1. 工程结构：

workspace

main project

pods project

Model

View

Controller

Workspace:工作空间，是project的集合。在iOS开发中，可以通过cocoapods集成第三方的库以节约开发成本，在工程结构中，为了方便维护和测试，这些第三方库往往是以一个工程的形式集成到项目中，因此需要workspace来统一管理这些第三方库和主工程。

Main project:主工程，以MVC结构创建，分模块实现主工程中每个模块的功能。

Pods project:第三方继承的库，以辅助工程的形式共存在workspace中。

重要的目录文件说明：

* 1. AppDelegate.h / Appdelegate.m：执行App生命周期的一系列函数，包括了App启动、进入后台，返回前台，收到消息、完成加载等，可以在这里处理相关的一些以app为对象的操作，比如收到消息推送、注册第三方的sdk等动作。
  2. Assets.xcassets：资源文件夹。资源文件夹本质上是一个路径，任何存放在这个文件夹内部的图片、pdf或者其他可支持的资源都可以通过XXWithName:的方法来获取，简而言之，这个文件夹就是一个资源路径默认文件夹。
  3. LaunchScreen.storyboard：App启动界面，可以直接在这里定义App的启动界面，也可以在Assets.xcassets中设置LaunchImage，但是两种方法不能同时存在。
  4. OrderBreakfast.app：编译产生的二进制文件。

2．模块划分：

2.1 Tabbar：App的界面结构是一个TabbarController，TabbarController统一管理了三个Tab，也就是对应了三个主要模块，TabbarController负责控制Tabbar的切换，定义其下部的Tabbar item的颜色、图标、文字等属性。每个Tabbar Item也是一个Controller，本App采用了传统的导航树转场模式，因此每个模块应用了iOS自带的UINavigationController，用进栈和出栈的方法来处理页面跳转。因此下面每个模块的Controller都由一个UINavigationController持有。

2.2 首页模块：（对应模块名：Home）

HomeMainController.h/HomeMainController.m：首页两个包含了2个按钮的主界面。

CreateOrderController.h/CreateOrderController.m：我要下单界面，本质上是一个列表界面。

CreateOrderCell.h/CreateOrderCell.m：我要下单列表的单元子视图，被他的父视图所调用。

CreateOrderAddressCell.h/CreaOrderAddressCell.m：我要下单的地址子视图。

2.3 订单模块：（对应模块名：GetOrder）

OrderMainController.h/OrderMainController.m：查看订单列表的主控制器，他持有了2个OrderListViewController，用于显示2个不同样式的列表。

OrderListViewController.h/OrderListViewController.m：订单列表的控制器，用于显示订单列表。

SearchViewController.h/SearchViewController.m：搜索订单列表，后续可以扩展为搜索其他比如用户等信息的列表。

2.4 个人模块：（对应模块名：Me）

MeMainController.h/MeMainController.m:个人中心，包含了和个人相关的操作。

MyAddressListController.h/MyAddressListController.m：我的地址管理列表，用于显示用户已经添加了的用户地址。

MyAddAddressController.h/MyAddAddressController.m：新建地址列表。

3. 通用模块：为了将自定义的通用模块和系统模块区分，我们在项目中为所有通用模块添加了LC（Logical Common）前缀。

3.1网络模块：LCNetworkCenter,AFNetworking库已经对iOS原生的NSURLSession类进行了封装，而为了使用方便，我们对AFNetworking进行二次封装，抽离出Http中常用get和post的请求方法，用接口的路径作为参数，并且将接口获取到的json数据自动转化为iOS直接可用的model。在json转model的方法中，我们使用了比较方便的YYModel。

3.2 宏定义模块：LCLibConstants：将项目中常用的比较长的语句抽离，做成宏定义集合，比如屏幕宽度#define SCREEN\_WIDTH [[UIScreen mainScreen] bounds].size.width 等，方便修改。

3.3 用户模型：LCAccount：由于用户的个人数据多且复杂，又经常会在各种场景中用到，因此将用户个人抽离做成一个模块，统一管理一些琐碎的用户数据，比如用户的登录态、性别年龄、是否已经绑定手机号码等信息。

4. 主要的第三方库介绍：

4.1 AFNetworking：网络访问库，用于实现网络请求

4.2 Masonry：第三方的界面约束库，用于对界面进行自动布局，适应各种屏幕大小

4.3 MLTransition：全局左滑返回上一层手势

4.4 SDWebImage：通过url的方式访问图片，并且实现了图片的缓存管理

4.5 YYKit：对已有iOS常用类进行一些通用方法的扩展

4.6 WMPageController：分页显示不同Controller的一个第三方库