Java程序设计文档

**项目名称** 中医问诊app

**项目成员** 孙琦 江越天

**组员学号**084315123 084315128

**班级专业** 15软件工程

**日 期** 2017年7月1号

**南京中医药大学**

目录

[一、 项目背景 4](#_Toc486934836)

[二、 需求分析 4](#_Toc486934837)

[2.1 用户需求分析 4](#_Toc486934838)

[2.2 开发环境分析 5](#_Toc486934839)

[三、 系统设计 5](#_Toc486934840)

[3.1 系统总体设计 5](#_Toc486934841)

[3.2 功能模块设计 5](#_Toc486934842)

[3.3 类的设计 6](#_Toc486934843)

[3.4 界面设计 10](#_Toc486934844)

[四、 数据库设计 10](#_Toc486934845)

[4.1 数据库需求分析 10](#_Toc486934846)

[4.1.1数据字典 10](#_Toc486934847)

[4.2 概念结构设计 13](#_Toc486934848)

[4.2.1 ER图 13](#_Toc486934849)

[4.3 逻辑结构设计 13](#_Toc486934850)

[4.3.1关系模式 13](#_Toc486934851)

[4.3.2视图 14](#_Toc486934852)

[4.3.3关系图 15](#_Toc486934853)

[4.4 物理结构设计 16](#_Toc486934854)

[4.4.1建立索引 16](#_Toc486934855)

[4.5 数据库实施 16](#_Toc486934856)

[4.5.1创建数据库模式 16](#_Toc486934857)

[4.5.2装入数据 19](#_Toc486934858)

[五、 算法设计 25](#_Toc486934859)

[5.1问题描述 25](#_Toc486934860)

[5.2算法流程图 26](#_Toc486934861)

[5.3代码描述 26](#_Toc486934862)

[六、 运行测试 29](#_Toc486934863)

[七、 总结体会 35](#_Toc486934864)

# 项目背景

目前使用中医治疗手段解决一些亚健康问题，已经得到社会极大的关注与认可，而国内存在着挂号难、挂号久等令人烦恼的医疗现状，尽管有很多致力于帮助挂号和排队就诊的app和服务，但并不能很好地改善问题。

现如今，移动应用正在重塑人们的日常生活，手机成了日常生活必不可少的用品，而手机上的app就为用户的日常出行、活动、娱乐提供了极大的便利。

根据这一现状，本着为亚健康人群提供方便快捷中医诊疗方式的理念，通过用户回答问诊问题，软件实现中医简易机器问诊，根据中医相关理论及标准，对问题及答案的设计进行标准化设定，并且给出诊断和推荐方剂后提供相关治疗器材的在线一键购买。app的辅助功能还提供健康保健知识的推送和保健器材的购买，为用户提供“在家治病”的有效方法。

# 需求分析

## 2.1 用户需求分析

中医自诊app是一款移动端软件，用户可以通过本软件进行一些简单的中医自我诊疗，根据用户的回答，主要通过舌象和脉象去数据库中进行匹配，并给出相应的方剂推荐。除了主要的简单自我诊疗功能，软件还提供了养生小知识的推送和商店购买健康养生器材的功能。

1自诊模块 用户通过回答中医自诊单上的问题，系统根据用户回答，产生相应的编号，再去数据库中方剂的编号进行匹配，从而产生相应的方剂推荐，推荐的方剂包括方剂名，方剂出处，方剂主治，方剂功效，以及方剂的用药。

2 商店模块 用户可以根据自己的喜好和需要，在商店中挑选心仪的健身保健器材，提供购物车功能暂存商品以便用户一次购买多个商品，同时有包邮限制，当用户不足包邮价时，系统自动会根据当前需凑够的邮费通过动态规划算法向用户进行包邮推荐。

