系统，将共享单车应用瘫痪的可能性降至最低。

虽然现在共享单车已经很流行，提供服务的商家众多，但是目前共享单车的商业模式仍然是B2C的模式，而不同于滴滴和Uber的C2C的模式，简单说就是，现在使用市面上的共享单车应用只能租用服务商提供的单车，而无法将自己的单车在平台上进行租用。这还不同于滴滴或者Uber，因为自己的单车在平台上租用后，被别人骑走，自己又得去找回费时费力，所以起初只能是小范围将服务进行展开，比如说用户出租的单车的租用范围限制在用户所在的小范围内，比如一个校区或者小区内。

1.3 本文的主要研究内容

本文主要的研究对象就是基于Android平台，使用Bmob云存储进行数据的存储和传输，使用百度地图提供的地图相关的功能，开发出一个适用于小范围内共享单车的系统。租车平台是在Android6.0平台下进行开发的，使用第三方的Bmob云存储提供的云与端实时数据同步，数据库建表，数据的增删改查，用户注册的邮件认证的功能。基于百度地图进行地图方面的开发，使用百度地图提供的个性化地图服务，可以在基础地图的基础上根据自己的应用需求定制地图服务，定位服务，基于GPS、WIFI、基站的综合定位服务，具有定位精度高、覆盖率广、网络定位请求流量小，定位速度快等优势，出行服务，提供步行、骑行、公交、驾车等多种类型的路线规划方案,然后开发一个基于安卓平台的租车系统，对整个租车的流程进行代码实现。

主要研究内容如下：

首先对现在共享单车的使用状况进行分析，对现有共享单车的应用进行分析，发现应用的优缺点，优点在系统中进行借鉴，缺点在系统中进行改进。进行系统设计以及各个模块的设计，对设计中需要用到的知识内容进行学习，对设计的系统进行代码实现。

然后是关于在地图上生成可租用的自行车图标、进行路线规划、用户骑行结束怎么保存骑行的路线。在地图中生成图标需要提供经纬度信息，由于这个系统并没有正真和自行车进行绑定，所以要进行模拟自行车，当要在地图上生成可租用自行车图标(marker)时，一次性生成30个图标，先生成两个随机数，分别是经度随机数和纬度随机数，然后将30分成四个部分，每个部分对应当前定位的经纬度和经度随机数、纬度随机数的加、减的四种随机组合中的一种组合。当点击地图上的一个marker时，通过marker的经纬度和当前定位的经纬度进行路径搜索，最后显示路线规划的结果。在用户骑行的过程中，每两秒上传一次用户的当前的经纬度，然后在骑行结束时，就上传了骑行过程的一个个点，最后根据这些点生成骑行的路线。

1.4 本文的组织结构

论文一共分为五个章节，相应的每一章节内容如下：

第一章为整个论文的绪论部分，在绪论部分中，主要对租车平台的研究背景，租车平台的研究意义，租车平台的研究现状和租车平台的研究内容进行介绍；

第二章

第2章 租车平台相关概念及技术概述

2.1 安卓平台相关概念及技术

在安卓平台中，有基本的四大组件，它们分别是Activity、Service、BroadcastReceiver和ContentProvider，本租车平台只用到了上述四大组件中的两个组件，分别是Activity和Service组件，以下分别对这两个组件进行介绍。

Activity是Android应用的重要组成单元之一，而Activity又是Android应用最常见的组件之一，不同的Activity向用户呈现不同的操作界面。Android引用的多个Activity组成Activity栈，当前活动的Activity位于栈顶。



图 1 Activity类图

图1显示了部分Android提供的Activity类，自己可以个性化的定制自己所需的Activity，建立自己的Activity也需要继承Activity基类，有时在不同的应用场景可能需要继承Activity的子类，比如列表Activity类ListActivity。

当Activity处于Android应用中运行时，它的活动状态由Android以Activity栈的形式管理，当前活动的Activity位于栈顶。每个Activity都有可能从活动状态转入非活动状态，也可能从非活动状态转入活动状态。

如图2所示，Activity大致会经过如下4种状态。

运行状态：当前Activity位于前台，用户可见，可以获得焦点。

暂停状态：其他Activity位于前台，该Activity依然可见，只是不能获得焦点。

销毁状态：该Activity不可见，失去焦点。

当一个Activity启动另一个Activity时，常常会有一些数据需要传过去，在两个Activity之间会有一个负责携带信息的类：Intent，因此我们主要将需要交换的数据放入Intent即可。

