



毕业设计论文

基于安卓的知心助理 APP 的开发

教学单位：计算机学院

专业名称：软件工程（软件开发技术）

学 号：2014030402011

学生姓名：杜敏

指导教师：梁瑞仕（副教授）

指导单位：计算机学院

完成时间：2018 年 4 月 30 日

电子科技大学中山学院教务处制发

基于安卓的知心助理 APP 的开发

摘要

目前安卓开发正对我们的生活造成巨大的影响，我们的生活已经离不开如微信、QQ、微博等应用。这些应用极大的便利了我们的生活，为我们的工作生活带来了很多的乐趣，丰富和充实了我们的生活。该系统是正是利用 Android 开源平台开发的一款助理类 APP——知心 App, 利用当今主流的编程语言 Java 进行代码的编写，利用 Google 最新开发的 Android Studio 为开发工具，通过了解安卓平台的系统架构和应用程序开发流程，最终成功开发出了该款应用。该款应用主要包括了人脸识别登录、网易新闻、欧酷天气、图灵机器人、作者个人信息、应用更新等模块。此款应用采用了 ButterKnife 依赖注入技术、Gson 数据解析框架、OkHttp 网络处理框架、科大讯飞语音识别技术、图灵机器人技术、一登刷脸登录技术、Bmob 登录注册技术、蒲公英应用发布以及自动更新等技术。其中特色功能有刷脸登录、图灵机器人、应用更新。此款应用较好的跟上时代的步伐，运用了一些当前热门的人工智能技术来增添新意，具有一定的实用价值。此款应用也严格按照软件开发标准流程进行设计与开发，代码结构清晰，用户体验较好。

关键词：Android; Android Studio; 人脸识别; 语音识别; 图灵机器人;

Development Of ZhiXin Assistant App Based On Android

Abstract

At present, Android development is having a tremendous impact on our lives. Our lives are inseparable from such applications as WeChat, QQ, and Weibo. These applications have greatly facilitated our lives, brought a lot of fun to our work life, and enriched and enriched our lives. The system is an assistant APP developed by Android open source platform - Zhixin App, using the current mainstream programming language Java to write code, using Google's newly developed Android Studio as a development tool, and understanding the system architecture of the Android platform. With the application development process, the application was eventually successfully developed. This application mainly includes face recognition login, NetEase news, cool weather, Turing robot, author personal information, application updates and other modules. This application uses ButterKnife dependency injection technology, Gson data parsing framework, OkHttp network processing framework, science and technology news voice recognition technology, Turing robot technology, a login face login technology, Bmob login registration technology, dandelion application release, and automatic updates And other technologies. Among them, there are brush face login, Turing robot and application update. This application is better to keep pace with the times, using some of the current popular artificial intelligence technology to add new ideas. This application also strictly follows the software development standard process for design and development. The code structure is clear and the user experience is better.

Key Words: Android; Android Studio; Face Recognition; Speech Recognition; Turing Robot;

目录

| | |
|----------------------------------|----|
| 第 1 章 绪论 | 1 |
| 1.1 课题背景..... | 1 |
| 1.1.1 人工智能 | 1 |
| 1.1.2 大数据 | 1 |
| 1.2 目的意义..... | 2 |
| 1.3 论文主要工作 | 2 |
| 第 2 章 相关技术和理论基础 | 3 |
| 2.1 开发框架技术 | 3 |
| 2.1.1 ButterKnife 依赖注入框架..... | 3 |
| 2.1.2 Gson 框架 | 3 |
| 2.1.3 OkHttp 网络框架 | 4 |
| 2.2 第三方服务平台 | 4 |
| 2.2.1 科大讯飞语音识别与图灵机器人 | 4 |
| 2.2.2 Bmob 后端服务平台以及蒲公英应用平台 | 5 |
| 2.2.3 GitHub 项目管理工具 | 5 |
| 第 3 章 系统分析 | 7 |
| 3.1 功能需求分析 | 7 |
| 3.1.1 用户需求分析 | 7 |
| 3.1.2 功能需求分析 | 7 |
| 3.2 非功能需求分析 | 8 |
| 3.3 可行性分析 | 8 |
| 3.3.1 经济可行性 | 8 |
| 3.3.2 技术可行性 | 8 |
| 3.3.3 操作可行性 | 8 |
| 第 4 章 系统设计 | 9 |
| 4.1 总体设计..... | 9 |
| 4.1.1 功能模块图 | 9 |
| 4.1.2 用例图 | 10 |
| 4.1.3 数据流图 | 10 |
| 4.2 详细设计..... | 11 |
| 4.2.1 登录注册模块 | 11 |
| 4.2.2 网易新闻模块 | 11 |
| 4.2.3 图灵机器人模块 | 12 |
| 4.2.4 欧酷天气模块 | 12 |

| | |
|----------------------------|----|
| 4.2.5 引导页、欢迎页模块 | 12 |
| 4.2.6 开发者详情模块 | 12 |
| 4.2.7 应用更新模块 | 12 |
| 第 5 章 系统实现与测试 | 14 |
| 5.1 系统实现 | 14 |
| 5.1.1 登录、注册模块实现 | 14 |
| 5.1.2 图灵机器人模块实现 | 17 |
| 5.1.3 网易新闻模块实现 | 19 |
| 5.2 系统测试 | 21 |
| 5.2.1 登录注册功能测试 | 21 |
| 5.2.2 图灵机器人、网易新闻模块测试 | 23 |
| 5.2.3 欧酷天气模块测试 | 24 |
| 第 6 章 总结和展望 | 25 |
| 6.1 本文总结 | 25 |
| 6.2 未来展望 | 26 |
| 参考文献 | 27 |
| 附录 | 28 |
| 致谢 | 34 |

图目录

| | |
|-----------------------------|----|
| 图 2-1 ButterKnife 示例代码..... | 3 |
| 图 2-2 OkHttp 示例代码..... | 4 |
| 图 2-3 Git 命令示例代码..... | 6 |
| 图 4-1 系统结构图..... | 9 |
| 图 4-2 MVP 架构图..... | 9 |
| 图 4-3 功能模块图..... | 10 |
| 图 4-4 用例图..... | 10 |
| 图 4-5 用户注册数据流图..... | 10 |
| 图 5-1 登录流程图..... | 14 |
| 图 5-2 判断登录是否成功..... | 15 |
| 图 5-3 登录结果回调函数..... | 15 |
| 图 5-4 注册流程图..... | 16 |
| 图 5-5 用户注册函数..... | 16 |
| 图 5-6 图灵机器人流程图..... | 17 |
| 图 5-7 语音识别函数..... | 18 |
| 图 5-8 语音合成函数..... | 18 |
| 图 5-9 语音处理适配器类..... | 19 |
| 图 5-10 语音处理适配器类(续)..... | 19 |
| 图 5-11 网易新闻流程图..... | 20 |
| 图 5-12 新闻处理适配器类..... | 21 |
| 图 5-13 账号登录测试效果图..... | 22 |
| 图 5-14 刷脸登录测试图..... | 22 |
| 图 5-15 图灵机器人对话测试图..... | 23 |
| 图 5-16 网易新闻效果图..... | 23 |
| 图 5-17 城市列表效果图..... | 23 |
| 图 5-18 天气详细信息效果图..... | 23 |

表目录

| | |
|---------------------|---|
| 表 3-1 项目详细开发环境..... | 8 |
|---------------------|---|

第 1 章 绪论

1.1 课题背景

目前市场上主流的手机操作系统有两种，那就是 Android 和 IOS。Android 系统是谷歌开发的并且最重要的是开放源代码，而 IOS 则是苹果公司研发的，不开放源代码。就当今用户使用手机情况来看，安卓手机占据大约 60% 的手机市场，而苹果手机大约占 30% 的市场。全世界的企业大部分还是中小企业，经济能力有限，因此大部分厂商还是喜欢用开源免费的。因此，我的这个应用就是基于 Android 平台开发的。想必大家都用过智能手机，也用过普通的手机软件，早在五年之前安卓市场非常火爆，但是当时的软件还是非常的普通，完全不具备现在的人工智能与大数据技术。现在的应用都在慢慢的被大数据与人工智能技术所熏陶，比如说微博、微信等应用，各种个性化推荐信息、智能提示非常方便用户使用。

