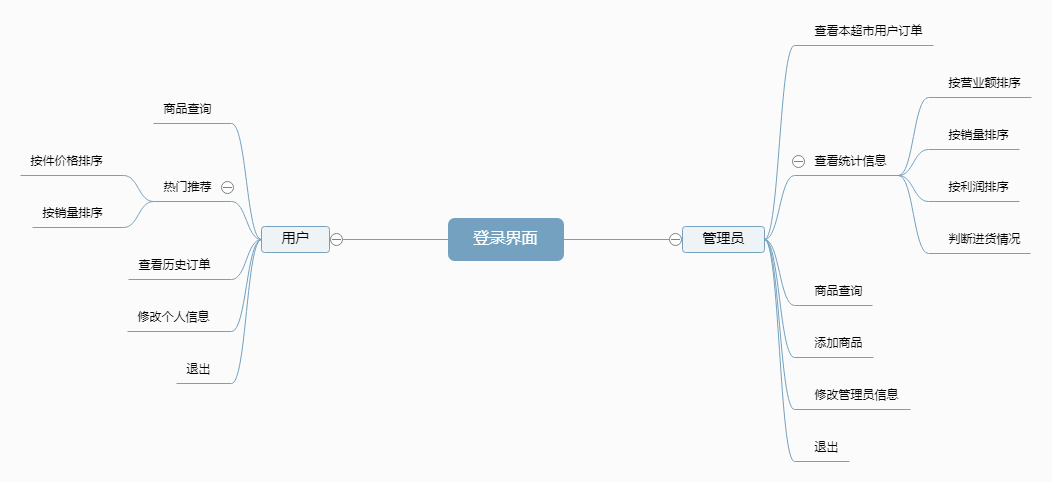
设计报告（电商第五组）

【**系统总体设计**】

**1.系统总体功能描述：**

 该项目基于文件的数据储存方式，主要聚焦于实现一个简易的网上超市购物系统。打开该系统，首先选择顾客或管理员身份进行登录。登录时对账号长度、字符具有纠错机制，密码有三次输入机会。进入顾客主界面后，有商品查询、热门推荐(可以按销量和价格进行排序)、修改个人信息、查看历史订单、退出系统的功能。可以通过商品名或超市名进行查询；会根据销量排序或售价排序进行推荐；在购买商品时，通过添加商品的序号来进行添加商品，可多次添加商品；生成订单时系统会根据不同的超市自动拆分订单，并计算出所有订单总价；购买时若余额不足或商品库存不足则会购买失败，若购买成功则会自动扣除余额并且实现商品销量和库存的实时变化。进入管理员主界面后，具有商品查询、添加商品、统计信息、查看用户订单、修改个人信息、退出系统的功能。商品查询后可修改售价和库存；添加商品时有名称、编号的查重机制；修改电话和邮箱时具有纠错机制；查看用户历史订单时会根据所有订单计算营业额和利润。

**2.接口定义规范：**见名知意，没有歧义，自定义函数名称首字母大写，单词不宜太长，按照函数的实际功能来定义接口。

**3. 系统的数据结构描述：**系统处理的数据，最多共计14个文件，分别为1个储存用户信息的文件，1个储存管理员信息的文件，1个储存商品信息的文件；5个用于记录用户历史订单的文件，3个用于储存超市历史订单的文件，以及3个分别存放三家超市营业额和利润的文件。程序运行中的结构体共有三种，分别是顾客类型结构体、商品类型结构体、管理员类型结构体。

**4. 系统的开发环境及运行环境描述（包含软硬件环境）：**笔记本电脑、Dev-C++6.3、文件

【**模块详细设计**】

**1.功能概述与模块的输入输出：**

**（1）登录：**

首先读取储存顾客或管理员信息的文件，放入一个结构体数组；用一个相同类型结构体去接收输入的账号，若长度和字符符合要求，则遍历该数组的账号信息与该输入账号比对；同理密码也是如此遍历比较。