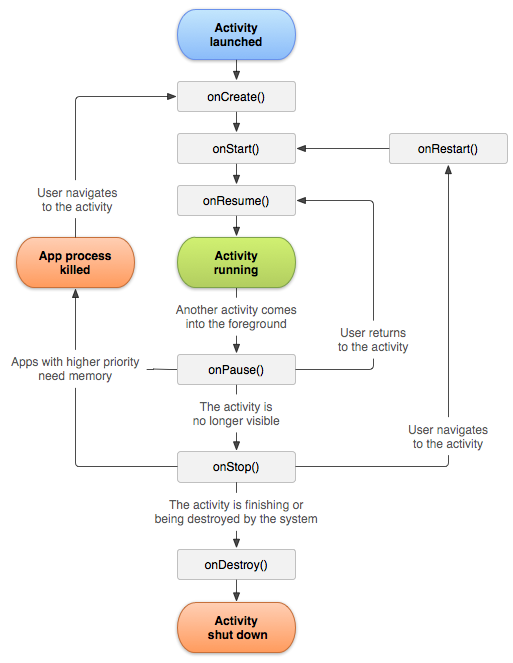


图 2 Activity生命周期

Service是Android四大组件中与Activity最相似的组件，它们都代表可执行的程序，Service与Activity的区别在于：Service一直在后台运行，它没有用户界面，所以绝不会到前台来。一旦Service被启动起来之后，它就与Activity一样，它完全具有自己的生命周期。如果某个程序组件需要在运行时向用户呈现某种界面，或者该程序需要与用户交互，就需要使用Activity；否则就应该使用Service。要实现自己的Service时，都要继承Service基类，并在AndroidManifest.xml文件中配置该Service。

如图3所示，Service的生命周期分为非绑定Service的生命周期和被绑定生命周期，当通过Context的startService()方法启动Service时即对应非绑定生命周期，访问者与Service之间没有关联，即使访问者退出了，Service也仍然运行。当通过Context的bindService()方法启动Service时即对应被绑定生命周期，访问者与Service绑定在一起，访问者一旦退出，Service也就最终终止了。

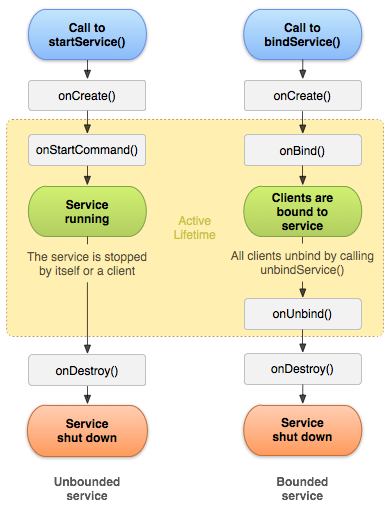


图 3 Service生命周期

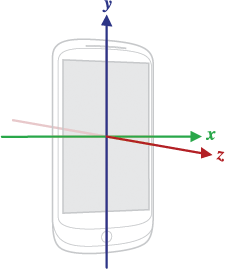


图 4 安卓手机坐标系

安卓中传感器有8种左右，5.0的安卓版本又新增了心率传感器，本系统中只用到了其中的方向传感器，当然所用的百度地图API中还用到了陀螺仪传感器。如图4所示，当一个设备被放在其默认方向上时，x轴是水平指向右的，Y轴是垂直向上的，Z轴是指向屏幕正面之外的，即屏幕背面是Z 的负值。当设备运动或者旋转时，这些坐标轴是不会改变的，它们是跟随手机的。

方向传感器返回三个角度，分别是手机顶部朝向与正北方的夹角即与Y轴的夹角，手机顶部或底部翘起的角度即与Z轴的夹角，手机左侧或右侧翘起的角度即与X轴的夹角。通过安卓的传感器管理类SensorManager类返回当前方向传感器的值，然后传给定位数据建造器，对地图上的定位方向数据进行实时的更新，以显示当前用户的朝向，为用户按地图上的路线导航提供便利。

2.2 百度地图相关概念及技术

百度地图提供了丰富的类和方法，本系统只用到了其中的几个方面。

百度地图提供了丰富的类和方法，本系统只用到了其中的几个方面。



1、地图

地图展示和地图操作功能；

地图展示包括：普通地图(2D和3D)和实时交通图

地图操作：可通过接口或手势控制来实现地图的点击、双击、长按、缩放、旋转、改变视角等操作。

地图展示：开放高清4K地图显

图 5 基础地图

本系统在主界面放置基础地图，可显示建筑物和实时交通图，也可通过对地图进行一系列操作，比如拖拽，双击放大地图，点击指南针改变视角，缩小等。



1. 地理编码功能

将地理坐标和地址之间相互转换

正向地理编码：实现了将中文地址或地名描述转换为地球表面上相应位置的功能。

反向地理编码：将地球表面的地址坐标转换为标准地址

图 6 地理编码

本系统使用了地理编码的反向地理编码功能，通过点击地图上的某一辆单车，可以获取单车的经纬度，然后进行反向地理编码，获得单车的中文地址，比如：江苏省南京市江宁区佛城西路。



3、路线规划

支持公交信息查询、公交换乘查询、公交/驾车/骑行/步行路线规划

步行路线检索：支持步行路径的规划

图 7 路线规划

本系统使用了其中的步行路线检索功能，当用户点击地图中的单车图标(Marker)时，将Marker的当前位置信息和用户的当前位置信息传给步行路线规划的类，然后就会得到一个步行路线，得到这个路线之后就可以通过折线覆盖物来绘制出步行的路线。



4、地图覆盖物

百度地图支持多种地图覆盖物，帮助展示更丰富的地图。目前所支持的地图覆盖物有：定位图层、地图标注(Marker)、几何图形(点，折线、弧线、多边形等)、地形图图层、POI检索结果覆盖物、路线规划结果覆盖物、热力图层，瓦片图层。

