

摘要

本次课程设计通过使用 Android Studio 软件完成一个安卓应用程序的设计。依据选题要求，本论文基于一个蛋糕店的场景进行综合管理模块的 APP 开发。需要完成蛋糕管理、会员管理、订单管理与查询管理四个主要模块。

依据考察内容，假设学校校内新开设一家蛋糕店，该店员希望学生为其设计一个蛋糕预定 APP，主要功能是蛋糕管理，如蛋糕的增加和删除，通过查询功能搜索特定的蛋糕；会员管理，如用户登记相应的信息可以成为本店会员；订单管理，如会员通过 APP 进行蛋糕的选购和下订单，店员可以在管理后台看到哪些用户下了哪些蛋糕的订单，以及订购顺序，便于其进行蛋糕的制作和配送等。通过使用 Intent、Activity、Fragment 以及 ListView 技术完成前台页面逻辑的设计，后台使用 SQLite 的技术，完成本次考察内容。

关键词： 蛋糕店；移动应用开发；管理程序设计；前后端交互

1、需求分析

(1) 蛋糕管理

蛋糕管理是本次程序设计的主要部分，店员通过管理员面板设置蛋糕的相关参数，可以设置蛋糕的名称，当天的库存以及销售价。用户可以通过查询面板对特定蛋糕进行查询，满足个性化的消费需求。

(2) 会员管理

会员管理是本次程序设计的重要组成部分，用户的蛋糕查询，下订单和退订单都是基于会员管理模块。通过会员管理，用户可以在该蛋糕店注册成为会员，为了简化会员注册的流程，仅需记录会员的账号和密码；会员姓名以及会员的手机号即可完成一位会员数据记录的建立。

(3) 订单管理

订单管理也是本次程序设计的重要组成部分，用户选购的订单，以及管理员对蛋糕制作顺序的安排都基于订单管理。并且通过会员的手机号与下订单的蛋糕建立数据库关系，便于管理员观看是哪位用户下单了什么蛋糕，在配送时，可以通过用户预留的手机号码进行联系。

(4) 查询管理

查询管理是本次程序设计的一般组成部分，通常，一个蛋糕店中的蛋糕种类是比较多的。用户如果在繁多种类的蛋糕找到一款想购买的蛋糕通常来说比较麻烦。此时，用户通过查询模块，就可以获得他们希望选购的蛋糕，提升用户的使用体验。

2、总体设计

根据蛋糕店场景的实际情况和考察要求，将页面分为综合操作；商品展示和用户管理三个页面。其中，综合操作页面下又分为蛋糕查询，蛋糕预定和订单退订三个子页面，使用 Fragment 技术实现。商品展示中，使用 ListView 带出蛋糕店中库存的蛋糕信息，在用户管理中，用户可以通过输入自己预留的电话号码查询自己所订购的蛋糕记录，同时也有用户注册按钮，便于普通用户注册成为会员。而在另一端的管理员界面，可以通过核验管理员账号和密码对全体成员蛋糕订购情况进行观察，对会员信息进行管理，对商店蛋糕进行管理。该程序最终能达到基本应用于现实生活的场景。

