



中南大學
CENTRAL SOUTH UNIVERSITY

应用基础实践二

医院看病预约应用

学生姓名：_____

学 号：_____

指导教师：_____ 杨希

学 院：_____ 计算机学院

专业班级：_____

日 期：_____ 2022/1/10

目录

一、	问题描述.....	1
二、	项目环境.....	2
三、	需求分析.....	2
四、	概要设计.....	3
五、	android 详细设计.....	8
六、	web 服务器详细设计.....	21
七、	运行结果.....	24
八、	调试报告.....	34
九、	心得体会.....	38
	参考文献.....	39
	附录：源码.....	39

一、 问题描述

1、 课程目的

通过指导学生上机实践,让学生学会综合运用所学过的《计算机程序设计基础》、《数据结构》、《算法》、《Java 语言与系统设计》、《数据库》、《Web 技术》、《移动应用开发》等课程的基础知识,从而能够熟练掌握开发市面上比较流行的移动应用开发的基本技能,设计和开发出具有一定规模的 Android 客户端应用+JSP (PHP) 服务器端编程+MySQL 数据库的典型移动互联网应用。

2、 课程设计的基本要求

知识: 了解 HTML、HTML5、CSS、JavaScript、JSP、Android 等开发技术在实际移动互联网应用开发过程中的基本用法;了解移动互联网应用开发从需求分析、系统设计、模块设计、开发以及调试的各个环节。掌握典型移动互联网应用各个环节的技术以及具体运用方式;了解移动互联网典型应用开发是如何结合 Web 相关技术 (HTML、HTML5、CSS、JavaScript、JSP 等)、移动客户端开发技术 (Android 为例) 以及数据库技术 (以 MySQL 为例) 进行开发。

能力: 通过对 Web+Android+MySQL 移动应用开发的案例的设计、编码和调试工作,将对移动互联网应用开发的理性认识转变为感性认识,初步体会和理解计算机相关学科的软件项目工程的具体概念,体会工程的复杂性;学会将理论知识与实际生产结合,学会用工程的角度去提出问题、分析问题和解决问题;初步建立移动互联网应用软件开发项目开发项目需求分析、总体设计、模块设计、编码实现和调试的完整概念,提高实际动手能力,为具备大型移动互联网应用开发软件系统从设计到开发的能力打下坚实基础。

素质: 通过实际动手设计和开发多个完整的移动互联网应用实例,培养理论知识的综合运用能力和动手编程开发的能力,培养软件工程开发的流程化管理观念;在实习中理解并遵守软件工程开发的职业道德和规范,履行责任,提高职业规范素质。

3、 具体内容与要求

1) 用 Android 开发看病预约客户端。

- 2) 用 JSP 或 PHP 开发预约挂号 Web 服务器。
- 3) 用 MySQL 做预约挂号 Web 服务器的后台数据库。
- 4) 可以预约挂号、查看医生介绍及日程安排列表等功能。
- 5) 可以实现患者注册、登录功能。
- 6) 在线挂号支付。

二、 项目环境

- 1) 数据库

MySQL 8.0.27

- 2) Web 服务器

集成环境：IDEA

部署服务器：Tomcat 9.0.55

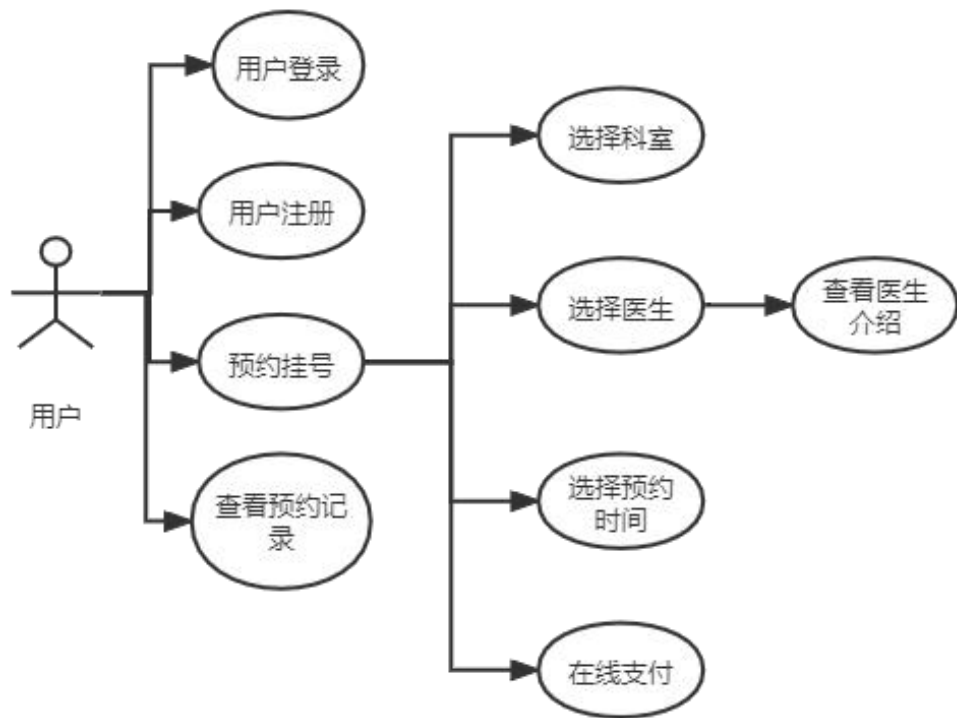
- 3) Android

集成环境：Android Studio

compileSdk 31

三、 需求分析

本应用作为医院看病预约应用，用户为患者，针对患者的基本需求，包括注册和登录、预约挂号、查看预约记录等功能。用例图如下：



四、 概要设计

1、 数据库

数据库中包含 5 个表，分别为：

- 1) user 用户表存储用户相关信息
- 2) doctor 医生表存储医生相关信息
- 3) depart 科室表存储科室相关信息
- 4) schedule 排班表存储医生的排班信息
- 5) myorder 预约记录表存储用户的预约记录