目前处于人工智能和大数据飞速发展的时代，为了让用户体会到大数据与人工智能的魅力，感受新时代的气息，我在应用中添加了人脸识别和图灵机器人以及语音识别等功能模块。目前提供人脸识别服务的平台比较多，有百度 AI 平台，阿里云平台，腾讯云平台，Face++，一登等。但是从个人选择上来讲，一登接入比其他的平台简单，并且是免费的。智能聊天机器人采用图灵公司开发的新一代的机器人。语音识别功能采用的是科大讯飞的语音识别技术，相对来说科大讯飞的语音技术在国内应该是顶尖的。这些是具有人工智能功能的应用是传统的应用所无法替代的，因此为了适应时代发展的潮流，我在普通应用的基础之上有了自己的创新。利用最新的 Android 开发技术结合人工智能技术开发具有良好用户体验以及更加智能的安卓应用。

1.1.1 人工智能

人工智能技术简称 AI，它的研究主要是便于机器模拟人的思维，能够像人一样思考，最终利用机器人更好地为人类服务。

1.1.2 大数据

大数据技术，它的研究主要是为了处理当今社会生活中产生的大批量的数据。目前主流的大数据平台有 Hadoop 和 Spark。Hadoop 平台主要是支持大批量数据的离线处理，而 Spark 平台主要是基于内存的高速实时数据流处理。两者在性能上

有较大的差距，Spark 的处理速度大约是 Hadoop 的 10 倍。当然两者分别具有各自的特色，适用于不同的应用场景。

1.2 目的意义

目前正处在人工智能和大数据的时代，为了让用户感受到人工智能的存在，我在应用中添加了人脸识别和图灵机器人以及语音识别等功能模块。这些是传统的应用所不具备的功能，因此我在其基础之上有了自己的创新，让应用更加智能化。具有良好的用户体验是衡量一个安卓应用品质的很重要的标准。只有拥有良好的用户体验的应用才能长期生存下去。新的时代需要不断创新，要把最新的技术运用到 Android 应用之上，让 App 具有新意，更加符合人们的使用需求。因此，我设计了这款 App 帮助用户更好的体验智能化的生活。

1.3 论文主要工作

目前正是人工智能和大数据技术火热的时候，为了在让传统的安卓应用具有新时代的特点，我在本次开发中加入了一些人工智能的元素，希望能给用户带来全新的用户体验。

本课题的主要开发流程包括需求分析、概念设计、详细设计、编码测试、应用上线。需求分析阶段主要采用市场调研和问卷调查两种方式。概念设计阶段主要是总体结构设计，采用 MVP 三层架构技术。其中详细设计阶段主要设计应用的各个功能模块，如引导页模块、登录注册模块、网易新闻模块、图灵机器人模块、开发者个人信息模块、应用更新等模块。编码实现阶段采用 Java 语言开发，运用 Android Studio 开发工具进行开发。后期的编码测试采用真机测试，记录日志信息，修复 Bug。应用开发完成后，进行应用打包，最后发布到蒲公英应用发布平台。

第2章 相关技术和理论基础

2.1 开发框架技术

在此应用中主要用到了 ButterKnife 依赖注入框架、Gson 数据解析框架、OkHttp 网络处理框架。

2.1.1 ButterKnife 依赖注入框架

（1）ButterKnife 介绍

ButterKnife 依赖注入框架主要用于自动化生成控件绑定代码，不需要手动的写 findViewById 和点击事件，应用插件即可一键生成绑定代码，极大地减少了程序员繁琐的工作量。在项目中我使用了该框架，极大地减少了我的繁琐的工作量。

（2）项目配置

在 AndroidStudio Module 中的 build.gradle 中添加以下依赖：

```
compile 'com.jakewharton:butterknife:8.8.1'
```

```
annotationProcessor 'com.jakewharton:butterknife-compiler:8.8.1'
```

（3）示例代码

```
class ExampleActivity extends Activity {
    @BindView(R.id.title) TextView title;
    @BindView(R.id.subtitle) TextView subtitle;
    @BindView(R.id.footer) TextView footer;

    @Override public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.simple_activity);
        ButterKnife.bind(this);
        // TODO Use fields...
    }
}
```

图 2-1 ButterKnife 示例代码

2.1.2 Gson 框架

（1）Gson 介绍

Gson 是 Google 公司发布的一个开放源代码的 Java 库，主要用途为序列化 Java 对象为 JSON 字符串，或反序列化 JSON 字符串成 Java 对象。在项目中，我主要用 GSON 处理天气数据，将城市数据数据转化为对应的实体类对象；

处理新闻数据，将新闻数据转换为对应的新闻实体类；处理图灵机器人接口数据，将返回数据转换成一个实体类对象。

（2）项目配置

在 AndroidStudio Module 中的 build.gradle 中添加以下依赖：

```
compile 'com.google.code.gson:gson:2.8.2'
```

2.1.3 OkHttp 网络框架

（1）OkHttp 介绍

OkHttp 是一款处理网络请求的框架，具有较高的网络处理性能，可以用于文件的上传和下载、加载图片、处理 GET/POST 请求。在本项目中，我主要用于获取 JSON 数据，加载图片。

（2）项目配置

在 AndroidStudio Module 中的 build.gradle 中添加以下依赖：

```
compile 'com.squareup.okhttp3:okhttp:3.10.0'
```

（3）示例代码

```
OkHttpClient client = new OkHttpClient();
String run(String url) throws IOException {
    Request request = new Request.Builder()
        .url(url)
        .build();
    Response response = client.newCall(request).execute();
    return response.body().string();
}
```

图 2-2 OkHttp 示例代码

2.2 第三方服务平台

在此应用中主要用科大讯飞语音识别技术、图灵机器人技术、一登刷脸登录技术、Bmob 登录注册技术、蒲公英应用发布以及更新、GitHub 项目管理等技术。

2.2.1 科大讯飞语音识别与图灵机器人

（1）语音识别技术

语音识别包括听写、语法识别功能。语音识别技术就是把人说的话转换成相应的文本数据。语法识别是基于语法规则，将与语法一致的自然语言音频转换为文本输出的技术。

科大讯飞开发者文档：http://doc.xfyun.cn/msc_android/index.html

（2）图灵机器人技术

图灵机器人是图灵公司开发一款机器人，具备与人交流的能力，可以和人进行简单的对话。主要应用领域是聊天机器人、客服机器人和服务机器人。

图灵机器人开发者文档：https://www.kancloud.cn/turing/web_api/522992

2.2.2 Bmob 后端服务平台以及蒲公英应用平台

（1）Bmob 后端云数据库

Bmob 后端云数据库主要用于实时数据与文件存储，轻松实现应用与云端数据连通。数据存储除了常见的文本类型，还可以存储图片、视频、音频、地理位置等信息。在本次项目中，我主要用于存取用户注册信息，将注册的账号密码存在云数据库中，然后登录的时候从云数据库中获取过来账号密码进行比较，判断是否登录成功。

Bmob 开发者文档：<http://doc.bmob.cn/data/android/>

（2）蒲公英应用发布平台

蒲公英应用平台是一款应用发布的平台，可以将开发好的应用安装包发布到该平台，然后该平台会生成一个应用下载页面，包含应用介绍、应用截图、下载应用的二维码。该平台也提供了应用检测更新的功能，只需要集成 SDK 就可以实现应用检测更新和下载更新。在本次项目中，我主要用到了应用发布和应用更新的功能。