3 健康养生推文模块 软件为用户提供有关健康养生美容的推文，用户在查看后可以对推文进行点赞，或者差评，这些评价亦可以取消。

4 个人中心模块 在个人中心中用户除了添加一些个人信息以外还可以在个人中心中查看历史阅读记录和历史购买记录。

## 2.2 开发环境分析

1. 硬件环境

Cpu: Intel (R) Core(TM) i5

内存：4G

1. 软件环境

64位 window7操作系统。JDK和Android studio软件、SQlite Expert环境。

1. 运行环境

Android手机SDK版本23以上

# 系统设计

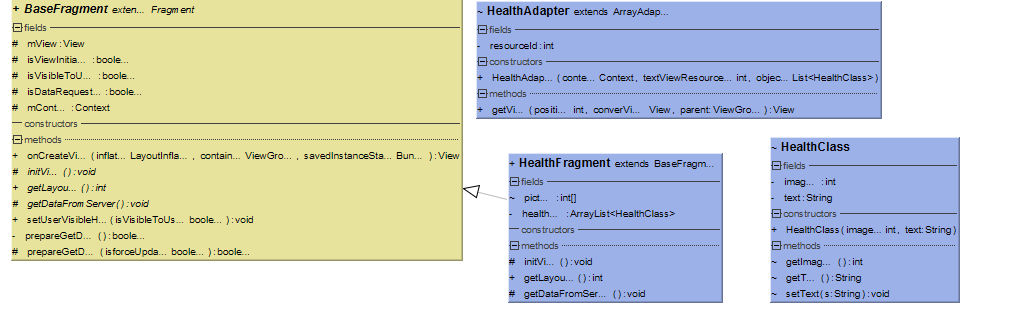
## 3.1 系统总体设计

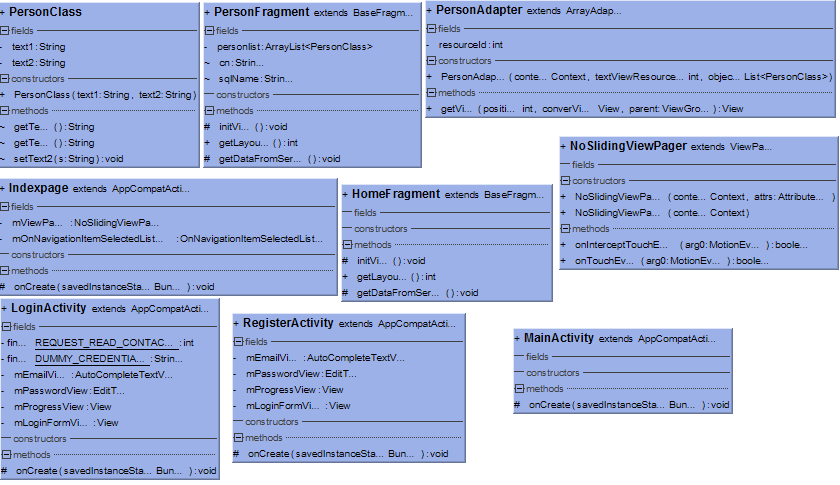
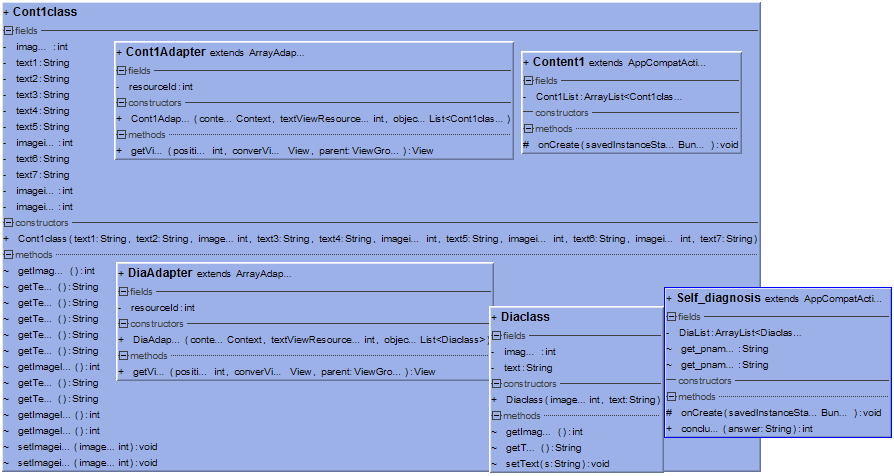
实现自动化的简易自诊系统，让用户可以通过症状得出病症和推荐方剂，与此同时，为了使软件更加多元化，配有养生小知识的推送和健康养生器材的购买功能。用户还可以在个人中心中查看历史购买记录和阅读记录，以便对历史行为进行查看。也可以为后期开发对用户行为的挖掘留下数据

## 3.2 功能模块设计

1. 自诊模块：用户通过回答中医问诊单上的一系列问题，系统根据问题进行编号，根据用户的不同回答得到不同的编号，在去数据库中和方剂所对应的编号进行匹配，若匹配到方剂则，将其信息从数据库中取出并推荐给用户。
2. 商店模块：用户在商店中选取心仪的商品，可以选择立即购买也可以先加入购物车，待挑选好所有想购买的商品时再一并购买，在购物车中，若用户加错了商品或者不想购买时，可以选择清除订单，随即购物车被清空，当用户决定购买时，若不足包邮价，系统会提示用户去凑单，用户亦可选择直接购买，此时系统会在总价上加上邮费。当用户选择去凑单时，系统会根据需凑满的邮费，给用户一系列的推荐商品，这些推荐商品的总价是无限逼近包邮价格的，用户可以根据自己的情况选择去购买其中推荐的商品，当凑满包邮价时，用户不必支付邮费，而支付密码即为用户的登录密码。
3. 养生文章推荐模块：为使软件更加多元化，全面化，系统提供健康养身小知识的推送，用户可以根据自己的喜好和文章的阅读量进行阅读，根据自身的实践和看法可以给文章进行点赞或者差评，若不小心点错亦可将操作取消，而这些评价也可为其他读者提供一个方法可信度的评判标准。
4. 个人中心模块：在个人中心中用户除了可以填写自己的一些个人信息之外，还可以查看以往在何时阅读了推文，在何时购买了商品。即为用户提供了两个历史记录的查看功能。