以上表的部分数据如下：

user： 主要包括用户姓名、账号、密码

uid	uname	upsw
001	李明	123
002	王欢	123
003	余科	123
004	姜朝阳	123
006	安逸	123
007	庄念	234

doctor:主要包括医生编号、医生姓名、职务、介绍、所属科室、性别详细信息

did	dname	dlevel	dinfo	departid	sex	ddetail
1	王刚	主治医生	擅长皮肤过敏治疗、红斑狼疮、过敏性皮肤病、白癜风、银屑病、脱发、性病等的诊疗和皮肤病理诊断	1	1	医学博士，现为中南大学湘雅二医
2	许可依	副主任医师	擅长急性肠胃炎治疗	2	0	医学博士，现为中南大学湘雅二医
3	许华	主治医生	擅长急性肠胃炎治疗	3	0	医学博士，现为中南大学湘雅二医
4	柳琦	主治医生	擅长荨麻疹、特应性皮炎、皮肤病理、痤疮、湿疹、荨麻疹、白癜风、癣等及疑难皮肤病的诊治	1	0	医学博士，现为中南大学湘雅二医
5	吴珍	副主任医师	擅长过敏性皮炎、皮肤病理、痤疮、湿疹、荨麻疹、白癜风、癣等及疑难皮肤病的诊治	1	1	医学博士，现为中南大学湘雅二医
6	张智康	主治医生	擅长皮肤病理、痤疮、湿疹、荨麻疹、白癜风、癣等及疑难皮肤病的诊治	4	0	医学博士，现为中南大学湘雅二医
7	王星融	主治医生	擅长皮肤病理、痤疮、湿疹、荨麻疹、白癜风、癣等及疑难皮肤病的诊治	1	1	医学博士，现为中南大学湘雅二医
8	余敬廉	主治医生	擅长急性肠胃炎治疗	2	1	医学博士，现为中南大学湘雅二医
9	张三	主治医生	擅长急性肠胃炎治疗	2	1	医学博士，现为中南大学湘雅二医

depart:主要包括科室编号和科室名

departid	departname
1	皮肤科
2	肠胃科
3	血液内科
4	肿瘤科
5	感染科
6	呼吸内科
7	神经外科
8	麻醉科
9	肝脏外科

scschedule: 主要包括医生编号、排名时间、挂号费用

sid	did	time	price
1	1	2022-01-14 下午	20
2	1	2022-01-15 上午	20
3	4	2022-01-15 上午	10
4	5	2022-01-16 下午	10
5	4	2022-01-15 下午	10
6	4	2022-01-16 上午	10
7	5	2022-01-17 上午	10

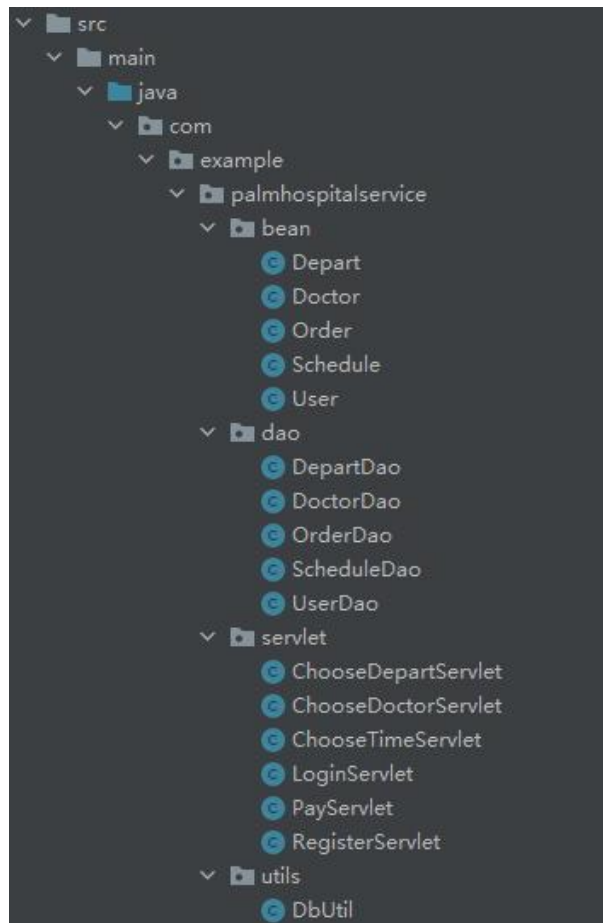
myorder: 预约记录主要包括用户账号、排班编号

oid	sid	uid
1#1	1	1
1#4	1	4
3#1	3	1
3#2	3	2

2、web 服务器

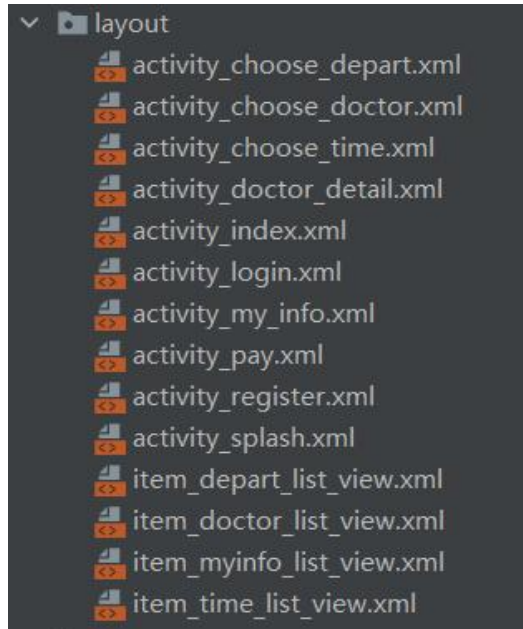
web 服务器中的设计主要包括四个包：servlet、dao、bean、utils

- 1) servlet 处理客户端发送来的请求，业务逻辑处理、并返回结果。
- 2) dao，即数据访问层，对数据库进行处理。
- 3) bean，存放相关的实体。
- 4) utils，存放相关的工具类。

