《小禾苗留守儿童APP项目》

**详细设计**

**Version 1.0**

**2018年09月**

**版本历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 日期 | 备注 |
| 1.0.0 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[《小禾苗留守儿童APP项目》 1](#_Toc523948984)

[第一章引言 5](#_Toc523948985)

[1.1编写目的 5](#_Toc523948986)

[1.2 项目背景 5](#_Toc523948987)

[1.3 定义 6](#_Toc523948988)

[1.API 6](#_Toc523948989)

[2. SOAP 6](#_Toc523948990)

[3. Java 7](#_Toc523948991)

[4. Ubuntu 7](#_Toc523948992)

[5. SDK 8](#_Toc523948993)

[1.4 参考资料 8](#_Toc523948994)

[第二章总体设计 9](#_Toc523948995)

[2.1、小禾苗留守儿童平台逻辑架构设计 9](#_Toc523948996)

[2.2.物理架构设计 10](#_Toc523948997)

[2.3.技术架构设计 11](#_Toc523948998)

[1.View: 11](#_Toc523948999)

[2.Model: 11](#_Toc523949000)

[第三章 系统数据结构详细设计 13](#_Toc523949001)

[3.1、系统数据结构类图 13](#_Toc523949002)

[3.1.1、类图 13](#_Toc523949003)

[3.1.2、类说明 13](#_Toc523949004)

[3.2.、后台ERP相关数据表说明 22](#_Toc523949005)

[3.3、XML数据接口说明 22](#_Toc523949006)

[第四章 系统业务详细设计 24](#_Toc523949007)

[4.1、系统功能结构 24](#_Toc523949008)

[4.2、系统功能设计 25](#_Toc523949009)

[4.3、配置参数设计 25](#_Toc523949010)

[4.4、路径设计 25](#_Toc523949011)

[第五章 部署设计 25](#_Toc523949012)

[5.1、系统部署图 25](#_Toc523949013)

[5.2运行环境 26](#_Toc523949014)

[5.2.1 智能终端环境 26](#_Toc523949015)

[5.2.2客户机器环境 26](#_Toc523949016)

[5.2.3开发环境要求 26](#_Toc523949017)

## 第一章引言

### 1.1编写目的

编写本设计的目的是为了准确的阐述小禾苗留守儿童APP项目系统的具体实现思路和方法，即系统的详细架构和实现逻辑，主要包括程序系统的结构以及各层次中每个程序的设计考虑，实现方法。

### 1.2 项目背景

农村留守儿童在我国呈现出数量不断增加、问题不断突出的趋势，他们是社会上的弱势群体，正处于受教育的关键期，由于缺少父母关爱，其学习、健康、安全、道德和心理等都面临着诸多问题。长期以来，留守儿童的保护体制维持着一贯的传统，在形式与创意上难以得到突破。

随着留守儿童数量的增多，舆情事件的发生率逐渐增长，即使政府出台了关爱保护留守儿童的政策，但相关法律法规对留守儿童的保护依旧存在滞后性。

Palindromic项目组通过对其背后的根源的梳理研究，发现对于留守儿童的关怀不能时时刻刻洞察儿童的情绪、心理和身体变化，儿童的行为习惯和异常状况也不能立即发现，同时派出专门的人力监管也会出现人力资源不足、过度监管限制儿童自由发展等问题，因此团队决定运用科技的力量改变滞后的现状，最大限度降低监管的存在感，并通过实时收集信息交由核心算法分析处理来最大限度增强监管力度。我们利用从留守儿童地区收集到的数据，完善信息数据库，通过APP收录兼备理性与感性的数据，从而实时了解与掌握留守儿童的相关资讯，达成留守儿童家庭系统内部以及其与外部社会的互通，从而在根本上解决这一社会痼疾。

### 1.3 定义

#### 1.API

应用程序接口（API），，又称为应用编程接口，就是软件系统不同组成部分衔接的约定。由于近年来软件的规模日益庞大，常常需要把复杂的系统划分成小的组成部分，编程接口的设计十分重要。程序设计的实践中，编程接口的设计首先要使软件系统的职责得到合理划分。良好的接口设计可以降低系统各部分的相互依赖，提高组成单元的内聚性，降低组成单元间的耦合程度，从而提高系统的维护性和扩展性。