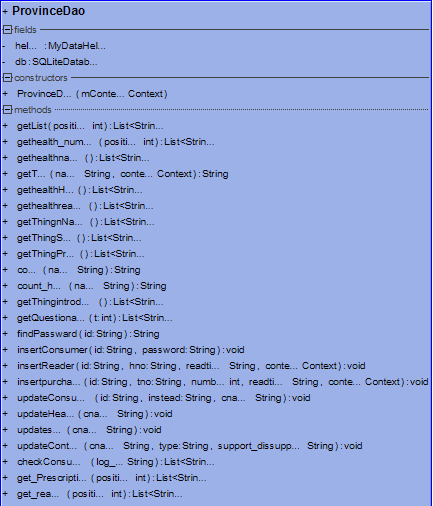
## 3.3 类的设计

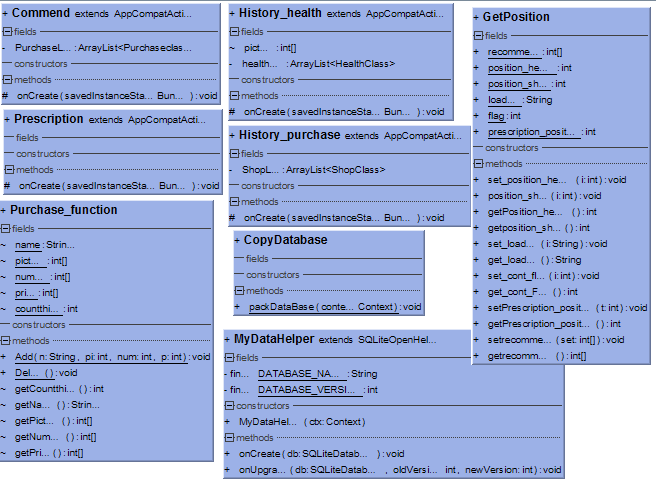
UML图：

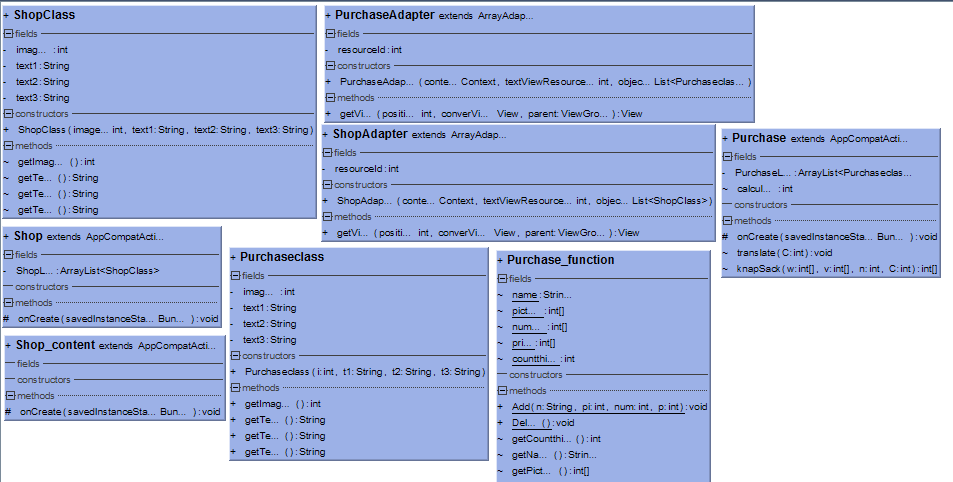


为数据库操作封装的类：

通过建立类对象，传参获取数据库内容或者对数据库进行操作







## 3.4 界面设计

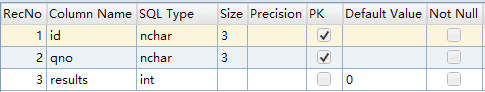
软件共使用14个activity和3个fragment即17个界面，其中主要的界面都是通过ListView产生一个可滑动效果，它的每一个item上是设置主要包括TextView和ImageView，整体使用LinearLayout每个item子布局使用RelativeLayout，在自诊单部分通过选项得出方剂，使用了自定义布局的AlertDialog和RadioButton组，在登录注册以及个人中心也使用了一些EditText之类的Button、ImageButton之类基本控件。在一开始的登录界面使用了ScrollView和AutoCompleteTextView让文字有一个动画滑动的效果。

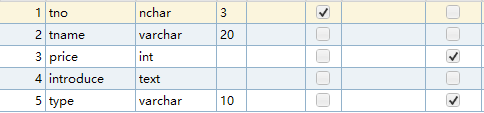
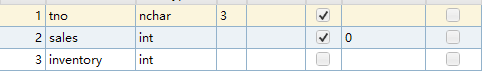
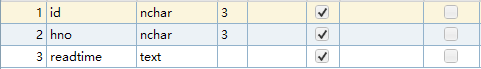
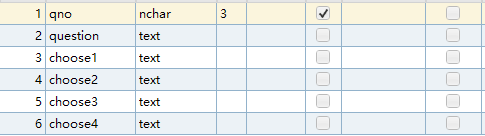
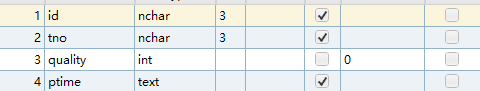
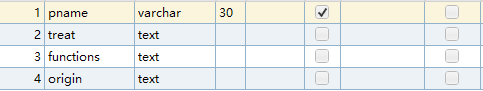
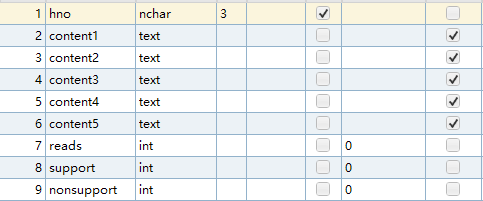
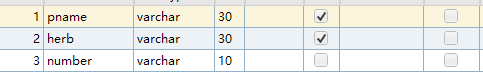
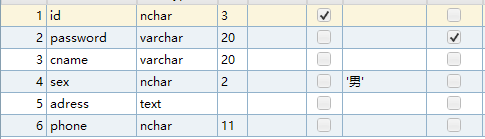
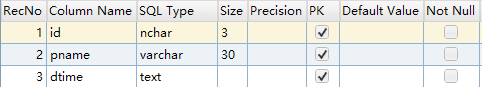
# 数据库设计

