**一、实验题目**

外卖小助手

**二、实验目的**

1.掌握 Android 中的菜单及导航框架。

2.掌握自定义布局。

3.掌握 Android 中的数据存储。

**三、总体设计**

**1.需求分析**

外卖小助手的主要用户群体包括上班族、学生、家庭用户等，他们需要一个方便快捷的外卖订购平台。用户需求具体分析如下：

**便捷性**：能够快速、方便地下单，无需外出即可享受美食；**多样性**：有多种餐饮选择，包括不同类型的菜系、价格区间；**食品安全与质量**：用户关心食物的卫生和质量，希望餐食安全可靠；**实时跟踪**：能够实时跟踪订单状态和配送进度；**优惠和促销**：能够通过优惠券、折扣和会员特权等方式获得价格优惠；**支付便利**：用户需要多种支付选项，如在线支付、到付等；**客户服务**：在遇到问题时能够获得及时有效的客户支持。

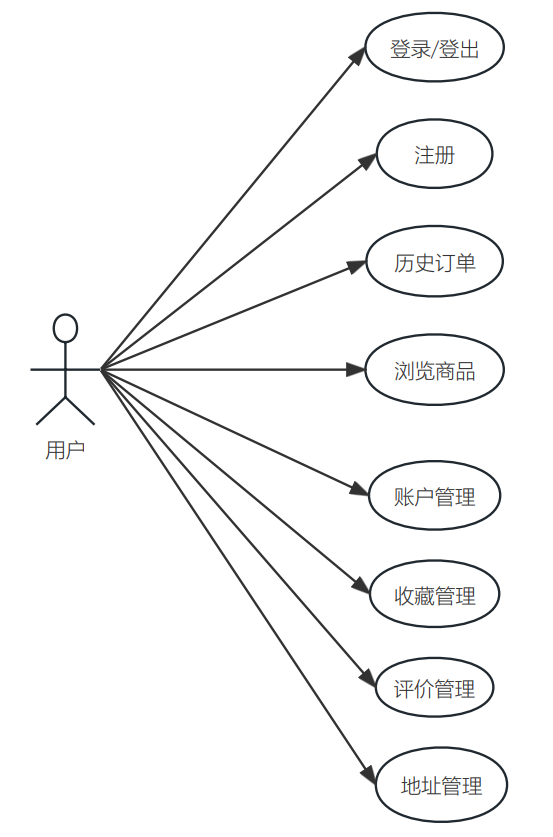


图1 用例图

**2.模块功能介绍**

由于时间有限和个人能力不足，只实现了小部分的需求。

（1）登录注册模块

该模块负责用户的注册、登录和身份验证。新用户使用手机号进行注册时，如果账户已经存在跳转密码登录界面，不存在则自动创建账户并初始化密码为“123456”。

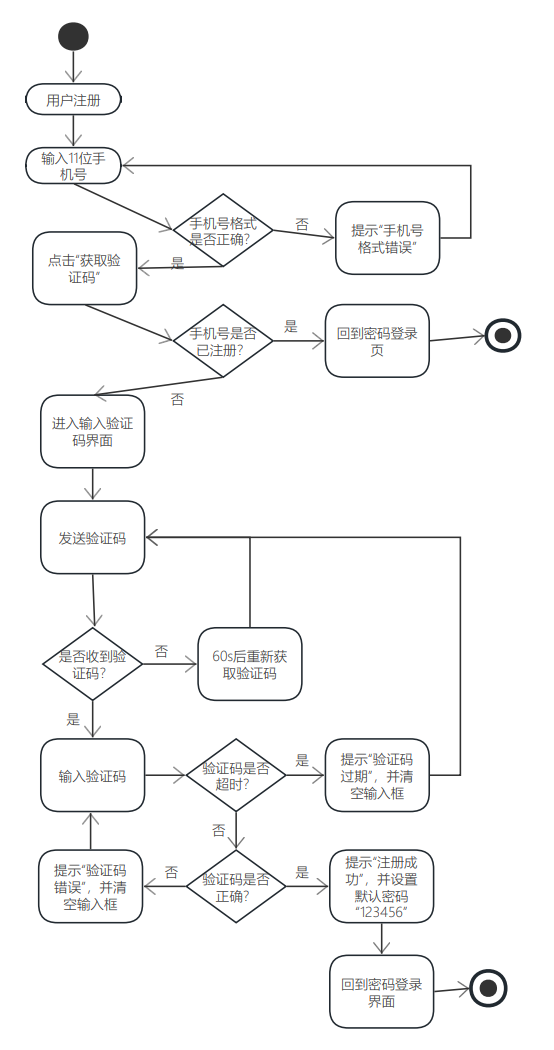


图2 用户注册活动图

（2）商品展示模块

首页展示所有用户默认地址周围的商家店铺及其基础信息，并按照美食、甜点饮品、超市便利、蔬菜水果进行分类；对每一个店铺提供的商品进行分类，便于用户查找；用户可以将自己想要的商品添加进购物车，也可以收藏该店铺。