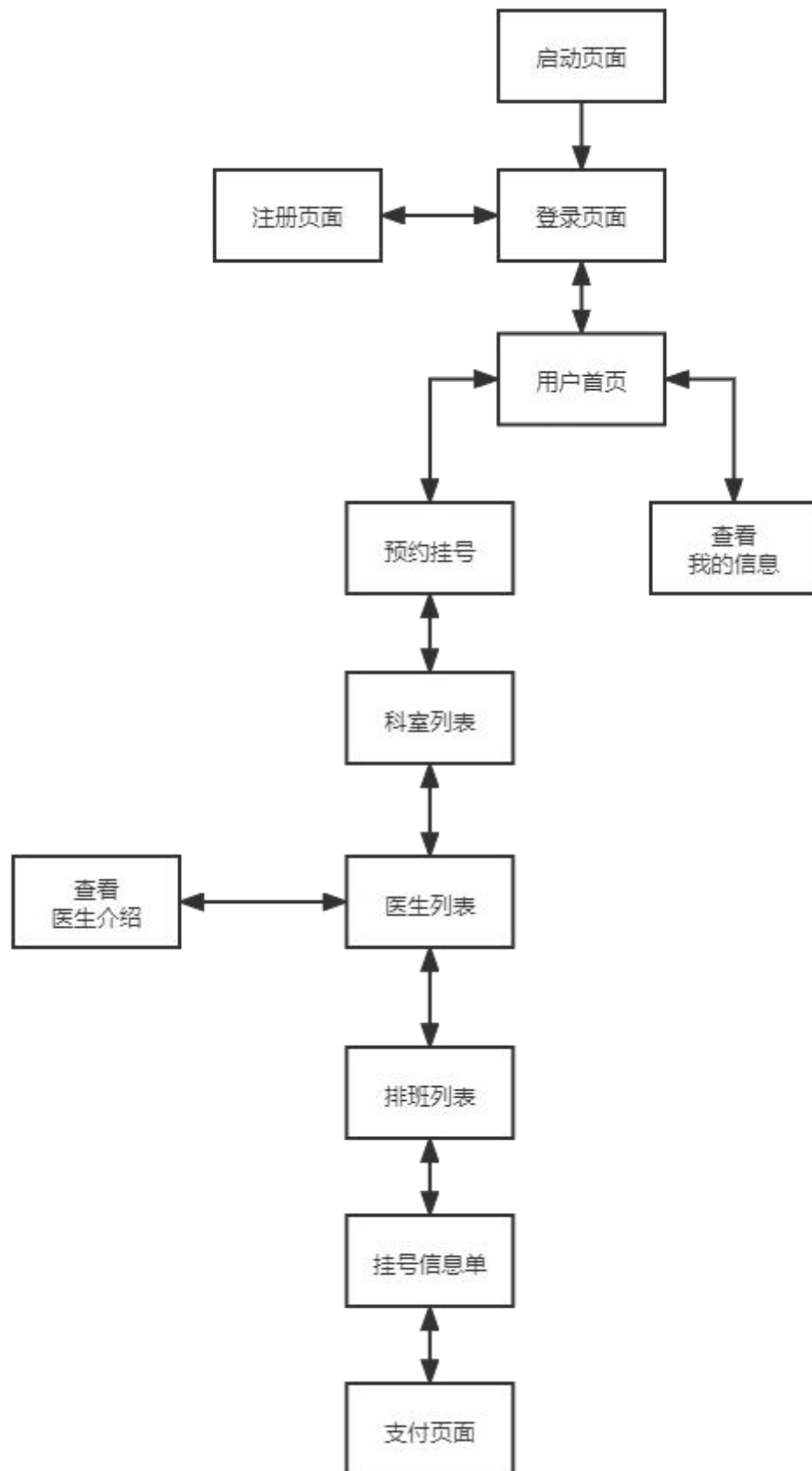


3、 android

- 1) res 资源中， layout 存放布局、 mipmap 和 drawable 存放相关图片资源。 layout 主要包括：注册、登录、首页、查看我的信息、点击预约后呈现的：选择科室、选择医生、选择排班时间、确认预约信息以及支付等等。

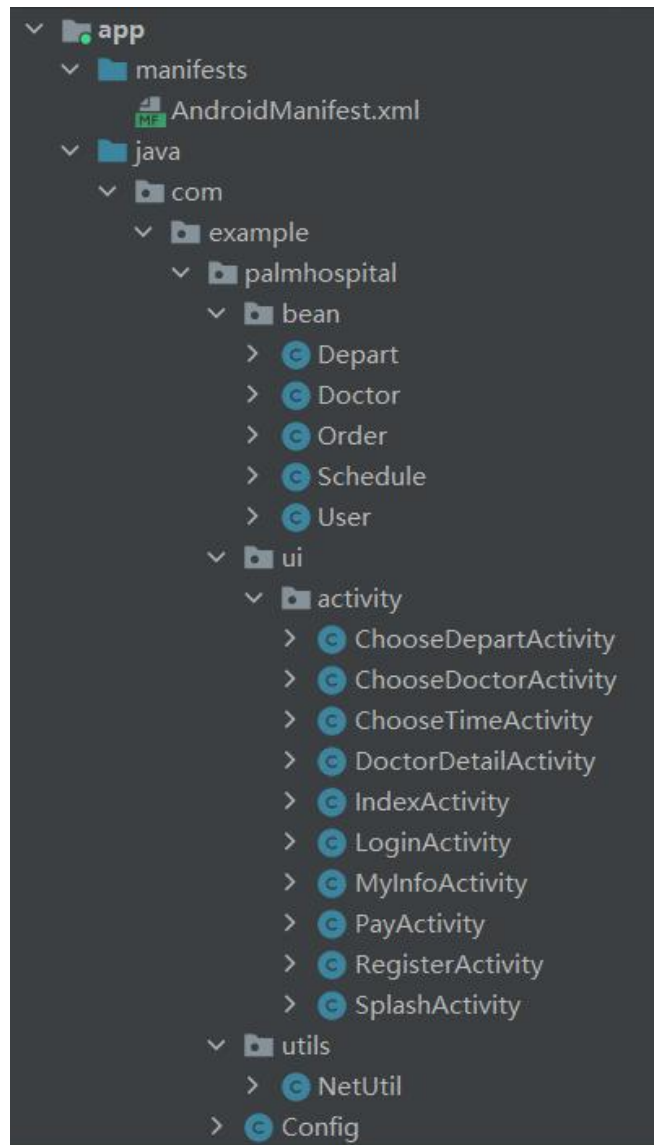


其中的页面功能关系如下：



2) java 代码中主要包括三个包：ui、bean、utils

- ui 即界面，存放 activity，每个 activity 与 layout 对应。
- bean，存放实体类，与服务器部分基本一致
- utils，存放工具类，将部分操作抽象到这里，进行复用