本软件采用的是Android自带的轻量级数据库SQLite，使用可视化数据库软件SLQLite Expert 创建表和约束，以及添加需存储的数据，在程序中首先封装一个类CopyDatabase给定外部数据库的存储路径通过二进制文件读写的方式将原文存储在assets文件夹下的数据库文件写入路径为data/data/项目名…中然后通过封装类MyDataHelper 继承自SQLiteOpenHelper给出数据库的名称和版本号，最后封装类ProvinceDao并以MyDataHelper和SqliteDatabase为成员进行数据库的连接，并在类中写入所需数据库操作的方法。

## 4.1 数据库需求分析

### 4.1.1数据字典





## 4.2 概念结构设计

### 4.2.1 ER图

## 4.3 逻辑结构设计

### 4.3.1关系模式

根据ER图直接转化为逻辑结构得出以下关系模式：

Consumer（id，password，cname，sex，address，phone）

主码：id

Answer（id，qno，results）

主码：（id，qno），外码：id，qno

Diagnosis（id，pname，dtime）

主码：（id，pname，dtime） 外码：id，pname

Health（hno，content1，content2，content3，content4，content5，

reads，support，nosupp）

主码：hno

Prescription（pname，treat，functions，origin，herb，number）

主码：（pname，herb ）

Purchase（id，tno，quality，ptime）

主码 (id，tno，ptime) 外码：id，tno

Questionaire（qno，question，choose1，choose2，choose3，choose4）

主码：qno

Reader（id，hno，readtime）

主码：(id,hno,readtime) 外码：id，hno

Things（tno，tname，price，introduce，type，sales，inventory）

主码：（tno，sales）

据分析，其中的Thing和Prescription中有非主属性对码的部分函数依赖，将其模式分解，规范化到BC范式：

Sale（tno，sales，inventory）

主码：(tno，sales) 外码：tno

Things（tno，tname，price，introduce，type）

主码：tno

Details（pname，herb，number）

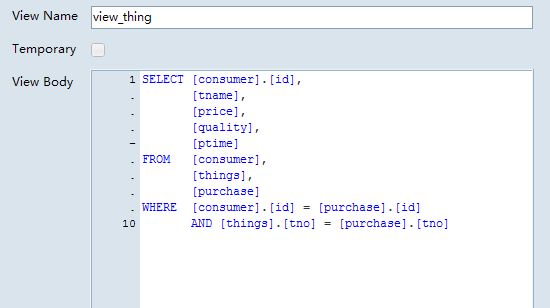
主码：(pname，herb) 外码：pname

Prescription（pname，treat，functions，origin）

主码：pname

### 4.3.2视图

在进行历史记录的查询时需要进行三表连接，并且不需要进行插入操作，建立视图简化操作方便查询。



### 4.3.3关系图

## 4.4 物理结构设计

### 4.4.1建立索引

因为在统一界面显示文章时需要不断的查找内容，所以在养生推文上建立了索引以便快速查找。

同样的在具体的商店信息显示的时候，为了提高程序的复用性，商店里面的内容是根据上一个活动的item的position值去数据库中查找相应的商品信息的，所以为了提高查找效率，在things的信息上添加了索引。

## 4.5 数据库实施

### 4.5.1创建数据库模式

CREATE TABLE answer(

id nchar(3),

qno nchar(3),

results int default 0 check(results in(0,1,2,3,4)),

primary key(id,qno),

foreign key(id)references consumer(id),

foreign key(qno)references questionaire(qno));

CREATE TABLE consumer (

id nchar(3) primary key,

password varchar(20) not null,

cname varchar(20),

sex nchar(2) default'男'check(sex in('男','女')),

adress text,

phone nchar(11));

CREATE TABLE details(

pname varchar(30),

herb varchar(30),

number varchar(10),

primary key(pname,herb),

foreign key(pname)references prescription(pname));

CREATE TABLE diagnosis(

id nchar(3),

pname varchar(30),

dtime text,

primary key(id,pname,dtime),

foreign key(id)references consumer(id),

foreign key(pname)references prescription(pname));

CREATE TABLE health(

hno nchar(3) primary key,

content1 text not null,

content2 text not null,

content3 text not null,

content4 text not null,

content5 text not null,

reads int default 0 check(reads>=0),

support int default 0 check(support>0),

nonsupport int default 0 check(nonsupport>0));

CREATE TABLE prescription(

pname varchar(30) primary key,

treat text,

functions text,

origin text);