本次程序设计的总体结构图如图 1 所示。

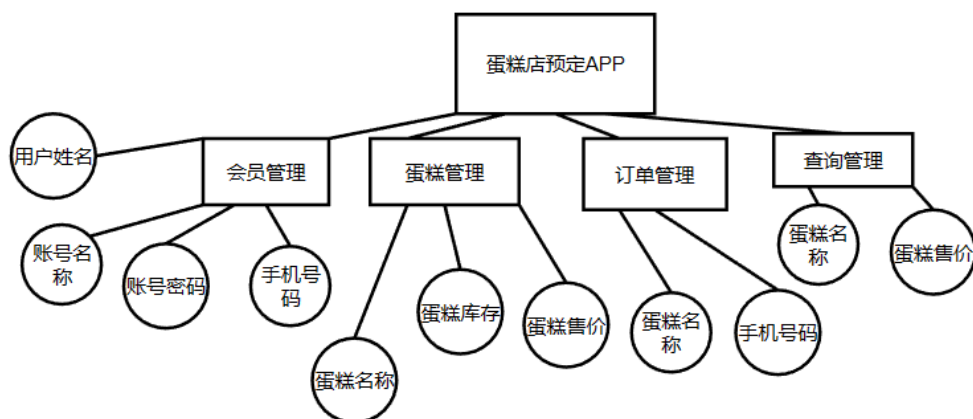


图 1 蛋糕店预定 APP 总体结构图

同时，本次程序设计考虑到前后端的交互，需要进行数据库设计。本次程序设计的数据库关系 E-R 图如图 2 所示，数据库表的结构如图 3 所示。

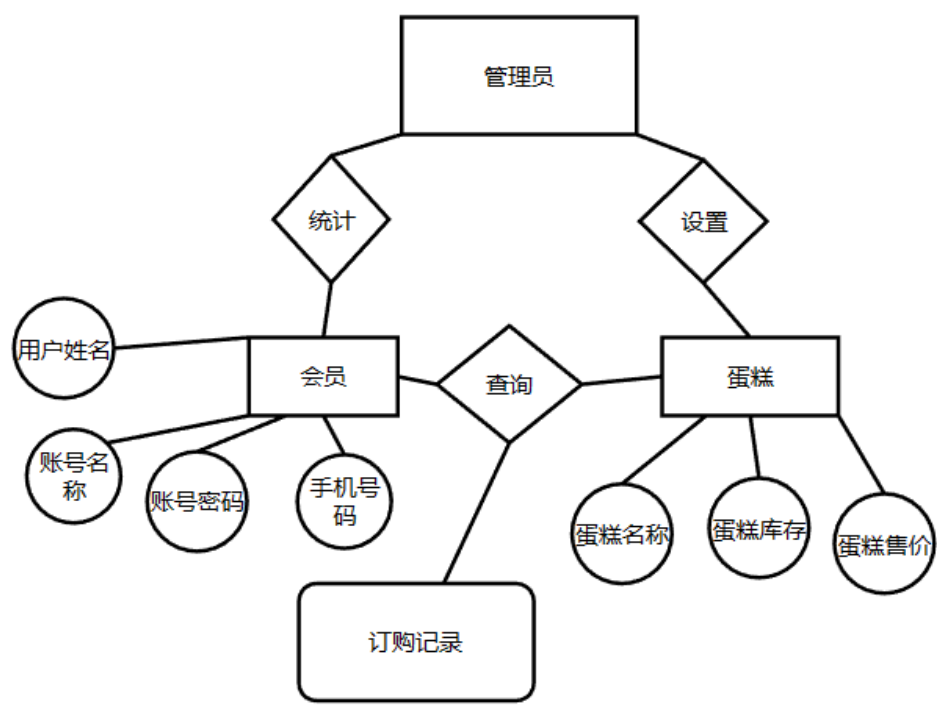


图 2 蛋糕店预定 APP 数据库 E-R 图

▼ Tables		
▼ Cake		
id	INTEGER	"id" INTEGER
name	TEXT	"name" TEXT
storage	INTERGER	"storage" INTERGER
willbuy	INTERGER	"willbuy" INTERGER
price	TEXT	"price" TEXT
▼ Checkout		
id	INTEGER	"id" INTEGER
usernumber	TEXT	"usernumber" TEXT
cakeName	TEXT	"cakeName" TEXT
cakeNumber	TEXT	"cakeNumber" TEXT
▼ People		
id	INTEGER	"id" INTEGER
username	TEXT	"username" TEXT
password	TEXT	"password" TEXT
name	TEXT	"name" TEXT
number	TEXT	"number" TEXT
pd	INTERGER	"pd" INTERGER

图 3 数据库表字段结构

3、详细设计

由于本次考察为分小组完成，按照分工要求，我负责页面布局设计模块。在布局模块中，我首先进行了一个主页面的设置，该页面使用 Acitivity 实现，页面中央设置了蛋糕店的 Logo，以及一个提示进入的按钮。此部分代码如图 4 所示。

```

1 package com.example.john.myapplication;
2
3 import ...
4
13
14 public class LoginActivity extends AppCompatActivity {
15     Button login_ok;
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.layout_login);
20         getWindow().addFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS); // 设置状态栏透明
21         login_ok= findViewById(R.id.login_ok);
22         SharedPreferences setting = getSharedPreferences( name: "myActivityName", mode: 0);
23         boolean user_first = setting.getBoolean( s: "FIRST", b: true);
24         if(user_first){// 第一次
25             setting.edit().putBoolean( s: "FIRST", b: false).apply();
26             Toast.makeText( context: LoginActivity.this, text: "欢迎使用广商蛋糕店APP", Toast.LENGTH_LONG).show();
27             mydatabasehelper dbHelper = new mydatabasehelper( context: this, name: "CakeStore.db", factory: null, version: 1);
28             SQLiteDatabase db= dbHelper.getWritableDatabase();
29             importdataclass.importcake(db); // 导入信息
30         }else{// 并非第一次登陆
31             Toast.makeText( context: LoginActivity.this, text: "欢迎回来", Toast.LENGTH_LONG).show();
32         }
33         login_ok.setOnClickListener(view -> {
34             Intent intent=new Intent( packageContext: LoginActivity.this,MainActivity.class);
35             startActivity(intent);
36         });
37     }
38 }