五、 android 详细设计

1、 启动页面

用户进入软件，首先有个启动页面，程序应用名称、应用 logo。用户点击左上角“跳过”或等待 3s 自动跳过。



具体实现对应 activity 为 SplashActivity，当“跳过”被点击时，直接跳转到登录页面，否则，调用 Handler 的延迟 3s 执行跳转

```
public class SplashActivity extends AppCompatActivity {

    private Handler mHandler = new Handler();
    private Button mBtnSkip;

    private Runnable mRunnableToMain = new Runnable() {
        @Override
        public void run() {
            toLoginActivity();
        }
    };

    private void toLoginActivity() {
        Intent intent = new Intent(this, LoginActivity.class);
        startActivity(intent);
        finish();
    }
}
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_splash);

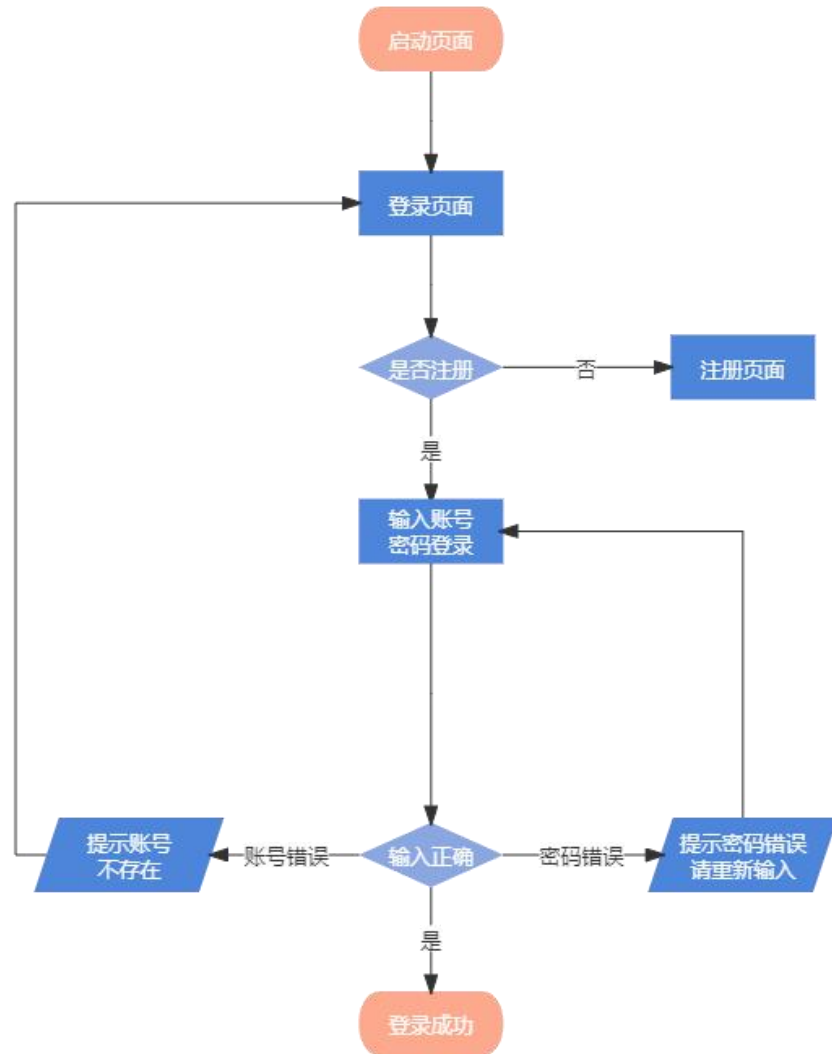
    mBtnSkip = (Button) findViewById(R.id.id_btn_skip);
    mHandler.postDelayed(mRunnableToMain, 3000);

    mBtnSkip.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            mHandler.removeCallbacks(mRunnableToMain);
            toLoginActivity();
        }
    });
}
```

2、用户注册和登录

用户经过启动页面之后，进入登录页面，用户根据自身需求选择：跳转到注册界面，或直接输入账号密码进行登录。

具体逻辑流程图如下：



在注册界面和登录界面，当注册或登录按钮被点击后，客户端首先会简单验证信息。

- 信息是否都填上
- 注册界面中，两次输入的密码是否相同。

当以上条件都满足时，才会将信息提交给服务器端，进行核验，这样子能减少服务器的压力。相关提示都采用 toast 提示在界面上。具体界面如下：

掌上医院	注册账户
请输入账号	请输入姓名
请输入密码	请输入账号
	请输入密码
	请重复输入密码
登录	提交
注册账号	已经注册? 直接登录

3、android 网络操作

android 和 web 服务器的数据交互，涉及到 android 的网络相关操作，可以选择使用第三方库或者自己编写简单的代码实现。我采用了自己编写代码，并抽取到工具类 NetUtil 中。

首先，创建配置类 Config，里面存储了访问服务器用的 url 中基础的部分 baseUrl：<http://192.168.2.102:8080/PalmHospitalService/>，这样在其他类中，只需要将 url 加上这一部分，修改代码很方便。

接着编写工具类 NetUtil：由于数据量不大，我们都采用 Get 的请求方式，将参数添加在 url 末尾。编写函数 requestDataByGet：参数 url，进行 HttpURLConnection 的相关设置，获取请求的返回码，判断是否成功等等

```
public static String requestDataByGet(String urlString) {  
    String result = null;  
    try {  
        URL url = new URL(urlString);  
        HttpURLConnection connection = (HttpURLConnection) url.openConnection(  
    );  
  
        connection.setConnectTimeout(30000);           // 设置超时时间  
        connection.setRequestMethod("GET");           // 请求的方法类型: GET POST  
        connection.setRequestProperty("Content-Type", "application/json");  
        // 获取到的数据格式
```

成功

```
connection.setRequestProperty("Charset", "UTF-8");
connection.setRequestProperty("Accept-Charset", "UTF-8");
connection.connect();           // 发起连接
int responseCode = connection.getResponseCode();    // 获取请求的返回码
if (responseCode == HttpURLConnection.HTTP_OK) {    // 返回码是 200,

    InputStream inputStream = connection.getInputStream();
    result = streamToString(inputStream);
} else {
    String responseMessage = connection.getResponseMessage();
    //Log.e(TAG, "requestDataByPost: " + responseMessage);
}
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
return result;
}
```

其中涉及到将输入流转换成字符串，也编写一个函数 streamToString:

```
public static String streamToString(InputStream is) {
    try {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        byte[] buffer = new byte[1024];
        int len;
        while ((len = is.read(buffer)) != -1) {
            baos.write(buffer, 0, len);
        }
        baos.close();
        is.close();
        byte[] byteArray = baos.toByteArray();
        return new String(byteArray);
    } catch (Exception e) {
        Log.e("LoginActivity", e.toString());
        return null;
    }
}
```

编写完成之后，在 `activity` 中对服务器进行请求，就可以先写好 `url`，然后直接调用 `NetUtil` 网络工具类即可。

4、用户首页

登录成功之后，出现首页，用户可以选择预约挂号、查看我的信息、退出账号等三个功能，代码实现上，只要为三个按钮添加监听事件即可。



退出账号，即退回到登录界面，不再赘述，下面主要讲解预约挂号和查看我的信息两个功能。

5、预约挂号

点击预约挂号之后，呈现所有科室，用户可以根据自己的需要进行选择。



这里涉及到的数据流向：数据库->服务器->客户端。由于网络操作较慢，我们需要将网络操作写在另外的线程中，当数据得到之后，显示在 ui 上，这里还涉及到 ListView。

1) 从服务器获取数据

编写一个类继承 AsyncTask，完成异步任务

```
public class DepartAsyncTask extends AsyncTask<Void,Void, String> {

    @Override
    protected String doInBackground(Void... params) {
        result = NetUtil.requestDataByGet(baseUrl);
        return result;
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        Gson gson = new Gson();
        departs = gson.fromJson(result, new TypeToken<List<Depart>>()
        {}.getType());
        departListView.setAdapter(new DepartListAdapter( departs));
    }
}
```

➤ 重写 doInBackground 函数，表示异步任务执行过程中，我们向服务器发送

请求。

- 重写 `onPostExecute`，表示异步任务执行完成之后，进行的操作，我们将从服务器获取到的数据从 `json` 格式转变为 `gson` 格式，然后利用适配器显示在 `ui` 上，具体实现描述在下面。

