

《智能驾驶助手 APP》 项目指导书 —— 车机版

V 1.1

王治强编写

大连东软信息学院
智能与电子工程学院
电子信息工程系

2020 年 7 月

目 录

| | |
|------------------|----|
| 1 项目设计目的..... | 3 |
| 2 内容及要求..... | 3 |
| 2.1 内容概述..... | 3 |
| 2.2 基本功能要求..... | 3 |
| 3 程序开发要求..... | 8 |
| 3.1 开发技术要求..... | 8 |
| 3.2 开发工具要求..... | 9 |
| 3.3 过程管理要求..... | 9 |
| 3.4 开发成果物要求..... | 9 |
| 4 开发周期与进度安排..... | 10 |
| 5 提供开发资料..... | 11 |
| 5.1 资源一览..... | 11 |
| 5.2 关键技术..... | 11 |
| 6 开发阶段..... | 13 |
| 6.1 需求分析..... | 13 |
| 6.2 构想设计..... | 13 |
| 6.2 详细设计..... | 14 |
| 6.3 编码调试..... | 14 |
| 7 系统测试..... | 15 |
| 8 项目管理..... | 16 |
| 9 项目考核评价..... | 16 |

1 项目设计目的

面向企业人才需求，提高Java语言和安卓开发技能，锻炼分析解决问题能力，训练项目设计、开发、测试等完整产品开发过程。培养学生QCD（质量、成本、期限）意识，学习产品开发的过程管理知识，训练创新与开发能力。

2 内容及要求

2.1 内容概述

面向智能驾驶助手主题，开发一款具备导航、音乐播放、蓝牙通信等功能的APP软件。该软件采用普通安卓手机作为运行载体，以横屏使用方式，模拟汽车中控的导航、音响系统而设定开发目标。包含主界面菜单功能、导航功能、汽车状态模拟、音乐播放、语音助手、系统设置等功能。

关键技术上涵盖安卓四大组建、RecyclerView等常用控件、Intent、多进程、线程使用、蓝牙通信、数据存储等。

项目实现过程包含需求分析、程序设计、编码实现、测试、文档编写。

以下文档中UI图片仅作参考设计参考，具体式样由各个开发组自行设计。



2.2 基本功能要求

- (1) 主界面：显示天气状态，根据GPS位置信息自动获取所在城市天气，可以手动设置固定显示某个城市天气，定时刷新天气信息（比如，每隔1小时）；

显示当前日期时间；显示其他功能菜单；语音助手开关；手机连接状态等等。



- (2) 导航功能：调用百度或者高德导航SDK实现简易地图导航功能，可以在地图显示当前位置。或者intent直接调用百度地图进行导航。最好直接调用SDK，可以获取行驶距离，从而产生动态数据。



- (3) 音乐播放功能：可以播放本地或者网络音乐，最好是播放网络音乐，比如调用网易云音乐api接口。可以搜索音乐，添加收藏。可以通过蓝牙/Wifi连接手机，从手机app同步播放列表（比如设置按钮，覆盖手机端收藏列表，手机端也可以设置同样功能，覆盖车机端收藏列表）。



- (4) 语音助手：使用百度api或者其他语音识别api接口，实现语音转文字，从而实现固定的命令执行。比如：“打开导航”、“播放音乐”、“天气预报”、“当前时间”等等。



- (5) 汽车状态显示：显示汽车状态，车窗开关状态、启动停止状态、运行速度、剩余电量（用手机电量替代）、可行驶距离、行驶里程等等。可以手动改变状态，同时也接受手机端控制。（控制内容可自由发挥）



- (6) 行驶记录：汽车被启动后，开始生成一条行驶数据，记录当前位置，当前里程，然后根据定位信息，每隔1分钟或者计算车移动距离，如果大于100米记录一个路径点，用于恢复行驶路径，停止时记录终止位置。用于把信息同步给手机端，查看汽车行驶记录。车机端有记录列表查看页面，理想状态是点击一条记录后，可以打开地图在地图上显示行程轨迹。（本功能需要在后台运行，不影响音乐播放等其他功能，这里是难点！）
- (7) 系统设置功能：设置蓝牙列表，连接蓝牙音箱，打开车内热点等



