

项目实现文档

目录

1	功能实现.....	1
1.1	登陆与交互功能模块.....	1
1.1.1	功能描述.....	1
1.1.2	性能描述.....	1
1.1.3	输入.....	1
1.1.4	输出.....	1
1.1.5	程序逻辑.....	1
1.1.6	限制条件.....	2
1.2	首页功能模块.....	2
1.2.1	功能描述.....	2
1.2.2	性能描述.....	2
1.2.3	输入.....	2
1.2.4	输出.....	2
1.2.5	程序逻辑.....	2
1.2.6	限制条件.....	3
1.3	背单词功能模块.....	3
1.3.1	功能描述.....	3
1.3.2	性能描述.....	3
1.3.3	输入.....	3
1.3.4	输出.....	3
1.3.5	程序逻辑.....	3
1.3.6	限制条件.....	3
1.4	复习功能模块.....	4
1.4.1	功能描述.....	4
1.4.2	性能描述.....	4
1.4.3	输入.....	4
1.4.4	输出.....	4
1.4.5	程序逻辑.....	4
1.4.6	限制条件.....	4
1.5	资讯功能模块.....	5
1.5.1	功能描述.....	5
1.5.2	性能描述.....	5
1.5.3	输入.....	5
1.5.4	输出.....	5
1.5.5	程序逻辑.....	5
1.5.6	限制条件.....	5
1.6	个性化功能模块.....	5
1.6.1	功能描述.....	5
1.6.2	性能描述.....	6
1.6.3	输入.....	6

1.6.4	输出.....	6
1.6.5	程序逻辑.....	6
1.6.6	限制条件.....	6
2	技术实践.....	7
2.1	开发依赖的外部 API.....	7
2.2	数据存储与传输.....	9
2.2.1	SQLite.....	9
2.2.2	MySQL.....	9
2.2.3	Handler 消息传递机制.....	9
2.2.4	SharedPreferences.....	9
2.2.5	数据压缩解压与解析.....	9
2.2.6	OkHttp 网络连接.....	10
2.3	用户界面技术.....	11
2.3.1	transition.....	11
2.3.2	makeSceneTransitionAimation.....	12
2.3.3	night mode.....	12
2.3.4	immersionbar.....	13
2.3.5	BasePopup.....	13
2.3.6	autofittextview.....	14
2.3.7	MPAndroidChart.....	14
2.3.8	material.....	15
2.3.9	androidx.recyclerview.....	16
2.3.10	material-calendarview.....	19
2.4	Android 组件与功能.....	20
2.4.1	Acitivity&Fragment.....	20
2.4.2	Service.....	20
2.4.3	BroadcastReceiver.....	20
2.5	媒体.....	21
2.5.1	MediaPlayer.....	21
2.5.2	WebView.....	21
2.5.3	VideoView.....	22
2.5.4	ImageView.....	22
2.5.5	Glide.....	22
2.5.6	dkplayer.....	22
2.6	后端开发技术.....	23

1 功能实现

1.1 登陆与交互功能模块

1.1.1 功能描述

登录：用户可以通过手机验证码登录，第一次登录系统自动注册；

欢迎页：用户打开 app 时会首先进入欢迎页，欢迎页使用 Bing 图片，每日一句使用金山词霸的每日英文短句，每日都会更新，用户也可以分享；

计划制定：用户可以在系统所给出的词书中进行选择，并设置自定义学习计划。

1.1.2 性能描述

一般操作的响应时刻在 1s 内，数据下载和数据解析并写入数据库时，响应时间在 20s 内。

1.1.3 输入

登录界面：输入手机号及验证码

欢迎页：点击分享

计划制定：选择词书制定每日学习量

1.1.4 输出

登录界面：获得验证码，进入主页面

欢迎页：将每日英文短句分享到各社交平台

计划制定：更改学习计划内的词书和学习量

1.1.5 程序逻辑

- a. 判断是否输入正确手机号
- b. 使用云平台发送验证码
- c. 输入验证码，判断验证码是否正确
- d. 进入主页，获取一句每日英文短句
- e. 点击分享，唤起分享页面
- f. 将获得的每日英文短句以文本形式发送到选择的平台
- g. 选择词书
- h. 确定每日学习量
- i. 在翻译网站获取相应词书的单词资源包
- j. 加载单词资源包