Bmob 开发者文档：http://www.pgyer.com/doc/view/sdk_android_guide

2.2.3 GitHub 项目管理工具

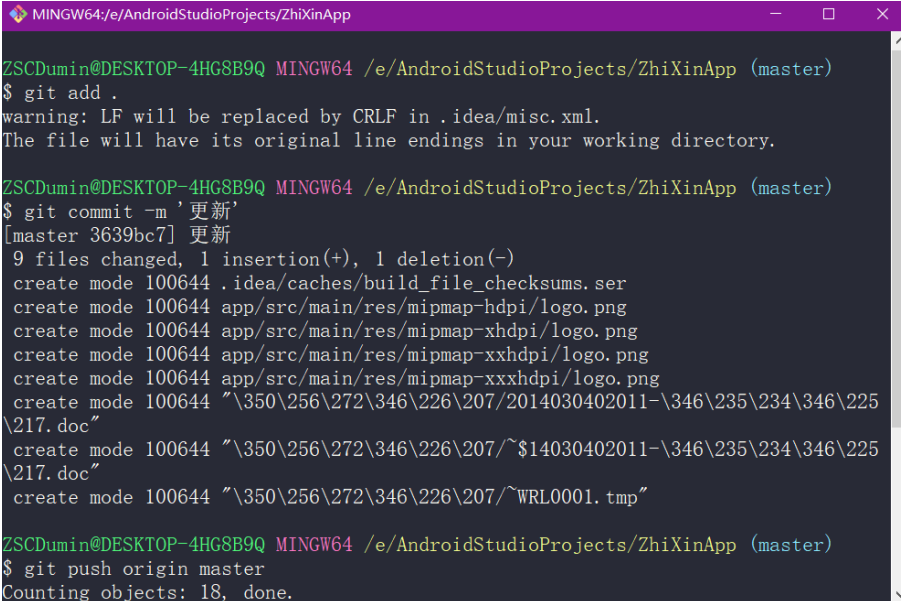
（1）GitHub 网址：<https://github.com>

（2）Git 常用命令

- `git init` //初始化项目
- `git status` //查看项目状态
- `git add` //添加文件到 `git` 追踪列表里
- `git commit` //提交到本地仓库
- `git log` //提交记录
- `git clone` //复制项目到本地
- `git branch` //查看或者创建分支
 - `git branch net` //创建 `net` 分支：
 - `git branch -d net` //删除分支：
- `git checkout` //签出一个分支或者一个路径，在开发中常用于分支切换和恢复文件。

- `git checkout net` //切换 net 分支:
- `git checkout -- SayHello.java` //恢复文件:
- `git merge` //合并分支;
 - `git merge net` //合并 net 分支:
- `git tag` //列出项目的标签和创建项目的标签
 - `git tag -a v1.0 -m "这里写相关信息"` //为版本打一个标签
 - `git show 标签名` //来查看该分支的具体信息
- `git help` //帮助
- `git push` //推送
 - `git push 仓库名 分支名`
 - `git push origin master:master` 或 `git push origin master` //本地 master 分支提交到 origin 的 master 分支，分支相同的可简写
- `git push origin :net` //删除 origin 仓库的 net 分支
- `git pull` //更新最新的代码
- `git ignore` //忽略文件
- `fork` //远程复制别人的仓库到自己的仓库

(3) 示例代码



```

MINGW64:/e/AndroidStudioProjects/ZhiXinApp
ZSCDumin@DESKTOP-4HG8B9Q MINGW64 /e/AndroidStudioProjects/ZhiXinApp (master)
$ git add .
warning: LF will be replaced by CRLF in .idea/misc.xml.
The file will have its original line endings in your working directory.

ZSCDumin@DESKTOP-4HG8B9Q MINGW64 /e/AndroidStudioProjects/ZhiXinApp (master)
$ git commit -m '更新'
[master 3639bc7] 更新
9 files changed, 1 insertion(+), 1 deletion(-)
create mode 100644 .idea/caches/build_file_checksums.ser
create mode 100644 app/src/main/res/mipmap-hdpi/logo.png
create mode 100644 app/src/main/res/mipmap-xhdpi/logo.png
create mode 100644 app/src/main/res/mipmap-xxhdpi/logo.png
create mode 100644 app/src/main/res/mipmap-xxxhdpi/logo.png
create mode 100644 "\350\256\272\346\226\207\2014030402011-\346\235\234\346\225\217.doc"
create mode 100644 "\350\256\272\346\226\207\~$14030402011-\346\235\234\346\225\217.doc"
create mode 100644 "\350\256\272\346\226\207\~WRL0001.tmp"

ZSCDumin@DESKTOP-4HG8B9Q MINGW64 /e/AndroidStudioProjects/ZhiXinApp (master)
$ git push origin master
Counting objects: 18, done.

```

图 2-3 Git 命令示例代码

第3章 系统分析

3.1 功能需求分析

随着大数据、人工智能的迅速发展，我们的生活变得越来越便利。传统的应用只能通过账号密码来登录，用户体验不是太好。而如今的 App 变得越来越智能化了，比如支付宝、淘宝、微信等。支付宝现在正运用最先进的大数据与人工智能技术开启了刷脸登录与刷脸支付的时代。为了让 App 具有新时代的特点，不得不运用大数据与人工智能的技术，这也是未来发展的趋势。经过市场的调查，发现当前大部分的应用还是不具备新时代的特点，因此我觉得我这个项目是符合市场需求的。

3.1.1 用户需求分析

此软件要求能够运行在 Android 5.0 以上的系统终端上，相对来说使用群体还是占据很大一部分，但是用户对软件也有一定的要求：

- (1) 简单易学
- (2) 运行稳定
- (3) 操作简单

3.1.2 功能需求分析

本项目应当具有以下功能：

- (1) 应用引导页、欢迎页
- (2) 账号密码登录、注册
- (3) 刷脸登录
- (4) 网易新闻
- (5) 图灵机器人
- (6) 欧酷天气
- (7) 开发者个人信息
- (8) 应用更新
- (9) 应用打包发布

3.2 非功能需求分析

- (1) 开发环境为 Android Studio
- (2) 要求系统响应不超过 3 秒
- (3) 要求应用兼容 Android 5.0 以上的主流手机平台
- (4) 多渠道打包

3.3 可行性分析

3.3.1 经济可行性

本软件基于开源的 Android 系统开发，采用 Android Studio 免费软件进行开发，运用第三方平台提供的免费服务实现特定功能，开发成本相对较低。

3.3.2 技术可行性

表 3-1 项目详细开发环境

| | |
|-------------------|----------------------|
| Windows 版本 | Windows 10 家庭版 64 位 |
| Android Studio 版本 | 3.1.1 |
| Android 编译 SDK 版本 | 26 |
| Android 测试版本 | 华为荣耀青春版 (Android7.0) |
| Gradle 版本 | 4.4 |
| Maven 版本 | 3.5.3 |
| Git 版本 | 2.16.2 |

本软件基于开源的 Android 系统开发，使用主流的编程语言 Java 进行编码，因此该软件在技术上可行。

3.3.3 操作可行性

该软件采用主流的 Android 软件应用开发风格，并使用 Google 全新的设计语言 Material Design，界面友好，简单易懂，为用户带来全新的视觉体验。

第 4 章 系统设计

4.1 总体设计

系统结构图主要包含应用引导页、应用欢迎页、登录、注册、网易新闻、图灵机器人、欧酷天气、作者信息、应用更新。其中登录包括账号密码登录和刷脸登录、网易新闻包括新闻列表和新闻详情页、图灵机器人包含显示对话列表和讯飞语音识别、欧酷天气包括城市列表和天气详情、应用更新包括检测更新和下载更新。



图 4-1 系统结构图

系统架构主要采用 MVP 架构模式，其中 M 表示模型层，V 表示视图层，P 有点类似于 MVC 中的控制层。

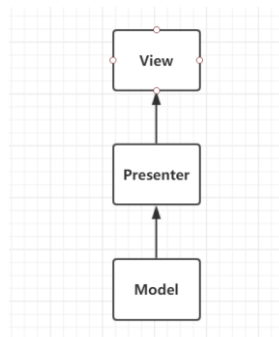


图 4-2 MVP 架构图

4.1.1 功能模块图

系统功能模块主要包括引导页模块、网易新闻模块、用户注册模块、用户登录模块（包含账号密码登录和刷脸登录）、图灵机器人模块、欧酷天气模块、开发者信息模块、应用更新模块。