CREATE TABLE purchase(

id nchar(3),

tno nchar(3),

quality int default 0 check(quality>0),

ptime text,

primary key(id,tno,ptime),

foreign key(id)references consumer(id),

foreign key(tno)references things(tno));

CREATE TABLE questionaire(

qno nchar(3)primary key,

question text,

choose1 text,

choose2 text,

choose3 text,

choose4 text);

CREATE TABLE reader(

id nchar(3),

hno nchar(3),

readtime text,

primary key (id,hno,readtime),

foreign key(id)references consumer(id),

foreign key (hno)references health(hno));

CREATE TABLE sale(

tno nchar(3),

sales int default 0 check(sales>=0),

inventory int,

primary key(tno,sales),

foreign key(tno)references things(tno));

CREATE TABLE things(

tno nchar(3) primary key,

tname varchar(20),

price int not null check(price>0),

introduce text,

type varchar(10)not null);

CREATE VIEW view\_thing as

select consumer.id,tname,price,quality,ptime

from consumer,things,purchase

where consumer.id=purchase.id and things.tno=purchase.tno;

CREATE INDEX index\_health on health(content1);

CREATE INDEX index\_price on things (price,introduce);

### 4.5.2装入数据

由于插入数据较多，尤其数据库中存有养生推文，所以此处给出部分插入代码，完整代码另附。

insert into questionaire values('001','身体','畏热','畏寒',null,null);

insert into questionaire values('002','脚','凉','热',null,null);

insert into questionaire values('003','头痛','脑后','头顶','左侧','右侧');

insert into questionaire values('004','目','涩','酸','疲劳','胀');

insert into questionaire values('005','口','苦','渴','欲饮','不欲饮');

insert into questionaire values('006','耳','耳鸣','耳聋',null,null);

insert into questionaire values('007','胸胁','胀','痛','心烦','心悸');

insert into things values('001','纯铜艾灸盒随身灸家用无烟仪器',12,'家用艾灸','艾灸');

insert into things values('002','竹制艾灸盒',38,'艾灸，竹制','艾灸');

insert into things values('003','家用双头梅花针',16,'防弱视近视脱发','推拿');

insert into things values('004','木制艾灸床',260,'宫寒背腹腰腿疼','艾灸');

insert into things values('005','坐灸仪艾炙凳',780,'家用妇科宫寒养生','艾灸');

insert into things values('006','拔罐器17罐子真空罐',99,'驱寒暖宫','拔罐');

insert into things values('007','高清针灸穴位人体模型',120,'人体模型','针灸');

insert into things values('008','中医号脉枕诊脉包',12,'家用问诊','家用');

insert into things values('009','泗滨砭石枕头',179,'家用艾灸','家用');

insert into things values('010','0-6岁小儿推拿书正版教材',30,'养生类书籍','书籍');

insert into things values('011','中草药书中药书籍',18,'中草药类书籍','书籍');

insert into things values('012','中医四大名著黄帝内经金匮要略',149,'中医书籍','书籍');

insert into things values('013','多功能仿中医手法颈肩按摩器材',635,'家用按摩器材','推拿');

insert into things values('014','数码经络按摩仪理疗仪',70,'家用按摩器材','推拿');

insert into sale values('001',230,500);

insert into sale values('002',545,120);

insert into sale values('003',234,140);

insert into sale values('004',263,366);

insert into sale values('005',233,144);

insert into sale values('006',20,100);

insert into sale values('007',250,102);

insert into sale values('008',223,106);

insert into details values('麻黄汤','麻黄','9g');

insert into details values('麻黄汤','桂枝','6g');

insert into details values('麻黄汤','杏仁','6g');

insert into details values('麻黄汤','灸甘草','3g');

insert into details values('大青龙汤','麻黄','12g');

insert into details values('大青龙汤','桂枝','6g');

insert into details values('大青龙汤','灸甘草','6g');

insert into details values('大青龙汤','杏仁','6g');

insert into details values('大青龙汤','石膏','12g');

insert into details values('大青龙汤','生姜','9g');

insert into details values('大青龙汤','大枣','3g');

insert into details values('香苏散','香附','120g');

insert into details values('香苏散','紫苏','120g');

insert into details values('香苏散','灸甘草','30g');

insert into details values('香苏散','陈皮','60g');

insert into details values('止嗽散','白前','100g');

insert into details values('止嗽散','炒甘草','375g');

insert into details values('止嗽散','陈皮','60g');

insert into details values('止嗽散','抄桔梗','100g');

insert into details values('止嗽散','百部','100g');

insert into details values('止嗽散','紫菀','100g');

insert into details values('止嗽散','荆芥','100g');

insert into prescription values('麻黄汤','发汗解表、宣肺平喘','恶寒发热、头疼身痛、无汗而喘、舌苔薄白、脉浮紧','伤寒论');

insert into prescription values('大青龙汤','清热除烦、发汗解表','外感风寒、发热恶寒、寒热俱重、脉浮紧、身疼痛、不汗出而烦躁','伤寒论');

insert into prescription values('香苏散','疏散风寒、理气和中','外感风寒、内有气滞、形寒身热、头痛无汗、不思饮食、苔薄白、脉浮','太平惠民合合剂方局');

