**整体开发流程**

app采用：云服务器+原生态app+第三方API+爬虫

**一、界面（5+7+4）**

1.1app启动页面，登录选择界面，qq登录注册页面，账号登录页面，账号注册页面

1.2首页，复习页面，练习页面，个人中心，单词背诵页面，练习题页面，浏览页面

1.3个人基本信息页面，方向页面，得分页面，设置页面

1. **界面实现过程**

**2.1app启动页面：**

判断用户是否登录：

BmobUser.*getCurrentUser*(User.class);

如果登录就直接跳转到首页，未登录就跳转到登录页面

使用定时器定时跳转

**2.2登录选择界面：相对布局**

界面上面一个app图标，下面两个登录选择，第三方qq登录，账号登录

**2.3第一qq登录时，需要进行基本信息的注册：**

包括用户的基本信息，用户基本信息是由，

用户名：

密码：

邮箱：

学校：

等级：

使用正则表达式，将其约束规范

**2.4账号登录**

用户可以根据自己是否注册有账号，可以输入账号，密码进行登录，也可以点击注册进行注册，账号密码使用正则表达式，对输入的信息进行规范约束

**2.5账号注册**

包括用户的基本信息，用户基本信息是由，

用户名：4-10个字符组成

密码： 4-10字符组成

邮箱： 邮箱必须输入贵方的邮箱才可以

学校： 后面必须是“大学”“学院”

等级：

以上使用正则表达式进行规范约束，必须输入相应的规范数据才可以进行注册

**2.6首页**

记录当前的等级所需要背诵的单词数量，复习的单词数量，可以进行打卡，记录袭击背诵的数量，可以在首页进行选择，是进行背诵单词，还是进行复习，还是进行练习四六级题，可以在首页进行修改当前的计划，可以修改到四级，六级，雅思，托福

**2.7复习**

复习页面主要可以进行单词的双向翻译，可以进行中文-》英文的翻译，英文-》中文翻译,可以进行单词的复习，根据当前用户类型的不一样进行单词的联系，可以选择今日背诵单词，也可以进行选择复习，随机背诵单词，复习题还可以选择复习，选择做联系题

**2.8练习**

选类型题进行做，可以选择翻译题，选择匹配题，选择完形题，选择阅读题，选择阅读题，选择作文题，进行完成，在页面可以进行分享到微信朋友圈，微信群，qq群聊等,在头部导航栏，可以进行外部资源的访问

**2.9个人中心**

个人中心，用户可以对自己的基本信息，进行设置，可以查看自己的基本信息，方向，得分等等，设置里面可以对学校，方向，词汇难度进行设置，可以退出登录

**2.10单词**

单词主要采用materidesign框架，进行界面的搭建，单词界面可以进行单词意思的翻译，进行单词的收藏，单词的发硬，对于掌握的单词，可以进行斩，斩或收藏以后，单词将添加到数据库，单词收藏本上面，以便复习的时候进行调用

，还可以进行下一个，上一个等背诵

**2.11练习题**

练习题主要采用materidesign框架，进行界面的搭建，用户可以在这里进行练习题，对于题可以进行收藏，可以提交答案，查看答案，对于翻译题，作文无法进行答案的校验，只能参考答案，对于匹配题，完形题，阅读题，则可以进行答案的校验，采用正则表达式，对答案进行验证，校验，在用户提交答案以后，该题将会进入数据库的联系题收藏本中，复习的时候可以使用

**2.12个人基信息**

显示用户的基本信息，包括学校，等级，方向，词汇难度，得分等等

1. **第三方登录实现过程：调用sdk**
2. 首先是到腾讯开发者平台注册账号
3. 再进行申请appId

（3）在开发的时候注册进行app

Tencent.*createInstance*("101552311", this.getApplicationContext());