1.1.6 限制条件

- a. 一台内存大小不低于 100MB 的 Android 手机
- b. 安卓版本不小于 Android 6(API 23)。
- c. 设备需要联网

1.2 首页功能模块

1.2.1 功能描述

随心单词：用户可以点击刷新按钮，刷新首页单词；

单词夹：用户可以创建单词夹，将单词存入单词夹中用以学习，也可以将单词夹中的单词删除；

搜索：用户可以模糊搜索当前所选词库中的单词。

1.2.2 性能描述

一般操作的响应时刻在 1s 内，数据下载和数据解析并写入数据库时，响应时间在 20s 内。

1.2.3 输入

随心单词：点击刷新按钮；点击单词背景；进入单词详情后页面点击添加到单词夹

单词夹：创建单词夹，删除单词夹内单词，删除单词夹

搜索：在输入框输入单词

1.2.4 输出

随心单词：刷新单词；点击单词背景会进入单词详情页面；点击添加到单词夹会跳出添加到哪个单词夹的提示

单词夹：创建单词夹，删除单词夹内单词，删除单词夹

搜索：若该单词在学习计划的词书中，会跳出该单词的解释，否则显示暂无查询结果

1.2.5 程序逻辑

- a. 首先获取计划词书内的单词
- b. 随机获得一个单词展示在首页
- c. 点击刷新会更换单词
- d. 点击单词背景进入单词详情展示
- e. 点击添加到单词夹，若无单词夹，提示暂无单词夹，若已经建立过单词夹，会弹出一个窗口，询问存入哪个单词夹
- f. 点击存入单词夹内的单词可以跳转到单词详情，点击移除按可以移除该单词夹内的该单词
- g. 同一个单词夹内不能存入同一个单词

1.2.6 限制条件

- a.一台内存大小不低于 100MB 的 Android 手机
- b.安卓版本不小于 Android 6(API 23)。
- c.设备需要联网

1.3 背单词功能模块

1.3.1 功能描述

对选择的单词可以选择认识/模糊/不认识，如果选择认识，直接掠过这个单词；选择模糊，App 弹出提示；选择不认识，则会详细显示单词的释义，加强对单词的理解。之后背一定数量的单词之后，会进入复习模式，即提示单词的英文，需要我们选择适当的释义。

1.3.2 性能描述

一般操作的响应时刻在 1s 内，数据下载和数据解析并写入数据库时，响应时间在 20s 内。

1.3.3 输入

点击认识、不认识、模糊、复习模式选择释义

1.3.4 输出

如果选择认识，直接掠过这个单词；选择模糊，App 弹出提示；选择不认识，则会详细显示单词的释义，加强对单词的理解。

背一定数量的单词之后会进入复习模式，若选择释义错误会提示错误，并跳至该单词的详细解释页面；选择正确会回到学习单词的界面

1.3.5 程序逻辑

- a. 首先获取计划词书内的单词，随机获得一个单词进行展示
- b. 如果选择认识，直接掠过这个单词；选择模糊，App 弹出提示；选择不认识，则会详细显示单词的释义
- c. 背一定数量的单词之后会进入复习模式，若选择释义错误会提示错误，并跳至该单词的详细解释页面；选择正确会回到学习单词的界面
- d. 学习完计划数量的单词可以进行打卡

1.3.6 限制条件

- a.一台内存大小不低于 100MB 的 Android 手机
- b.安卓版本不小于 Android 6(API 23)。
- c.设备需要联网

1.4 复习功能模块

1.4.1 功能描述

拍照取词：用户可以通过选择相机或相册中的单词，利用百度 OCR 识别文本得到单词列表进行背诵复习；

多词匹配：将汉语和英语相匹配，之后可以进行回顾；

单词速过：快速播放需要复习的单词，可以选择暂停，之后可以进行回顾。

1.4.2 性能描述

一般操作的响应时刻在 1s 内，数据下载和数据解析并写入数据库时，响应时间在 20s 内。

1.4.3 输入

拍照取词：使用相机拍取单词或扫描本地图片

多词匹配：点击英文/中文后点击对应的中文/英文

单词速过：点击暂停和返回

1.4.4 输出

拍照取词：利用百度 OCR 识别文本得到单词列表，并可以添加到单词夹

多词匹配：配对正确会消失，配对错误会提示点错了哦

单词速过：暂停速过功能，返回主页面

1.4.5 程序逻辑

- a. 拍照取词：获得相机和本地相册的使用权，利用百度 OCR 识别文本得到单词列表并展示
- b. 多次匹配：在计划的词书内随机选取 5 个单词，对中英文进行拆分后配对，配对正确会消失，配对错误会提示点错了哦。如果在点了英文/中文之后再点击英文/中文同样会提示点错了哦。全部配对正确会进入本次匹配的 5 个单词的简要解释页面进行再次巩固
- c. 单词速过：在学习过的单词中选出 6 个，进行快速的播放，全部复习之后会进入本次速过的 6 个单词的简要解释页面进行再次巩固