2) 将数据显示在 `ui` 上

我们采用 `ListView`，在 `layout` 中，写一个 `ListView`，然后再创建一个 `layout` 文件：`item_depart_list_view`，里面是详细的每一个 `item` 的布局。具体步骤如下：

1. 在 `Layout` 中创建 `ListView`
2. 创建每一行的 `layout`
3. 创建每一行的数据
4. 用 `adapter` 将数据填充到每一行的视图中

接着，编写一个类继承 `BaseAdapter`，主要是实现 `getView`，就是对 `item` 的控件进行具体的设置内容，其中还需要为每个 `item` 添加点击监听事件，并将被点击的科室 `depart` 数据传递给下一个 `activity`。

```
public class DepartListAdapter extends BaseAdapter{
    private List<Depart> mdeparts;
    DepartListAdapter(List<Depart> departs){
        mdeparts = departs;
    }
    @Override
    public int getCount() {
        return mdeparts.size();
    }

    @Override
    public Object getItem(int i) {
        return mdeparts.get(i);
    }

    @Override
    public long getItemId(int i) {
        return mdeparts.get(i).getDepartid();
    }
}
```

```
@Override
public View getView(int i, View view, ViewGroup viewGroup) {

    if(view == null){
        LayoutInflater inflater = (LayoutInflater) getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        view = inflater.inflate(R.layout.item_depart_list_view,null);
    }
    TextView departName = view.findViewById(R.id.depart_name_tv);
    departName.setText(mdeparts.get(i).getDepartname());

    departName.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(ChooseDepartActivity.this, ChooseDoctorActivity.class);
            intent.putExtra("depart", new Gson().toJson(mdeparts.get(i)));
            startActivity(intent);
        }
    });

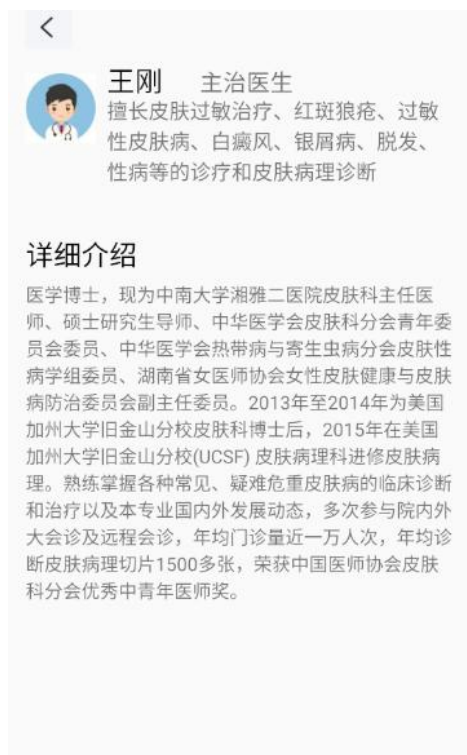
    return view;
}
```

选择科室之后，呈现这个科室的所有医生：



实现的时候，需要向服务器传递科室的 id，即 departid，然后服务器返回 List<Doctor>，然后将所有的医生信息呈现在界面上。

点击医生后，会跳转到医生介绍中：



点击左上角的返回按钮，继续乖乖的医生列表。点击预约的按钮，进入排班信息列表：

这个页面会传递医生 id 给服务器，服务器返回该医生的所有排班信息，即 List<Schedule>:



上面部分是医生的简单介绍，下面的排班详情，包括排班时间和挂号费用。点击任意一个排班，即可进行预约。

确认信息

患者姓名:	李明
医生姓名:	王刚
预约科室:	皮肤科
预约时间:	2022-01-15 上午
挂号费用:	¥ 20.0

确认支付

取消预约

点击确认支付，即可进行支付；点击取消预约即可，取消预约。然后返回刚刚到排班信息列表。

6、查看我的信息

点击我的信息之后，跳转到我的页面

<

李明，欢迎您！

预约记录

时间	科室	医生姓名
2022-01-15 上午	皮肤科	柳琦

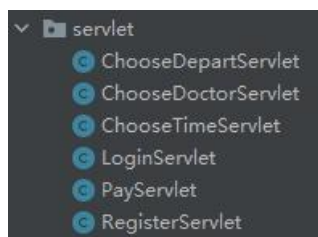
上面是对用户的简单欢迎语。下面是用户的所有预约记录。当点击进入这个界面时，

客户端会将用户 id 发送给服务器，服务器从数据库中查询到所有该用户的预约记录，然后返回到页面上，即 `List<Myorder>`。也是用 `ListView` 呈现数据，与上面的内容类似。

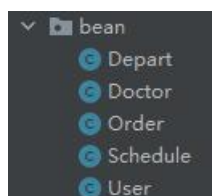
六、 web 服务器详细设计

服务器主要是进行接收客户端的请求、对数据库中的数据进行增删改查、返回数据给客户端。

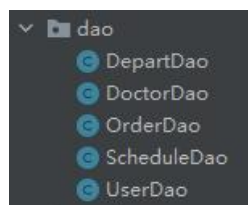
- 编写了 6 个 Servlet，用于接收请求。



- 对于数据库中存储的数据，编写了 5 个实体类：



- 对于每个数据表，编写了 5 个 Dao，对相关数据表进行操作。



下面以处理登录请求为例，说明代码实现过程，其他功能的逻辑基本类似。

1、 bean

针对数据库中的数据，我们编写用户类 `user`，里面包括的成员变量有 `uid`、`uname`、`upsw`。还包括构造函数、`getter`、`setter`、`toString` 函数等。

2、 Servlet

Servlet 中，重写 `doGet` 和 `doPost` 函数，分别处理 `Get` 和 `Post`。流程如下：

- 设置 `request` 和 `response` 的数据格式

- 接收参数
- 调用 Dao，进行数据处理，返回数据
- 判断返回的数据，处理不同的情况，并返回结果给客户端

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html;charset=UTF-8");

    // 接收参数
    String uid = request.getParameter("uid");
    String upsw = request.getParameter("upsw");

    UserDao userDao = new UserDao();
    User user = userDao.login(uid, upsw);

    PrintWriter writer = response.getWriter();
    String result = "";

    if (user == null) { // 登录失败
        response.setStatus(HttpServletResponse.SC_UNAUTHORIZED);
    }
    else { // 登录成功
        response.setStatus(HttpServletResponse.SC_OK);
        result = new Gson().toJson(user, User.class); // 将 user
        // 的数据变成 json 格式，传输给客户端
        writer.write(result);
        writer.flush();
        writer.close();
    }
}
```

而在 doPost 中，直接调用 doGet 即可

