以记住账号密码。效果如下图所示：

登录界面

注册界面

修改密码界面

用户主界面：

主要操作按键设置较大的圆角矩形，便于操作，并通过添加 android:drawableTop 属性附带贴图，使得功能更加形象。对于下方的通知界面，系统自动获取当前时间，并从数据库导出当天和其后一天的核酸检测通知。其中，核酸通知的两个按钮通过颜色改变表示选中与未选中，绿色为未选中状态，白色为选中状态，当选中其中一个按钮时，另一个按钮便为非选中状态。

注意，核酸检测者界面中心的按钮为扫码按钮，管理员界面中心的按钮为发布通知按钮。效果如下图所示：

普通用户界面

核酸检测者界面

管理员界面

二维码出示界面

点击二维码出示界面后，系统会根据用户的学号生成对应的二维码，并查询出对应的用户信息。点击修改信息后会弹出一个 Dialogue 界面，其每个 EditText 的默认值均设为用户原信息，用户在此基础上进行信息的修改。为了使点击确认修改后只刷新用户信息而不刷新此界面，我们将用户信息以 ListView 的形式显示，当点击确认修改按钮后，调用 notifyDataSetChanged()函数可直接无刷新地更新用户信息。效果如下图所示：

二维码出示界面

修改信息对话框

核酸信息查询界面

用户管理界面、核酸信息统计界面以及未做核酸查询界面的功能大抵相同，我们在这里统一展示。其总体布局均为查询框+ListView 的形式，页面下方设有翻页键和跳转页面键，当页面为第一页时只有向后翻页键，当页面为最后一页时只有向前翻页键。

修改用户界面还设有修改用户类型以及删除用户的按键，无论对 ListView 执行何种操作，结束后均会调用 notifyDataSetChanged()函数直接无刷新地更新信息。

核酸信息统计界面以及未做核酸查询界面的查询条件以及列表呈现的属性有所不同，这一设计也较为符合实际情况。其中时间的 EditView 的默认值设置为当天。对于未做核酸查询，要么是导员使用，要么是班长或班干部使用，所以只需根据学院和班级去查询，有时也需要电话询问没做核酸的同学，如果想查询具体某位同学是否做了核酸，可以根据其学号去核酸信息统计模块查询。核酸信息统计模块的设计主要为了对有问题的检测试管及时追查以及提供学生具体的检测信息，所以只需根据窗口号及学号进行查询。效果如下图所示：

用户管理界面　　未做核酸查询界面　　核酸信息统计界面

核酸通知发布界面：

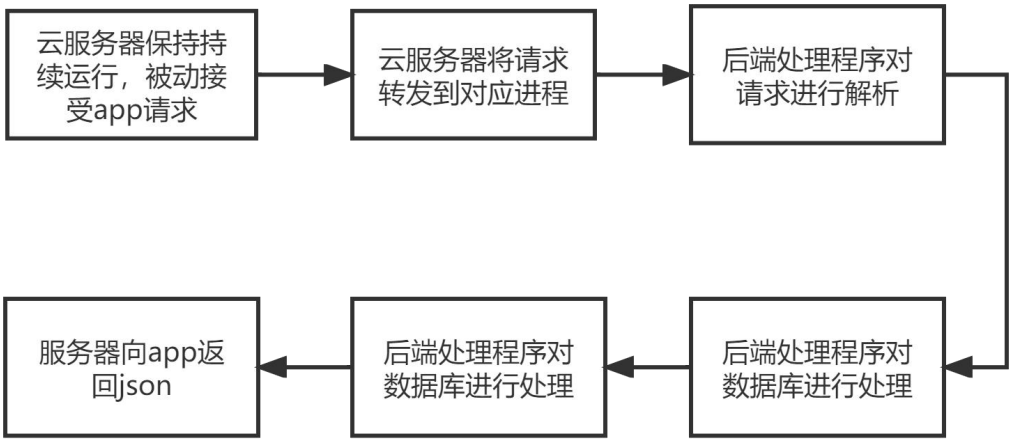
日期 EditText 的默认值设置为当天的后一天，因为一般都是提前一天发布通知。效果如下图所示：

核酸通知发布界面

3.2 后端设计

3.2.1 后端总体设计

后端整体处理过程如下



3.2.2 线程池设计

在网络服务器中，有着多个进程，对于我们需要使用的每一个进程，给予其一个特定的进程号。对于我们的 app，我们在后端中的进程号设置为 8107，对于 app 发送来的请求，我们使用域名 api.gcgzs.club 来进行请求，我们将 api.gcgzs.club 进行 DNS 解析到服务器 110.40.174.65 的默认 http 端口（80），在服务器中，我们使用 nginx 进行线程池分发，将 api.gcgzs.club 转发到 8107 端口，具体配置文件（nginx.conf）如下

```
server {
listen 80;
charset utf-8;
server_name api.gcgzs.club; #你的域名
location /
{
proxy_set_header Host $host;
proxy_set_header X-Real-Ip $remote_addr;
proxy_set_header X-Forwarded-For $remote_addr;
proxy_pass http://110.40.174.65:8107;
}
}
```