图 8 地图覆盖物

本系统使用了其中的定位图层，地图标注，几何图形，路线规划结果覆盖物等。当用户出租自己的单车时，就会上传用户现在所处位置信息用于生成单车的地图标注，所处位置信息需要定位图层的支持。步行路线规划需要路线规划结果覆盖物的支持，记录用户的骑行路线信息需要几何图形的支持。



1. 定位

采用GPS、WIFI、基站、IP混合定位模式，能够提供定位精度高、覆盖率广、网络定位请求流量小、定位速度快的定位功能

图 9 定位

百度地图会根据当前网络状况自动选择定位模式，或者多种定位同时进行，在定位时有多种参数可以进行设置，也可以选择是否使用GPS进行定位，定位时间间隔设置，设备方向信息等。

2.3 Bmob云存储相关概念及技术

轻松搭建应用数据库、支持可视化操作，Bmob提供可视化的云端数据表设计界面，轻松建库建表。支持10种不同数据类型存储，如字符串，整型，数组等。通过云推送，可以随时随地向应用程序的用户推送通知或消息，与用户保持积极互动，提升用户留存率，活跃度以及用户体验。用户是应用程序的核心，Bmob提供了一整套用户注册登录的功能，支持邮箱注册登录，手机验证码登录，第三方授权登录方式。在软件架构层面，Bmob提供了传输层面、应用层面、表层面、ACL角色管理、发布层面等不同粒度的权限控制的方式，确保用户数据安全。为了防止程序异常导致数据丢失的情况，Bmob提供了数据库中的数据实时/定时在线备份，持续保持应用数据，在需要时，可以及时还原数据。

**第3章 概要设计**

本章对系统进行概要设计，包括系统功能模块设计、接口设计、运行设计、出错处理设计等内容[10][11][12]。

第3章 基于安卓平台的租车系统的设计

3.1 系统总体设计

整个系统主要包括用户功能模块和单车相关模块，用户功能模块包括用户欢迎部分、用户登录部分、用户注册部分、个人中心部分、找回密码部分，个人信息设置部分，单车相关模块主要包括所有可用单车展示部分、出租单车部分、我的行程部分、租用单车、单车信息展示部分。欢迎界面主要是和单车主题相关的停留几秒的flash，用户的登陆注册用于提供给用户进行登陆和注册，个人中心里面包括出租单车、我的行程、设置的入口。主界面用于展示所有的单车，以及



图 10 租车系统功能结构示意图

提供租用单车的入口。如图10所示。

3.1.1 系统整体运行流程

系统运行流程如下：

首先将打包好的app下载安装到手机客户端，打开应用，展示给用户的是一张基础地图，在基础地图上显示着现在可租用的共享单车。

然后如果用户点击左上角的按钮或者地图上的单车想进行租用，当前用户如果没有登陆，则会进入登陆界面，若还未进行过注册，则可以选择进行注册后进行邮箱验证后再登陆，当前用户如果已经登录，则点击左上角的按钮会进入个人中心，进行用户信息的展示，点击地图上的单车进行租用则会进行单车信息展示界面，如果用户想骑车，则可以点击骑行，就可以骑行单车，骑行结束后点击结束骑行会进入主界面又回到开始的状态。如果用户在主界面通过左上角的按钮进入个人中心，在个人中心会有用户信息的展示和出租单车、我的行程、设置等入口。用户可以选择相应的入口进入使用相应的功能。如图11显示的是租车系统运行的示意图：



图 11 租车系统运行流程示意图

3.1.2 系统设计过程

在租车系统设计的过程中，涉及到用户功能的设计以及租车功能的设计，具体过程为：

1. 分析当前现有的共享单车系统，并加上本系统的需求，根据需求确定出租和租车所需的功能流程，确定数据类型和表结构。
2. 搭建出租单车和租车过程中地图框架，在进行系统设计之前，先把百度地图关于地图定位、安卓方向传感器进行方向定位、反向地理编码等相关的地图环境搭建好。
3. 根据百度地图返回的数据进行单车系统流程的设计，在系统设计之前先随机生成模拟共享单车的坐标，然后在基础地图上生成Marker。
4. 对系统进行功能设计，设计整个系统的功能框架。
5. 将租车系统集成到租车系统当中，进一步设计用户功能，分析用户所需的数据，在Bmob云数据中建立用户数据表，最后测试用户的功能流程是否正常。

3.2 用户功能模块设计

用户功能模块主要是对用户提供功能的界面,用于获取用户的输入信息以及和用户进行交互的部分。在MVC中主要是视图部分。

3.2.1 单车显示模块

此模块主要是对所有可以使用的单车在基础地图上进行显示。当用户进入该模块时，会根据当前网络状况进行实时定位，用户也可点击左下角的定位圆形按钮进行定位，当用户水平转动方向时，定位图标的箭头会跟当前用户手机顶端所指方向一致，当用户点击基础地图上的某一个单车图标时，会根据当前用户定位与单车图标位置进行步行路径规划，以提供用户进行步行的导航，点击单车图标的同时，会在地图的顶端显示弹出框，弹出框的首行是通过单车的经纬度进行反地理编码的中文地址，中间分别是骑行预计花费，步行距离，步行时间，提供用户参考是否要寻车，最下面一行是租用，当点击租用时，会进行单车详细信息显示的模块。