1.4.6 限制条件

- a. 一台内存大小不低于 100MB 的 Android 手机
- b. 安卓版本不小于 Android 6(API 23)。
- c. 设备需要联网

1.5 资讯功能模块

1.5.1 功能描述

悦听：用户可以查看英语新闻、聆听英语音频、查看对应原文与翻译，在页面内支持自动滚动播放，手动定位播放，自动回滚播放位置；

悦看：用户可以观看英语演讲，视频支持自动播放，自动暂停播放，横屏播放。

1.5.2 性能描述

一般操作的响应时刻在 1s 内，数据下载和数据解析并写入数据库时，响应时间在 20s 内。

1.5.3 输入

悦听：点击任意一新闻；手动拖拽英文，点击左边的播放;手动拖拽英文，松手。

悦看：点击播放、暂停；点击全屏；在全屏模式下翻转手机

1.5.4 输出

悦听：新闻自动播放，英文同步滚动；定位至该处，音频同步；跳转到正在播放的位置

悦看：视频播放、暂停；视频由竖屏转为横屏；视频自动旋转

1.5.5 程序逻辑

a.参考绝大多数项目中使用的 RecyclerView+播放器的方式实现

1.5.6 限制条件

a. 一台内存大小不低于 100MB 的 Android 手机

b.安卓版本不小于 Android 6(API 23)。

c.设备需要联网

1.6 个性化功能模块

1.6.1 功能描述

数据统计分析：用户可以查看学习时间，单词输入量，单词占比，打卡日历，在手机通知栏中也会显示今日单词学习任务；

备份：用户可以将学习数据进行备份；

恢复：用户可以将学习数据从云端恢复至本地；

单词列表：查看所有单词释义与读音；

学习计划：修改单词匹配个数，修改单词速记个数，
修改词书计划；夜间模式：修改所有界面颜色为黑夜风格。

1.6.2 性能描述

一般操作的响应时刻在 1s 内，数据下载和数据解析并写入数据库时，响应时间在 20s 内。

1.6.3 输入

数据统计分析：点击首页开始背单词，完成 50 个

备份：点击云端备份

恢复：在未备份情况下点击云端恢复，在备份情况下点云端恢复

单词列表：点击单词后面的黑色条形框

夜间模式：点击纽扣按钮

1.6.4 输出

数据统计分析：学习单词数变为 50，坚持天数+1，铜板+10

备份：提示云端备份完成

恢复：提示暂无备份；一段时间后提示恢复成功

单词列表：黑色条形框消失，出现释义

夜间模式：应用主题变为黑色

1.6.5 程序逻辑

- a. 制定学习计划
- b. 完成每日学习计划进行打卡，获得铜板奖励，将学习过的单词进行记录
- c. 学习提醒：获取通知栏权限，设置时间，时间到达后在通知栏弹出学习信息
- d. 通知栏单词：获取通知栏权限，在通知栏简要显示单词信息，可以进行单词切换和发音等行为
- e. 单词列表：导入词书内所有单词，将释义进行遮盖，点击遮盖处显示释义
- f. 夜间模式：另外设置一套控件的颜色，设置监听时间，按下纽扣按钮后切换方案

1.6.6 限制条件

- a. 一台内存大小不低于 100MB 的 Android 手机
- b. 安卓版本不小于 Android 6(API 23)。
- c. 设备需要联网

2 技术实践

2.1 开发依赖的外部 API

视频列表功能，可以参考绝大多数项目中使用的 RecyclerView+播放器的方式实现。

拍照取词功能，可以使用百度 OCR 实现。

UI 部分主要使用 android.transition,shared element,android.material,night mode 等图形库。

在数据传输部分使用 Handler, SQLite, OkHttp 等。

在单词资源包和 API 方面，主要是使用了百度、有道的相关包和向导。

2.1.1 百度 OCR

提供复习专栏中拍照取词的图像识别功能，将手机相机或者手机相册中的图片，上传到百度接口，获得识别的单词结果，再对单词结果进行本地数据库的匹配，将数据库已有单词显示在列表中。

HTTP 方法：POST

请求 URL：https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic

URL 参数：

参数	值
access_token	通过 API Key 和 Secret Key 获取的百度 access_token

Header：

参数	值
Content-Type	application/x-www-form-urlencoded

Body 中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	说明
image	和 url/pdf_file 三选一	string	图像数据，base64 编码后进行 urlencode，要求 base64 编码和 urlencode 后大小不超过 4M，最短边至少 15px，最长边最大 4096px，支持 jpg/jpeg/png/bmp 格式
url	和 image/pdf_file 三选一	string	本次功能实现不使用 URL

参数	是否必选	类型	说明
pdf_file	和 image/ url 三选 一	string	本次功能实现不使用 PDF 文件
pdf_file_ num	否	string	需要识别的 PDF 文件的对应页码
language_ _type	否	string	识别语言类型，本次功能实现使用 ENG
detect_dir ection	否	string	是否检测图像朝向，默认不检测，即：false。
detect_la nguage	否	string	是否检测语言，默认不检测。
paragraph	否	string	是否输出段落信息
probabilit y	否	string	是否返回识别结果中每一行的置信度

2.1.2 有道 API 调用

在路径变量上注明英音/美音，以及单词的英文，即可得到音频的资源。

[https://dict.youdao.com/dictvoice?type=\[1 表示英音, 0 表示美音\]&audio=\[这里填写具体的单词\]](https://dict.youdao.com/dictvoice?type=[1 表示英音, 0 表示美音]&audio=[这里填写具体的单词])