1. 实现接口：

对用户登录情况的处理，对于用户登录失败，登录成功，取消登录等

****

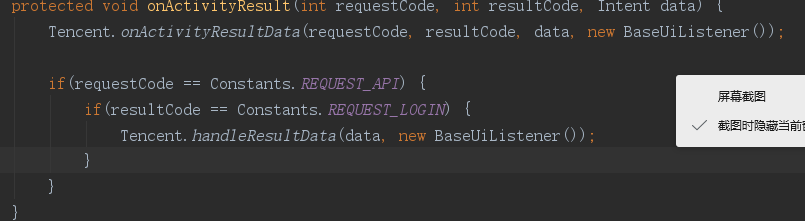
1. 实例化监听器：

对用户登录的情况，可以进行对用户的基本信息的获取，包括名字，头像等基本信息

****

1. 回调验证：

用于判断用户的登录情况，是否已经拉取授权等等

****

1. 登录：

传入登录的监听实例，拉取第三方qq，界面显示当前qq用户吗，用户验证授权登录

**4**

1. **第三方的有道api调用过程 第三方sdk**
2. 在有道开发者平台注册appid
3. app中注册

YouDaoApplication.*init*(this, "6c8f597c03a9e45d");

1. 对资源查询设置：

保存对翻译的word,进行翻译的方向，翻译的资源等等

//查词对象初始化，请设置source参数为app对应的名称（英文字符串）  
Language langFrom = LanguageUtils.*getLangByName*(from);  
//若设置为自动，则查询自动识别源语言，自动识别不能保证完全正确，最好传源语言类型  
//Language langFrom = LanguageUtils.getLangByName("自动");  
Language langTo = LanguageUtils.*getLangByName*(to);  
  
TranslateParameters tps = new TranslateParameters.Builder()  
 .source("ydtranslate-demo")  
 .from(langFrom).to(langTo).build();  
  
Translator translator = Translator.*getInstance*(tps);

1. 资源回调：

通过查询第三方，进行翻译input，加入监听器实例

translator.lookup(input, "requestId", new TranslateListener() {  
@Override  
 public void onError(TranslateErrorCode translateErrorCode, String s) {  
 }  
 @Override  
 public void onResult(final Translate translate, final String s, final String s1) {  
 translateWord=translate;  
 runOnUiThread(new Runnable() {  
 @Override  
 public void run() {  
 intiData();  
 }  
 });  
 }  
 @Override  
 public void onResult(List<Translate> list, List<String> list1, List<TranslateErrorCode> list2, String s) {  
 }

1. 资源解析：

对于返回的Translate数据进行解析，解析翻译的意思，当前单词的发音资源等等

private String listStr(List<String> list) {  
 StringBuilder sb = new StringBuilder();  
 if (list != null) {  
 for (String s : list) {  
 sb.append(s).append("\n");  
 }  
 }  
 return sb.toString();  
}  
public String webMeans(Translate translate) {  
 StringBuilder sb = new StringBuilder();  
 List<WebExplain> explains = translate.getWebExplains();  
 if (explains != null) {  
 for (WebExplain s : explains) {  
sb.append(s.getKey()).append(":").append(listStr(s.getMeans())).append("\n");  
 }  
 }  
 return sb.toString();  
}

1. 手机媒体调用：

上面几步，进行数据的获取解析，可以获取到，单词发音的资源，手机调用媒体，加载资源进行发音

//发音  
public synchronized void playVoice(String speakUrl) {  
 YouDaoLog.*e*(AudioMgr.*PLAY\_LOG* + "TranslateDetailActivity click to playVoice speakUrl = " + speakUrl);  
 if (!TextUtils.*isEmpty*(speakUrl) && speakUrl.startsWith("http")) {  
 AudioMgr.*startPlayVoice*(speakUrl, new AudioMgr.SuccessListener() {  
 @Override  
 public void success() {  
 YouDaoLog.*e*(AudioMgr.*PLAY\_LOG* + "TranslateDetailActivity playVoice success");  
 }  
 @Override  
 public void playover() {  
 YouDaoLog.*e*(AudioMgr.*PLAY\_LOG* + "TranslateDetailActivity playover");  
 }  
 });  
 }  
}

1. **后端云调用过程：sdk**
2. bomb云端注册appid
3. app中进行注册：

使用appi进行注册，只有注册以后才可以使用

Bmob.*initialize*(this, "b73504c6580b52655febd75a7f88af5c");

