## 一、功能模块

项目目前分为二大模块，分别为：导出模块，自动创建模块

## 二、重要模块功能及相关类

自动创建模块：实现自动创建模板，控制器，视图。

重要的相关类与接口：

Carousel:实现图片的轮播效果，默认的轮播间隔为5秒

Indicator：指示当前图片是第几页，可以修改CircleNavigator改变其样式。

文章模块：显示文章，分为列表显示和详情显示。

重要的相关类与接口：

ArticleListActivity:显示文章列表

ArticleDetailActivity:通过webview显示文章详情，后面可以增加相关文章链接和评论功能。

活动模块：显示活动列表和活动详情，提供报名接口，此模块也用在发现模块中。

重要的相关类与接口：

ActListActivity:显示推荐活动列表

ActDetailActivity:显示活动的详情

动态模块：显示用户发送的动态，只显示有图片的动态

重要的相关类与接口：

MomentDetailActivity：动态详情，可以对动态评论及点赞，并可以对评论进行回复。

推荐路书：官方提供一些路线供用户使用

重要的相关类与接口：

RecommendRouteDetailActivity:显示路书的详情，可以对路书进行评论和使用。

俱乐部模块：推荐俱乐部和显示同城的俱乐部

重要的相关类与接口：

LocClubFragment:显示同城的俱乐部，通过定位获取到的位置进行判断

RecClubFragment:显示推荐的俱乐部，后台进行设置

SearchActivity:用于搜索俱乐部

发现骑友模块：用于寻找朋友和搜索朋友，推荐规则和页面显示需要一些改进

重要的相关类和接口：

FriendsActivity:使用RecyclerView显示推荐的骑友，推荐规则由后台定义。

运动模块：较为复杂的模块之一，用于记录和显示用户数据及轨迹，有启动，暂停及停止三个状态，停止运动时判断运动距离是否超过1000m，若超过则保存用户数据，否则不记录。在运动过程中也可以通过地图页面查看自己的运动轨迹。

重要的相关类和接口：

SportFragment:用户数据显示和与用户交互的页面

SportService：后台进程，用于后台定位及数据计算。

MapActivity：显示和追踪用户轨迹，显示用户数据

个人中心模块：二维码生成和扫描使用了谷歌的ZXing开源框架，开源在QRcode模块中查看相关代码。

更多模块：在附近车店的功能使用了百度的LBS云提供的服务，通过在后台设置相应的POI，实现了在某个位置确定附近范围的车店的功能，并通过Uri挑战到百度地图或者高德地图实现路线导航。

通用辅助工具类介绍：

除app模块之外，其他模块均是抽取出来的通用模块可以供其他的项目使用，imagepicker模块是一个图片选择器框架，在第三方的开源框架下有所更改，高仿微信图片选择的效果；

indicator模块是专门用过ViewPager指示器的一个框架，目前只定义了圆形的指示器，可以根据实际需求进行更多的拓展；

lib模块是通用的开发组件集合，为了便于开发而抽取出来的通用工具类，一些方法和类在此项目中并没有用到，例如CrashHandler等，每个包下封装了不同的通用类，可以通过集成或其他方式进行使用，比如widget包下封装了滚轮选择器等等；

map模块是对百度地图的一些封装，用以降低百度地图的耦合性，同时实现更为简便的操作，比如定位，路线规划，百度鹰眼服务等；

photoview是一个图片浏览器框架，在第三方开源框架的基础上进行修改，实现了比第三方较为复杂的功能，比如点击关闭等；

Qrcode模块是用于进行二维码相关操作的框架，封装了扫描二维码和生成二维码的相关类；

xrecyclerview模块是使用第三方开源框架，实现的是对RecyclerView的再封装和功能的增加，通过该框架可以轻易实现下拉加载和上拉加载更多等功能。

## 三、附加说明

由于一些原因，项目采用了MVC与MVP混合的架构模式，主要采用的是MVC架构模式，部分页面使用的是MVP架构模式，主要功能区分按照包名区分，项目除了基本的网络请求，页面显示等业务外，短信验证码使用的是Mob提供的短信验证码服务，Bug监测和应用升级使用的是腾讯bugly，图片加载使用的是Glide，网络加载模块使用了AsyncHttpClient和Retrofit两个第三方框架，为实现缓存的更好实现，预计后期将AsyncHttpClient逐步替换为Retrofit，同时配合Retrofit使用了RxJava和RxAndroid两个异步加载框架。即时通信方面使用了环信以及使用了环信开源的EaseUI，地图模块使用的百度地图的技术支持，其中还使用了一些其他的第三方开源框架。

路线追踪：路线追踪有两种实现方式，一种是利用百度提供的鹰眼SDK，一种是利用定位SDK计算当前点的位置记录并连接。第一种方式能够实现较为准确的定位，但是耗能较多，第二种方式在定位上不够准确，但是耗能相对较小。目前项目使用的是定位SDK，路线追踪的大体实现思路为：在后台获取到当前点，判断是否在这个点上停留，若停留状态则不记录当前点，若停留则记录当前点，再根据距离计算工具计算出两个记录点的距离相加，计算出总距离和时间求出平均距离。

路线规划：路线规划使用的是百度地图提供的路线规划相关的SDK，目前存在的问题是如果使用步行规划的方式中间无法添加途经点，使用驾车的方式可以添加途经点但由于驾车与骑车存在差异，会造成路线不准确的情况，该问题尚未解决，目前使用的依然是驾车的方式进行路线规划。使用路线规划过程中较为复杂的是选点列表的编辑与路线点之间数据交互，当编辑选点列表时注意路线的重新规划以及起止点和途径点的顺序安排。