词书 API

<http://ydschool-online.nos.netease.com/>

2.1.3 金山词霸每日一句 API 调用

以固定形式请求数据，对金山词霸每日一句 API 网址进行请求

<https://open.iciba.com/dsapi/>

2.1.4 必应图片 API 获取

以固定形式请求数据，对必应图片 API 网址进行请求

<https://www.bing.com/HPIImageArchive.aspx?format=js&idx=0&n=1>

2.2 数据存储与传输

2.2.1 SQLite

手机端单词各项数据存储，为实现离线背单词功能，大部分单词相关的数据均存储在本地 SQLite 数据库。本地 SQLite 数据库设计了每日一句信息表 `dailydata`、单词收藏夹对应表 `folderlinkword`、单词释义表 `interpretation`、学习时长表 `learntime`、每日学习情况表 `mydate`、单词短语表 `phrase`、例句表 `sentence`、用户信息表 `user`、用户配置表 `userconfig`、单词学习情况表 `word`、单词收藏夹信息表 `wordfolder`，用于数据存储与使用。

2.2.2 MySQL

后端使用 MySQL 管理数据，主要涉及云端记录备份和恢复功能的数据管理，同时对用户名和密码进行管理。

2.2.3 Handler 消息传递机制

Handler 是一个消息分发对象，进行发送和处理消息，并且其 `Runnable` 对象与一个线程的 `MessageQueue` 关联。Handler 可以调度消息，将一个任务切换到某个指定的线程中去执行。因为子线程不允许访问 UI，所以引入 Handler 来解决子进程无法访问 UI 的问题，可以理解作为一种消息传递。

我们在很多需要多线程处理数据的地方使用到了 Handler。我们在更改计划活动中，开启一个线程分析数据，分析完成利用 Handler 传递消息给主线程；播放音频的子线程播放完毕返回给主线程；图表显示活动处理完数据后也会通过 Handler 传递消息；还有每日单词，学习提醒、登录、OCR 处理、云同步、欢迎、音频处理等等涉及多线程消息传递的，都用到了 Handler。

2.2.4 SharedPreferences

安卓的 `SharedPreferences` 是一个轻量级的存储类，特别适合于保存软件配置参数。（是用 xml 文件存放数据，文件存放在 `/data/data/<package name>/shared_prefs` 目录下）本项目 App 主要对个人配置信息，包括登录状态、夜间模式设置、提醒配置、通知栏配置、个性化设置，在 `SharedPreferences` 中进行保存、维护和读取。

2.2.5 数据压缩解压与解析

压缩解压

ZIP 是一种较为常见的压缩形式，在 Java 中要想实现 ZIP 的压缩需要导入 `java.util.zip` 包，可以使用此包中的 `ZipFile`、`ZipOutputStream`、`ZipInputStream`、`ZipEntry` 几个类完成。
ZipEntry：在每一个压缩文件中都会存在多个子文件，那么这每一个的子文件在 JAVA 中就使用 `ZipEntry` 表示。
ZipOutputStream：如果要想完成一个文件或文件夹的压缩，要使用 `ZipOutputStream` 类完成，`ZipOutputStream` 是 `OutputStream` 的子类。
ZipFile：是一个专门表示压缩文件的类，在 JAVA 中，每一个压缩文件都可以使用 `ZipFile` 表示，还可以使用 `ZipFile` 根据压缩后的文件名称找到每一个压缩文件中的 `ZipEntry` 并将其进行解压缩操作。

ZipOutputStream: “压缩文件输出流”，用于将程序中的压缩流写出到磁盘上。通常需要使用 ZipFile ZipInputStream/ZipOutputStream ZipEntry 级 File 完成解压缩操作。

本项目中封装了 zip 工具类，主要用于通过 url 获取的有道单词包资源的压缩与解压。

解析

Gson 是 Google 提供的用来在 Java 对象和 JSON 数据之间进行映射的 Java 类库。可以将一个 JSON 字符串转成一个 Java 对象，或者反过来。它拥有快速、高效，代码量少、简洁，面向对象，数据传递和解析方便的特点。Gson 提供了 fromJson() 和 toJson() 两个直接用于解析和生成的方法，前者实现反序列化，后者实现了序列化。在集合的序列化和反序列化方面，因为泛型的类型擦除，Java “分辨”不出 List<String> 和 List<User>，对 Java 而言它们的类型都是 List.class。为了解决上面的问题，Gson 为我们提供了 TypeToken 来实现对泛型的支持，所以当我们希望使用将以上的数据解析为 List<String> 时需要将 List<String> “套在” new TypeToken<T>().getType() 中。

本项目中主要使用 Gson 将解压好的单词资源数据包解析成封装好的单词实体类的集合，然后遍历集合，将每一个单词，对应的音标，图片，释义，短语，例句等存入数据库中。