**（2）顾客：**

**商品查询：**首先遍历商品文件的信息，将其逐个读入存放商品的结构体数组；用一个相同类型的结构体去接收输入的商品名或超市名，然后遍历该数组进行比较，如果有正确结果就输出。

**热门推荐：**以销量为例，通过两重循环进行排序。第一层循环从结构体数组中第一个迭代到倒数第二个，每迭代到一个销售量，就进行第二次循环，从该位置的后一个迭代到最后一个，两者进行比较，数大者排前面。如此经过这两轮循环，即可完成销售量由大到小排列。

购买：选择序号将商品加入购物车数组中，每次只能添加一种商品的一个，可反复添加。添加完商品后，可选择生成订单，系统会迭代购物车数组，将商品按不同超市分别输出，并通过时间函数显示当前时间，并输出订单号，由此即达到根据不同的超市自动差拆分订单的功能。购买时，若余额与商品库存足够，则从余额自动扣除钱款，实际修改了顾客结构体数组中的信息，然后将修改后的信息再覆盖原顾客文件。同理，即可实现商品库存和销量的实时变化。

**查看历史订单：**以每行的形式读入存放顾客历史订单的文件，并将其逐行输出。

**修改信息：**修改信息实际就是对顾客结构体中某个属性重新赋值，并将修改后的信息重新覆盖到原顾客信息文件。

**（3） 管理员：**

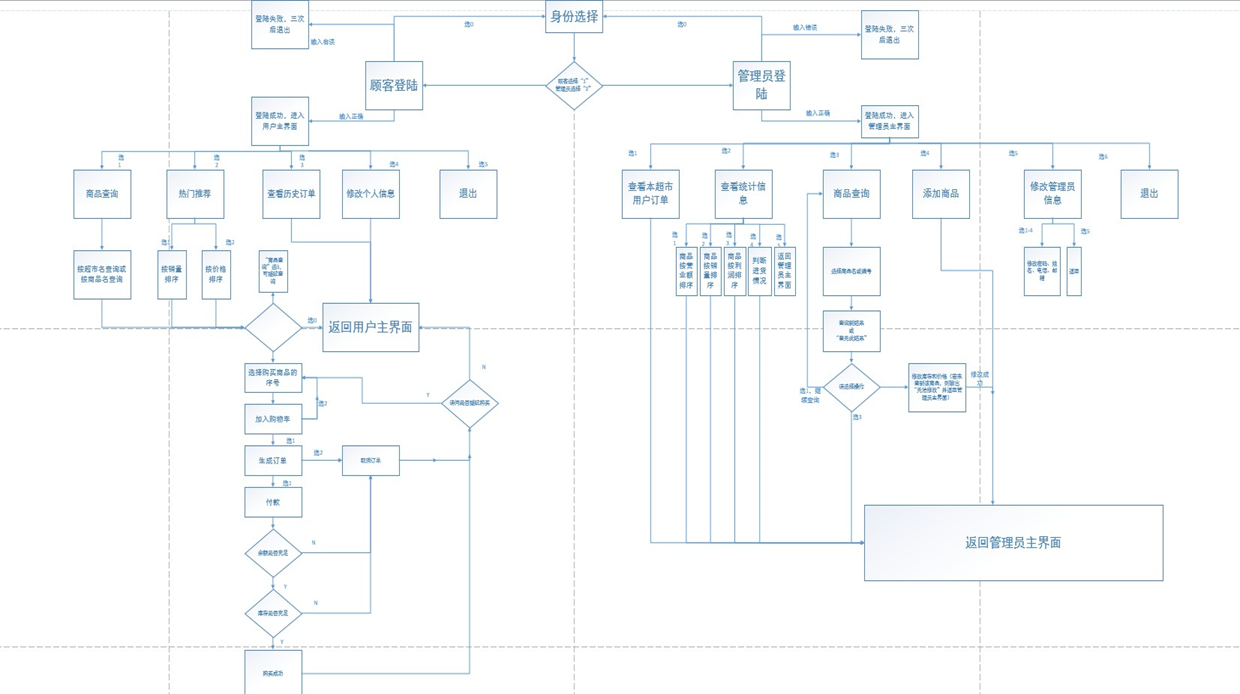
**查看历史订单：**以每行的形式读入存放超市历史订单的文件，并将其逐行输出。

**查看统计信息：**通过同样的数值排序方式，对营业额、利润、销量进行排序并输出；查看进货情况时，计算销量与库存的比例，满足某些情况时，会分别辨别滞销、进货、无三种情况。

**商品查询：**首先遍历商品文件的信息，将其逐个读入存放商品的结构体数组；用一个相同类型的结构体去接收输入的商品名或编号，然后遍历该数组进行比较，如果有该商品或编号且是该商店的商品就输出。若查到该商品，可以选择修改其库存和售价，也是一个文件读写操作。

**添加商品：**逐一添加商品名、编号、售价、进价、库存，商品名和编号不能和该超市已有的重复。价格和库存不能为负数，均有纠错机制。若选择发布商品，则在商品文件中追加一行该商品的信息

**修改信息：**修改信息实际就是对管理员结构体中某个属性重新赋值，并将修改后的信息重新覆盖到原管理员信息文件。

**2.处理流程：**

**3.模块接口描述：**

void Menu(); //选择身份菜单函数

void Account\_customer(); //顾客登录函数

void Customer(); //顾客主界面函数

void Change\_information\_customer(); //顾客修改信息函数

void Change\_information\_manager(); //管理员修改信息函数