该软件的功能模块图如下：

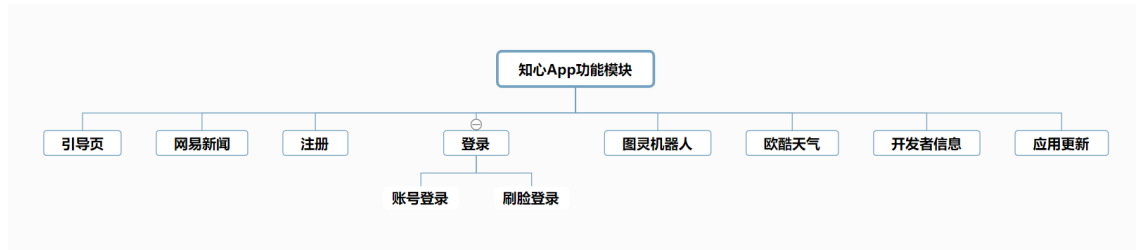


图 4-3 功能模块图

4.1.2 用例图

系统用例主要包含账号登录、刷脸登录、注册、查看天气信息、与机器人对话、查看网易新闻、查看开发者详情、检测更新、下载更新等。

用例图如下：

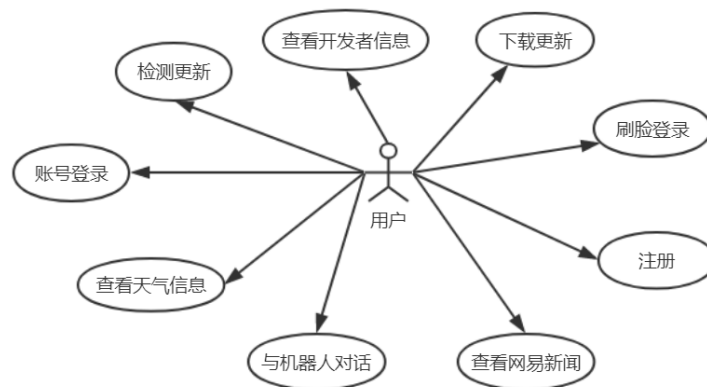


图 4-4 用例图

4.1.3 数据流图

首先，用户填写注册信息，然后系统审核信息是否有效，若数据经过系统审核以后没有问题，则将注册信息写入后端数据库，否则提示注册信息有误，请重新填写。



图 4-5 用户注册数据流图

4.2 详细设计

对于引导页模块，采用图片滑动形式，引导用户了解如何使用该软件以及相关的功能。

对于登录模块，采用两种登录方式：账号密码登录和刷脸登录。账号密码登录的话采用 Bmob 后端云数据库实现，刷脸登录采用的是一登刷脸服务。

对于注册模块，同样采用的是 Bmob 后端云数据库。

对于网易新闻模块，采用的是网易新闻的接口。

对于欧酷天气模块，采用的是天气预报的接口。

对于图灵机器人模块，采用的图灵智能机器人的接口。

对于开发者个人信息模块，采用列表显示出来。

对于应用更新模块，采用蒲公英应用发布平台的 App 更新技术。

4.2.1 登录注册模块

1、登录模块处理逻辑如下：

- (1) 用户输入账号密码进行登录
- (2) 通过回调函数获取登录返回结果
- (3) 根据返回结果判断登陆是否成功，若登录成功则显示用户信息，否则提示登录失败。
- (4) 登陆成功将用户的姓名和电话显示出来

2、注册模块处理逻辑如下：

- (1) 用户输入账号密码进行注册
- (2) 通过回调函数获取注册返回结果
- (3) 根据注册返回结果判断是否注册成功，若注册成功则返回登录页面，否则提示注册失败

4.2.2 网易新闻模块

1、网易新闻模块处理逻辑如下：

- (1) 通过天气接口的 URL 获取返回的 JSON 数据
- (2) 将 JSON 数据转换为 News 对象
- (3) 将数据显示在页面上

4.2.3 图灵机器人模块

1、图灵机器人模块的处理逻辑如下：

- (1) 将语音数据转换为文本数据
- (2) 通过图灵接口的 URL 获取返回的 JSON 数据
- (3) 将 JSON 数据转换为 TalkBean 对象
- (4) 获取对象中的内容，再将其转换为语音输出

4.2.4 欧酷天气模块

1、欧酷天气模块处理逻辑如下：

- (1) 通过城市 URL 获取城市 JSON 数据
- (2) 将 JSON 数据转换为城市对象
- (3) 通过天气 URL 获取天气 JSON 数据
- (4) 将天气 JSON 数据转换为天气对象
- (5) 在页面显示天气信息

4.2.5 引导页、欢迎页模块

1、引导页、欢迎页模块处理逻辑如下：

- (1) 判断应用是不是第一次打开
- (2) 判断本地数据库是否有记录，若是第一次打开的话则显示引导页，并将记录写到本地数据库
- (3) 否则直接进入欢迎页

4.2.6 开发者详情模块

1、开发者详情模块处理逻辑如下：

- (1) 直接加载布局文件的内容显示出来

4.2.7 应用更新模块

1、应用更新模块的处理逻辑如下：

- (1) 点击检测更新按钮
- (2) 系统比较本地版本和服务器上的版本
- (3) 若本地版本等于服务器版本，则提示用户“您的应用已是最新版本！”
- (4) 若本地版本低于服务器版本，则提示用户“检测到新版本，是否更新？”

- (5) 若用户点击“取消”，则退出更新
- (6) 若用户点击“确定”，则后台服务自动下载应用更新包
- (7) 下载完成后，系统提示用户是否立即安装
- (8) 若用户点击“取消”，则退出安装程序
- (9) 若用户点击“确定”，则等待安装完成