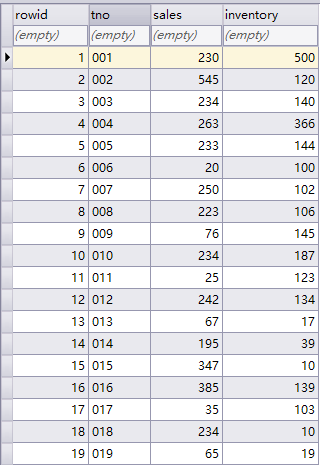
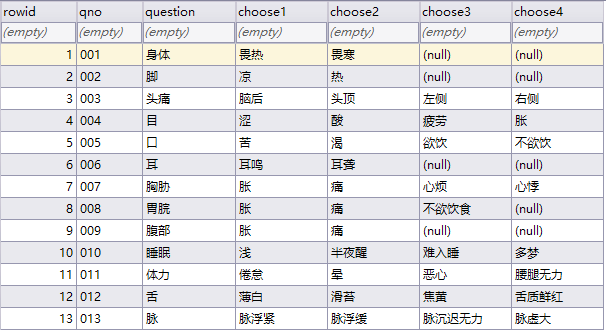
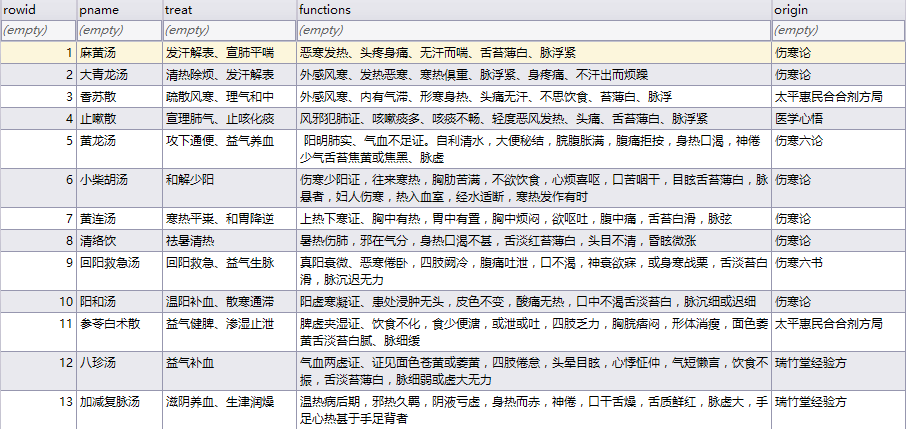
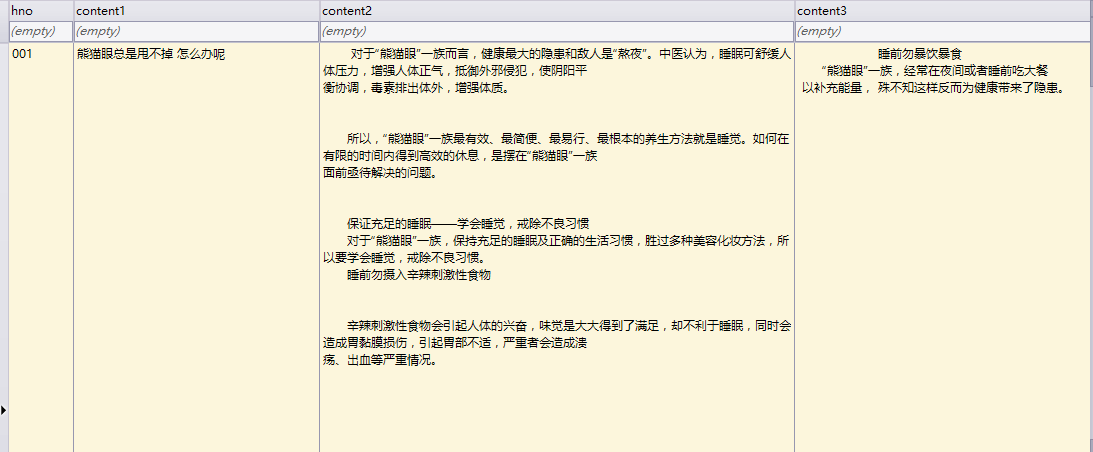
insert into prescription values('止嗽散','宣理肺气、止咳化痰','风邪犯肺证、咳嗽痰多、咳痰不畅、轻度恶风发热、头痛、舌苔薄白、脉浮紧','医学心悟');

insert into prescription values('黄龙汤','攻下通便、益气养血',' 阳明肺实、气血不足证。自利清水，大便秘结，脘腹胀满，腹痛拒按，身热口渴，神倦少气舌苔焦黄或焦黑、脉虚','伤寒六论');

insert into prescription values('八珍汤','益气补血','气血两虚证、证见面色苍黄或萎黄，四肢倦怠，头晕目眩，心悸怔仲，气短懒言，饮食不振，舌淡苔薄白，脉细弱或虚大无力','瑞竹堂经验方');

insert into prescription values('加减复脉汤','滋阴养血、生津润燥','温热病后期，邪热久羁，阴液亏虚，身热而赤，神倦，口干舌燥，舌质鲜红，脉虚大，手足心热甚于手足背者','瑞竹堂经验方');