```
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    doGet(request, response);
}
```


3、Dao

在 UserDao 中，主要包括登录的查询和注册的添加。对数据库的查询，我们采用工具类 dbutils。针对不同功能，编写不同的 sql 语句。

```
public class UserDao {
    private QueryRunner queryRunner = new QueryRunner();

    public User select(String uid) {
        User user = new User();
        try {
            user = queryRunner.query(DbUtil.getConnection(),"select * from user where uid=?",new BeanHandler<User>(User.class),uid);

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return user;
    }

    public int addUser(User user){
        int res = 0;
        try {
            res = queryRunner.update(DbUtil.getConnection(),"INSERT INTO user values(?,?,?)",user.getUid(),user.getUsername(),user.getUpsw());
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return res;
    }

    public User login(String uid,String upsw){
        User user = select(uid);
        if(user!=null && user.getUpsw().equals(upsw)){ // 该账号存在，能查询到用户且密码正确
            return user;
        }
        else return null;
    }
}
```

```
}
```

- 在登录 `login` 中，首先调用 `select` 查询 `uid` 的用户，如果用户存在且密码和输入的密码相同，则登录成功，我们就返回这个用户信息 `user`，否则返回 `null`。
- 在注册 `register` 中，我们先调用 `select` 查询 `uid` 的用户，如果用户存在，说明这个 `uid` 已经被注册，注册失败；否则，插入用户提交的信息，注册一个新用户。

七、 运行结果

1、 启动页面

启动程序，进入启动页面：上面含有程序应用名称、应用 `logo`。用户点击左上角“跳过”或等待 3s 自动跳过，直接进入登录页面。



2、 登录页面

之后进入登录页面：



掌上医院

请输入账号

请输入密码

登录

注册账号

当用户输入的信息不完整时，提示“账号或密码不能为空”



掌上医院

001

请输入密码

登录

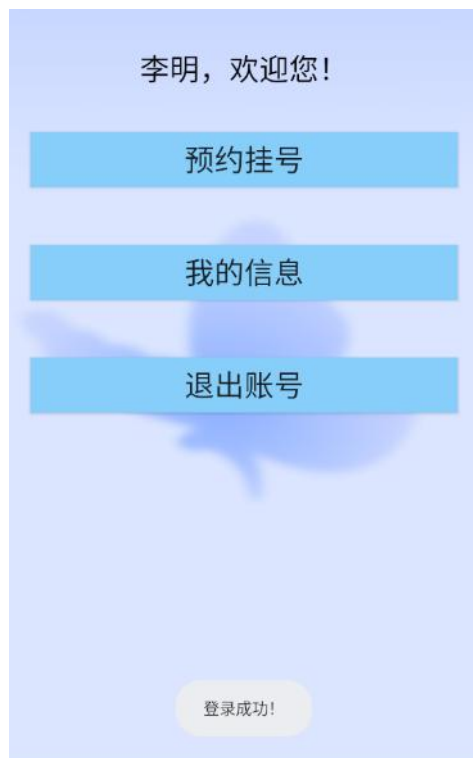
注册账号

账号或密码不能为空

当输入的账号或密码有误时，提示信息如下：



当输入信息正确时，提示信息如下，并跳转到用户首页。



在登录页面中，点击下方的“注册账号”，可进入注册界面。



注册账户

请输入姓名

请输入账号

请输入密码

请重复输入密码

提交

已经注册? 直接登录

3、注册页面

用户信息输入不完整时，提示信息如下：



注册账户

123

请输入账号

请输入密码

请重复输入密码

提交

已经注册? 直接登录

所填信息不能为空!

用户输入的两次密码不相同时，提示信息如下：



注册账户

Tom

001

...

....|

提交

已经注册? 直接登录

两次输入的密码不相同!

当用户输入的账号已经被注册时，提示信息如下：



注册账户

Tom

001

...

....|

提交

已经注册? 直接登录

账号已存在!

当以上情况均不存在时，注册成功，提示如下，并跳转到登录界面：



掌上医院

请输入账号

请输入密码

登录

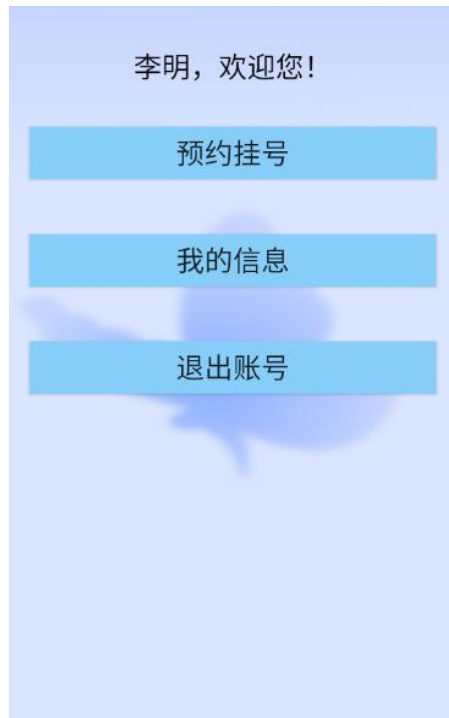
注册账号

注册成功!

当用户已经注册过，可点击下方的“已经注册？直接登录”，直接跳转到登录界面。

4、用户首页

用户首页包含上方的简单欢迎语与下方的三个功能，点击按钮，即可实现对应的功能。



李明，欢迎您！

预约挂号

我的信息

退出账号

5、预约挂号


呈现所有的科室，供用户选择。




点击具体的科室之后，呈现该科室所有的值班医生。如下图，左边为皮肤科，右边为肠胃科：



点击医生的信息，进入医生的详细介绍页：





王刚 主治医生
擅长皮肤过敏治疗、红斑狼疮、过敏性皮肤病、白癜风、银屑病、脱发、性病等的诊疗和皮肤病理诊断

详细介绍

医学博士，现为中南大学湘雅二医院皮肤科主任医师、硕士研究生导师、中华医学会皮肤科分会青年委员会委员、中华医学会热带病与寄生虫病分会皮肤性病学组委员、湖南省女医师协会女性皮肤健康与皮肤病防治委员会副主任委员。2013年至2014年为美国加州大学旧金山分校皮肤科博士后，2015年在美国加州大学旧金山分校(UCSF) 皮肤病理科进修皮肤病理。熟练掌握各种常见、疑难危重皮肤病的临床诊断和治疗以及本专业国内外发展动态，多次参与院内外大会诊及远程会诊，年均门诊量近一万人次，年均诊断皮肤病理切片1500多张，荣获中国医师协会皮肤科分会优秀中青年医师奖。

点击右边的预约按钮，查看该医生所有的排班情况：





柳琦 主治医生
擅长荨麻疹、特应性皮炎、皮肤病理，痤疮、湿疹、荨麻疹、白癜风、癣等及疑难皮肤病的诊治

排班详情

2022-01-15 上午	¥ 10.0
2022-01-15 下午	¥ 10.0
2022-01-16 上午	¥ 10.0

点击具体的某一排班，进入预约信息确认界面：

确认信息

患者姓名：	李明
医生姓名：	柳琦
预约科室：	肠胃科
预约时间：	2022-01-15 上午
挂号费用：	¥10.0

确认支付

取消预约

点击左边的确认支付，返回到排班信息列表，并提示支付成功：

<



柳琦 主治医生

擅长荨麻疹、特应性皮炎、皮肤病理，
痤疮、湿疹、荨麻疹、白癜风、癣等及
疑难皮肤病的诊治

排班详情

2022-01-15 上午	¥10.0
2022-01-15 下午	¥10.0
2022-01-16 上午	¥10.0

预约挂号成功！

点击右边的取消预约按钮，退出预约界面



6、查看我的信息



八、 调试报告

1、 android 访问网络不被允许

报错如下：Cleartext HTTP traffic to localhost not permitted