第5章 系统实现与测试

5.1 系统实现

该项目主要包含引导页模块、登录注册模块、网易新闻模块、图灵机器人模块、开发者个人信息模块、应用更新等模块。以下是各个模块的设计与实现。

5.1.1 登录、注册模块实现

（1）登录模块流程

首先用户输入账号密码，然后用户点击登录按钮以后，系统判断账号密码是否正确；若账号密码正确，则进入主界面，否则提示“账号或者密码错误，请重新登录”。

登录模块流程图如下：

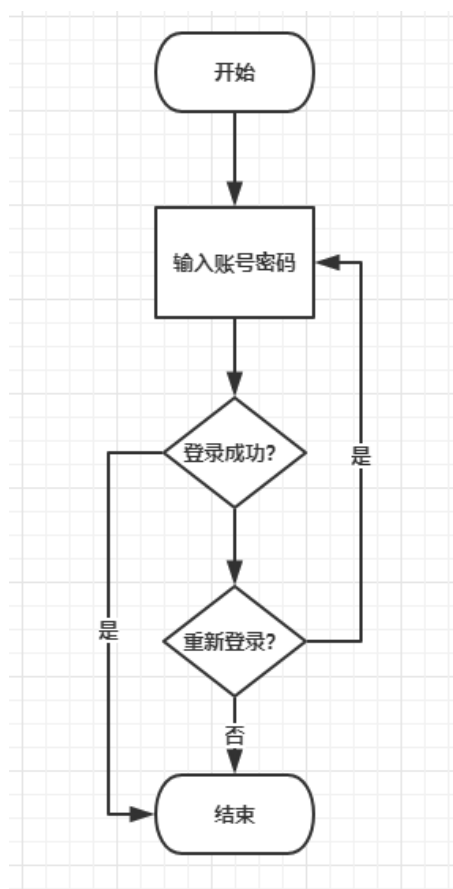


图 5-1 登录流程图

（2）登录模块主要代码

该函数用于判断登录是否成功。首先用户点击登录按钮以后，获取用户从页面输入的账号密码，然后系统去云数据库查找所有注册了的账号密码，接着用该查询到的账号密码和输入的账号密码进行比较，看看是否一致；若账号密码正确，则进入主界面，否则提示“登录失败，账号或者密码错误！”。

```
public void isLoginSuccess() {
    if (userAccount.length() > 0 && userPassword.length() > 0) {
        BmobQuery<Person> query = new BmobQuery<Person>();
        query.findObjects(this, new FindListener<Person>() {
            String user_account = userAccount.getText().toString().trim();
            String user_password = userPassword.getText().toString().trim();
            int flag = 0;
            @Override
            public void onSuccess(List<Person> list) {
                for (int i = 0; i < list.size(); i++) {
                    String account = list.get(i).getUserAccount();
                    String password = list.get(i).getUserPassword();
                    if (account.equals(user_account) && password.equals(user_password)) {
                        flag = 1;
                        Toast.makeText(UserLoginActivity.this, "登录成功!", Toast.LENGTH_SHORT).show();
                        Intent intent = new Intent(UserLoginActivity.this, MainActivity.class);
                        Bundle bundle = new Bundle();
                        bundle.putString("login_type", "with_account");
                        intent.putExtras(bundle);
                        startActivity(intent);
                        break;
                    }
                }
            }
        });
        if (flag == 0) {
            userAccount.setText("");
            userPassword.setText("");
            Toast.makeText(UserLoginActivity.this, "登录失败,账号或密码错误!", Toast.LENGTH_SHORT).show();
        }
    }
}
```

图 5-2 判断登录是否成功

该函数主要用于处理刷脸登录返回的状态信息，若返回的状态信息为登录成功，则获取该用户的用户信息，包括用户名、电话号码、用户头像等信息；若返回的状态信息为登录失败，则提示刷脸登录失败，请尝试用账号密码登录。

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    switch (resultCode) {
        case SDKConfig.LOGIN_SUCCESS:
            // 使用人脸登录后获取的一登用户信息
            String user_info = Cache.getCached(UserLoginActivity.this, SDKConfig.KEY_APPINFO);
            String phone_num = "";
            String user_icon = "";
            String user_name = "";
            try {
                JSONObject obj = new JSONObject(user_info);
                phone_num = obj.getString(SDKConfig.KEY_PHONENUM);
                user_icon = obj.getString(SDKConfig.KEY_AVATAR);
                user_name = SuperIDUtils.judgeChina(obj.getString(SDKConfig.KEY_NAME), 10);
            } catch (JSONException e) {
                e.printStackTrace();
            }
            Intent intent = new Intent(UserLoginActivity.this, MainActivity.class);
            Bundle bundle = new Bundle();
            bundle.putString("login_type", "with_face");
            bundle.putString("user_name", user_name);
            bundle.putString("user_icon", user_icon);
            bundle.putString("phone_num", phone_num);
            intent.putExtras(bundle);
            startActivity(intent);
            finish();
            break;
        default:
            Toast.makeText(UserLoginActivity.this, "刷脸登录失败,请尝试用账号密码登陆!", Toast.LENGTH_SHORT).show();
            break;
    }
}
```

图 5-3 登录结果回调函数

（3）注册模块流程图

首先，用户在页面输入账号密码并点击注册按钮，然后系统根据输入的账号去后台数据库进行比对。若已存在该账号，则提示“注册失败!”；否则提示“用户注册成功!”。

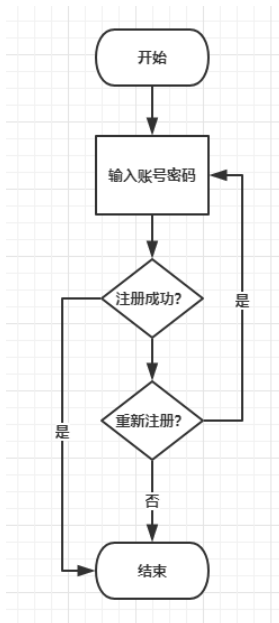


图 5-4 注册流程图

（4）注册模块代码如下：

首先，用户在页面输入账号密码并点击注册按钮，然后系统判断输入的账号和密码是否为空，若为空则提示“账号或密码为空，请重新注册!”；否则根据后台数据库返回状态信息进行判断。若状态信息为成功，则提示“用户注册成功!”；若状态信息为失败，则“提示用户注册失败!”。

```
public void onViewClicked() {
    final String user_account = userAccount.getText().toString().trim();
    String user_password = userPassword.getText().toString().trim();

    //判断用户名和密码是否为空,如果为空则不能进去。
    if (user_account.length() > 0 && user_password.length() > 0) {
        person = new Person();
        person.setUserAccount(user_account);
        person.setUserPassword(user_password);
        //插入方法
        person.save(this, new SaveListener() {
            @Override
            public void onSuccess() {
                // TODO Auto-generated method stub
                userAccount.setText("");
                userPassword.setText("");
                Toast.makeText(UserRegisterActivity.this, "恭喜您,注册成功!", Toast.LENGTH_SHORT).show();
                Intent intent=new Intent(UserRegisterActivity.this,UserLoginActivity.class);
                startActivity(intent);
                finish();
            }

            @Override
            public void onFailure(int code, String msg) {
                // TODO Auto-generated method stub
                Toast.makeText(UserRegisterActivity.this, "很抱歉,您注册失败! : " + msg, Toast.LENGTH_SHORT).show();
            }
        });
    } else {
        Toast.makeText(UserRegisterActivity.this, "账号或密码不能为空!", Toast.LENGTH_SHORT).show();
    }
}
```

图 5-5 用户注册函数

5.1.2 图灵机器人模块实现

（1）图灵机器人流程图

首先，初始化语音机器人，接着机器人发出“欢迎使用图灵机器人的”语音，然后用户用对着麦克风讲话向机器人提问，与此同时语音信息也将以文本的形式用聊天列表的方式显示出来。接下来，通过图灵机器人的接口来获取问题回答的 JSON 数据，接着把数据转换成文本和语音形式，最后将其显示在聊天列表中和语音播放出来。

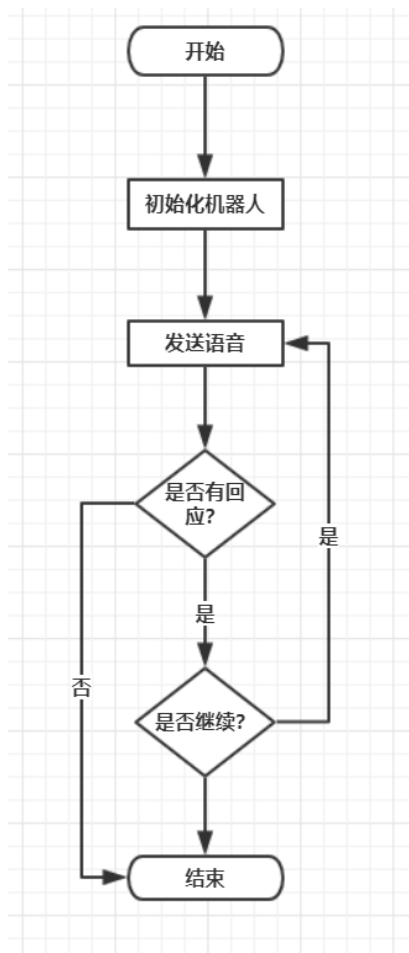


图 5-6 图灵机器人流程图

（2）图灵机器人实现代码

该函数为语音识别的函数，首先创建了一个语音识别的对象，并设置了一些相关的参数信息，比如口音、语言等。接下来设置接口回调函数，在回调函数中获取语音识别结果的 JSON 数据，然后将其转换为普通的文本数据并添加到聊天列表。最后显示接收语音输入的对话框。