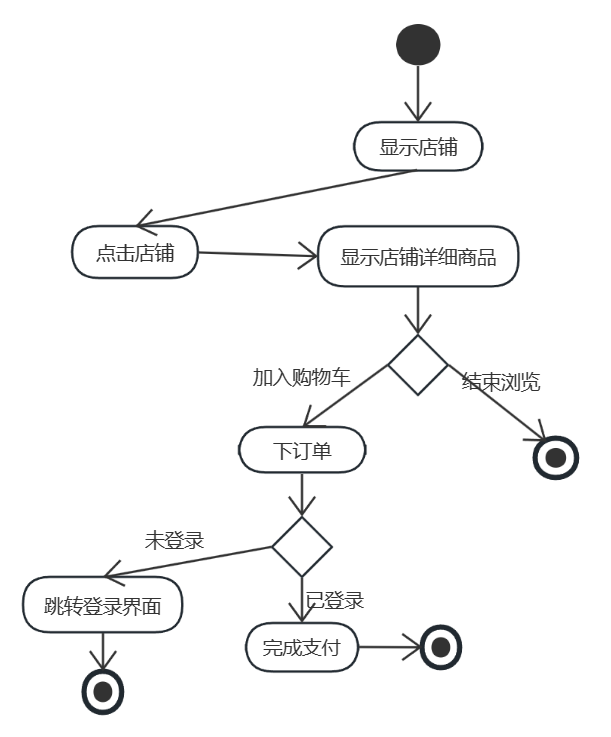


图3 浏览商品活动图

（3）历史订单模块

已登录的用户可以查看自己的全部历史订单记录，并选择性的对已完成的订单进行评价。

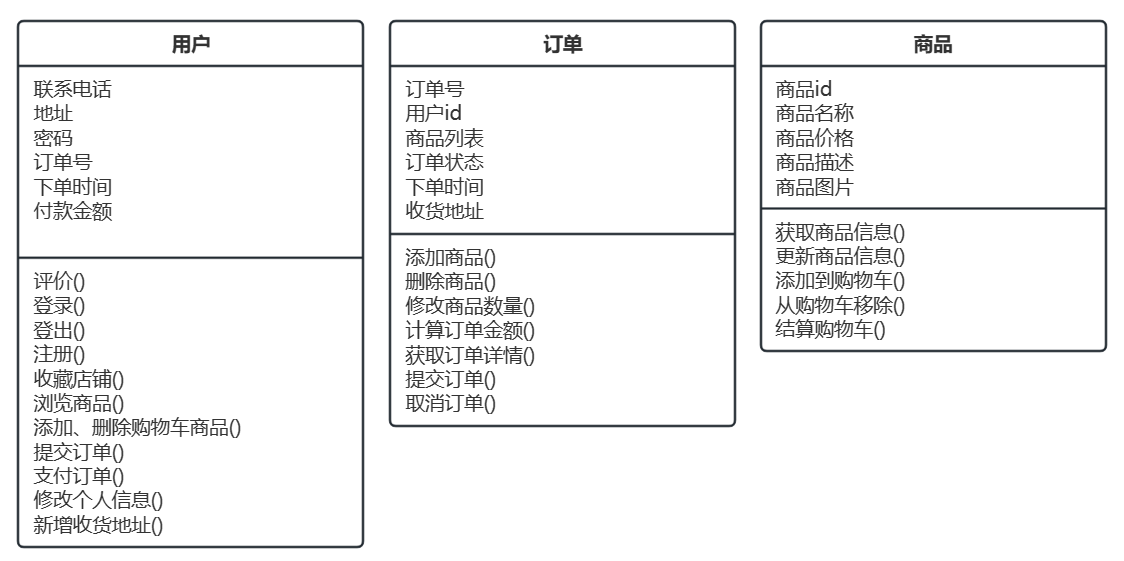


图4 总体设计类图

（4）个人中心模块

维护用户的账户信息，比如修改用户名和账号密码、注销账号等；对收货地址进行管理，可以切换当前收货地址和添加新的收货地址；用户可以在此对自己的评论和收藏进行统一管理。