Nginx.conf 配置文件

对于网络服务器，由于需要持续的运行后端程序来对前端请求进行响应，我们需

要使用 docker 容器来维持后端程序的持续运转，在测试过程中，我们可以使用 tmux 或 screen 来维持项目的持续运转

```
^Cydhlw@VM-4-15-ubuntu:~/ydhlw$ python3 manage.py runserver 0:8107
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly un
Run 'python manage.py migrate' to apply them.
March 06, 2023 - 06:27:21
Django version 3.2.16, using settings 'ydhlw.settings'
Starting development server at http://0:8107/
Quit the server with CONTROL-C.
```

将后端项目运行在 8107 端口上

3.2.3 接口设计

对于前端的不同请求，将其转发到指定进程后，对于不同的请求，我们分别进行解析

我们的接口规则选用了 restful 规则，格式为:

域名/接口名/数据

例如：api.gcgzs.club/load/?num=2014020228&pd=123456

对于每一个前端请求，后端都会进行对应的响应，并且返回一个 json

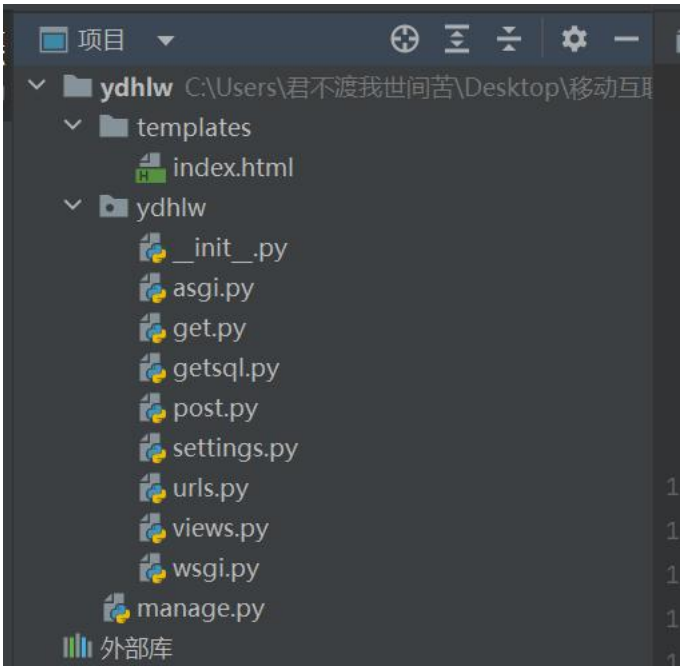
下图为收到请求时的后端响应

```
[06/Mar/2023 05:13:25] "GET /yes/?date=2022-12-23 HTTP/1.1" 500 66753
[06/Mar/2023 06:23:43] "GET /load/?num=2014020228&pd=123456 HTTP/1.1" 200 53
[06/Mar/2023 06:23:43] "GET /search_get/?q=2014020228 HTTP/1.1" 200 182
[06/Mar/2023 06:23:43] "GET /get_report/?day=2023-03-06 HTTP/1.1" 200 37
[06/Mar/2023 06:23:45] "GET /no/?data1=2023-03-06&data2=&data3= HTTP/1.1" 200 1159
[06/Mar/2023 06:23:43] "GET /search_get/?q=2014020228 HTTP/1.1" 200 182
[06/Mar/2023 06:23:43] "GET /load/?num=2014020228&pd=123456 HTTP/1.1" 200 53
```

后端响应

3.2.4 后端程序设计

后端程序选用了基于 Python3.x 的 Django 框架，Django 框架是一个轻量级，热重载速度快的后端框架，也是现有的主流开发框架之一，我们将后端不同的请求处理分别用不同的 Django APP 来处理，整体的后端内容如下



后端整体结构

3.2.5 数据库设计

对应后端数据库部分，我们使用了 Mysql 8.0 版本云数据库，通过 pymysql 接口进行访问，数据库整体设计如下

表 user_info

字段名	字段	限制条件
学号	user_num	数字，主键
密码	password	字符串
姓名	name	字符串
班级	class	字符串
电话	telephone	数字，长度
学院	college	字符串
类型	title	数字，长度

表 user_record

字段名	字段	限制条件