```

public void startListen(View v) {
    //1.创建RecognizerDialog对象
    RecognizerDialog mDialog = new RecognizerDialog(this, null);
    //2.设置accent、language等参数
    mDialog.setParameter(SpeechConstant.LANGUAGE, "zh_cn");
    mDialog.setParameter(SpeechConstant.ACCENT, "mandarin");
    sbuff = new StringBuffer();
    //3.设置回调接口
    mDialog.setListener(new RecognizerDialogListener() {
        //最终的识别结果
        @Override
        public void onResult(RecognizerResult recognizerResult, boolean b) {
            String result = recognizerResult.getResultString();
            String spreak_word = pareData(result);
            sbuff.append(spreak_word);
            if (b) {
                askContent = sbuff.toString(); //得到最终结果
                TalkBean askBean = new TalkBean(askContent, -1, true); //初始化提问对象
                mList.add(askBean);
                answers = "这个问题我们机器要开个会,等商量出来再告诉你";
                new GetMessage().start();
            }
        }
        @Override
        public void onError(SpeechError speechError) {
        }
    });
    //4.显示dialog, 接收语音输入
    mDialog.show();
}

```

图 5-7 语音识别函数

该函数为语音合成函数，主要将文本数据转换为语音数据。首先，初始化一个语音合成对象，然后设置相关参数，比如发音人、语速、音量、云端等。最后开始语音合成。

```

public void compose(String speak) {
    //1. 创建 SpeechSynthesizer 对象 , 第二个参数: 本地合成时传 InitListener
    SpeechSynthesizer mTts = SpeechSynthesizer.createSynthesizer(this, null);
    //2.合成参数设置
    mTts.setParameter(SpeechConstant.VOICE_NAME, "xiaoyan"); //设置发音人
    mTts.setParameter(SpeechConstant.SPEED, "60"); //设置语速
    mTts.setParameter(SpeechConstant.VOLUME, "80"); //设置音量, 范围 0~100
    mTts.setParameter(SpeechConstant.ENGINE_TYPE, SpeechConstant.TYPE_CLOUD); //设置云端
    mTts.setParameter(SpeechConstant.TTS_AUDIO_PATH, "./sdcard/iflytek.pcm");
    //3.开始合成
    mTts.startSpeaking(speak, null);
}

```

图 5-8 语音合成函数

这是一个语音适配器类，继承自基础的适配器类，主要用于将聊天信息列表数据和 ListView 数据进行适配。其中 getCount 返回列表的大小，getItem 表示获取某一项列表数据，getItemId 返回某一项列表项的 Id，getView 的话根据布局文件返回一个视图对象。

```

class MyAdapter extends BaseAdapter {
    @Override
    public int getCount() {
        return mList.size();
    }
    @Override
    public TalkBean getItem(int position) {
        return mList.get(position);
    }
    @Override
    public long getItemId(int position) {
        return position;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        ViewHolder holder;
        if (convertView == null) {
            convertView = View.inflate(getApplicationContext(), R.layout.list_item, null);
            holder = new ViewHolder(convertView);
            convertView.setTag(holder);
        } else {
            holder = (ViewHolder) convertView.getTag();
        }
        TalkBean item = getItem(position);
        if (item.isAsk) {
            //提问
            holder.tvAsk.setVisibility(View.VISIBLE);
            holder.llAnswer.setVisibility(View.GONE);
            holder.tvAsk.setText(item.content);
        } else {

```

图 5-9 语音处理适配器类

```

            if (item.isAsk) {
                //提问
                holder.tvAsk.setVisibility(View.VISIBLE);
                holder.llAnswer.setVisibility(View.GONE);
                holder.tvAsk.setText(item.content);
            } else {
                holder.tvAsk.setVisibility(View.GONE);
                holder.tvAnswer.setVisibility(View.VISIBLE);
                holder.tvAnswer.setText(item.content);
                //图片
                if (item.imageId > 0) {
                    holder.ivPic.setVisibility(View.VISIBLE);
                    holder.ivPic.setImageResource(item.imageId);
                } else {
                    holder.ivPic.setVisibility(View.GONE);
                }
            }
        }
        return convertView;
    }
}

```

图 5-10 语音处理适配器类(续)

5.1.3 网易新闻模块实现

(1) 网易新闻模块

首先，通过接口获取网易新闻 JSON 数据，然后将 JSON 数据转换对应的新闻实体类对象，接下来将数据采用列表的形式显示在新闻主界面。接着用户可以浏览新闻，选择不同类型的新闻，这里大致分为四种类型：头条、NBA、汽车、笑话。当用户点击某一条新闻的时候，进入具体的新闻详情页。

网易新闻模块流程图如下：

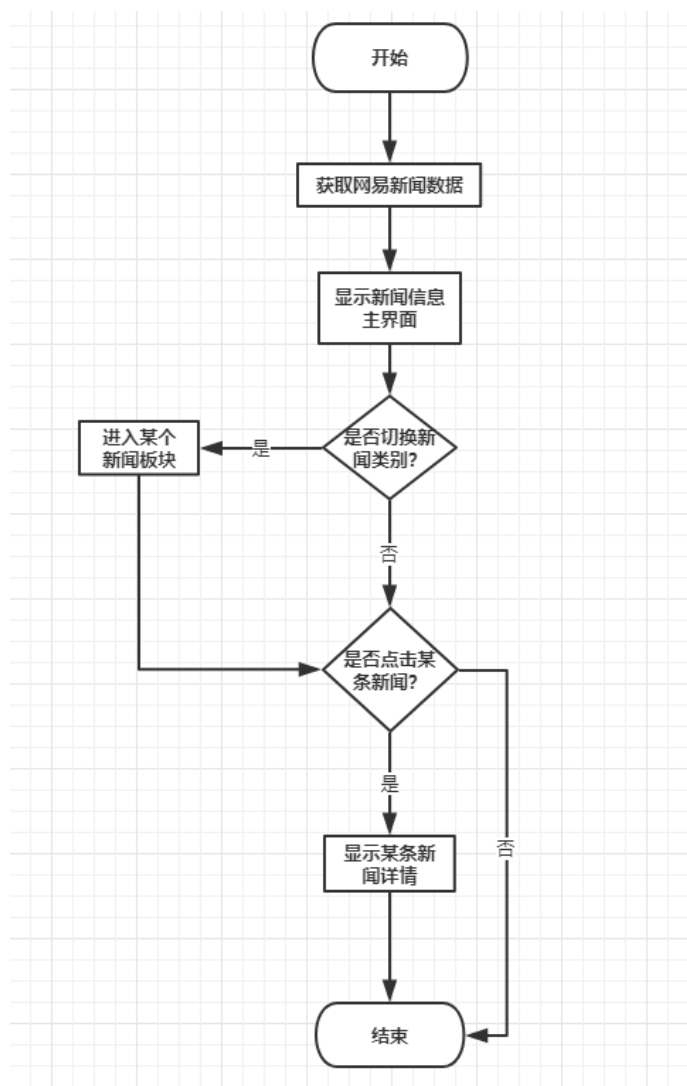


图 5-11 网易新闻流程图

（2）网易新闻模块主要代码

这是一个新闻列表适配器类，继承自 `RecyclerView.Adapter`。其中的 `setData` 函数表示设置数据集，然后在 `onCreateViewHolder` 函数中先判断视图的类型，如果为列表项，则加载新闻列表项对应的布局文件，否则加载分割线相对应的布局文件。