#### 2. SOAP

SOAP简单对象访问协议是交换数据的一种协议规范，使用在计算机网络Web服务中，交换带结构信息。SOAP为了简化网页服务器从XML数据库中提取数据时，节省去格式化页面时间，以及不同应用程序之间按照HTTP通信协议，遵从XML格式执行资料互换，使其抽象于语言实现、平台和硬件。此标准由IBM、Microsoft、UserLand和DevelopMentor在1998年共同提出，并得到IBM，莲花（Lotus），康柏（Compaq）等公司的支持，于2000年提交给万维网联盟，目前SOAP 1.1版是业界共同的标准，属于第二代的XML协定。

#### 3. Java

Java是一种广泛使用的计算机编程语言，拥有跨平台、面向对象、泛型编程的特性，广泛应用于企业级Web应用开发和移动应用开发。

#### 4. Ubuntu

Ubuntu是以桌面应用为主的Linux发行版，Ubuntu由Canonical公司发布，他们提供商业支持。它是基于自由软件，其名称来自非洲南部祖鲁语或科萨语的“ubuntu”一词，意思是“人性”、“我的存在是因为大家的存在”，是非洲传统的一种价值观。Ubuntu的开发由英国Canonical有限公司主导，南非企业家Mark Shuttleworth所创立。Canonical通过销售与Ubuntu相关的技术支持和其他服务来产生收益。Ubuntu项目公开承诺开源软件开发的原则；鼓励人们使用自由软件，研究它的运作原理，改进和分发。

#### 5. SDK

软件开发工具包（SDK）一般是一些被软件工程师用于为特定的软件包、软件框架、硬件平台、操作系统等创建应用软件的开发工具的集合。

它或许只是简单的为某个编程语言提供应用程序接口的一些文件，但也可能包括能与某种嵌入式系统通讯的复杂的硬件。一般的工具包括用于调试和其他用途的实用工具。SDK还经常包括示例代码、支持性的技术注解或者其他的为基本参考资料澄清疑点的支持文档。