```

图 4 LoginActivity 界面代码

在主要业务设计中，我将功能依据类型划分为三个 Fragment 控件。在代码包中对应 barFragment1-3. 其中,barFragment1 中有蛋糕查询,蛋糕预定和订单退订三个子 Fragment。对应文件是 barFragment1_1-barFragment1_3。均为在屏幕中间设置提示词，用户输入框和提交按钮，进行相应业务的处理，考虑论文篇幅限制，蛋糕预定 Fragment 的关键源代码如图 5 所示，相关联的 XML 布局文件使用 LinearLayout 线性布局样式，代码如图 6 所示。

```

19 public class barFragment1_2 extends Fragment {
20     EditText editText,usernum;
21     Button button;
22     boolean pd=false;
23     String bookname="";
24     String bianhao="";
25     int bookkucun=0;
26     @Override
27     public void onActivityCreated(@Nullable Bundle savedInstanceState) {
28         super.onActivityCreated(savedInstanceState);
29         editText= getActivity().findViewById(R.id.fragment1_2_info);
30         usernum= getActivity().findViewById(R.id.fragment1_2_number);
31         button= getActivity().findViewById(R.id.fragment1_2_button);
32         button.setOnClickListener(view -> {
33             guiHuan();
34             editText.setText("");
35         });
36     }
37     @
38     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
39         return inflater.inflate(R.layout.fragment1_2, container, attachToRoot: false);
40     }
41     public void guiHuan(){
42         mydatabasehelper dbHelper = new mydatabasehelper(getActivity(), name: "CakeStore.db", factory: null, version: 1);
43         dbHelper.getWritableDatabase();
44         final SQLiteDatabase db= dbHelper.getWritableDatabase();
45         Cursor cursor =db.query( table: "Cake", columns: null, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
46         if(cursor.moveToFirst()){
47             do {
48                 String username =cursor.getString(cursor.getColumnIndex( s: "name"));
49                 if(username.equals(editText.getText().toString())){
50                     pd=true;
51                     bookkucun=cursor.getInt(cursor.getColumnIndex( s: "storage"));
52                     bookname=cursor.getString(cursor.getColumnIndex( s: "name"));
53                     bianhao=cursor.getString(cursor.getColumnIndex( s: "price"));
54                     break;
55                 }
56             }
57         }
58     }
59 }

```

图 5 蛋糕预定界面关键源代码

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     android:layout_width="match_parent"
3     android:layout_height="match_parent"
4     android:orientation="vertical"
5     android:gravity="center"
6     >
7
8     <TextView
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="请输入你想订购的蛋糕"
12        android:textSize="30dp"/>
13
14    <EditText
15        android:id="@+id/fragment1_2_info"
16        android:gravity="center"
17        android:layout_width="wrap_content"
18        android:layout_height="wrap_content"
19        android:hint="蛋糕名"
20        android:ems="10"/>
21
22    <EditText
23        android:id="@+id/fragment1_2_number"
24        android:gravity="center"
25        android:layout_width="wrap_content"
26        android:layout_height="wrap_content"
27        android:hint="手机号"
28        android:ems="10"/>
29
30    <Button
31        android:id="@+id/fragment1_2_button"
32        android:layout_width="wrap_content"
33        android:layout_height="wrap_content"
34        android:text="提交" />
35
36 </LinearLayout>
```

图 6 蛋糕预定界面 XML 布局文件

在第二个界面中，其功能是列出蛋糕店库存的所有蛋糕，并展示它们的库存和售价信息，该界面我使用了 ListView 控件，通过与负责蛋糕管理的同学联系，了解到其设计的蛋糕表的结构，通过数据库查询语句，将每条蛋糕记录以列表的形式展示到用户界面中。该界面在代码包中对应的文件是 barFragment2，源代码如图 7 所示，相应的 XML 布局文件如图 8 所示。