```
W/System.err: java.io.IOException: Cleartext HTTP traffic to localhost not permitted
W/System.err:     at com.android.okhttp.HttpHandler$CleartextURLFilter.checkURLPermitted(HttpHandler.java:127)
W/System.err:     at com.android.okhttp.internal.huc.HttpURLConnectionImpl.execute(HttpURLConnectionImpl.java:462)
W/System.err:     at com.android.okhttp.internal.huc.HttpURLConnectionImpl.connect(HttpURLConnectionImpl.java:131)
W/System.err:     at com.example.palmsystem.ui.activity.LoginActivity.requestDataByPost(LoginActivity.java:115)
W/System.err:     at com.example.palmsystem.ui.activity.LoginActivity.access$600(LoginActivity.java:28)
W/System.err:     at com.example.palmsystem.ui.activity.LoginActivity$1.onClick(LoginActivity.java:66)
W/System.err:     at android.view.View.performClick(View.java:7448)
W/System.err:     at com.google.android.material.button.MaterialButton.performClick(MaterialButton.java:1119)
W/System.err:     at android.view.View.performClickInternal(View.java:7425)
W/System.err:     at android.view.View.access$3600(View.java:810)
W/System.err:     at android.view.View$PerformClick.run(View.java:28305)
W/System.err:     at android.os.Handler.handleCallback(Handler.java:938)
W/System.err:     at android.os.Handler.dispatchMessage(Handler.java:99)
W/System.err:     at android.os.Looper.loop(Looper.java:223)
W/System.err:     at android.app.ActivityThread.main(ActivityThread.java:7656) <1 internal call>
W/System.err:     at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:592)
W/System.err:     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:947)
I/LoginActivity: null
```

查阅资料发现，android 9.0 之后对于 http 请求有限制。需要进行以下操作：

1. 创建安全配置文件

- a) 在 res 文件夹下创建 xml/network-security-config 文件
- b) 增加 cleartextTrafficPermitted 属性

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <base-config cleartextTrafficPermitted = "true"/>
</network-security-config>
```

2. 添加安全配置文件：在 AndroidManifest.xml 中的 Application 申明

```
<uses-permission android:name="android.permission.INTERNET" />
```

2、 android 网络访问不成功

报错：Failed to connect to localhost/127.0.0.1:8080

```
W/System.err: java.net.ConnectException: Failed to connect to localhost/127.0.0.1:8080
    at com.android.okhttp.internal.io.RealConnection.connectSocket(RealConnection.java:147)
W/System.err:     at com.android.okhttp.internal.io.RealConnection.connect(RealConnection.java:116)
    at com.android.okhttp.internal.http.StreamAllocation.findConnection(StreamAllocation.java:186)
    at com.android.okhttp.internal.http.StreamAllocation.findHealthyConnection(StreamAllocation.java:128)
    at com.android.okhttp.internal.http.StreamAllocation.newStream(StreamAllocation.java:97)
    at com.android.okhttp.internal.http.HttpEngine.connect(HttpEngine.java:289)
    at com.android.okhttp.internal.http.HttpEngine.sendRequest(HttpEngine.java:232)
    at com.android.okhttp.internal.huc.HttpURLConnectionImpl.execute(HttpURLConnectionImpl.java:465)
    at com.android.okhttp.internal.huc.HttpURLConnectionImpl.connect(HttpURLConnectionImpl.java:131)
```

解决过程:

首先, 我先用浏览器输入了服务器对应的 url, 发现是成功访问的, 说明 web 服务器没有问题。接着检查 android, 意识到 127.0.0.1 表示的是本机的 ip 地址, 而在 android 的模拟机中, 不能写这个 ip 地址, 应该写具体的 ip 地址。将 127.0.0.1 替换为我的电脑的 ip 地址即可运行成功。

3、判断字符串为空

在登录界面时, 客户端首先判断用户输入的信息是否位置, 我在判空时, 才用了 `uid==null||upsw==null`, 发现没法实现正确的判空逻辑。改成了如下代码:

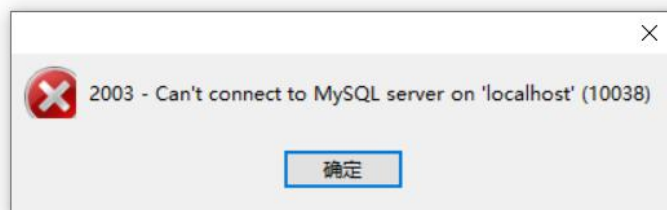
```
if(TextUtils.isEmpty(uid)||TextUtils.isEmpty(upsw)){
    Log.i("LoginActivity","账号或密码不能为空");
    Toast.makeText(getBaseContext(),"账号或密码不能为空",Toast.LENGTH_SHORT).show();
    return;
}
```

发现成功运行。

4、数据库连接失败

1) 问题

运行 web 项目, 弹出如下错误信息, 显示无法连接到 MySQL:



检查电脑中已经启动的 Window 服务：

```

C:\Users\Anwen>net start
已经启动以下 Windows 服务：

Agent Activation Runtime_a7c66
Application Information
AppX Deployment Service (AppXSVC)
AVCTP 服务
Background Tasks Infrastructure Service
Base Filtering Engine
CAJ Service Host
Client License Service (ClipSVC)
CNG Key Isolation
COM+ Event System
Connected User Experiences and Telemetry
Contact Data_a7c66
CoreMessaging
Credential Manager
Cryptographic Services
DCOM Server Process Launcher
Delivery Optimization
Device Association Service
DHCP Client
  
