1. 系统的构成
2. 逻辑框简图



消费者APP

扫一扫

超市显示终端

服务器service

超市数据库服务器

４

８

６

７

３

１

２

５

注：单向箭头只是获取的一个过程；

　　双向箭头是一个来回，请求与应答。

1. 服务流程说明

①：请求——由自助结账人手动在终端上触发，向服务器请求开辟一个购物账号，就像购物的购物车。

　　返回——服务器收到终端开辟新的购物单申请后，创新一个单号，并把单号，还有其他的信息，如安全key等。最后在终端上，把信息以二维码的形式显示出来。

②：消费者打开自助ＡＰＰ扫描在超市终端上的二维码信息。

③：请求——自助ＡＰＰ扫描到的信息并向服务器，提交信息认证。把自己的ＩＤ也填入购物单信息内。

　　返回－－返回这张购物单内的商品。当然在这里是空的。

　　　　④：请求——在这里是每隔１秒的询问，新的购物单信息是否完整，在信息完整后，就是询问是否有新的购物商品添加。

返回——购物单信息，与添加的商品信息。在这个定单下有

⑦：一一解析有没有没填入价格、与名称的商品，发现没有就用条码信息向本地的数据库询问。

　　　　　　本地数据库返回；

⑧：把商品信息不完整的填入完整，并在终端上显示。

⑤：消费用户用ＡＰＰ扫描商品的barcode形码；

⑥：把新扫描的商品条码，向购物单里发送添加商品的请求。

　　返回购物单最新的信息情况。并可以显示。

最后：ＡＰＰ进入结账，过程，支付宝或微信支付——支付完成——终端也显示原始界面。

准备进入下一轮的自助结账循环过程

详细的实现过程

①：用Django 编写服务器。完成的模块有注册、登录、退出、新购物单的生成、添加商品、添加　　商品价格与，名称等详细信息。

②：超市终端，准备用可以安装ubuntu，python3.5　的嵌入式硬件。这里这里的界面，数据库查询，服务器交互，都已完成。

③：手机端没有做。

④：商品信息的添加，现在是用ＰＣ的网页模拟可以实现。

1. 附图终端