### 1.4 参考资料

本项目的经核准的计划任务书或合同、上级机关的批文；

《“小禾苗”留守儿童数据可视化平台创业项目书》

属于本项目的其他已发表的文件；

《需求说明》

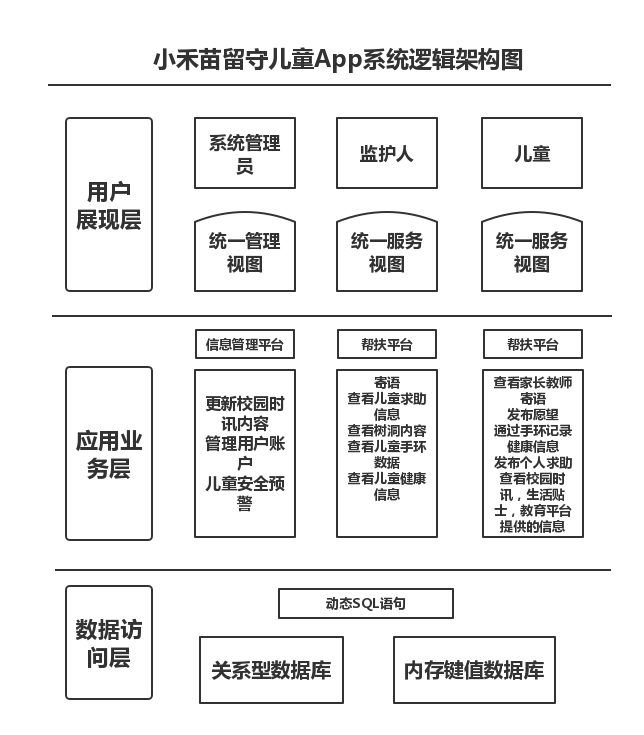
《概要设计》

本文件中各处引用到的文件资料，包括所要用到的软件开发标准。

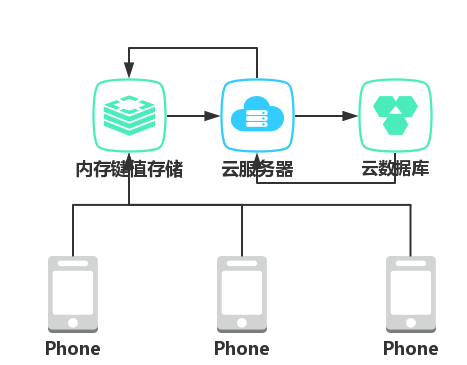
《Java语言编码规范(Java Code Conventions)》

## 第二章总体设计

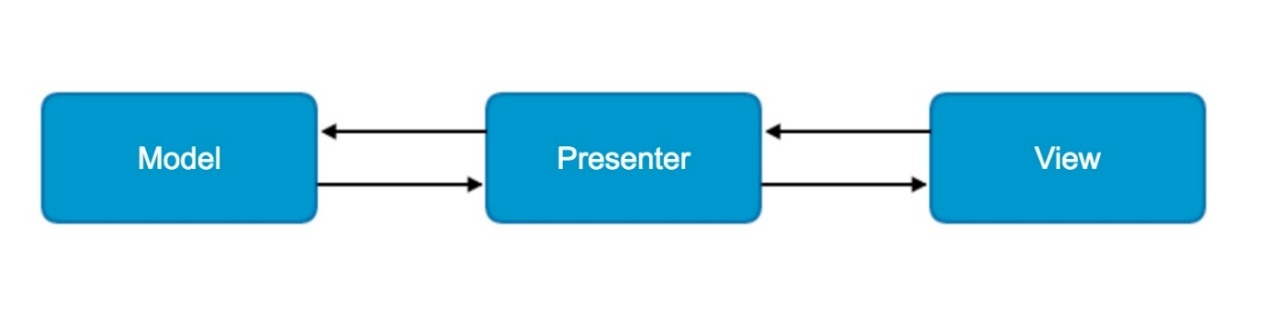
### 2.1、小禾苗留守儿童平台逻辑架构设计

****

### 2.2.物理架构设计



### 2.3.技术架构设计



对于一个客户端我们需要对它抽象出各个层次，在MVP架构中它将UI界面与数据进行隔离

1.View: 对于View层也是视图层，在View层中只负责对数据的展示，提供友好的界面与用户进行交互。在Android开发中通常将Activity或者Fragment作为View层。

2.Model: 对于Model层也是数据层。它区别于MVC架构中的Model，在这里不仅仅只是数据模型。在MVP架构中Model它负责对数据的存取操作，例如对数据库的读写，网络的数据的请求等。

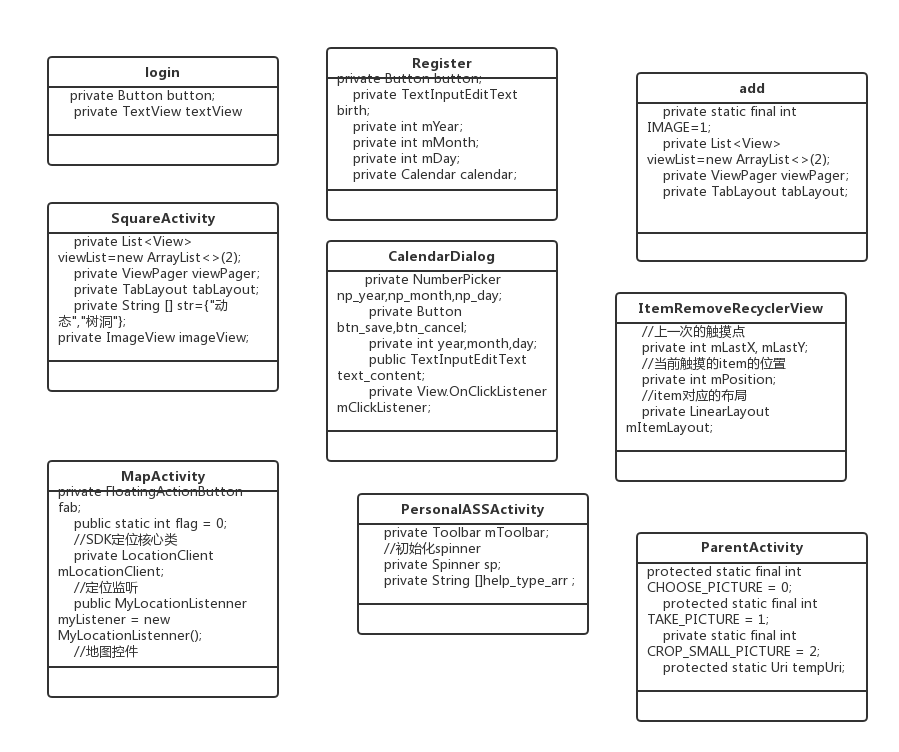
**3.Presenter:**对于Presenter层他是连接View层与Model层的桥梁并对业务逻辑进行处理。在MVP架构中Model与View无法直接进行交互。所以在Presenter层它会从Model层获得所需要的数据，进行一些适当的处理后交由View层进行显示。这样通过Presenter将View与Model进行隔离，使得View和Model之间不存在耦合，同时也将业务逻辑从View中抽离。