2.2.6 OkHttp 网络连接

OkHttp 是一个处理网络请求的开源项目，Android 当前最火热网络框架，由移动支付 Square 公司贡献，用于替代 HttpURLConnection 和 Apache HttpClient(android API23 6.0 里已移除 HttpClient)。其具有以下优点：1.支持 HTTP2/SPDY（SPDY 是 Google 开发的基于 TCP 的传输层协议，用以最小化网络延迟，提升网络速度，优化用户的网络使用体验）；2.socket 自动选择最好路线，并支持自动重连，拥有自动维护的 socket 连接池，减少握手次数，减少了请求延迟，共享 Socket,减少对服务器的请求次数；3.基于 Headers 的缓存策略减少重复的网络请求；4.拥有 Interceptors 轻松处理请求与响应（自动处理 GZip 压缩）

本项目封装了 OkHttp 工具类，该工具类中包含异步 Get, Post 请求方法，调用时十分简单，下面以异步 post 请求为例：

```
OkHttpUtils.sendOkHttpResponse(url, requestBody, new Callback() {  
    //重写回调方法  
    @Override  
    public void onFailure(Call call, IOException e) {  
        //发送失败后的处理  
    }  
  
    @Override  
    public void onResponse(Call call, Response response) throws IOException {  
        //发送成功，可通过 response.body().string()获取返回信息  
    }  
});
```

本项目主要使用 OkHttp 完成登录注册，备份恢复功能，向服务器发送请求，并处理所响应的信息。

2.3 用户界面技术

2.3.1 transition

在 Android 4.4 Transition 就已经引入了，但在 Android 5.0(API 21)之后，Transition 被更多的应用起来。相对于 View Animation 或 Property Animator，Transition 动画更加具有特殊性，Transition 可以看作对 Property Animator 的高度封装。不同于 Animator，Transition 动画具有视觉连续性的场景切换。

在我们的项目中，我们频繁地使用动画来启动 Activity。我们在 BaseActivity 内封装了两个方法,分别达到窗口缩放式(Explode)出现和滑动式(Slide)出现的动画效果

//滑动式，参数是滑动方向

```
public void windowSlide(int position) {
    getWindow().setEnterTransition(new Slide(position).setDuration(300));
    getWindow().setExitTransition(new Slide(position).setDuration(300));
    getWindow().setReenterTransition(new Slide(position).setDuration(300));
    getWindow().setReturnTransition(new Slide(position).setDuration(300));
}
```

//扩大式

```
public void windowExplode() {
    getWindow().setEnterTransition(new Explode().setDuration(300));
    getWindow().setExitTransition(new Explode().setDuration(300));
    getWindow().setReenterTransition(new Explode().setDuration(300));
    getWindow().setReturnTransition(new Explode().setDuration(300));
}
```

调用举例：

在启动 WordDetailActivity（单词详情页）时，会调用 windowSlide（Gravity.TOP），使窗口从上往下滑入



在启动 FinishActivity（完成今日任务结算页面）、WordFolderActivity（单词夹页面）、LoadWordActivity（加载单词界面）、MatchActivity（多次匹配）、SpeedActivity（单词速过）时，会调用 windowExplode（）方法，使界面显示一个切入的动画，退出窗口时，会显示切出的动画。

2.3.2 makeSceneTransitionAimation

这是 ActivityOptions 的一个静态方法，在 API 21 中添加。

它可以创建一个 ActivityOptions 以使用跨活动场景动画在活动之间进行转换。该方法携带一个共享元素的位置到启动的 Activity。的位置 sharedElement 将用作退出过渡的震中。共享元素在启动的 Activity 中的位置将是其进入 Transition 的中心。

我们在首页的单词搜索功能采用了这个方法，点击搜索按钮，搜索按钮会平移到顶部平滑转变为搜索框。（这类似于 powerpoint 的平滑效果）

调用：

```
ActivityOptionsCompat activityOptionsCompat2 =
ActivityOptionsCompat.makeSceneTransitionAnimation(getActivity(),
    tranSearchView, "imgSearch");
```

参数说明：

tranSearchView 是指定的过渡到已启动的试图

“imgSearch”是目标 Activity 中使用的共享元素名称（这在 SearchActivity 的 xml 文件中以 android:transitionName="imgSearch"的形式声明）

2.3.3 night mode

我们的项目实现了一键切换黑夜模式。

1. 首先需要让主体继承 Theme.AppCompat.DayNight

```
<style name="AppTheme" parent="Theme.AppCompat.DayNight.DarkActionBar">
```



- 2.我们在资源文件中设立了两个 colors.xml 文件，分别存储白天和黑夜模式的颜色数值。
- 3.在用户个人页面的按钮接口处，调用 AppCompatActivity.setDefaultNightMode(mode)，可以设置全局的夜间模式。

2.3.4 immersionbar

自从 android4.4 开始，android 手机状态栏有了透明状态（沉浸式）。

沉浸式实现原理其实是使整个 activity 布局延伸到整个屏幕，然后使状态栏变成透明色，使得 APP 更加的美观。

引入相关库以后，只需要在需要的页面添加如下代码即可：

```
if (ConfigData.getIsNight()) {  
    // 沉浸式状态栏，设置深色  
    ImmersionBar.with(this)  
        .statusBarDarkFont(false)  
        .init();  
} else {  
    ImmersionBar.with(this)  
        .statusBarDarkFont(true)  
        .init();  
}
```