2.3 车机与手机 APP 通信互动

车机与手机之间可以通过蓝牙或者Wifi进行通信。通信方式上，车机作为服务端，手机主动连接车机。在使用Wifi情况下，车机通过Wifi热点功能开启热点

功能，手机连接到车机的热点与车机连接。蓝牙通信情况下，由手机搜索车机的蓝牙设备，并主动连接到车机与车机建立通信连接。两种通信方式任选其一实现即可。（由于使用蓝牙通信需要两部真实手机才能完成开发，使用Wifi可以使用虚拟机模拟实现，因此推荐使用Wifi通信）

与数据通信相关的功能有以下几方面：

- （1） 手机控制车机：将手机当作汽车的遥控器，可以实现启动停止、车门上锁开锁、后备箱开门关门等。

在实现上，可以是手机单方向发送控制指令数据给车机，车机接收到之后改变汽车状态数据，并在UI上改变显示状态。手机端也相应改变UI显示状态。

- （2） 手机寻找车机：模拟在停车场车主找不到汽车的情况下，使用手机寻找车机功能，前提是手机已连接上车机的Wifi热点，手机发送位置请求指令给车机，车机收到指令后返回当前汽车位置信息给手机。手机收到车机位置信息后，调用地图显示界面，在地图上标记出车主当前位置和汽车当前位置，可能的话可以标记导航路径。

- （3） 音乐列表同步：因为手机和车机端都有音乐播放功能，任意一端音乐列表发生变更的情况下，都可以同步到另一端。为实现简单，只要求实现增量同步。即，如果车机的音乐列表多一条音乐，则把多的一条音乐同步到手机端，反之亦然。车机和手机端都提供列表同步按钮。

实现思路：将本地列表逐条发送给另一端，另一端接收到之后，先查询数据库，如果数据库中没有，则添加，如果有则忽略即可。

- （4） 汽车行驶记录同步：车机端可以记录汽车的行驶路径数据，并存放在数据库中，当手机连接到车机，车机会把数据库中的数据同步到手机端，数据发送原理与音乐列表类似，在建立连接后，将数据逐条发送，手机端接收到数据后，查询数据库数据是否存在，如果不存在则添加数据库，如果已经存在则忽略。

数据传输方法上，使用Json将数据序列化之后，通过socket进行发送，接收到数据后，再将Json串解析成对象，进行数据操作。由于发送的数据种类比较多，需要设计数据指令，标记将要发送的是什么类型数据，从而可以分门别类进行数

据转换和处理。



3 程序开发要求

3.1 开发技术要求

- (1) Java语法强化：封装继承、接口、常用容器类等
- (2) 安卓四大组件工作原理及应用
- (3) 常用控件：TextView、Button、ImageView、Switch、RecyclerView (ListView)、ViewPager、WebView等
- (4) 布局管理：ConstraintLayout、LinerLayout等
- (5) 系统常用Intent、Service
- (6) 进程与线程工作原理及应用Thread、Runnable、AsyncTask
- (7) 通信：互联网访问Http等、Wifi、蓝牙（音频、控制）
- (8) AIDL跨进程通信
- (9) 数据存储：SharedPreferences、SQLite使用



3.2 开发工具要求

- (1) Android studio 使用技巧
- (2) Git/github/Gerrit: 版本管理
- (3) Astah: 时序图画法
- (4) Excel/PowerPoint/Word Office 软件: 流程图等其他设计图画法以及文档编辑方法

3.3 过程管理要求

开发过程管理采用简化的 CMMI 管理过程, 需要包含如下过程:

- ☐ 构想设计
- ☐ 详细设计
- ☐ 编码实现
- ☐ 系统测试

3.4 开发成果物要求

- ☐ 需求式样书
- ☐ 概要设计书
- ☐ 项目计划书
- ☐ 详细设计书
- ☐ 测试式样书

- 代码
- 课题管理表
- bug 管理表
- 发表资料

4 开发周期与进度安排

项目开发周期：四周（20工作日）

开发起止日期：8月10日~9月4日



根据开发过程管理要求，划分为如下个阶段：

第一阶段（第一周时间）：需求分析与构想设计

根据项目指导书正确理解项目需求，并提出创新功能式样。

制作需求式样书，对需求进行详细定义。

对需求进行分析，做项目系统设计，并完成系统设计书。

根据系统设计做项目开发计划书。

第二阶段（第二周时间）：项目详细设计

根据系统设计，编写详细设计文档。

包括：UI设计、流程图设计等。

对设计中的难题问题，记录问题管理表。

对详细设计的成果物需要进行评审，并修改设计缺陷。

第三阶段（第三周时间）：项目编码实现

根据设计进行编码实现。并进行程序调试。

对编码中的问题，记录问题管理表。

第四阶段（第四周时间）：程序系统测试，发布程序

编写测试用例，进行程序测试，记录测试结果。

对存在的bug，记录bug管理表。

修改测试bug，并进行回归测试。

最后发布程序，填写发布管理表。

第五阶段（第四周时间）：对项目进行总结，进行项目发表答辩。

对程序文档进行整理，程序代码整理，成果物打包上交。

完成发表ppt。

5 提供开发资料

5.1 资源一览

本项目开发提供以下资源：

- （1） 相关安卓知识点讲解
- （2） 相关工具使用方法讲解
- （3） 开发流程、设计方法和管理文档编写方法讲解
- （4） 安卓相关技术点Demo程序
- （5） 提供各个阶段开发文档模板。

5.2 关键技术

- （1） ViewModel/LiveData→视图数据架构

使用ViewModel框架可以解决页面视图与数据解耦，便于程序设计模块化，是很好的设计思想。可以参考谷歌Jetpack框架，优化软件设计。同时，使用ViewModel可以确保数据安全，在手机横竖屏切换的时候，数据也不会丢失。

参考网络学习资源。

- （2） RecyclerView (ListView) →音乐列表/蓝牙设备列表等

在音乐列表、蓝牙设备列表等场合都需要使用RecyclerView 或者ListView（有点过时了），尽量使用RecyclerView。

提供课程Demo。

- （3） ViewPager+BottomNavigationView+Fragment→底部导航方式

使用BottomNavigationView 、ViewPager2与Fragment，可以很方便实现

类似微信底边导航的效果。

提供课程Demo。

- (4) Intent→页面跳转、数据传递、拨打电话等

Intent是安卓基础知识，在安卓课程中都学习过，面对具体需求的时候可以从网上找到相关Action、参考代码进行使用。

- (5) Service→音乐播放、蓝牙/Wifi后台通信

Service是安卓基础知识，属于安卓四大组件之一。在解决Activity和Service通信的时候引入AIDL技术。

参考AIDL课程Demo。

- (6) AIDL跨进程通信→音乐播放/蓝牙通信/Wifi通信等

Android Interface Definition Language (AIDL)

提供课程Demo。

- (7) 进程与线程→蓝牙/Wifi通信、数据库操作、网络访问等

使用Thread、Runnable、AsyncTask等相关类

- (8) 互联网访问Http→在线音乐

使用Volley库方便实现http访问

参考网络学习资源。

- (9) 蓝牙→音频输出、数据传输

使用BluetoothAdapter、BluetoothDevice、BluetoothServerSocket、BluetoothSocket相关类

提供课程Demo。

- (10) SharedPreferences→用户设置

保存用户配置信息。

提供课程Demo。

- (11) SQLite→音乐/汽车数据存储

数据库使用知识，使用SQLiteOpenHelper、SQLiteDatabase类实现数据库增删改查。

提供课程Demo。

- (12) Json→序列化、数据传输

类数据序列化方法。引入Json库，解析Json和生成Json字符串。

参考网络学习资源。

(13) WifiManger→Wifi热点设置

使用WifiManger可以打开、关闭热点，设置热点名称、密码等，在车机上不应该用手机自身功能去设置。

参考网址：<http://www.qs77.net/?post=2>

(14) Tcp Socket→网络通信

局域网环境通信，收发数据。

提供课程Demo。

6 开发阶段

6.1 需求分析

这个阶段主要完成对产品的功能式样的明确与定义。根据项目基本要求，和自主创新功能，书写项目需求文档。

每条功能点尤其是创新功能点，一定要考察可实现性，进行充分的调查后方可添加到需求式样书中。技术准备上开始解析代码，安装开发环境，了解开发流程。

文档要求：参见《需求式样书（模板）.xlsx》

要求细化各功能点内容，逐条目罗列。尽可能详细描述每个功能点，功能点要分大类和小类，注意分类有序填写。

思路指导：

6.2 构想设计

内容说明：

这个阶段又叫做概要设计或者系统设计，这个阶段主要完成概要设计书。根据需求中的功能点，对系统整体设计做分析与实现方案的明确。对整个系统进行模块划分，软件层次划分。原则是将大的模块按照功能，结构，人员分工等方面进行细分。遵循“低耦合，高内聚”的原则。

对于不确定的技术点，一定要进行调查分析，落实到切实可行的实现方案。可以用结构图，时序图，状态迁移图等方式进行设计表达。对会用到的全局变量等数据进行设计论证。对每个模块的接口进行明确与定义。