在MVP架构中将这三层分别抽象到各自的接口当中。通过接口将层次之间进行隔离，而Presenter对View和Model的相互依赖也是依赖于各自的接口。这点符合了接口隔离原则，也正是面向接口编程。在Presenter层中包含了一个View接口，并且依赖于Model接口，从而将Model层与View层联系在一起。而对于View层会持有一个Presenter成员变量并且只保留对Presenter接口的调用，具体业务逻辑全部交由Presenter接口实现类中处理。

## 第三章 系统数据结构详细设计

### 3.1、系统数据结构类图

#### 3.1.1、类图



#### 3.1.2、类说明

LogIn //登录

private Button button;

private TextView textView;

Register //注册

private Button button;

private TextInputEditText birth;

private int mYear;

private int mMonth;

private int mDay;

private Calendar calendar;

add //广场中添加动态类

private static final int IMAGE=1;

private List<View> viewList=new ArrayList<>(2);

private ViewPager viewPager;

private TabLayout tabLayout;

private String [] str={"动态","树洞"};

private TextView textView;

private ImageView imageView;

private Button button\_addActivties;

private Button button\_addTree;//photo

private Button button\_location;//location

SquareActivity //广场

private List<View> viewList=new ArrayList<>(2);

private ViewPager viewPager;

private TabLayout tabLayout;

private String [] str={"动态","树洞"};

private ImageView imageView;

public class MyViewHolder extends RecyclerView.ViewHolder {

public TextView tv3;

public TextView delete;

public LinearLayout layout;

}

public class MyAdapter extends RecyclerView.Adapter<MyViewHolder> {

private ArrayList<String> list1;

private OnItemClickListener mOnItemClickListener;

private LayoutInflater mInflater;

}

public class CalendarDialog extends Dialog implements

NumberPicker.OnValueChangeListener, NumberPicker.Formatter, NumberPicker.OnScrollListener {

/\*\*

\* 上下文对象 \*

\*/

Activity context;

private NumberPicker np\_year,np\_month,np\_day;

private Button btn\_save,btn\_cancel;

private int year,month,day;

public TextInputEditText text\_content;

private View.OnClickListener mClickListener;

}

public class CalendarActivity extends AppCompatActivity implements AdapterView.OnItemClickListener,OnItemClickListener {

private Toolbar mToolbar;

private CalendarDialog cd;

private ItemRemoveRecyclerView rv1;

private ArrayList<String> list1;

private SwipeRefreshLayout srl;

private CalendarView mcv;

private FloatingActionButton mFAB;

private int year,month,day;

private int data\_count = 0;

private String notice = null;

private Calendar ca = Calendar.getInstance();

private MyAdapter adapter;

private String arrStr[] ;

}

public class SpacesItemDecoration extends RecyclerView.ItemDecoration {

private int space;

public SpacesItemDecoration(int space) {

this.space = space;

}

}

public class ItemRemoveRecyclerView extends RecyclerView {

private Context mContext;

//上一次的触摸点

private int mLastX, mLastY;

//当前触摸的item的位置

private int mPosition;

//item对应的布局

private LinearLayout mItemLayout;

//删除按钮

private TextView mDelete;

//最大滑动距离(即删除按钮的宽度)

private int mMaxLength;

//是否在垂直滑动列表

private boolean isDragging;

//item是在否跟随手指移动

private boolean isItemMoving;

//item是否开始自动滑动

private boolean isStartScroll;

//删除按钮状态 0：关闭 1：将要关闭 2：将要打开 3：打开

private int mDeleteBtnState;

//检测手指在滑动过程中的速度

private VelocityTracker mVelocityTracker;

private Scroller mScroller;

private OnItemClickListener mListener;

}

public interface OnItemClickListener {

void ItemClick(View v, int pos);

void onDeleteClick(int position);

}

public class MapActivity extends AppCompatActivity {

private FloatingActionButton fab;

public static int flag = 0;

//SDK定位核心类

private LocationClient mLocationClient;

//定位监听

public MyLocationListenner myListener = new MyLocationListenner();

//地图控件

public MapView mMapView = null;

//百度地图对象

public BaiduMap mBaiduMap;