入库图片：



# 算法设计

## 5.1问题描述

问题描述：如何通过算法实现对用户的包邮推荐问题。

具体问题描述：在满100包邮的条件下，用户已购买一部分的商品，若想要在此基础上，凑够邮费的同时，又能获得最高的价值量（商品的销售量/10），并且能够推荐尽量多的商品，系统该如何对用户的进行商品的推荐。

这个问题可以通过0/1背包问题的模型解决，在这里为得到最逼近包邮条件的推荐商品即获得整体最优解，选择用动态规划来处理。

1. 背包容量即100-(目前购物车商品总价）
2. 商品重量即商品的单价
3. 商品的价值：本着销售量越高价值越大的设计理念，商品的价值即商品的销量/10；

将问题模型化后带入动态规划0/1背包问题算法进行计算，得出在推荐种类尽可能多的情况下（即每种商品只选择一次），在最逼近包邮价格的条件下，推荐商品的总价值量最大。

## 5.2算法流程图



## 5.3代码描述

监听事件中调用部分：

calculate=0;

for (int i =0;i<=purchase.getCountthing();i++)

{

calculate+=purchase.getPrice()[i]\*purchase.getNumber()[i];

}//计算当前商品的总价

translate(100-calculate);//传入背包容量

Intent intent = new Intent(Purchase.this, Commend.class);

startActivity(intent);

封装的方法：

void translate(int C)

{

ProvinceDao db=new ProvinceDao(Purchase.this);

List<String> getW=db.getweight();

List<String> getV=db.getvalue();

int w[]=new int [23];

w[0]=0;

int v[]=new int [23];

v[0]=0;

for(int i=1;i<23;i++)

{

w[i]=Integer.valueOf(getW.get(i-1));

v[i]=Integer.valueOf(getV.get(i-1))/10;

}

int get\_goodthing[];

get\_goodthing= knapSack(w,v,22,C);

GetPosition t=new GetPosition();

t.setrecommend(get\_goodthing);

}

Translate方法用来作为中转，将从数据库中接收的价格和销售量转化为int型数组，并处理后作为参数传入knapSack方法，同时将的出的结果即get\_goodthing[]存入GetPosition类中。以便在Commend 活动中调用。

int[] knapSack(int w[],int v[],int n,int C)

{

int V[][]=new int [n+1][C+1];

int i,j;

int x[]=new int[n+1];

x[0]=0;

for(i=0;i<=n;i++)

{

V[i][0]=0;

}

for( j=0;j<=C;j++)

{

V[0][j]=0;

}

for(i=1;i<=n;i++)

{

for(j=1;j<=C;j++)

{

if(j<w[i])

{

V[i][j]=V[i-1][j];

}

else

{

V[i][j]=V[i-1][j]>(V[i-1][j-w[i]]+v[i])?V[i-1][j]:(V[i-1][j-w[i]]+v[i]);

}

}

}

j=C;

for(i=n;i>0;i--)

{

if(V[i][j]>V[i-1][j])

{

x[i]=1;

j=j-w[i];

}

else x[i]=0;

}

return x;

}

通过动态规划，根据传入的参数进行计算得出x[]即对哪些物品进行推荐

Commend活动中获取推荐数组，并将其可视化的展现（有省略）：

int flag[]=t.getrecommend();//用flag数组接收推荐数

for (int i =1;i<23;i++)

{

if(flag[i]==1)

PurchaseList.add(new Purchaseclass(R.drawable.herb5,getname.get(i-1),"价格：￥"+getPrice.get(i-1),"包邮推荐"));

}//从1开始遍历数组，当值为1即物品被选中时，将其可视化的作为item显示；

}

}

相关代码（数据库中获取数值，有省略）：

List<String> proList = new ArrayList<String>();

db = helper.getReadableDatabase();

Cursor cursor = db.rawQuery("select sales from sale",null);

//sql语句获取sales并存入游标中

while (cursor.moveToNext())

{

String value = cursor.getString(0);

proList.add(value);}

//从游标中获取数据（商品价值）并存入list<String>中

Cursor cursor = db.rawQuery("select price from things", null);

//sql语句获取所有商品的价格

while (cursor.moveToNext())

{

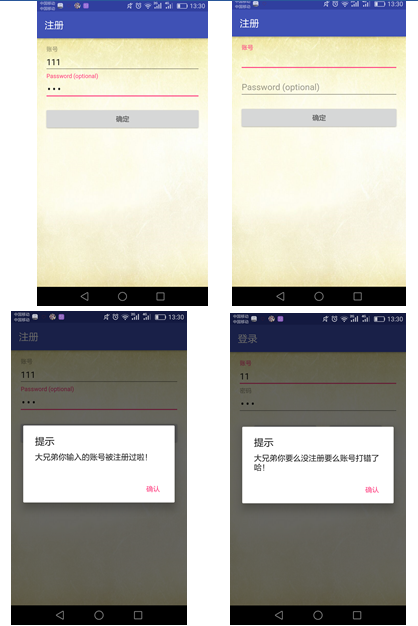
String value = cursor.getString(0);