```

1 package com.example.john.myapplication.ui.Fragment;
2
3 import ...
4
15
16 public class barFragment2 extends Fragment {
17     private mydatabasehelper dbHelper;
18     String showtext="";
19     @Override
20     public void onActivityCreated(@Nullable Bundle savedInstanceState) {
21         super.onActivityCreated(savedInstanceState);
22         dbHelper = new mydatabasehelper(getActivity(), name: "CakeStore.db", factory: null, version: 1);
23         dbHelper.getWritableDatabase();
24         final SQLiteDatabase db=dbHelper.getWritableDatabase();
25         showtext="";
26         Cursor cursor =db.query( table: "Cake", columns: null, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
27         if(cursor.moveToFirst()){
28             do {
29                 String name =cursor.getString(cursor.getColumnIndex( s: "name"));
30                 int kuchun =cursor.getInt(cursor.getColumnIndex( s: "storage"));
31                 String bianhao=cursor.getString(cursor.getColumnIndex( s: "price"));
32                 showtext+=name+"\t\t\t\t\t"+kuchun+"\t\t\t\t\t"+bianhao+"@";
33             }while (cursor.moveToNext());
34         }
35         String [] data=showtext.split( regex: "@");
36         ArrayAdapter<String> adapter = new ArrayAdapter<>(getActivity(), android.R.layout.simple_list_item_1, data);
37         ListView listView = getActivity().findViewById(R.id.fragment2_listview);
38         listView.setAdapter(adapter);
39     }
40
41     @
42     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) { //动态加载fragment布局
43         return inflater.inflate(R.layout.fragment_2, container, attachToRoot: false);
44     }

```

图 7 显示所有蛋糕的界面源代码

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     android:layout_width="match_parent"
3     android:layout_height="match_parent"
4     android:orientation="vertical"
5 >
6
7 <TextView
8     android:id="@+id/textView2"
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content"
11    android:textSize="30dp"
12    android:textColor="@color/black_de"
13    android:text="蛋糕名称\t\t\t\t\t库存\t\t\t\t\t价格" />
14
15 <ListView
16     android:id="@+id/fragment2_listview"
17     android:layout_height="match_parent"
18     android:layout_width="match_parent"/>
19
20 </LinearLayout>

```

图 8 显示所有蛋糕的界面 XML 布局文件

在第三个界面中，其功能是订单查询、新用户注册以及管理员界面，对于订单查询界面，用户可以通过输入对应的手机号码进行查询，结果返回与之对应的订单信息，显示在

查询框的下方，并给出订购蛋糕的总价（如果一个手机号码订购了多个蛋糕）；而下有一个“用户注册”的按钮，用于给初次访问该 APP 的用户，并且希望在该蛋糕店进行订购的用户进行注册，该按钮链接到负责人员管理同学所设计的增加用户对话框中，用以完成普通用户的新增；而在管理员界面中，通过设置提示文本，醒目背景颜色，表明该界面为管理员界面，管理员需要对账户进行验证：输入完管理员账号和密码后，需要点击验证按钮，如果验证通过，则弹出管理员控制面板，如果验证失败则给出提示，对于这部分界面，对应代码包中文件为 barFragment3 及 barFragment3_1 和 barFragment3_2，关键代码如图 9 所示，XML 布局文件如图 10 所示。

```
47 public void guiHuan(){
48     mydatabasehelper dbHelper = new mydatabasehelper(getActivity(), name: "CakeStore.db", factory: null, version: 1);
49     dbHelper.getWritableDatabase();
50     final SQLiteDatabase db= dbHelper.getWritableDatabase();
51     Cursor cursor =db.query( table: "Checkout", columns: null, selection: null, selectionArgs: null, groupBy: null, having: null, orderBy: null);
52     if(cursor.moveToFirst()){
53         do {
54             usernumber =cursor.getString(cursor.getColumnIndex( s: "usernumber"));
55             if (usernumber.equals(editText.getText().toString())){
56                 pd=1;
57                 bianhao+=cursor.getInt(cursor.getColumnIndex( s: "cakenumber"));
58                 abab += cursor.getString(cursor.getColumnIndex( s: "cakename")) + " ";
59                 show="您买了"+bianhao+"元的蛋糕,它们是: "+abab;
60             }
61         }while (cursor.moveToNext());
62     }
63     if (pd==1){
64         showstatus.setText(show);
65         Toast.makeText(getActivity(), text: "您当前的订单花费了"+bianhao+"元",Toast.LENGTH_SHORT).show();
66         bianhao=0;
67         pd=0;
68     }
69     else
70         Toast.makeText(getActivity(), text: "你还没有下订单",Toast.LENGTH_SHORT).show();
71     editText.setText("");
72     abab="";
73 }
74 }
```