```

发现其中没有 MySQL。

2) 原因：

因为之前设置了 MySQL 启动为手动

	Microsoft Windows SMS 路...	根据...	手动(触发...	本地服务
	Microsoft 键盘筛选器	控制...	禁用	本地系统
	Microsoft 云标识服务	支持...	手动	网络服务
	mysql		手动	本地系统
	Net.Tcp Port Sharing Service	提供...	禁用	本地服务
	Netlogon	为用...	手动	本地系统
	Network Connected Devic...	网络... 正在...	手动(触发...	本地服务

3) 解决方法

启动 MySQL

```

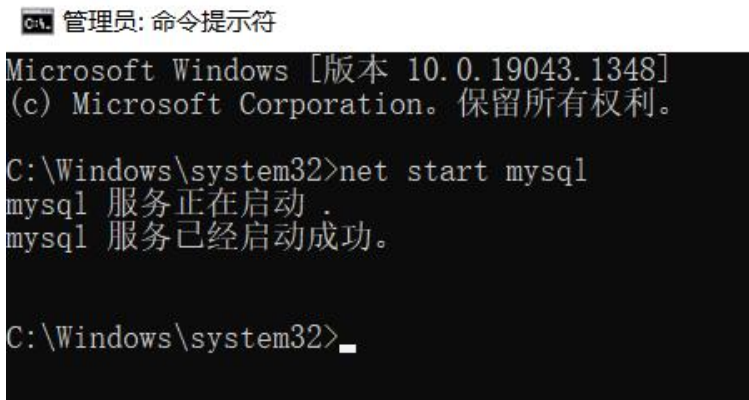
C:\Users\Anwen>net start mysql
发生系统错误 5。

拒绝访问。

C:\Users\Anwen>_
  
```

提示拒绝访问。

检查发现需要以管理员身份启动命令提示符。



```
C:\Windows\system32>net start mysql
mysql 服务正在启动 .
mysql 服务已经启动成功。

C:\Windows\system32>
```

可以发现，启动成功。

5、注册用户，数据库数据没有变化

1) 问题

在实现“注册账号”过程中，需要添加用户。在实现过程中发现，Dao 中修改操作返回 1，说明已经添加了用户，但是查询数据库发现用户数量没有发生变化。

2) 原因

检查后发现，忘记对该事务进行提交。

3) 解决方法

补充代码 DbUtils.commit()即可：



```
@Override
public int updatePsw(String userid, String password) {
    DbUtils.begin();
    int res = userDao.update(userid,password);
    DbUtils.commit(); // 整个事务完成，进行事务提交
    return res;
}
```

6、端口被占用

1) 问题

运行项目，报错：Port already in use :1099


```
Using CATALINA_BASE: "C:\Users\Anwen\AppData\Local\JetBrains\IntelliJ\Idea2021.3\tomcat\b90e5797-8691-469f-9c2e-520c86b28682"
Using CATALINA_HOME: "E:\cs\apache-tomcat-9.0.55"
Using CATALINA_TMPDIR: "E:\cs\apache-tomcat-9.0.55\temp"
Using JRE_HOME: "E:\cs\java\jdk1.8.0_152"
Using CLASSPATH: "E:\cs\apache-tomcat-9.0.55\bin\bootstrap.jar;E:\cs\apache-tomcat-9.0.55\bin\tomcat-juli.jar"
Using CATALINA_OPTS: ""
错误: 代理抛出异常错误: java.rmi.server.ExportException: Port already in use: 1099; nested exception is:
    java.net.BindException: Address already in use: JVM_Bind
Disconnected from server
```

2) 解决方法:

在命令提示符中输入如下命令，查找端口为 1099 的进程，并关闭。

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19043.1348]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Anwen>netstat -aon|findstr 1099
TCP    192.168.43.11:1099    103.192.225.131:443    ESTABLISHED    20564

C:\Users\Anwen>taskkill -f -pid 20564
成功: 已终止 PID 为 20564 的进程。

C:\Users\Anwen>_
```

九、 心得体会

1、 更加熟练掌握相关知识

在本次课程设计中，我将 android 安卓端 + web 服务器端 + MySQL 数据库联系起来，其中 web 和 MySQL 是之前所学的内容，而 android 是临时学。通过本次课程设计，我发现之前学的知识还很不扎实，在这次实战中，边复习边写，快速地掌握了数据库的增删改查、web 服务器端接收请求、处理数据以及 android 布局设计、与用户的交互、网络操作等等，提升非常大，相关知识内容收获丰富，具体都提现在上面的报告中。

2、 提升了自学能力

这次实验，涉及到了不少我之前没有学过的知识点，基本是自己独立一个人完成，在这个过程中，查阅书籍、查阅教程、阅读官方文档和博客并学会应用等，克服了很多困难，也在这个不断学习的过程中锻炼了自己的自学能力。

3、 自顶向下，理清头绪

一开始写课程设计的时候，觉得很复杂，功能繁杂，需要写的部分非常多。但冷静

下来, 仔细分析可以分为三大部分: android 布局、web 服务器处理数据、android 和 web 进行通信。android 包括: layout、activity、bean、utils 等等。web 服务器包括: servlet、bean、dao、utils 等, servlet 接收用户发送的请求, 并返回结果; Dao 进行数据库相关操作。就这样子自顶向下, 逐层划分模块, 就能将大问题化简为许多小问题, 理清头绪之后就可以更顺畅地完成实验。

4、学会看报错, 善于利用输出进行调试

当程序发生报错时, 首先认真看报错信息, 给的报错信息能具体到某一行。对于这一行涉及到的相关代码、函数, 再进行检查, 逐渐找到源头。还可以充分利用输出语句, 不断缩小出错代码的位置、查看某些变量的内容等等。这些良好的调试习惯能帮助我们快速超出问题。而我一开始没有非常认真地看报错信息, 耽误了不少时间, 在实验过程中, 逐渐熟悉报错, 调试越来越快。

5、丰富界面, 优化用户体验

在设计完后, 要多多使用与测试, 在体验功能的过程中, 能发现一些不方便的地方, 针对这些地方再进行改进界面设计等等。比如我一开始在用户登录之后, 直接进行预约挂号, 这显然是不符合用户体验的, 在测试过程中, 我发现了这个问题, 并增加了用户首页, 将用户的三个功能做了一个汇总, 这样子用户使用起来更加舒服顺畅。

参考文献

- [1]郭霖, 第一行代码 android(第 2 版)[M], 北京, 人民邮电出版社, 2016.12
- [2] 尼古拉斯·C·扎卡斯, JavaScript 高级程序设计 (第 4 版)[M], 北京, 人民邮电出版社, 2005
- [3] Bill Phillips Brian Hardy, Android 编程权威指南(第三版) [M], 北京, 人民邮电出版社 2017.

附录: 源码

具体看提交的源代码，其中：

PalmHospitalService，是 web 服务器端，由 IDEA 运行

PalmHospital，是 android 客户端，由 andorid Studio 运行