proList.add(value); }

从游标中获取数据（商品价格）并存入list<String>中

# 运行测试

测试截图：











# 总结体会

通过这次大作业，将我这学期所学的大部分知识都融合在了一起，包括数据库、java、安卓、算法，让我将在书本上所学的知识运用到了实践中，并且完成了一个小型的app。

这段时间经历了以下这个过程：

设计—上网学习—出bug—改bug—再出bug—再改bug—以为没bug了进行数据库连接—bug层出不穷—上网学习—再改…..（最后三步反复进行）

我感觉，开发的过程就是一个学习，实践，出错，改错的循环反复过程，接下是从Java，数据库，算法三个不同的层面总结这段开发过程。

**Java总结：**

Android的界面布局和java很类似但又有不同，他有专门的xml文件进行存放不同活动的布局，在其中可以设置摆放在其中的控件的一些设置比如大小，留白等等。

在开发过程中遇到的第一个问题就是怎样实现底部导航栏的切换，虽然android studio中用相应的模板但是还是需要进行创建fragment以及继承fragment的类，经过一段时间查询网上论坛和询问他人后，后终于实现了这个功能，他需要将切换的界面写在不同的fragment中因为底部导航栏的切换其实是在一个活动上不同碎片的切换

在界面的设计上，主要使用的控件是ListView，而其子布局上控件内容的设置和取出由一个class进行封装，再重写一个适配器。如此可以使同一个布局根据不同的条件触发显示不同的内容，实现代码的复用。在布局相似的情况下同一个布局进行封装后可以给不同的activity使用。

由于Dialog使用的方便性，在个人中心以及自诊信息修改的时候，使用的是自定义布局的AlertDialog，从而实现在Dialog中加入自己想要的控件，并可以在xml文件中对其字体格式等进行设置。

通过Dialog用户输入的信息在获取后若想及时刷新TextView上的信息需要通过适配器对象调用adapter\_p.notifyDataSetChanged()，实现数据的即时更新。

而在ListView的Item中控件上设置点击事件不能写在activity中而是写在ListView的适配器中，并且在适配器中对item上TextView的修改可以直接通过其对象修改并也不需要调用方法就能实现及时刷新。

在activity中判断RadioGroup中被选中的RadioButton可以使用RadioGroup的getCheckedRadioButtonId方法获取选中的按钮的id，并判断是等于R.id.XX控件的id。

在使用静态数组为程序暂存数据的时候要初始化并给其开辟好空间，否则后期在赋值时会直接更改引用导致之前暂存的数据丢失，甚至出现数组越界导致程序中断的情况

因为经常在不同的activity中需要传递一些position值，我封装了一个类，类中存有各种静态变量，及其Set和Get方法，通过类对象实现不同activity之间数值的传递。

同样的原理，为了实现购物车的功能，将用户先选定的商品信息用Purchase\_function(自定义)中的static变量和数组实现缓存。通过将指针countthing值置为-1来实现清除购物车。

**数据库总结：**

在数据库方面，因为之前一直使用的SQL2010，第一次使用SQLite及其可视化工具SQLite Expert。它语法的使用和SQL2010差别不大但有些细节上的差别比如SQL2010中nvchar（max）可以提供大量文本存储而在sqlite中提供text可存储大量多行文本。以下是在开发过程中数据库嵌入前端应用程序的一些学习总结。

在连接时遇到第一个问题就是拷贝外部数据库到程序中，后来通过学习网上的一些论坛以及询问了高年级的同学后了解了，要在程序中首先封装一个类CopyDatabase给定外部数据库的存储路径通过二进制文件读写的方式将原文存储在assets文件夹下的数据库文件写入路径为data/data/项目名…中然后通过封装类MyDataHelper 继承自SQLiteOpenHelper给出数据库的名称和版本号，最后封装类ProvinceDao并以MyDataHelper和SqliteDatabase为成员进行数据库的连接，并在类中写入所需数据库操作的方法。

Android中的SqliteDatabase提供了很多数据库操作的方法，我主要使用了其中的execSQL()方法和rawQuery（）方法，后者主要用来进行数据库查询，前者主要用来进行数据库的插入，更新操作，将其定义在事务中。后者会返回一个Cursor对象存储从表中查询出的数据集。

在程序中我把所有的数据库操作语句封装成方法，根据其功能和要求添加一定的参数和返回对象。再把这些对象统一封装在一个类里，当要调用的时候再通过定义类对象进行方法的调用从而提高程序的封装性。

为了提高程序的复用性，app中的健康养生推文模块，将推文的内容分成5个部分存在数据库中，在加载页面的时候通过自定义的GetPosition类对象获取上一个ListView的position从而确定从数据库中调出哪一篇推文的内容信息。

**算法总结：**

1、算法的代码不是很复杂，但是在实现的时候有些细节，比如数组下标越界的问题还是需要注意一下，如果不注意还是会出现一些难以预料的中断。

2、因为问题中想要获得是一个整体最优解，所以在权衡过暴力，贪心和动态规划的优缺点以后选择了使用动态规划来获得一个整体最优解。

3、在数据库调用和可视化展现的时候也同样需要注意数组下标的问题以及从数据库中取出数默认为String类型的问题，需要用一个方法对从数据库中取出的数进行预处理，将其中的数从下标为1的位置开始存起。

4、可视化的展现是使用了listView，需要将其中每个item布局的控件的操作用一个类进行封装，并且重写adapter类以之前的类作为类型。

最后在activity中调用相应的适配器给；listview对象。并且根据推荐数组选择是否将商品加入ArrayList<类型>对象中（即是否进行推荐）。