# 移动应用开发设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修改人 | 学号 | 修改日期 |
| V1.0 | 林志超 | 20T0209069 | 2022年6月20日 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[移动应用开发设计说明书 1](#_Toc106634670)

[一、 目的 3](#_Toc106634671)

[二、 产品介绍 3](#_Toc106634672)

[产品名称： 3](#_Toc106634673)

[运行环境： 3](#_Toc106634674)

[服务人群： 3](#_Toc106634675)

[三、 需求分析 3](#_Toc106634676)

[四、 基本设计 3](#_Toc106634677)

[理念： 3](#_Toc106634678)

[服务端（后端）： 3](#_Toc106634679)

[移动端（前端）： 4](#_Toc106634680)

[五、 详细设计 4](#_Toc106634681)

[用户界面： 4](#_Toc106634682)

[基本代码： 5](#_Toc106634683)

[六、 总结与反思 8](#_Toc106634684)

## 目的

本说明书是移动应用开发课程的软件设计说明书，本说明书用于指导软件设计思想，让开发人员能方便快捷的进行移动应用开发。

## 产品介绍

### 产品名称：

彩云待办（ic-todo-list）；

### 运行环境：

基于Android操作系统的移动电话，智能穿戴设备，智能电视等；

### 服务人群：

适用各年龄阶段所有人员的移动应用软件。

## 需求分析

当今时代离不开云，但是当前的待办基本都是离线但是当前的待办基本都是离线代办，效率低下，无法在多端同步。

## 基本设计

### 理念：

本软件的基本设计思想：采用前后端分离的设计方式。本设计模式有助于帮助项目维护。

### 服务端（后端）：

后端使用PHP语言实现，设置了六个接口——登录、注册、列表增加、列表修改、列表删除以及列表查询。

六个接口用于操作MySQL数据库及数据的增、删、改、查。

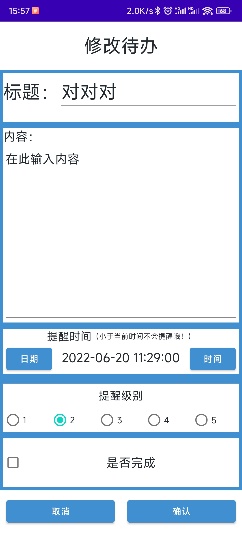
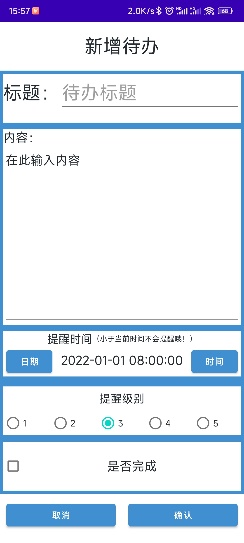
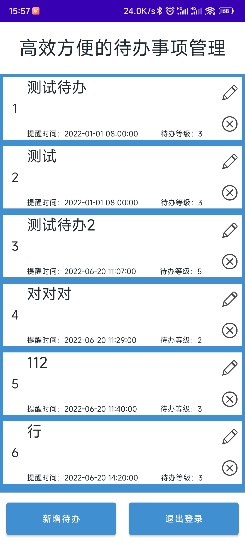
### 移动端（前端）：

前端使用AndroidStudoi开发，设计了四个页面——登录页面、注册页面、代办列表页面以及待办列表详情增、删、改、查。

## 详细设计

### 用户界面：

用户打开软件后进入主界面。



用户没有账号密码，点击【没有账号？】进入注册页面；

成功注册账号后，输入账号密码进入系统主页面；

进入系统主页面展示代办列表；

点击一条待办，可以打开待办详情；

点击新增代办，也可打开待办详情；



在主页面可以删除代办。

新增代办以后可以进行设置提醒。当到达提醒时间，系统弹出全局弹框提醒用户。

### 基本代码：

由于篇幅有限，这里只展示重要且核心的代码块。

服务内新建提醒代码：

manager = (AlarmManager) getSystemService(*ALARM\_SERVICE*);  
long time = intent.getLongExtra("Time", 0);  
Intent it = new Intent(this, AlarmReceiver.class);  
pi = PendingIntent.*getBroadcast*(this, 0, it, PendingIntent.*FLAG\_IMMUTABLE*);  
manager.set(AlarmManager.*RTC\_WAKEUP*, time, pi);  
number++;  
Log.*d*("res", "服务开启快结束了" + number);  
return super.onStartCommand(intent, flags, startId);