3.2.2 用户注册模块

此模块是本系统提供用户进行注册的模块，当新用户使用此租车系统时先要进行注册。注册时，用户需要填写用户名、密码和邮箱，这三项必不可少，邮箱必须是可以正常接收邮件的邮箱，由于用户的认证，当用户进行注册后，需要去邮箱进行注册，已确定是本人注册，不然无法登陆。当注册成功后会进行登陆界面进行登陆。

3.2.3 用户登录模块

此模块是本系统提供用户进行登陆的模块，当用户注册后或者退出登陆后，要进行用户登录，在登录时用户需要提供注册时的用户名和注册时的密码，然后系统调用Bmob提供的用户登录验证功能将数据传送到云端进行用户的验证，将验证结果返回给用户，用户信息验证通过即登录成功后进入单车显示模块，用户信息验证不通过即登录失败会提示用户失败信息。如果用户登录时忘记密码可以点击忘记密码按钮，进入忘记找回密码模块。

3.2.4 个人中心模块

此模块是本系统提供用户的个人信息的展示，主要是提供用户进行出租自己的单车，展示用户的头像，用户名，性别，用户可以从这个模块进入用户信息设置模块，进入查看用户的行程模块。点击最下面红色的退出登陆按钮，用户可以退出登陆，然后进入主界面。

3.2.5 个人资料设置模块

此模块是本系统提供用户的个人信息的设置，主要是对用户个人资料的设置以及显示。在本模块中，用户可以设置个人的信息，包括用户的头像、用户的昵称、用户的个人邮箱、用户的性别、用户的手机号码。用户的邮箱由于注册时进行验证是否是本人注册，昵称是一个不可重复的字符串。

3.2.6 我的行程模块

此模块是本系统提供用户的行程的展示，主要是提供给用户查看已经骑行的的所有行程，用一个列表展示用户所有已经骑行的行程，每一项有骑行结束的日期和时间，具体到分钟，自行车的编号，骑行时间，骑行花费。每一条有关于用户每一次骑行的详细信息，可以提供让用户了解每一次详细的骑行，当用户想更加细致的了解某一次骑行时，用户可以点击某一条骑行记录，然后就可以查看这一次骑行的更加详细的信息，会有这一次骑行中，开始骑行的位置，结束骑行的位置，和骑行过程中的路径，是一条平滑的曲线展示在地图中。

3.3 单车相关功能模块设计

租车功能模块是整个租车系统中核心部分，涉及到整个系统中用户定位，步行路线规划，两点之间计算最短路径距离，地图覆盖物。本模块主要运行在后台当中，在MVC中承担控制器和模型的部分。

3.3.1 模块总体设计

在本模块中，主要是进行单车的相关操作。涉及到单车的信息的维护以及实时更新单车的状态和骑行时的状态。在用户未骑单车时，用户可见所有可租用的单车，用户可以从现有可租用单车中租用一辆单车进行骑行，当有一个用户租用了云端数据库中的一辆单车时，要将这辆单车设置为其他用户不可见状态，然后更新云数据库中的单车的状态，在用户骑行的过程中，系统每两秒上传用户位置信息，相当于在地图上打点，用于记录用户的行程，便于后期绘制用户骑行行程曲线。当用户骑行结束时，要进行计费服务，并同时更新云数据库中刚才用户使用的那辆单程状态信息，从不可租用状态变为可以租用状态，然后骑行弹出框消失，用户回到主界面。具体流程如下：



图 12 单车租用流程图

3.3.2 单车租用模块设计

本模块是对租车系统中用户租用单车的功能进行简要介绍。当用户点击地图中一个单车Marker时，会根据当前用户定位与单车图标位置进行步行路径规划，以提供用户进行步行的导航，点击单车图标的同时，会在地图的顶端显示弹出框，弹出框的首行是通过单车的经纬度进行反地理编码的中文地址，中间分别是骑行预计花费，步行距离，步行时间，提供用户参考是否要寻车，最下面一行是租用，当点击租用时，会进行单车详细信息显示的模块。

3.3.2 单车信息显示模块设计

本模块是对租车系统中用于想租用的单车详细信息的展示进行简要介绍。首先显示的标题的单车信息，下面有以下一系列关于单车的详细信息，左边显示要显示的信息名字，右边是对应的信息内容，有打车的所属车主，也就是出租这辆车的用户的昵称；单车类型，单车类型有以下几种，通勤（普通）自行车、旅行自行车、山地自行车、公路自行车、小轮车、死飞车、折叠自行车、速降山地车、越野公路车、计时赛自行车、场地赛自行车、斜躺车、协力车、淑女自行车、亲子自行车、水上自行车、海滩自行车；单车的出租日期，具体到秒，提供用户参考到现在为止过了多长时间；出租时长，用户用现在时间减去单车出租日期，然后再减去出租时长就是这辆单车剩余能骑行的时间；出租这辆单车的用户的联系方式，方便用户骑行时联系车主；锁密码，用于用户开锁；单车描述，主要是关于单车的一些附加信息或者倡导绿色出行的一些标语。