```
public class NewsListAdapter extends RecyclerView.Adapter {
    private static final int TYPE_ITEM = 0;
    private static final int TYPE_FOOTER = 1;
    public final String TAG = "NewsListAdapter";
    private ArrayList<NewsBean> newsList;
    private Context context;
    private OnItemClickListener onItemClickListener;
    //是否显示脚布局
    private boolean showFooter = true;
    public void setData(ArrayList<NewsBean> newsList) {
        this.newsList = newsList;
        this.notifyDataSetChanged();
    }
    public NewsListAdapter(Context context) {
        this.context = context;
    }
    @Override
    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        if (viewType == TYPE_ITEM) {
            View root = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.layout_news_item, parent, false);
            ItemViewHolder vh = new ItemViewHolder(root);
            return vh;
        } else {
            View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.layout_listview_footer, null);
            view.setLayoutParams(new ViewGroup.LayoutParams(ViewGroup.LayoutParams.MATCH_PARENT,
                ViewGroup.LayoutParams.WRAP_CONTENT));
            return new FooterViewHolder(view);
        }
    }
}
```

图 5-12 新闻处理适配器类

5.2 系统测试

该系统为安卓项目，测试采用华为荣耀 8 真机，安卓系统版本为 7.0。主要包括账号登录、刷脸登录、注册、语音对话、获取新闻信息、获取城市信息、获取天气详细信息、开发者个人信息展示、应用更新等功能的测试。

5.2.1 登录注册功能测试

1、账号登录测试：

- (1) 测试账号：18979429542
- (2) 测试密码：123
- (3) 测试截图如下：



图 5-13 账号登录测试效果图

2、刷脸登录测试：



图 5-14 刷脸登录测试图

5.2.2 图灵机器人、网易新闻模块测试

1、图灵机器人对话测试：

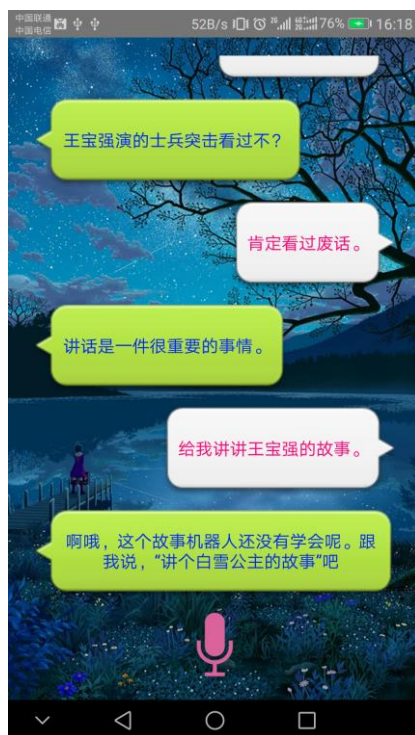


图 5-15 图灵机器人对话测试图

2、网易新闻测试效果图：



图 5-16 网易新闻效果图

5.2.3 欧酷天气模块测试

1、欧酷天气城市列表测试效果图：



图 5-17 城市列表效果图



图 5-18 天气详细信息效果图

第 6 章 总结和展望

6.1 本文总结

这个项目大概花了一个多月的时间才弄完的，查阅各种资料，虽然中途出现了各种问题，但是最终收获还是很多的。刚开始做这个项目的时候，搭建各种环境花费了一番功夫，好在之前自学了一段时间的安卓，最后顺利把环境搭建好了。这个项目主要包含了账号密码登录、刷脸登录、账号密码注册、网易新闻、图灵机器人、欧酷天气、开发者详情、应用更新等模块。其中涉及到的技术主要有 **ButterKnife** 注解、**Gson** 数据解析、**OkHttp** 网络处理、**Bmob** 后端云数据库、一登人脸识别、图灵机器人、科大讯飞语音识别、蒲公英应用发布与自动更新。在上学期学的安卓知识终于派上用场了，这次的项目总体来说还是不错的，加入了新时代的元素，比如说人工智能和大数据。人工智能和大数据是目前最火的技术，代表了未来发展的趋势，因此我在项目里加入了这些元素。项目开发一个很重要的问题就是规范，开发必须要按照标准的流程来，制定相应的代码规范，这样才能有效地减少开发人员的代码修改。在这次开发中，我引进了项目源代码管理仓库 **GitHub**，将开发过程记录到其中了。**GitHub** 是一个团队开发利器，可以方便地进行代码管理和并行开发，有效减少了代码之间的复制。在开发这个项目的时候遇到了一些比较莫名其妙的问题，有一些是依赖包的版本问题，还有一些是 **android** 系统版本的问题。我到现在也还没有把安卓手机兼容性问题解决掉，这个问题有点棘手，市场上的手机种类太多了。在刚开发安卓项目的时候，只知道去百度找资料，那时候还不会翻墙，不知道去谷歌搜索。后来学会了翻墙技术，就可以去谷歌查找更加专业的资料了，身为一个程序员只会百度的话那也太低级了。在遇到问题的时候，先去查看运行输出的错误日志，然后根据错误信息分析错误原因在哪里，最后才百度或者谷歌看看别人如何解决的。在这次项目开发中，发现自己还有许多不足之处，希望在日后不断学习，能够有所改善。本次项目依然存在许多不足之处，比如说程序安全性和健壮性不够，操作界面不够美观，功能不够完善，代码结构不够清晰。希望在以后的工作中能够做得更好。

1、项目管理总结

项目管理工具目前主流的是 SVN 和 GitHub，个人推荐还是 GitHub 比较好用。

2、项目中遇到问题如何解决

遇到问题，首先分析错误日志，然后分析错误的可能原因，最后看书或者去网上搜索找到解决方案。

3、如何获取开发方面的优质资源

推荐几个学习网站：StackOverFlow，掘金，简书，慕课，网易云课堂。

6.2 未来展望

希望在以后的学习中能够学会更多的技术，比如目前流行的大数据与人工智能技术。利用大数据技术与人工智能技术让移动应用更加智能化，更加便利我们的生活。本系统还存在许多不足之处，比如说功能方面还可以更加完善，性能上还有待提高，由于时间关系，来不及全部完成。

1、可添加的功能

- 根据定位自动获取并显示天气信息
- 利用 QQ、微信或者其他第三方平台登录
- 添加好友聊天功能
- 语音唤醒机器人
- 百度地图定位当前位置
- 二维码识别、生成
- 文本、图片识别

2、解决兼容性问题

由于 Android 系统版本较多，不同的手机获取资源对应的包不同，以及不同手机尺寸的不同，所以会出现部分手机闪退现象。希望在以后学习更多的知识后能够解决这个问题。

参考文献

- [1]李刚. 疯狂 Android 讲义[M]. 北京: 电子工业出版社, 2013:25-42.
- [2]杨丰盛. Android 技术内幕[M]. 北京: 机械工业出版社, 2011:77-89.
- [3]杨云君. Android 的设计与实现[M]. 北京: 机械工业出版社, 2013:45-49.
- [4]柯元旦. Android 内核剖析[M]. 北京: 电子工业出版社, 2011:59-70.
- [5]丰生强. Android 软件安全与逆向分析[M]. 北京: 人民邮电出版社, 2013:78-90.
- [6]余成锋, 李代平, 毛永华. Android3.0 内存管理机制分析[M]. 计算机应用与软件, 2013:55-80.
- [7]佐冰冰. Android 平台下 Launcher 启动器的设计与实现[D]. 哈尔滨工业大学, 2012:108-150.
- [8]杜吉志, 徐明昆. Android 系统内存管理研究及优化[J]. 软件, 2012,24(5): 69-80.
- [9]马越. Android 的架构与应用[D]. 北京: 中国地质大学, 2008:330-357.
- [10]姚昱旻, 刘卫国. Android 的架构与应用开发研究[J]. 计算机系统应用, 2008,77(11): 99-111.
- [11]高巍. Android 操作系统软件自动化测试方案的设计与实施[D]. 北京: 北京邮电大学, 2012:440-479.
- [12]孙剑. Android 系统上应用程序按需加载机制的设计与实现[M]. 北京大学, 2011:99-110.
- [13]卢娜. 基于 Android 平台的手机桌面资讯系统的设计与实现[M]. 西安电子科技大学, 2011:290-300.
- [14]高焕堂. Google Android 应用框架原理与程序设计 36 计[M]. Misoo,2010:8-13.
- [15]钟茂生, 王明文. 软件设计模式及其使用[J]. 计算机应用, 2002,22(8): 32-33.
- [16] Gartner November Report[EB/OL]. <http://www.gartner.com/newsroom/id/2482816>, 2013.
- [17] B. John. A binary analysis of resultatives[C]. In Proceedings of Texas Linguistics Society Conference (TLSC), Stanford, California, USA, 1997, 43-58.
- [18] 吴亚峰, 索依娜. Android 核心技术与实例讲解. 北京: 电子工业出版社, 2010
- [19] 王珊, 陈红. 数据库系统原理教程. 北京: 清华大学出版社, 2004.
- [20] 马志强. 基于 Android 平台即时通信系统的设计与实现[D]. 北京交通大学 2009.
- [21] 熊刚. 基于 Android 的智能手机的设计与实现[D]. 武汉理工大学, 2010.

附录

1、AndroidManifest.xml 配置文件

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.zscdumin.zhixinapp">

    <!-- 开启摄像头 -->
    <uses-feature
        android:name="android.hardware.camera"
        android:required="false"/>
    <uses-feature
        android:name="android.hardware.camera.autofocus"
        android:required="false"/>
    <uses-feature
        android:name="android.hardware.camera.setParameters"
        android:required="false"/>

    <uses-permission android:name="android.permission.CAMERA"/>
    <!-- 阅读消息 -->
    <uses-permission android:name="android.permission.READ_SMS"/>
    <!-- 接收消息 -->
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <!-- 读写文文件 -->
    <uses-permission
        android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE"/
    >
    <!-- 网络状态 -->
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission
        android:name="android.permission.CHANGE_NETWORK_STATE"/>
    <uses-permission
        android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission
```

```
        android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission
        android:name="android.permission.CHANGE_WIFI_STATE"/>
    <uses-permission
        android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission
        android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission
        android:name="android.permission.READ_PHONE_STATE"/>

    <application
        android:name="org.litepal.LitePalApplication"
        android:allowBackup="true"
        android:icon="@mipmap/logo"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.Design.Light.NoActionBar">
        <meta-data
            android:name="com.baidu.lbsapi.API_KEY"
            android:value="hpsOsygaYNWКУazSIwAdDW31oqyt9HG"/>
        <meta-data
            android:name="PGYER_APPID"
            android:value="dbe34fedbd7c195aedbfbf1ad0e1836e">    <!-- 由蒲
            公英提供，稍后创建自己的 App id 后更新掉-->
        </meta-data>
        <meta-data
            android:name="SUPERID_APPKEY"
            android:value="4fda3d571675caa5415aac8f" />
        <meta-data
            android:name="SUPERID_SECRET"
            android:value="3a3a1697cc83f7fe9837b997" />

        <activity
            android:name=".activity.WelcomeActivity"
            android:launchMode="singleTask">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

                <category
```

```
        android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
<activity android:name=".activity.WelcomeGuideActivity"/>
<activity android:name=".activity.UserLoginActivity"/>
<activity android:name=".activity.NewsActivity"/>
<activity android:name=".activity.MainActivity"/>
<activity android:name=".activity.WeatherMainActivity"/>
<activity android:name=".activity.WeatherActivity"/>
<activity android:name=".activity.TulingActivity"/>
<activity android:name=".activity.UserRegisterActivity"/>
<activity
    android:name="com.isnc.facesdk.aty.Aty_FaceDetect"
    android:label="@string/app_name"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Holo.Light.NoActionBar">
</activity>
<activity
    android:name="com.isnc.facesdk.aty.Aty_NormalFaceDetect"
    android:label="@string/app_name"
    android:screenOrientation="portrait"

    android:theme="@android:style/Theme.Holo.Light.NoActionBar" >
</activity>
<activity
    android:name="com.isnc.facesdk.aty.Aty_AgreeItem"
    android:label="@string/app_name"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Holo.Light.NoActionBar">
</activity>
<activity
    android:name="com.isnc.facesdk.aty.Aty_Auth"
    android:label="@string/app_name"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.Holo.Light.NoActionBar">
</activity>
<activity
    android:name="com.isnc.facesdk.aty.Aty_EditUserinfo"
    android:label="@string/app_name"
    android:screenOrientation="portrait"
```

```
        android:theme="@android:style/Theme.Holo.Light.NoActionBar">
    </activity>
    <activity
        android:name="com.isnc.facesdk.aty.Aty_ClipPicture"
        android:label="@string/app_name"
        android:screenOrientation="portrait"
        android:theme="@android:style/Theme.Holo.Light.NoActionBar">
    </activity>
    <activity
        android:name="com.isnc.facesdk.aty.Aty_CountryPage"
        android:label="@string/app_name"
        android:screenOrientation="portrait"
        android:theme="@android:style/Theme.Holo.Light.NoActionBar">
    </activity>
    <activity
        android:name="com.isnc.facesdk.aty.Aty_SecureEmail"
        android:label="@string/app_name"
        android:screenOrientation="portrait"
        android:theme="@android:style/Theme.Holo.Light.NoActionBar">
    </activity>
    <service
        android:name=".service.AutoUpdateService"
        android:enabled="true"
        android:exported="true"/>

</application>
```

```
</manifest>
```