图 9 普通用户操作界面部分关键代码


```

1      <?xml version="1.0" encoding="utf-8"?>
2      <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3          android:layout_width="match_parent"
4          android:layout_height="match_parent"
5          android:gravity="center"
6          android:orientation="vertical">
7          <EditText
8              android:gravity="center"
9              android:layout_width="wrap_content"
10             android:layout_height="wrap_content"
11             android:id="@+id/fragment3_1_info"
12             android:hint="输入本人电话号码"
13             android:ems="10"/>
14          <Button
15              android:id="@+id/fragment3_1_button"
16              android:layout_width="wrap_content"
17              android:layout_height="wrap_content"
18              android:text="查询个人订单"/>
19          <TextView
20              android:id="@+id/fragment3_1_showname"
21              android:layout_width="wrap_content"
22              android:layout_height="wrap_content"
23              android:textSize="30px"
24              android:textColor="@color/black_de"
25              />
26          <Button
27              android:layout_margin="50pt"
28              android:layout_width="wrap_content"
29              android:layout_height="wrap_content"
30              android:layout_gravity="center"
31              android:id="@+id/admin_reg"
32              android:text="用户注册"/>
33      </LinearLayout>

```

图 10 普通用户操作界面 XML 布局文件

上述三个 Fragment 是组成本次程序设计的重要组成部分。每个 Fragment 所实现的功能由负责相应功能模块开发的同学完成，我负责这些 Fragment 的设计和后续蛋糕管理、用户管理、及侧边栏设计的功能接入，后续相应的管理模块和侧边栏设计由相应同学完成展示。

4、程序运行结果测试与分析

因为本程序最终运行结果测试需要所有负责同学都完成开发后才能测试运行，因此本文的程序运行结果测试与分析是在本程序完成最终开发的前提下进行测试的。

在本程序编写完成后，在模拟器设备上和实机设备上（实机设备型号为：小米 10）都

进行了测试。显示效果基本符合预期，程序运行结果截图如下图 11-图 14 所示。其中，如图 11 所示，该截图表示 APP 的第一个页面，在代码包文件中对应 LoginAcitivity。图 12 用来存放蛋糕店的商品列表信息，在代码包文件中对应 barFragement2。图 13 为用户管理界面设计，负责处理用户订单的查询和管理员界面的进入，在文件代码包中对应 barFragement3,barFragement3_1,barFragement3_2。图 14 为蛋糕预定功能模块的展示。