3.3.2 骑行模块设计

本模块是对租车系统中用户骑行状态的实时展示与记录。当用户选定一辆单车并骑行时，用户当前视野中只有当前的定位和骑行的详情。单车的图标将会消失，定位图标会随着用户位置的改变而实时更新，同时也可以显示用户当前朝向，骑行的详情有当前用户所处位置的经纬度反向地理编码的中文地址；根据骑行时间计算出的预计费用，帮助用户了解当前的骑行已花费用；骑行路程信息，就是离用户骑行始发地总共骑行的路程；骑行时间，从租用开始花费的时间，租车的费用是根据这个在1.0元的基础上每半小时0.5元的价格往上加的。还有一个最后的结束骑行按钮，帮助用户结束骑行，并上传到云端的数据库中。

3.4 数据库设计

为了实现整个租车系统，同时方便数据的存取、统一管理和单车系统业务的实际需求。本系统设计了两个表进行数据的保存，所有的数据和数据表都是存储的云端，存取、访问、更新都是应用“云与端”的数据连通。

\_User表主要用于保存用户的个人信息，主要是objectId（主键）、用户名、密码、性别、学校、积分、邮箱、注册时间、邮箱是否验证。表结构如下：

表1 \_User数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 中文描述 | 数据类型 | 是否为空 | 备注 |
| objectId | \_User表的Id | String | 否 | 主键，自动生成 |
| username | 用户名 | String | 是 |  |
| password | 密码 | String | 是 |  |
| sex | 性别 | Boolean | 是 | true代表男 |
| school | 学校 | String | 是 |  |
| incredits | 积分 | Number | 是 |  |
| email | 邮箱 | String | 是 |  |
| emailVerified | 邮箱是否验证 | Boolean | 是 |  |
| createdAt | 注册时间 | Date | 是 |  |

BikeInfo表主要用于保存单车的信息，主要有objectId（主键）、所属用户(外键)、单车位置、锁密码、出租时长、车主手机号码、单车是否被使用、单车类型、单车描述、出租时间。表结构如下：

表2 BikeInfo数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 中文描述 | 数据类型 | 是否为空 | 备注 |
| objectId | BikeInfo表的Id | String | 否 | 主键，自动生成 |
| user | 单车所属用户 | Pointer | 是 | 外键，实际值为\_User表的objectId |
| location | 单车经纬度信息 | GeoPoint | 是 |  |
| unlockPass | 锁密码 | String | 是 |  |
| rentTime | 出租时长 | String | 是 |  |
| phoneNumber | 车主手机号码 | String | 是 |  |
| isUsed | 单车是否被使用 | Boolean | 是 |  |
| bikeType | 单车类型 | String | 是 |  |
| bikeDetail | 单车描述 | String | 是 |  |
| createdAt | 出租时间 | Date | 是 |  |

RouteInfo表主要用于保存用户骑行路程的信息，主要有objectId（主键）、单车编号、骑行花费、骑行时间、骑行时记录的经纬度信息、骑行的用户。表结构如下：

表3 BikeInfo数据表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 中文描述 | 数据类型 | 是否为空 | 备注 |
| objectId | BikeInfo表的Id | String | 否 | 主键，自动生成 |
| user | 骑行所属用户 | Pointer | 是 | 外键，实际值为\_User表的objectId |
| routeList | 骑行时记录的经纬度信息 | Array | 是 |  |
| bikeId | 单车编号 | String | 是 | 实际值为BikeInfo的objectId |
| bikeType | 单车类型 | String | 是 |  |
| createdAt | 骑行时间 | Date | 是 |  |

3.5 本章小结

本章主要是讲述整个租车系统的设计过程。作为整个论文的重点部分，首先对整个租车系统的设计，涉及到用户使用的功能部分，以及单车租用功能部分。然后，分为两个大模块：用户功能模块和单车租用模块进行介绍。详细介绍了用户功能相关模块的设计，同时对单车租用功能模块中使用到的相关定位，地理信息获取，步行路线规划进行了详细介绍，叙述租用单车，骑行单车，结束骑行的整个流程。最后讲述了整个系统中所使用到的数据表的结构。通过详细的设计过程，为接下来的系统实现提供相关的依据。

1. 基于安卓平台的租车系统的实现和测试

通过基于安卓平台的租车系统的设计，让我们在理论角度更加明确整个系统的运行流程以及相关模块的内容。整个系统是基于安卓平台、百度地图和Bmob云存储的，作为“云与端”互联的应用运行在安卓系统的智能手机上。下面将对实现过程进行详细讲述。

4.1 系统开发工具和平台

操作系统：Windows 7

数据存储：Bmob后端云存储

开发工具：Android Studio，Android SDK，JDK 1.8，github

4.2 用户功能模块的实现

4.2.1 用户欢迎界面

本界面主要是在用户打开本租车系统时进行显示。通过Timer运行一个耗时任务，使得欢迎界面在手机上停留2秒，相关的界面如下图：



图 13 欢迎界面

4.2.2 单车显示模块

在本模块中，通过MapView显示基础地图，通过读取云端的数据在地图上显示单车Marker，通过LocationService的start方法进行定位，将定位圆形按钮放到MapView中，当点击左下角的定位圆形按钮，定位的模式和定位的图像都会该表。第一个图定位处于正常模式，第二个图定位处于跟随模式，第三个图定位处于罗盘模式。相关界面如下图：