即根据当前是否是黑夜模式设置状态栏的颜色。

2.3.5 BasePopup

BasePopup 是一个开源的便捷的弹窗库

（项目地址：<https://github.com/razerdp/BasePopup>），

我们引入此库后对它进行了进一步的封装，使得弹窗出现时从左侧滑入，离开时从右侧滑出。

MyPopWindow.java

```
@Override  
protected Animator onCreateShowAnimator() {  
    ObjectAnimator showAnimator = ObjectAnimator.ofFloat(getDisplayAnimateView(),  
        View.TRANSLATION_X, -getScreenWidth(), 0);
```

```

        showAnimator.setDuration(animatTime);
        return showAnimator;
    }

    @Override
    protected Animator onCreateDismissAnimator() {
        ObjectAnimator showAnimator = ObjectAnimator.ofFloat(getDisplayAnimateView(),
            View.TRANSLATION_X, 0, getWidth());
        showAnimator.setDuration(animatTime);
        return showAnimator;
    }

```

在 WelcomActivity 中新建一个权限弹出窗，用到自定义的 PopWindow，使弹窗更加美观

```

// 设置权限弹出框
welWindow = new MyPopWindow(this);

```

2.3.6 autofittextview

这是一个可以根据文本长度自动调整 view 长度的控件，我们在单词速过的模块中，在屏幕中心显示单词时使用了这个控件

2.3.7 MPAndroidChart

MPAndroidChart 是一个强大的图表库（项目地址
<https://github.com/PhilJay/MPAndroidChart>）



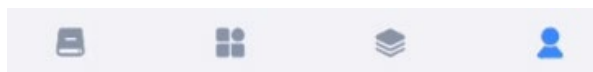
我们在统计用户行为数据时间使用了这个库，将用户背单词的数据可视化，达到较为美观的效果。

2.3.8 material

Material Design（材质设计）是 Google 在 2014 年 I/O 大会上发布的一种新的设计规范。

我们的项目较多地使用了这一设计规范，包含：

`android.material.bottomnavigation.BottomNavigationView` 用于完成底部导航栏



`android.material.tabs.TabLayout` 用于完成悦听、悦看两个板块的顶部导航栏



android.material.appbar.AppBarLayout 用于完成个人页顶部的 appBar



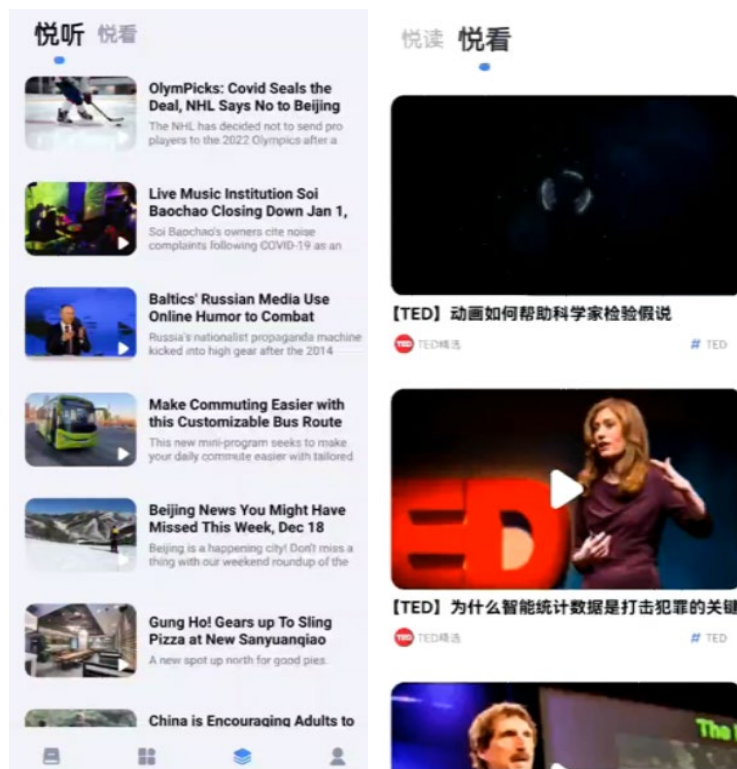
(这里一整个模块是一个 AppBar)

2.3.9 androidx.recyclerview

RecyclerView 是 Android 一个强大的控件，其不仅可以实现和 ListView 同样的效果,还有优化了 ListView 中的各种不足。

本项目大量使用了 RecyclerView 来展示有多个 item 内容的页面。





以实现多词匹配的页面为例，我们在这里实现一个瀑布流的效果，即两列单词，每列里每个卡片的宽度一致，长度自适应。

a.首先利用

```
StaggeredGridLayoutManager layoutManager = new StaggeredGridLayoutManager(2,  
StaggeredGridLayoutManager.VERTICAL);
```