2、build.gradle 配置文件代码

```
apply plugin: 'com.android.application'

apply plugin: 'kotlin-android'

apply plugin: 'kotlin-android-extensions'

android {
    compileSdkVersion 26
```

```
defaultConfig {
    applicationId "com.zscdumin.zhixinapp"
    minSdkVersion 15
    targetSdkVersion 26
    versionCode 1
    versionName "1.0"
    testInstrumentationRunner
    "android.support.test.runner.AndroidJUnitRunner"
    multiDexEnabled true
}
buildTypes {
    release {
        minifyEnabled false
        proguardFiles
        getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
}
}
```

```
dependencies {

    implementation 'com.pgyersdk:sdk:2.8.1'
    implementation 'com.qianwen:update-app:3.3.8'
    debugImplementation 'com.squareup.leakcanary:leakcanary-android:1.5.4'
    releaseImplementation
        'com.squareup.leakcanary:leakcanary-android-no-op:1.5.4'
    implementation 'com.isnc.facesdk:superid-sdk:1.6.9'
    implementation 'de.hdodenhof:circleimageview:2.2.0'
    implementation 'cn.bmob.android:bmob-sdk:3.4.6'
    implementation 'com.squareup.okio:okio:1.12.0'
    implementation 'com.jakewharton:butterknife:8.4.0'
    annotationProcessor 'com.jakewharton:butterknife-compiler:8.4.0'
    implementation 'com.squareup.okhttp:okhttp:2.7.5'
    implementation 'org.litepal.android:core:1.4.1'
    implementation 'com.android.support:recyclerview-v7:26.1.0'
    implementation 'com.android.support:cardview-v7:26.1.0'
    implementation 'com.android.support:design:26.1.0'
    implementation 'com.google.code.gson:gson:2.8.0'
    implementation 'com.github.bumptech.glide:glide:3.7.0'
    implementation 'com.squareup.okhttp3:okhttp:3.7.0'
```

```
implementation 'io.reactivex:rxandroid:1.1.0'
implementation 'io.reactivex:rxjava:1.1.0'
implementation fileTree(dir: 'libs', include: ['*.jar'])
implementation "org.jetbrains.kotlin:kotlin-stdlib-jre7:$kotlin_version"
implementation 'com.android.support:appcompat-v7:26.1.0'
implementation 'com.android.support.constraint:constraint-layout:1.0.2'
testImplementation 'junit:junit:4.12'
androidTestImplementation 'com.android.support.test:runner:1.0.1'
androidTestImplementation
'com.android.support.test.espresso:espresso-core:3.0.1'
}
```


致谢

经过一个多月的努力，我的毕业设计终于完成了，原以为毕业设计只是简简单单几天就可以搞完的，现在想想还是有些幼稚。毕业设计要想做好，还是得有些真本事的，仅仅靠大学课堂讲的那点知识是远远不够的。学习是一个循序渐进的过程，所谓活到老学到老大概就是指这个吧。在今后的工作中，我还得不断学习，不断的积累经验，不断提高自己的综合实力。在此，特别想感谢一下我的指导老师梁瑞仕教授，他在我的毕业设计选题、设计以及论文撰写和修改过程中为我指点迷津，使我开拓了思路，梁老师严谨的态度一直影响着我，在此谨向梁老师致以最诚挚的感谢。此外，也要感谢家人朋友的支持与帮助。