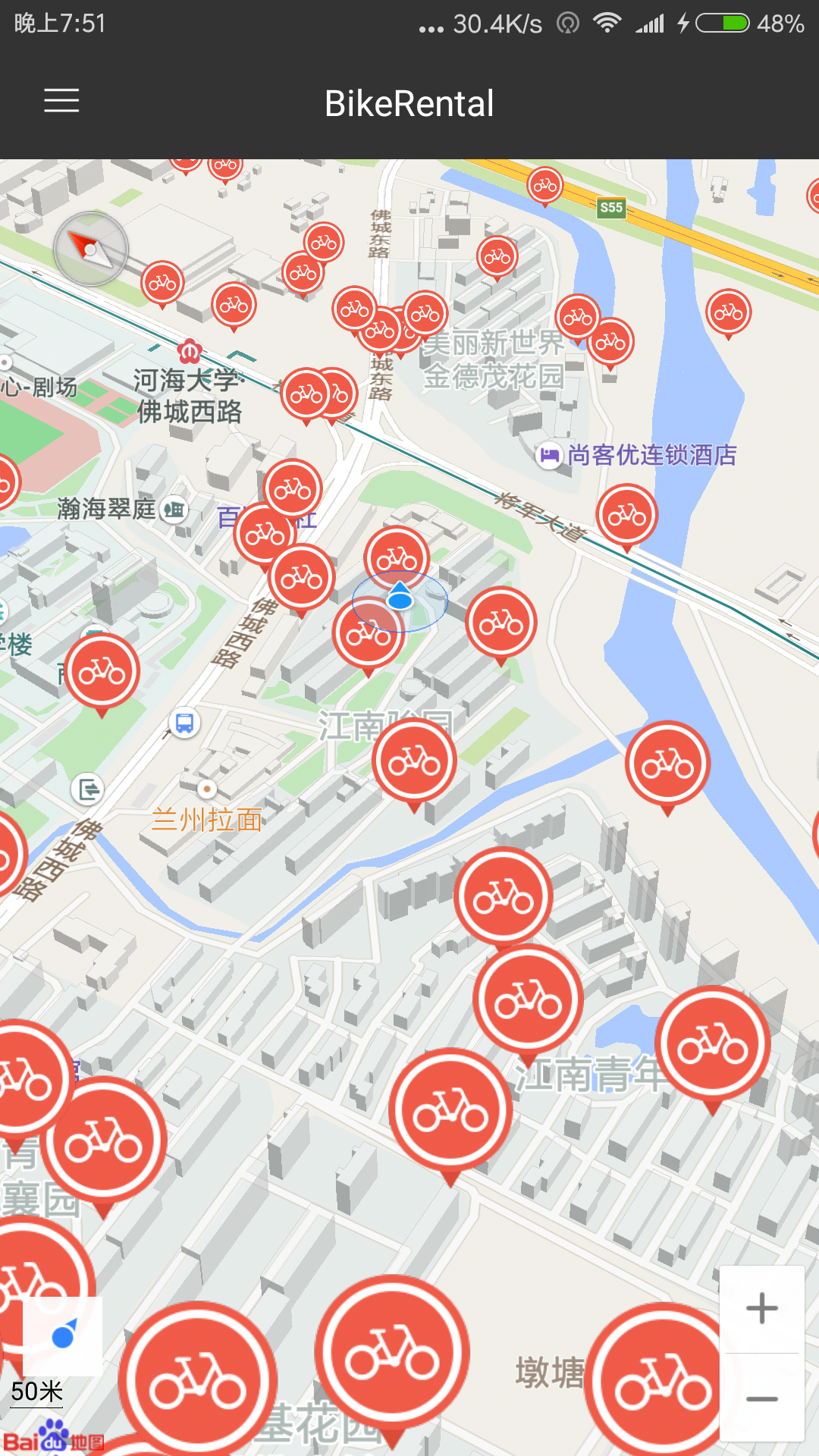


图 14 单车显示示意图

4.2.3 用户登录模块

在本模块中，用户可以进行输入用户名，密码，然后登陆，也可以点击注册进入用户注册模块，还可以点击最下面的忘记密码进行找回密码。也可以点击左上角的返回按钮返回到单车显示界面。相关界面如下图：