得到一个 2 列垂直分布的瀑布流的 layoutManager

b.在自定义的 MatchAdapter 中，

有三个关键的部分

ViewHolder

```
static class ViewHolder extends RecyclerView.ViewHolder {  
  
    View view;  
    CardView cardView;  
    TextView textView;  
  
    public ViewHolder(View itemView) {  
        super(itemView);  
        view = itemView;  
        cardView = itemView.findViewById(R.id.card_mt);  
        textView = itemView.findViewById(R.id.text_mt_word);  
    }  
}
```

要想使用 RecyclerView 就需要编写一个 Adapter 将数据适配到 RecyclerView 上，而为了节省资源提高运行效率，一般自定义类 ViewHolder 来减少 findViewById() 的使用以及避免过多地 inflate view，从而实现目标。

2.onCreateViewHolder

```
@Override  
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_match, parent,  
false);  
    final ViewHolder holder = new ViewHolder(view);  
    /.../  
}
```

这个部分的作用是初始化 ViewHolder

3. onBindViewHolder

```
public void onBindViewHolder(ViewHolder holder, int position) {  
    ItemMatch itemMatch = mItemMatchList.get(position);  
    holder.textView.setText(itemMatch.getWordString());  
    /.../  
}
```

这个部分的作用是将数据与 item view 进行绑定，这样滚动视图时，保证数据能更新。

c.最后为 recyclerView 分别设置 layoutManager 和 adapter，完成布局的设置

```
recyclerView.setLayoutManager(layoutManager);  
recyclerView.setAdapter(matchAdapter);
```

2.3.10 material-calendarview

这是一个有着 material design 风格的日历视图，（项目地址

<https://github.com/prolificinteractive/material-calendarview>）

我们用这个开源项目库完成了打卡日历功能的设计



2.4 Android 组件与功能

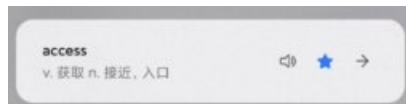
2.4.1 Activity&Fragment

在我们的项目中，主页面有一个 MainActivity 以及四个 Fragments 组成。由前面提到的 bottomNavigation 来实现四个 fragments 之间的切换。

而在每个 fragment 内，用户的操作可能会启动新的 Activity。如用户在个人页面点击通知栏单词设置，则会启动一个 LearnInNotifyactivity。

2.4.2 Service

Service 可以在后台完成一些工作。在我们的项目中，我们在实现通知栏显示单词时使用到了 Service



在 LearnInNotifyActivity 中，如果用户点击按钮打开，则调用 startService 方法启动服务，如果用户点击按钮关闭，则调用 stopService 关闭服务。

```
aSwitch.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
    @Override  
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
        if (isChecked) {  
            //.../  
            startService(ConfigData.getNotifyLearnMode());  
        } else {  
            //.../  
            stopService();  
        }  
    }  
});
```

2.4.3 BroadcastReceiver

BroadcastReceiver 即广播接收者。例如，他可以接收到系统开机完成的广播，以及系统电量不足的广播，以及系统收到短信的广播等等。

在我们的项目中，我们继承自 BroadcastReceiver 设计了 AlarmReceiver，用于实现英语单词学习的闹钟提醒。

```
public class AlarmReceiver extends BroadcastReceiver {
```

```

private static final String TAG = "AlarmReceiver";

@Override
public void onReceive(Context context, Intent intent) {
    /.../
    // 第二天的这个时间再提醒
    if (ConfigData.getIsAlarm()) {
        int hour = Integer.parseInt(ConfigData.getAlarmTime().split("-")[0]);
        int minute = Integer.parseInt(ConfigData.getAlarmTime().split("-")[1]);
        AlarmActivity.startAlarm(hour, minute, true, false);
    }
}
}
}

```

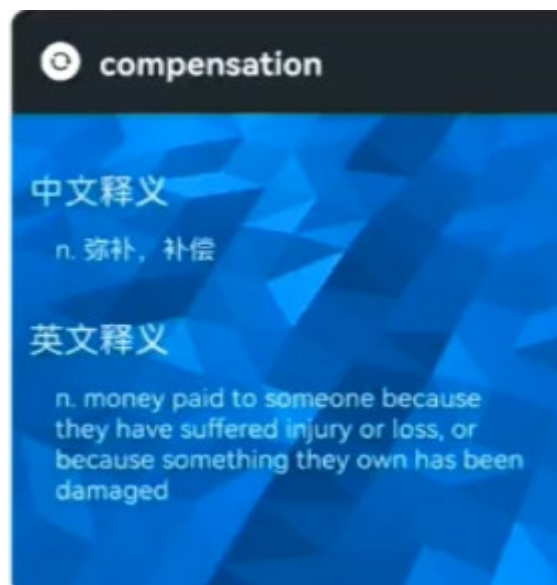
2.5 媒体

2.5.1 MediaPlayer

我们的项目是一个背单词 APP。因此涉及到单词播放的地方大量使用到了 MediaPlayer。除此之外，我们设计的“悦听”板块，实现了英语新闻的类歌词播放，同样也用到了 MediaPlayer