文档要求：

参见《概要设计书（模板）.docx》

6.2 详细设计

内容说明：

详细设计是对构想设计的进一步细化与面向编码的可实现，可操作的细节设计。要细化到主要函数级别，对函数的流程图进行定义。

在这个阶段，有部分重难点小功能，可以进行编码验证。涉及到通信协议的，需要弄清楚通信协议的规范和流程，相关库的使用方法。

细节内容，可以用流程图、时序图表达，如果需要调用库函数，要说明调用什么接口，参数如何传递。通信的数据，可以列表格说明每种数据含义。

文档要求：

参见《详细设计书（模板）.docx》

思路指导：

6.3 编码调试

在详细设计完成后进行编码调试阶段。

编码方式：

按照自顶向下的方式进行实现，根据详细设计书进行编码。先由一个技术负责人创建出程序框架，把需要的大部分文件创建出来，然后按照提前分配好的分工进行分头编码。每个人担当部分都完成后，再进行代码合并，调试。

编码注意点：

严格按照详细设计书中规定好的方法名进行编码，为了调试方便，要对重要的数据信息添加打印。

对发现详细设计的错误需要及时提出，并修改详细设计。

调试方法：

单体测试：在每个人自己的编码过程中，完成一个小功能点的时候，需要自己进行单体测试，确认功能的可用性。但是在没有别的模块配合的情况下，需要自己准备测试条件。

要求共通功能的负责人需要尽量早的完成模块功能，并提供给其余组员使用。比如主页面切换逻辑框架可以尽早搭建，便于成员分工完成各自的模块详细开发。

结合调试：个人负责的自功能确认好后，提交给程序合并的负责人进行合并，并整体运行调试，按照系统启动运行顺序逐步结合，结合一部分确认一部分，不要所有的都合并完后再调试，容易造成程序一下子不能运行，却找不到原因，而引起回退返工。

7 系统测试

在结合调试结束后进行系统测试阶段。

在进入系统测试之前，需要完成系统测试式样书的编写。这个工作可以在编码调试阶段的后期进行。编码调试阶段后期，由于主要调试工作集中在几名组员身上，可以由空闲的组员进行测试式样书的编写。测试式样书编写，一定要结合需求式样书进行编写。对所有的功能和非功能需求进行测试条目设计。

设计测试条目，不但要考虑正常操纵的测试内容，还要考虑异常甚至有破坏性的测试测试内容。

测试步骤：

按照系统测试式样书进行系统测试，将测试结果记录测试式样书。

对测试有问题的条目，一定要重点记录现象以及复现方法。可以进行截图等方式进行记录。

在修改完bug后，需要进行回归测试。

对测试出的bug需要记录bug管理表。

Bug调试方法：

要充分利用打印log的方式进行bug调试。

8 项目管理

项目管理包括多方面的内容：进度管理，需求管理，问题管理，bug管理，风险管理，质量管理，代码版本控制管理，人员管理等，所有与项目推进有关的内容。

在本次训练项目中，项目组长需要着重管理以下几方面：

- 进度管理：每个阶段应该做什么？什么时间做完？谁负责做？这些问题需要经常进行确认和管理。要求每天进行口头日报确认进度，每周填写项目周报。
- 问题管理：技术难题，资源不足等所有项目相关问题，都需要记录问题管理表。项目组长需要对管理表中问题指定解决的人选，并跟踪解决进度。
- 质量管理：为了确保软件质量，所有关键节点都需要设置评审会，对成果物质量进行把关。评审会上指出的问题，都需要记录并逐条修改，对所有问题需要有跟踪，无遗漏。
- 代码版本管理：本次开发使用Git作为版本控制工具，为了实现小组合作与代码review功能，使用Github作为代码仓库服务器。在代码与文档成果物管理方面，需要对版本管理进行操作流程和版本控制的规范，并做好备份工作。保证代码管理上不出现混乱。

9 项目考核评价

项目考核评价分为形成性考核和终结性考核。

形成性与终结性成绩各占50%，总分为100分。

形成性考核：

- 考核内容：技术能力，过程管理能力，项目管理沟通能力
- 考核对象：个人
- 考核方式：项目组长对组员评价，指导老师对组长评价

详细内容请参见：《形成性考核标准》

终结性考核：

- 考核内容：程序运行，项目文档，项目答辩

- 考核对象：项目组
- 考核方式：项目发表答辩会，通过评委进行打分评定。

详细内容请参见：《终结性考核标准》