//定位坐标

private LatLng latLngF;

private LatLng latLngM;

private LatLng latLng;

//判断首次定位

boolean isFirstLoc = true;

private static final int REQUEST\_CODE\_WRITE\_SETTINGS =1;

}

public class PersonalASSActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {

//初始化toolbar

private Toolbar mToolbar;

//初始化spinner

private Spinner sp;

private String []help\_type\_arr ;

}

public class MyASSActivity extends AppCompatActivity {

//初始化toolbar

private Toolbar mToolbar;

}

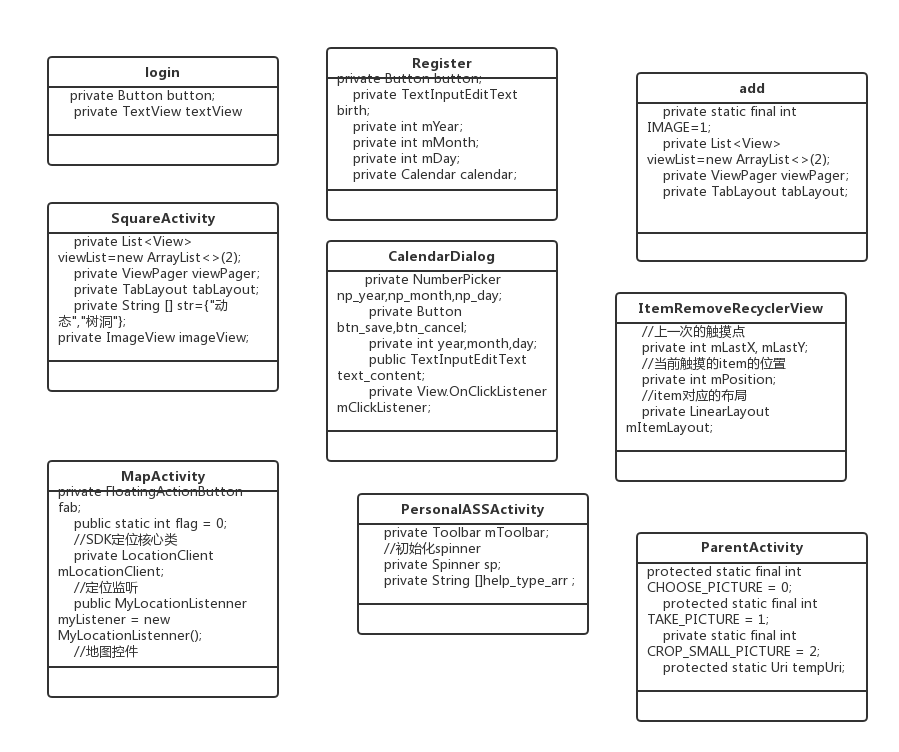
public class ParentActivity extends AppCompatActivity

{

SupportMapFragment mapViewD,mapViewM;

private static final String TAG = "ParentActivity";

private CircleImageView iv\_dad;



private CircleImageView iv\_mom;

private static int count;

private Toolbar mToolbar;

}

public class MapFragment extends FragmentActivity {

//定位坐标

private LatLng latLngD ;

private LatLng latLngM

}

public static String savePhoto(Bitmap photoBitmap, String path,

String photoName) {

String localPath ;

Bitmap output;

Canvas canvas ;

final int color ;

final Paint paint ;

final Rect src ;

final Rect dst;

final RectF rectF ;

}

public class CircleImageView extends ImageView {

Path path;

public PaintFlagsDrawFilter mPaintFlagsDrawFilter;// 毛边过滤

Paint paint;

}

DreamListActivity extends AppCompatActivity//愿望清单活动类

Private RecyclerDream rvDream; //愿望清单的RecyclerView

Private ImageView back; //返回图标

MessageActivity extends Activity //寄语板活动类

Private ImageView back; //返回图标

DirectActivity extends AppCompatActivity //打开APP时引导类

Private void date() //判断应用是否是安装后第一次打开

PreviewActivity extends Activity //第一次打开APP时候的Viewpager的轮播页

private ViewPager vpager\_one; //ViewPager

private ArrayList<View> aList; //ViewPager中添加内容的数组

private MyPagerAdapter mAdapter; //自定义适配器

private Button startButton; //跳转到登录页面的按钮

private ImageView myFirst;

private ImageView mySecond;