2.5.2 WebView

在单词的主页面，我们为主单词卡片实现了一个动态的壁纸效果。具体做法是利用 webview 加载 html，并利用 javascript 实现复杂的动态动画。



2.5.3 VideoView

VideoView 主要在“悦看”板块实现视频滚动自动播放功能的同时使用。

2.5.4 ImageView

在设置静态图片时，我们大量使用到了 ImageView。如单词详情的图片、按钮的背景图片等等。

2.5.5 Glide

Glide 是一个快速高效的 Android 图片加载库，注重于平滑的滚动。Glide 提供了易用的 API，高性能、可扩展的图片解码管道（decode pipeline），以及自动的资源池技术。

我们的项目在加载图片的地方也大量使用了 Glide 框架

例如，在“悦听”板块，加载英语文章的图片时，我们这样使用 Glide：

```
Glide.with(holder.mTitleImg.getContext())  
    .load(articleBean.getImgUrl()) //需要加载的图片 url  
    .placeholder(android.R.color.black)  
    .into(holder.mTitleImg);
```

2.5.6 dkplayer

dkplayer 是一个基于 IjkPlayer 的视频播放器，它能够支持重力感应自动全屏。

在阅读了 dkplayer 的部分源码后，我们能让它很好地适应我们的项目，并与 RecyclerView 结合实现了滚动自动播放的效果。



具体的做法是为 RecyclerView 设置监听事件，滚动停止时，通过遍历视图中的每一个 item，依据他们距离顶部的距离判断适合播放的 video，完成滚动停止的自动播放。


```

mRecyclerView.addOnScrollListener(new RecyclerView.OnScrollListener() {
    @Override
    public void onScrollStateChanged(@NonNull RecyclerView recyclerView, int newState) {
        super.onScrollStateChanged(recyclerView, newState);
        if (newState == SCROLL_STATE_IDLE) { //滚动停止
            autoPlayVideo(recyclerView);
        }
    }
});

private void autoPlayVideo(RecyclerView view) {
    if (view == null) return;
    //遍历 RecyclerView 子控件,如果 mPlayerContainer 完全可见就开始播放
    int count = view.getChildCount();
    L.d("ChildCount:" + count);
    for (int i = 0; i < count; i++) {
        View itemView = view.getChildAt(i);
        if (itemView == null) continue;
        VideoRecyclerViewAdapter.VideoHolder holder =
            (VideoRecyclerViewAdapter.VideoHolder) itemView.getTag();
        Rect rect = new Rect();
        holder.mPlayerContainer.getLocalVisibleRect(rect);
        int height = holder.mPlayerContainer.getHeight();
        if (rect.top == 0 && rect.bottom == height) {
            startPlay(holder.mPosition);
            break;
        }
    }
}

```

2.6 后端开发技术

本次项目后端采用的是 Django 框架。Django 是高水准的 Python 编程语言驱动的一个开源模型、视图、控制器风格的 Web 应用程序框架。Django 架构可以方便、快捷地创建数据库驱动的应用程序。Django 框架的核心组件中用于创建模型的对象关系映射和便捷生成的用户管理界面都在本次项目中得到了体现。本次后端开发实现了登录、注册、验证、云端资源上传与下载，以及日志记录等等接口，并实现了相应数据库的映射模型。本次项目后端已经部署到阿里云服务器。

悦词后台管理系统

用户名:

密码:

登录

悦词后台管理系统

首页, 悦词, 用户备份与恢复日志

开始输入以筛选...

悦词

用户+ 增加

用户备份与恢复日志+ 增加

认证和授权

用户+ 增加

组+ 增加

选择 用户备份与恢复日志 来修改

动作

执行24 个中 0 个被选

☐ 用户备份与恢复日志

☐ 15848978071 恢复操作 | 2021-12-30 04:52:54

☐ 15848978071 上传操作 | 2021-12-30 04:51:56

☐ 15848978071 恢复操作 | 2021-12-30 04:51:46

☐ 15848978071 上传操作 | 2021-12-30 04:51:25

☐ 15960005091 恢复操作 | 2021-12-28 14:03:19

☐ 13332445761 上传操作 | 2021-12-28 13:19:54

☐ 13332445761 恢复操作 | 2021-12-28 12:59:52

☐ 13332445761 恢复操作 | 2021-12-28 12:59:22

☐ 15960005091 恢复操作 | 2021-12-28 12:00:20

☐ 15960005091 上传操作 | 2021-12-28 11:59:27

☐ 15960005091 上传操作 | 2021-12-28 11:56:41

☐ 13332445761 恢复操作 | 2021-12-27 18:05:20

☐ 13332445761 上传操作 | 2021-12-27 17:51:55

☐ 13332445761 上传操作 | 2021-12-27 17:49:39

24