**3.数据库概念模型设计**

根据以上的需求分析和功能划分，经过分析和总结，提炼出用以描述用户业务需求的概念模型。

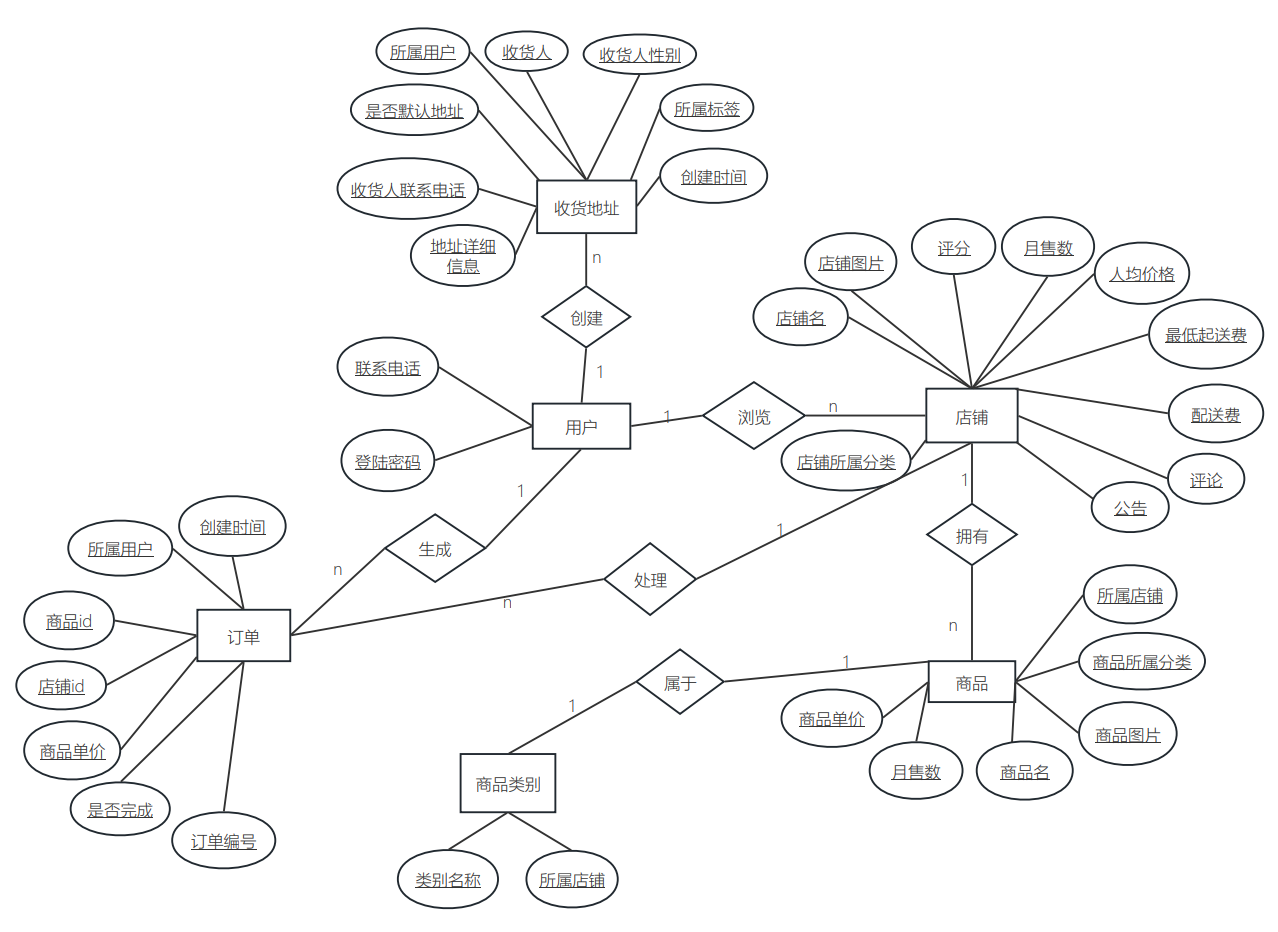


图5 ER图

**4.系统架构设计**

外卖小助手从架构上分为三层：表现层（UI）、业务逻辑层（BLL）以及数据层（DL）。

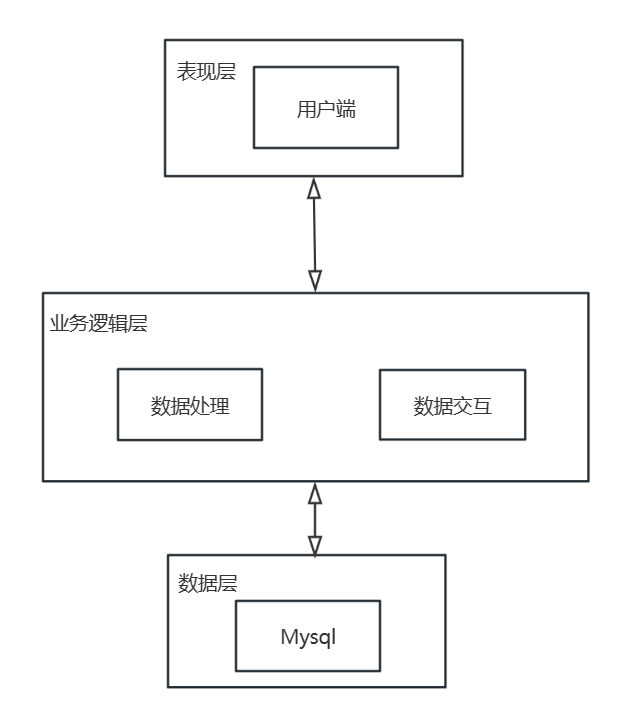


图6 系统架构设计图

**表现层（UI）**：主要完成外卖小助手的UI交互功能，一个良好的UI可以大大提高用户的使用体验。UI的界面设计也要适应不同尺寸的分辨率，以做到良好的兼容性。UI交互功能要求合理，用户进行交互操作时必须要得到与之相符的交互结果，这就要求表现层要与业务逻辑层进行良好的对接。

**业务逻辑层（BLL）**：主要完成外卖小助手的数据处理功能。用户从表现层传输过来的数据经过业务逻辑层进行处理交付给数据层，系统从数据层读取的数据经过业务逻辑层进行处理交付给表现层。

**数据层（DL）**：由于外卖小助手的数据是放在服务端的mysql数据库中，因此本属于服务层的部分可以直接整合在业务逻辑层中，所以数据层中只有数据库，其主要完成外卖小助手的数据存储和管理功能。

**四、详细设计**

**1.技术栈与开发工具**

前台：Android+Okhttp3

后台：SpringBoot+MyBatisPlus+Minio

开发工具：Idea+Navicat+Android Studio

Okhttp介绍：OkHttp是一个功能强大、易于使用的高性能HTTP客户端库，它通过连接管理、请求和响应处理、缓存机制、拦截器等机制来实现基本原理，从而提高请求的效率和可靠性。

Minio介绍：Minio 是一款功能强大、易于使用、安全可靠的分布式存储服务，适用于各种场景和需求。

**2.主要功能具体实现**

（1）注册功能

①前台实现

监听号码输入框，如果用户输入的手机号长度为11位且以“1”开头，关闭软键盘，设置获取验证码按钮颜色为黄色；如果手机号大于11位或者不是以1开头弹出提示“请输入正确的手机号码”，获取登录按钮变为灰色并禁用按钮；监听登录按钮点击事件，向后台提交用户输入的手机号码。