列表适配器代码：

@NonNull  
@Override  
public RecyclerView.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 if (viewType == TYPE\_EMPTY) {  
 View itemView = LayoutInflater.*from*(context).inflate(R.layout.*item\_isnull*, parent, false);  
 return new RecyclerView.ViewHolder(itemView) {  
 };  
 }  
 View itemView = LayoutInflater.*from*(context).inflate(R.layout.*item\_list*, parent, false);  
 return new MyViewHolder(itemView);  
}

列表请求接口代码：

public void onClick(DialogInterface dialog, int arg1) {  
 // *TODO Auto-generated method stub* dialog.dismiss();  
 OkHttpClient okHttpClient = new OkHttpClient();  
 RequestBody requestBody = new FormBody.Builder()  
 .add("lid", list.get(itemnums).getLid())  
 .build();  
 Request request = new Request.Builder()  
 .addHeader("content-type", "application/json")  
 .url("http://8.142.188.54/ic\_todo\_list/api/listdel.php")  
 .post(requestBody)  
 .build();  
 okHttpClient.newCall(request).enqueue(new Callback() {  
 @Override  
 public void onFailure(@NonNull Call call, @NonNull IOException e) {  
 Log.*d*("res", "连接失败：" + e.toString());  
 }  
  
 @Override  
 public void onResponse(@NonNull Call call, @NonNull Response response) throws IOException {  
 String res = response.body().string();  
 Log.*d*("res", "连接成功：" + res);  
 Gson gson = new Gson();  
 ListSimpleData listDelData = gson.fromJson(res, ListSimpleData.class);  
 }  
 });  
 list.remove(itemnums);  
 notifyItemRemoved(itemnums);  
 }  
 });  
 builder.setNegativeButton("取消", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 // *TODO Auto-generated method stub* dialog.dismiss();  
 }  
 });  
 builder.create().show();  
}

项目配置代码：

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="pro.sbmn.android.cloudstodo"  
 android:versionName="1.0.1">  
 <!--网络权限-->  
 <uses-permission android:name="android.permission.INTERNET" />  
 <uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />  
 <!--全局dialog弹窗权限-->  
 <uses-permission android:name="android.permission.SYSTEM\_ALERT\_WINDOW" />  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:networkSecurityConfig="@xml/networdconfig"  
 android:roundIcon="@mipmap/ic\_launcher"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.CloudsToDo"  
 tools:targetApi="31">  
 <meta-data  
 android:name="com.google.android.actions"  
 android:resource="@xml/networdconfig" />  
  
 <activity  
 android:name=".Activity.SigninActivity"  
 android:exported="true"  
 android:launchMode="singleTask"  
 android:screenOrientation="portrait">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity  
 android:name=".Activity.SignupActivity"  
 android:exported="true"  
 android:launchMode="singleTask"  
 android:screenOrientation="portrait" />  
 <activity  
 android:name=".Activity.MainActivity"  
 android:exported="true"  
 android:launchMode="standard"  
 android:screenOrientation="portrait" />  
 <activity  
 android:name=".Activity.ListDetailActivity"  
 android:exported="true"  
 android:launchMode="singleTask"  
 android:screenOrientation="portrait" />  
  
 <receiver android:name=".Alarms.AlarmReceiver" />  
 <service android:name=".Alarms.AlarmService" />  
 </application>  
</manifest>

## 总结与反思

本系统中我们首次采用了mvc分层次模式架构，极大的方便了系统的维护，除此之外，项目的结构逻辑合理，清晰简短，大大缩短了我们的开发周期。让我们的项目变得稳定可靠且易运行。