# 小程序后端开发过程难点与解决方法

## 服药提醒功能实现

* **模型部分**

患者patient模型保存服药提醒内容为JSONField类型，增加uid属性

* **视图函数**

**（1）发送服药提醒**

接收参数：接收监护人的openid，setTime，medicineName三个参数。

处理逻辑：通过openid查找到监护人，通过监护人绑定的患者的特殊id查找到指定患者，获取到患者uid，使用该uid通过Wxpusher公众号向指定患者发送服药提醒信息。

（2）修改患者uid

## 聊天功能实现

* **模型部分**

监护人guardian模型保存聊天内容为JSONField类型，增加flag属性

医生doctor模型中的监护人列表中的每一项增加一条flag属性

* **视图函数**

（1）**保存聊天记录**

接收参数： openid，opposite\_id，chatlist，identity四个参数。

处理逻辑：

如果identity为guardian，那么该openid是监护人的openid，通过openid查找到监护人，获取到监护人的特殊id为guardian\_id，并将保存聊天记录，然后通过opposite\_id查找到医生，获取医生的监护人列表，通过遍历列表中每一项的监护人特殊id是否与之前的guardian\_id匹配，找到指定监护人，将其flag置为true。

如果identity为doctor，那么该openid是医生的openid，通过openid查找到医生，判断医生是否存在。然后通过opposite\_id查找到监护人，将聊天记录保存到监护人的对应属性下。

1. **返回聊天记录**

接收参数： openid，opposite\_id，identity三个参数。

处理逻辑：

如果identity为guardian，那么该openid是监护人的openid，通过openid查找到监护人，返回其对应属性中保存的聊天记录。

如果identity为doctor，那么该openid是医生的openid，通过openid查找到医生，判断医生是否存在。然后通过opposite\_id查找到监护人，返回监护人对应属性中保存的聊天记录。

1. 将监护人的flag属性置为false
2. 将医生的监护人列表中的指定监护人对应的flag属性置为false

## 图片上传功能实现

* **模型部分**

user模型增加imageField类型的img属性，进行图片的保存。

* **视图函数**

接收参数：接收一个值为对应图片文件的image参数。

处理逻辑：从前端接收到image字段，对应的值为图片文件，然后将图片文件保存到对应user的img属性中，保存成功后返回服务器上该图片的访问路径到前端。