②后台实现

首先接收一个手机号码作为参数，然后使用随机生成验证码工具类生成一个6位的验证码，并将验证码存入数据库中。如果该手机号在数据库中不存在，则表示是新用户，会将验证码和手机号插入数据库；如果该手机号在数据库中已存在，则会更新该手机号对应的验证码。接下来接收一个用户登录请求对象作为参数。通过手机号查询数据库，获取该手机号对应的验证码。然后判断传入的验证码是否与数据库中的验证码一致。如果一致，则继续判断该手机号是否存在于用户表中。如果不存在，则表示是新用户，会将手机号和默认密码插入用户表；如果存在，则表示是已注册用户。无论是新用户还是已注册用户，登录成功后会将该手机号对应的验证码置空。

（2）登录功能

①前台实现

首先，通过判断输入的手机号长度是否为11位，并且以"1"开头，来确定手机号的格式是否正确。如果格式正确，则关闭软键盘，并设置登录按钮的颜色为亮黄色，同时启用登录按钮；如果格式不正确，则弹出提示信息，并将登录按钮的颜色设置为灰色，同时禁用登录按钮。接下来，在点击登录按钮时，会获取输入的手机号和密码，并封装成一个Map对象。然后使用OKHttpUtils发送POST请求到服务器的"/user/login"接口，并传递封装好的参数。在请求的回调中，根据服务器返回的结果进行处理。如果登录成功，则跳转到首页，并保存用户数据；如果登录失败，则弹出提示信息。

②后台实现

首先，根据用户输入的手机号查询数据库中的用户信息。如果查询结果为null，即表示该手机号对应的用户不存在，方法会返回字符串"error"。如果查询结果不为null，则表示该手机号对应的用户存在。接下来，代码比较查询到的用户对象的密码和用户输入的密码是否相等。如果相等，即表示密码正确，方法会返回字符串"success"。否则，表示密码错误，方法会返回字符串"error"。

（3）商品展示功能

①前台实现

通过post方法发送请求，获取到服务器返回的店铺数据。在请求成功后，将返回的json数据转换为List对象，并将其赋值给storeList变量。接下来，初始化一个ListView，并将storeList作为数据源，通过StoreAdapter适配器将数据显示在ListView上。当用户点击某个店铺时，会跳转到FoodActivity页面，并将所点击的店铺的id作为参数传递给FoodActivity。

②后台实现

根据店铺类别获取相应的店铺数据，并将结果以List的形式返回，如果没有店铺类别参数则返回所有的店铺数据。

（4）历史订单功能

①前台实现

首先使用SharedPreferences工具类判断是否已保存了用户数据，如果没有保存，则提示用户尚未登录。如果已保存了用户数据，则获取保存的手机号，并将其放入一个Map中作为后端接口的参数。在请求成功的回调中，首先判断返回的json数据是否为空。如果为空，则弹出一个Toast提示当前没有订单。如果json数据不为空，则将其解析为一个List对象，并将其赋值给orders变量。为orders集合中的每个Order对象的food字段赋值。

创建一个okHttpUtils对象，并发送请求获取所有食物数据。设置okHttpUtils对象的监听器，用于处理请求成功或失败的回调。在请求成功的回调中，将返回的json数据解析为一个List对象，并将其赋值给foodDOS变量。遍历orders集合，对每个Order对象的food字段进行赋值。通过比较foodDOS中的食物id和orderDO中的食物id来确定对应的食物，并将其赋值给orderDO的food字段。获取店铺数据同理，这里不再赘述。

②后台实现

首先根据传入的手机号码查询对应的用户信息，如果用户存在，则获取用户的ID。然后创建一个空的订单列表orders，用于存放所有与该用户有关的订单。接下来，通过用户ID查询所有与该用户有关的订单信息，并将这些订单信息存放在orderDOS列表中。为了获取所有不重复的订单编号，使用HashSet集合codes来存放订单编号。