private ImageView myThird; //加载到ViewPager中的图片

InitialActivity extends Activity //用于宣传的2秒自动跳转到首页活动类

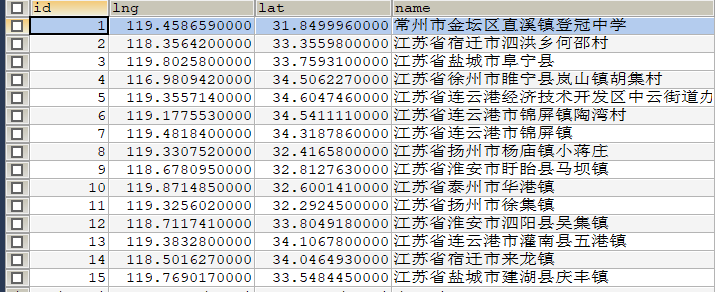
private Button skip; //跳过按钮

StartActivity extends AppCompatActivity /首页类

private Gridview grid\_view //首页图标排列

### 3.2.、后台ERP相关数据表说明





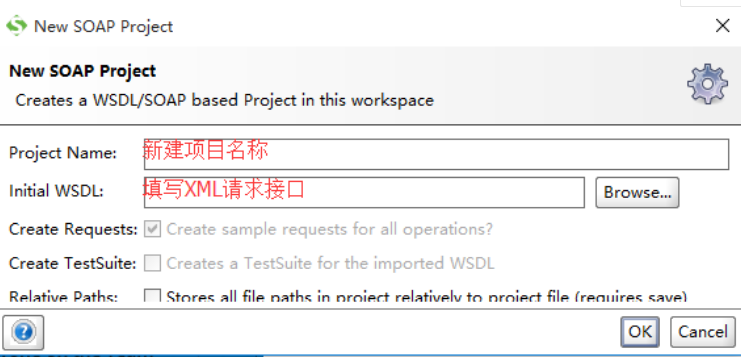
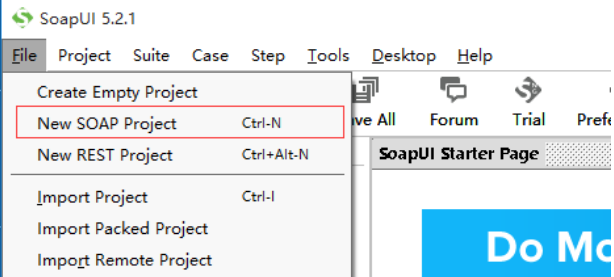
### 3.3、XML数据接口说明

（1）摘要

在运维开发过程中，经常要调用第三方产品数据，这时候就需要通过API访问接口，通常主流的API接口有RESTFUL API和XML API。RESTFUL API我们接触得比较频繁，使用起来也比较简单。本文主要通过XML接口的使用实例大概介绍如何调用XML API

（2）接口请求工具

Soap是一种简单的基于Xml的协议，它使应用程序通过Http来交换信息。通常请求XMl接口需要用到SOAPUI这个开源的测试工具。



## 第四章 系统业务详细设计

### 4.1、系统功能结构

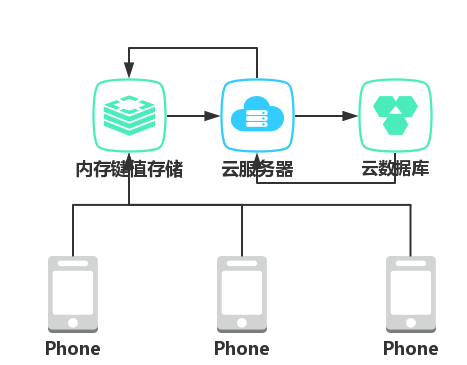
### 4.2、系统功能设计

### 4.3、配置参数设计

### 4.4、路径设计

## 第五章 部署设计

### 5.1、系统部署图



### 5.2运行环境

#### 5.2.1 智能终端环境

安卓智能手机

#### 5.2.2客户机器环境

1.高通骁龙801或以上微处理器（CPU）；

2.Android 8.0或以上操作系统；

#### 5.2.3开发环境要求

|  |  |  |
| --- | --- | --- |
| **项目** | **名称** | **版本** |
| 开发平台 | Windows | 10 |
| 开发工具 | Android studio，Intellij，Project studio | Android studio3.1.4 ; Intellij 3.5;  Project studio 1.6 |
| 代码管理工具 | Git | 2.18.0 |
| 开发环境 | SDK | 24.4.1 |