1. 用户注册：

传入用户用例，进行注册，实现匿名的SaveListener监听器，回调有对有错

user.setUserSchool(school);  
user.setUserHeaderPic("null");  
user.setUserLevel(level);  
user.setEmail(email);  
user.setUserSignature("null");  
user.setUserUseDate(date);  
user.signUp(new SaveListener<User>() {  
 @Override  
 public void done(User user, BmobException e) {  
 if (e == null) {  
 Intent intent=new Intent(RegisteActivity.this,LoginUser.class);  
 startActivity(intent);  
 finish();  
 } else {  
 Toast.*makeText*(RegisteActivity.this,"注册失败:用户名或邮箱已存在",Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
});

1. 数据查询：

写bql语句，这是它特定的sql语句，BmobQuery ，进行查询，结果集返回集合泛型，传入监听器的实例，对数据处理，解析

String bql ="select \* from practiceType where practiceTypeId= ?";  
new BmobQuery<PracticeType>().doSQLQuery(bql,new SQLQueryListener<PracticeType>(){  
 @Override  
 public void done(BmobQueryResult<PracticeType> result, BmobException e) {  
 if(e ==null){  
 List<PracticeType> list = (List<PracticeType>) result.getResults();  
 if(list!=null && list.size()>0){  
 PracticeType practiceType=list.get(0);  
 practiceTypeName=practiceType.getPracticeTypeName();  
 title.setText(practiceTypeName);  
 }else{  
 Log.*i*("smile", "查询成功，无数据返回");  
 }  
 }else{  
 Log.*i*("smile", "错误码："+e.getErrorCode()+"，错误描述："+e.getMessage());  
 }  
 }  
},practiceTypeId);

1. 数据更新：

将User对象的信息进行更新，必须以ObjectId为唯一标识，进行更新，传入一个监听器的实例

user.update(new UpdateListener() {  
 @Override  
 public void done(BmobException e) {  
 if (e == null) {  
 Toast.*makeText*(context, "更新成功", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(context, "更新失败", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
});

1. 用户注销

注销代码

BmobUser.*logOut*();

1. **App图标**

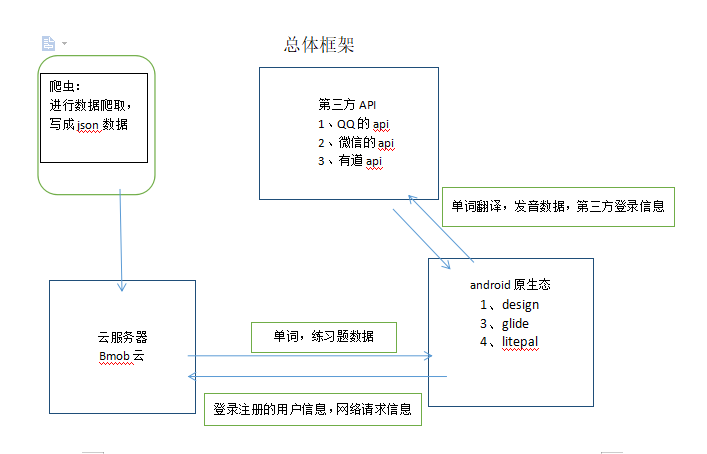
**使用ps进行P图，在百度上面进行获取资源**

1. **数据爬虫：python**

****

****

**八、整体框架：**

****

**总结：**

**技术难点：**

使用到几大技术：

1. 第三方登录验证：

采用qq登录，验证，拉取用户的授权

使用第三方qq的sdk进行信息获取，包括用户的名称，头像等等

1. 第三方的资源翻译

必须对当前app进行注册，采用第三方有道api进行资源的翻译，包括单词的翻译，单词的发音资源等等，使用有道的sdk资源

1. 网络爬虫

该网络爬虫，主要可以进行对四六级，雅思，托福等单词的数据进行爬取，首先爬取每个连接，有了连接在进行网页信息爬取，进行解析网页信息，将爬取的信息进行正则划分，将信息以json的方式，写入到文本中（采用深度优先）

1. 数据加载缓存，自动登录，动态交互

采用glide进行图片的三级缓存，把用户的基本信息写入到本地，用于下次打开自动登录，采用属性动画，以及materdesign框架的动态交互，使得用户体验更好

后台bmob云站账号：

用户名：[1477452395@qq.com](mailto:1477452395@qq.com)

**密码：123456789fengwen**

**微信开发者平台：**

**账号：[1477452395@qq.com](mailto:1477452395@qq.com)**

**密码：123456789fengwen**