void Manager(); //管理员主界面函数

void Account\_manager(); //管理员登录函数

void Read\_text(); //读取商品文件函数

void Search(); //顾客搜索查询函数

void Sort(); //排序主界面函数

void Salesvolume(); //销量排序函数

void Sellingprice(); //售价排序函数

void Buy(); //购买函数

void Time(); //显示当前时间函数

void Write\_seperate\_order(); //写用户历史订单函数

void History\_manager\_order(); //管理员查看该超市历史订单函数

void History\_customer\_order(); //顾客查看自己历史订单函数

void Add(); //添加商品函数

void Statistics(); //统计信息函数

void Manager\_search(); //管理员查询函数

void Manager\_profit(); //利润排序函数

void Manager\_salesvolume(); //销量排序函数

void Business\_volume(); //营业额排序函数

void Make\_seperate\_order(); //按超市自动拆分订单函数

void Write\_shop\_order(); //写超市历史订单函数

void Purchase\_status(); //进货情况函数

void Phone\_number(); //电话纠错机制函数

void Order\_volume(); //计算所有订单营业额及利润函数

void Email(); //邮箱纠错函数

**4. 每个功能的边界情况描述及处理方法：**

**添加购物车功能：**将商品加入购物车时，一次只能输入一个序号，也就是一次只能添加一种商品的一个。若要多个同种商品，则需反复添加。顾客添加购物车时，输入序号输入字符时，直接跳转回顾客主界面，没有错误提示，需要增加一个纠错机制。

**修改信息功能：**修改姓名、地址时没有纠错功能，输入任意数字、字符也可以修改成功；若不小心输错，只能修改成功后再次调用该功能进行修改。

**邮箱纠错机制：**邮箱格式识别时只能判断在输入字符中是否先后有@和“.”两个字符，若有多个“.”，则也可以修改成功。后续需要进一步完善纠错机制。

**电话纠错机制：**判断电话格式时只能识别是否为11字符，且都必须为数字，但若是开头为0等特殊情况则也可以修改成功。后续需要进一步完善纠错机制。

**修改商品信息：**修改商品售价和库存时，输入负数会报错，但输入符号会修改成功，后续需要进一步完善纠错机制。