然后，通过遍历codes集合，依次查询每个订单编号对应的订单信息，并将这些订单信息存放在sameCodeOrders列表中。在同一订单编号下，可能存在多个订单，所以同一订单编号对应的订单信息需要通过遍历sameCodeOrders列表进行处理。在处理过程中，首先获取每个订单的食物ID，然后根据食物ID查询食物信息，并将食物信息设置到订单对象的foodDO属性中。最后，将同一订单编号下的所有订单信息列表sameCodeOrders添加到orders列表中。最终返回orders列表，即为当前用户所有订单的列表。如果用户不存在，则返回null。

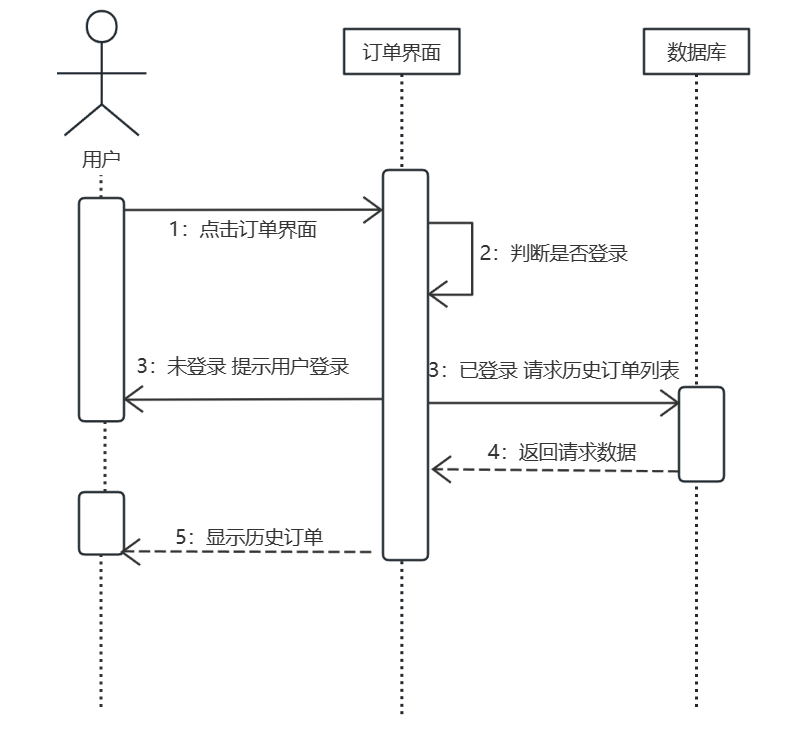


图7 历史订单时序图

（5）账户管理功能

checkPhoneNumber()方法用于动态显示手机号码。首先从SharedPreferences中获取保存的手机号码，然后对手机号码进行处理。将手机号码拆分为字符数组，然后使用循环将第4到第7位的数字替换为"\*"，最后将处理后的手机号码设置给一个按钮。

exitLogin()方法用于退出登录。当退出登录按钮被点击时，会弹出一个对话框询问用户是否确定退出。如果用户点击确定，会移除保存的手机号信息，并跳转到登录界面。

updateUserName()方法用于加载用户名并提供修改用户名的功能。首先从SharedPreferences中获取保存的手机号码，并将其设置给一个按钮。然后，当修改用户名按钮被点击时，可以在点击事件中添加相应的逻辑来实现修改用户名的功能。