而剩下的 AboutProject, FollowUs 和 MainEntrance 为侧边栏的页面组件，该部分详细展示由负责该模块的同学进行展示，本文不给出。



图 11 程序主页面

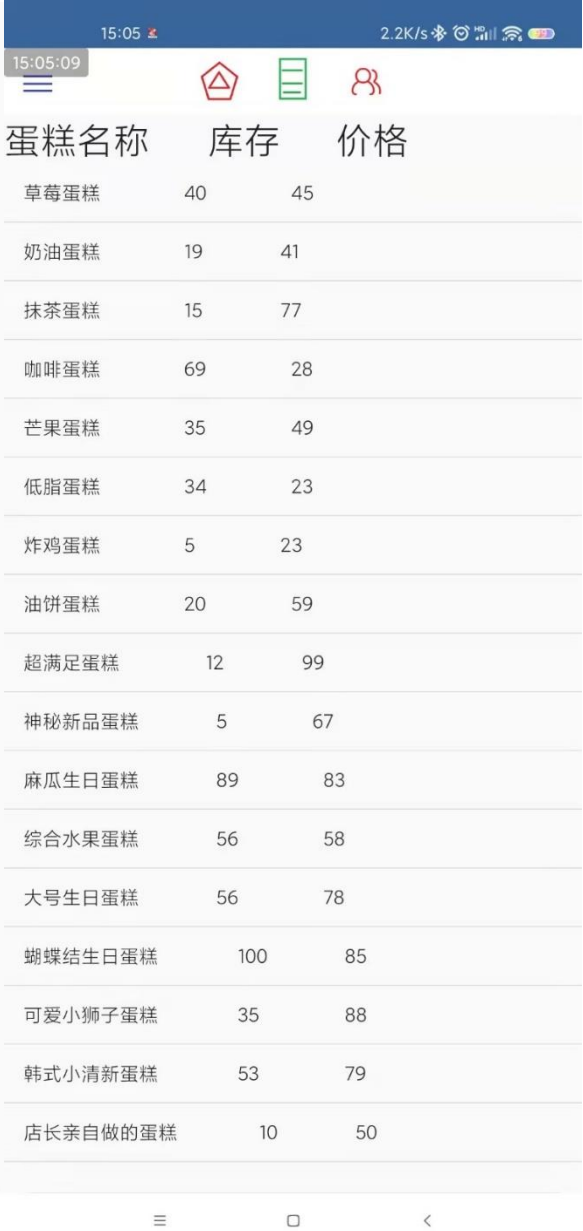


图 12 展示所有蛋糕信息页面



图 13 普通用户操作界面

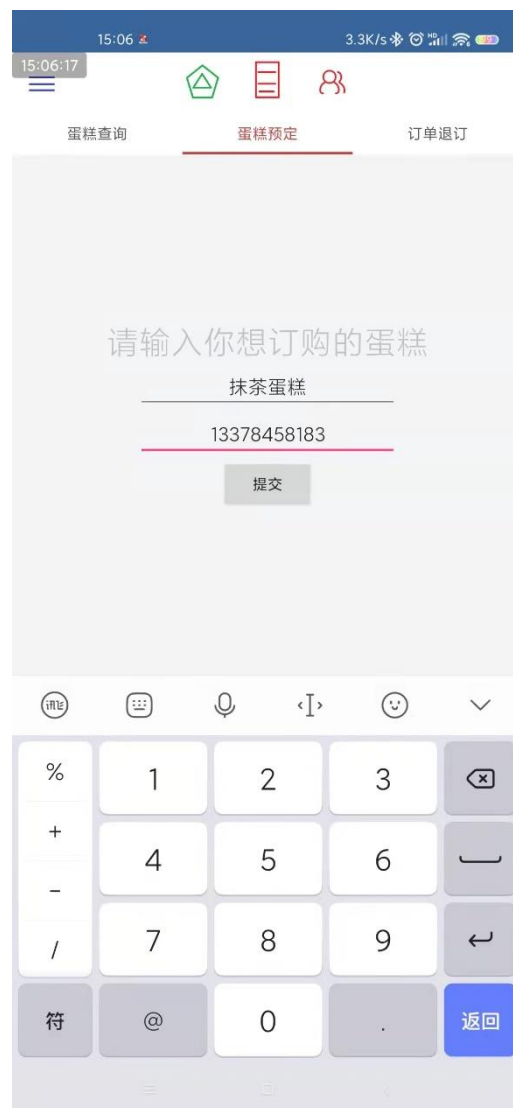


图 14 普通用户蛋糕预定界面

5、结论与心得

在本次考察设计中，我们小组设计了一个蛋糕店预定 APP，该 APP 对蛋糕店中常见的业务进行了程序设计，虽然界面相对简单，但是基本和主要的业务功能可以正常实现。例如一般管理系统中的增加、删除、修改和查询操作。在本项目的蛋糕管理和会员管理模块中均给出了实例展示。管理员可以对蛋糕店的会员和蛋糕店内的蛋糕分别进行上述操作。

我负责的界面设计中，使用了 Activity 和 Fragment 两种页面控件，Activity 主要应用在了初始界面和功能界面的切换，Fragment 用于功能界面之间的切换，而在 Fragment 中，切换页面的方式使用到了 Intent 的控件，基本将本学期所学的知识进行了简单的整合和应用。

但由于时间关系和个人学习能力的限制，要实现一个完整的商用级 APP 是相当复杂的，本案例仅给出了基本业务功能的设计和展示。使基本业务操作可以正常进行。但在现实情况，因为存在许多更详细的因素，例如配送员接单和配送，需要拉取用户的信息和电话号码，以及用户可以定制蛋糕的配料和成分，致使该程序的开发周期会更长，代码内容也更复杂，对于这部分内容，可以在后续的开发过程中逐步增加修改，还有就是如果某个蛋糕库存量只有 1 了，此时有两个用户进行下单，就需要在早下单的那个用户设置锁，将该蛋糕进行锁定，使其他用户不能再进行操作，避免两用户同时下单，而实际库存只有 1 个蛋糕，造成分配问题。