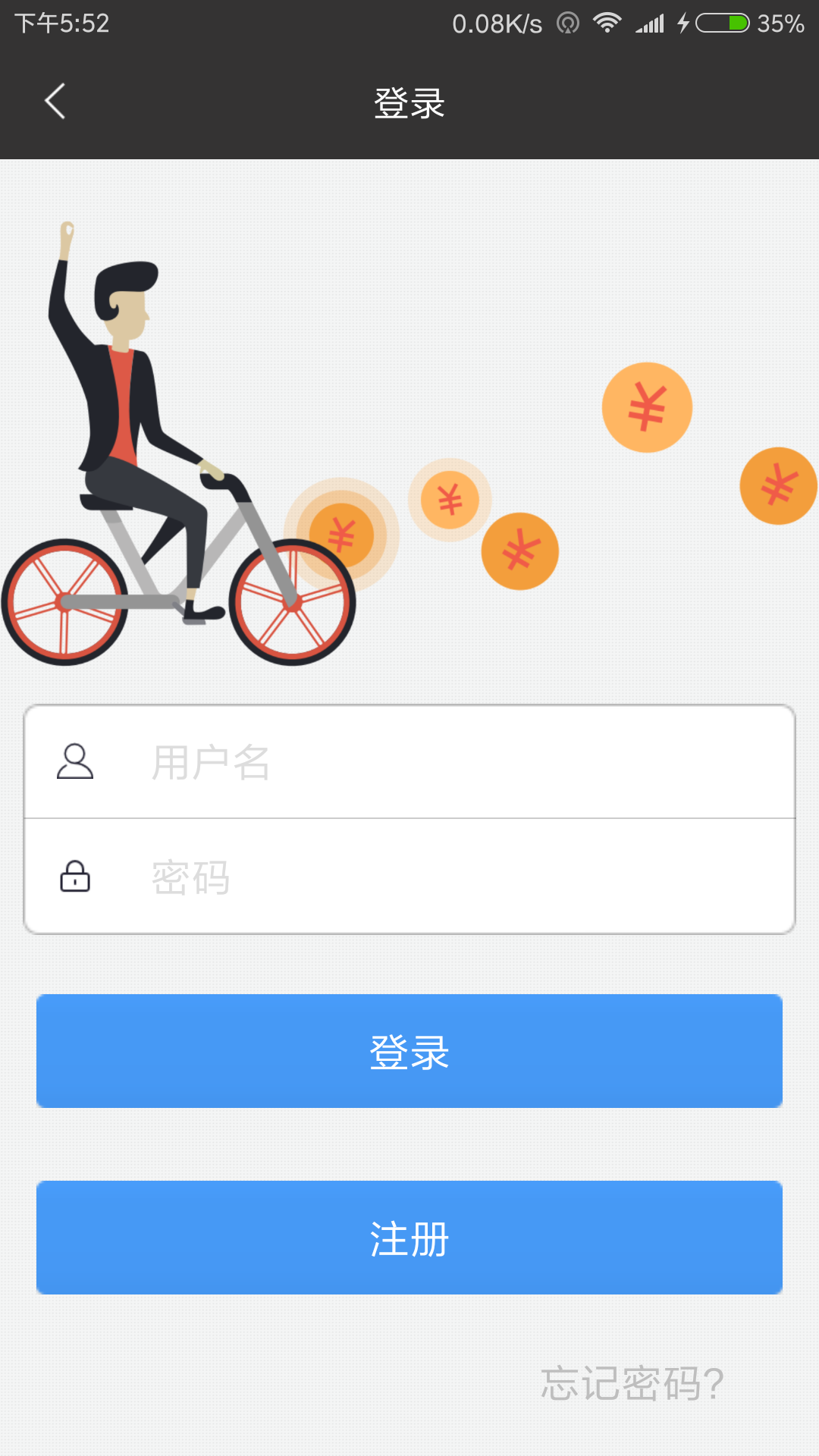


图 15 用户登录示意图

4.2.4 用户注册模块

在本模块中，用户可以输入相应的信息进行注册账号，也可以点击取消按钮或者左上角的返回按钮返回到登陆界面，点击注册后将回到登陆界面进行登陆，需要邮箱验证才能登陆成功。相关界面如下图：



图 16 用户注册示意图

4.2.5 找回密码模块

在本模块中，用户可以输入注册时的邮箱进行找回密码，点击确认之后去邮箱进行密码的更改。也可以点击左上角的返回按钮返回到登陆界面。相关界面如下图：



图 17 找回密码示意图

4.2.6 个人中心模块

在本模块中，显示了用户的头像，用户名，用户可以点击出租单车进行出租自己的单车，也可以点击我的行程查看骑行历史记录，点击设置可以设置用户的个人信息，点击退出登陆能过退出当前已经登录的用户，并返回主界面。



图 18 个人中心示意图

4.2.7 个人中心模块

4.3 单车相关功能模块实现

本模块中主要是单车相关功能的代码实现以及相关结果的显示。

4.3.1 单车显示模块

进行单车显示之前需要进行单车数据的生成，根据当前定位的数据，然后在半径为随机R的范围内生成大量的单车数据，相关代码如下：

|  |
| --- |
| private void rentAmoutBike(){  double random0=0.0,random1 = 0.0;  double varlon,varlat;  for(int i =0;i<30;i++){  random0 = (Math.random()/100000)\*1000;  random1 = (Math.random()/100000)\*1000;  random0 = Double.parseDouble(String .format("%.6f",random0));  random1 = Double.parseDouble(String .format("%.6f",random1));  varlon = longitude;  varlat = latitude;  switch(i%10){  case 1:  case 2:  case 3:  varlon = varlon-random0;  varlat = varlat-random1;  break;  case 4:  case 5:  varlon = varlon+random0;  varlat = varlat-random1;  break;  case 7:  case 8:  varlon = varlon-random0;  varlat = varlat+random1;  break;  default:  varlon = varlon+random0;  varlat = varlat+random1;  break;  }  if(varlon >100&&varlat>10){  newlocation = new BmobGeoPoint(varlon,varlat);  //bikeInfo = new BikeInfo("123456",newlocation,user,"110","山地车","我是一辆单车","20170506");  generateBikeInfo();  bikeInfo.save(this,new SaveListener(){  @Override  public void onSuccess() {  Log.e("出租成功出租成功出租成功","----");  }  @Override  public void onFailure(int i, String s) {  Log.e("出租失败出租失败出租失败",i+s);  }  });  }  }  } |
|  |