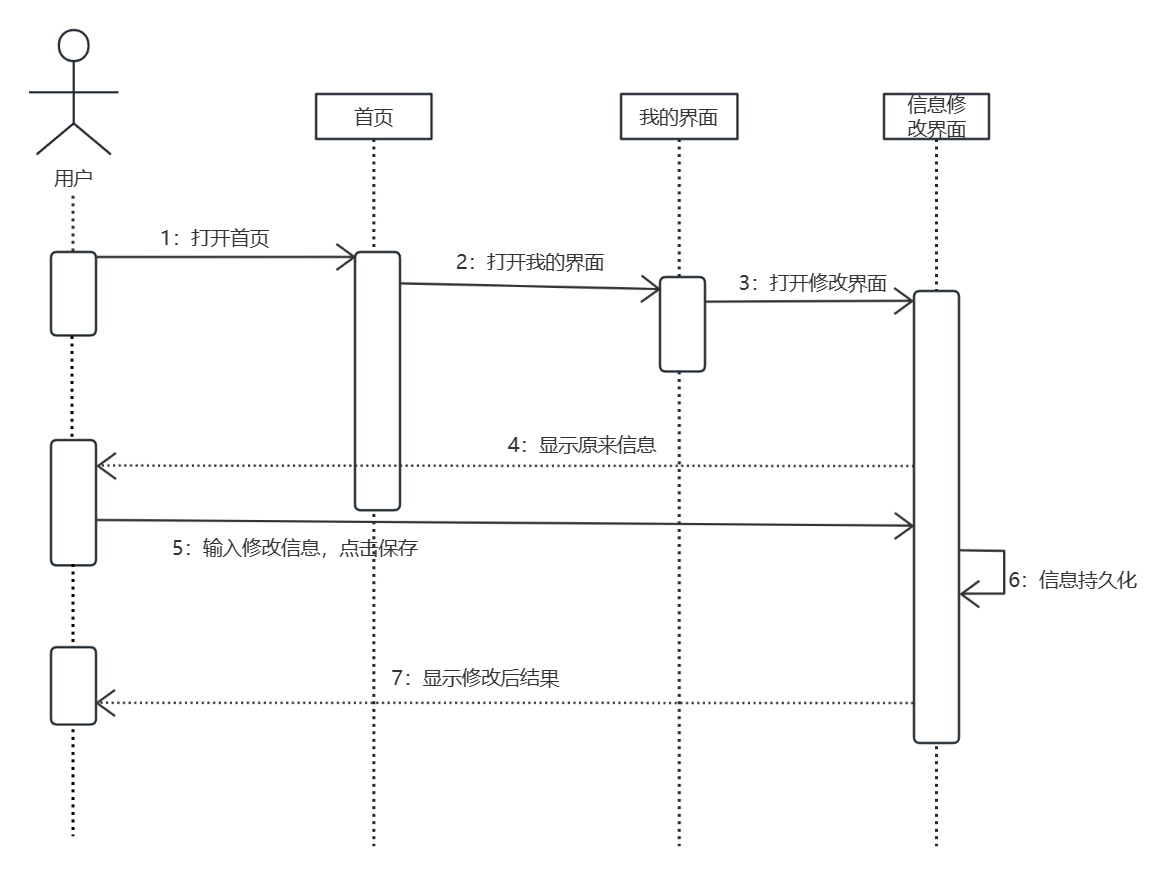


图8 修改账户信息时序图

（6）地址管理功能

①前台实现

新增地址：首先，从SharedPreferences中获取登录手机号，并将其设置到address对象中的loginPhoneNumber属性中。接着，通过点击按钮的监听器设置了保存地址按钮的点击事件。在点击事件中，首先判断地址、联系人、性别、手机号和登录手机号是否为空，只有当它们都不为空时，才将它们放入一个Map中。然后，通过网络请求发送该Map到服务器的/address/add接口。在请求的回调中，判断返回的结果，如果成功，则显示保存地址成功的提示，并跳转到我的地址页面；如果失败，则显示服务器出错的提示。

修改默认地址：点击地址列表中的某一项，向服务器发送请求设置该项为默认地址，并根据服务器返回的结果做出相应的提示和操作。

②后台实现

新增地址：首先根据用户手机号查询用户信息，然后将请求参数转换成地址对象，并设置默认值和关联的用户ID，最后将地址对象插入到数据库中。

修改默认地址：首先判断传入的地址对象是否为空，然后根据地址对象的用户ID查询该用户的所有地址。遍历所有地址，将其默认值设为0，然后判断传入的地址对象的ID是否与当前遍历的地址对象的ID相同，如果相同，则将其默认值设为1。最后更新所有地址。

**五、实验结果与分析**

1.当用户输入正确格式手机号和密码后，登录按钮解除禁用，变成黄色；点击按钮向后台提交登录请求，json数据为{"password":"123456","phoneNumber":"13712345678"}。



图9 登录界面

2.当用户没有登录时，无法获取到地址信息，提示“您还没有登录呢，请登录后查看地址”；在已登录时显示用户的默认地址；点击地址按钮进入选择页面，用户可以切换默认地址；点进新增地址页面，可以添加收货地址；在新增地址时，必须填写全部表单信息，否则会提示信息填写不全，且表单不被允许提交。

图10 未登录首页 图11 已登录首页 图12 切换默认地址

图13 添加收货地址 图14 错误填写表单提示

3.APP默认打开进入首页，首页展示该用户默认地址周围的所有店铺以及按照类别进行分类后的店铺信息；任意点进一个类别，展示该类别下所有店铺；点击店铺条目进入详情页，默认显示所有该店铺商品；点击商品分类时，显示该分类下的商品；点击商品条目“+”号，可以将该商品添加进购物车。

图15 展示所有店铺 图16 店铺详情 图17 添加商品进购物车

4.点击底部导航栏订单项，如果用户未登录提示“请登录后查看”，并跳转到登录界面；如果已登录，展示该用户所有的订单以及订单状态，用户可以对已完成的订单发表自己的评价。



图18 历史订单

**六、小结与心得体会**

通过本次课设，我获得了宝贵的实践经验和知识。在实际开发中，我将书本上的理论知识应用到实际项目中，真正理解了其中的原理和概念。同时，我也积极利用网络资源，查阅了许多博客和技术文档，以解决在开发过程中遇到的问题。

1.首先看到题目时，并不知道该从哪里入手进行开发，这是因为还没有进行开发前的需求建模工作；为了减少开发时间和成本，决定仿照美团外卖APP进行开发，**成熟软件往往已经经过了市场验证，用户反馈和需求被充分考虑**，因此可以更好地满足用户的需求。

2.本次开发中学习到了工程化的思想，像**工具类的使用**大大提高了代码的复用性和可维护性；安卓中封装了共享参数工具类和OKHttp网络请求工具类，**共享参数工具类**里面使用SharedPreferences进行存储数据，**OKHttp网络请求工具类**中定义了常用的请求方法，方便统一调用；springboot后台中封装了**随机生成验证码工具类**，当想改变生成验证码的规则时，可以直接在一个文件进行修改，方便管理。

3.在进行用户收货地址功能开发时，一开始没有考虑到用户可以帮忙给别人下订单，就用了手机号作为id查询；修改为**先根据用户的手机号查询到该用户的id，再通过用户id在地址表中查找数据**，这样用户就可以给他人下单商品了。

4.安卓开发中使用到了书本中讲到的适配器，它充当了数据源与显示器之间的桥梁，负责将数据适配到显示器上；具体实现思路：首先在xml文件布局上实现ListView，然后根据实际需求定制列表项，即实现ListView每行的xml布局（即item布局），接着定义一个HashMap构成的列表以键值对的方式存放数据，构造Adapter对象，设置适配器将LsitView绑定到Adapter上。

5.在使用OKHttp框架进行与后台联调时花费了很多时间，首先没有在manifests文件里面添加**联网权限**，导致一直请求失败；添加好联网权限后还是不通，使用postman工具测试后台接口没有问题，查找资料得知在使用OkHttp向后端发起请求时，将baseurl设置为"localhost"会导致连接失败或请求超时的问题。这是因为在Android设备上，"localhost"指的是设备本身，而不是指向电脑或服务器，需要提供一个有效的URL作为目标地址。如果**目标是本地的服务器，那么需要使用本机IPv4地址作为baseurl**，这样，OkHttp才知道将请求发送到本地服务器的正确位置。在cmd中使用ipconfig命令查看本机的ipv4地址，修改后成功获取到后台接口数据。

6.开发中使用到了MinIO存储图片，目前只有少量的图片需要存储，只使用了一个bucket；如果后续数据量大的时候，MinIO可以很容易地进行水平扩展，支持在集群中添加更多的存储节点，另外MinIO采用了分布式架构和并行化处理技术，可以提供高性能的数据读写能力，这对于美团外卖来说非常关键，因为它需要快速地加载和显示大量的图片数据。