然后就是对生成的单车数据显示在基础地图中，首先需要去云端查询所有可租用的单车数据，然后根据单车数据的坐标把单车都显示出来，同时进行定位，定位的同时方向传感器确定定位的方向，相关代码如下：

|  |
| --- |
| /\*\*  \* 查询单车信息并在地图上显示单车  \*/  private void queryBikeList(){  BmobQuery<BikeInfo> query = new BmobQuery<>();  query.setLimit(1000);  //query.setCachePolicy(BmobQuery.CachePolicy.CACHE\_ELSE\_NETWORK);  query.findObjects(this, new FindListener<BikeInfo>() {  @Override  public void onSuccess(List<BikeInfo> list) {  List<OverlayOptions> markerList = new ArrayList<OverlayOptions>();  Log.e("单车总数：",list.size()+"");  for(BikeInfo info:list){  LatLng point = new LatLng(info.getLocation().getLatitude(),info.getLocation().getLongitude());  MarkerOptions option = new MarkerOptions().position(point).icon(bitmap).zIndex(0).period(10);  option.animateType(MarkerOptions.MarkerAnimateType.grow);  markerList.add(option);  //baiduMap.addOverlay(option);  }  baiduMap.addOverlays(markerList);  }  @Override  public void onError(int i, String s) {  Log.e("错误错误",i+s);  //Toast.makeText(getApplicationContext(),"加载失败",Toast.LENGTH\_LONG).show();  }  });  } |

当点击Marker时要进行步行路径导航，相关代码如下：

|  |
| --- |
| /\*\*  \*  \* 步行路径导航  \*/  @Override  public void onGetWalkingRouteResult(WalkingRouteResult result) {  if (result == null || result.error != SearchResult.ERRORNO.NO\_ERROR) {  Toast.makeText(this,"加载失败",Toast.LENGTH\_LONG).show();  }  if (result.error == SearchResult.ERRORNO.AMBIGUOUS\_ROURE\_ADDR) {  return;  }  if (result.error == SearchResult.ERRORNO.NO\_ERROR) {  for(WalkingRouteLine line:result.getRouteLines()){  walkingRouteOverlay.setData(line);  }  walkingRouteOverlay.addToMap();  walkingRouteOverlay.zoomToSpan();  }  } |

查询数据并定位后的结果如下图所示：



图 19 查询单车数据并显示

4.3.2 单车信息显示模块



第5章 总结与展望

5.1 总结

5.1.1 论文总结

本文详细的讲述了基于安卓平台租车系统的关键技术，并且结合这些技术对一个完整的租车系统进行了实现。

本文首先介绍租车系统中的相关理论，包含百度地图的相关地理位置信息获取的方法，以及租车系统中用于步行路线规划的方法，以及基于用户定位点随机生成单车坐标，并在地图上显示单车Marker。

结合上述的相关内容，本文实现了一个可以完成租车业务流程的租车系统。本文中将租车系统分为用户功能模块和单车租用功能模块。然后分别介绍相关模块的设计与实现。

本次设计的租车系统能够实现租车、骑行等功能，能够提供许多单车供用户使用，能够适应手机的多种位置，并且能够达到定位的高精确度，使用Bmob后端云数据库进行数据操作，方便快捷，学习成本低，可以轻松实现应用“云与端”的数据连通。当然，本系统仍然存在一些不足，还需要进一步的改进。

5.1.2 个人总结

本次毕业设计中，不仅对安卓平台开发的相关理论和方法有了更加深入的了解，同时对百度地图和Bmob云存储平台也有初步的认识。在系统实现过程中，从学习安卓平台下的应用编程，到安卓平台下的Service使用，相关Activity，此外也涉及到Java多线程以及线程池和定时任务等相关知识的使用。经过本次的毕业设计，对安卓平台相关开发更加熟练。同时在本次毕业设计当中，还对云数据库操作数据、百度地图进行定位和步行路径规划、手机方向传感器进行方向的确定有了更多理解，通过实现，并且将它们组合起来进行应用，进一步锻炼了自己的能力。

5.2 展望

参考文献

[1]2017年中国共享单车发展现状分析 ，http://www.askci.com/news/hlw/20170323/13560594052.shtml

[2] 郭全中.[共享单车,能飞得起来吗](http://d.wanfangdata.com.cn/Periodical/hlwjj201611003" \t "http://s.wanfangdata.com.cn/_blank) [J].《互联网经济》，2016，(11)：17-19.

[3]百度地图开放平台 ，http://lbsyun.baidu.com/

[4]Android Reference ，https://developer.android.com/reference/packages.html

[5] Bmob后端云，http://www.bmob.cn/

[6] Android官方网站，http:s//www.android.com/

[7] Android Stdio 官方网站，https://developer.android.com/studio/index.html

[8] 百度地图 API wiki.lbsyun.baidu.com/cms/androidsdk/doc/v4\_3\_0/index.htm

[9] 疯狂 Android 讲义(第三版) 李刚 电子工业出版社 2015

[10] Bmob 开发文档 http://docs.bmob.cn/data/Android/b\_developdoc/doc/index.html

[11] 百度地图开发指南 http://lbsyun.baidu.com/index.php?title=androidsdk

[12] Github 指南 https://guides.github.com/activities/hello-world/