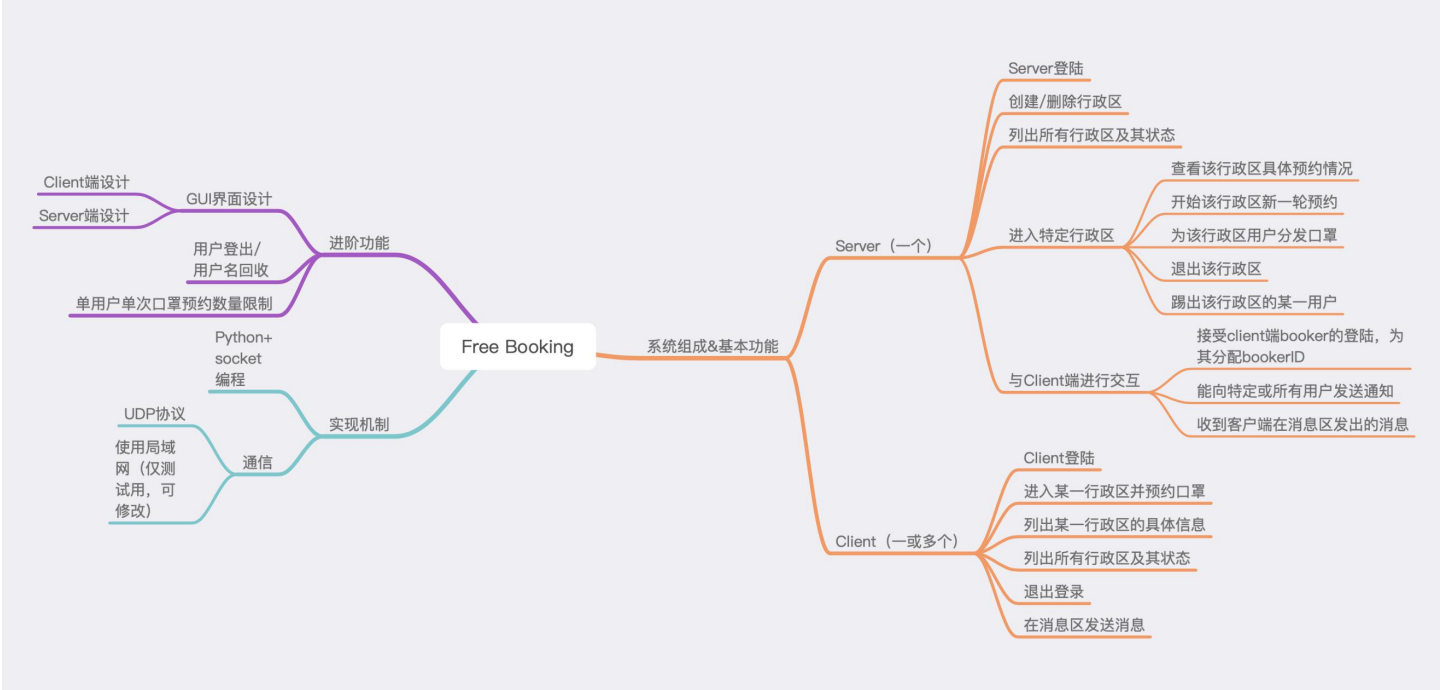


数据通信与计算机网络 PJ 设计报告

一. 系统结构

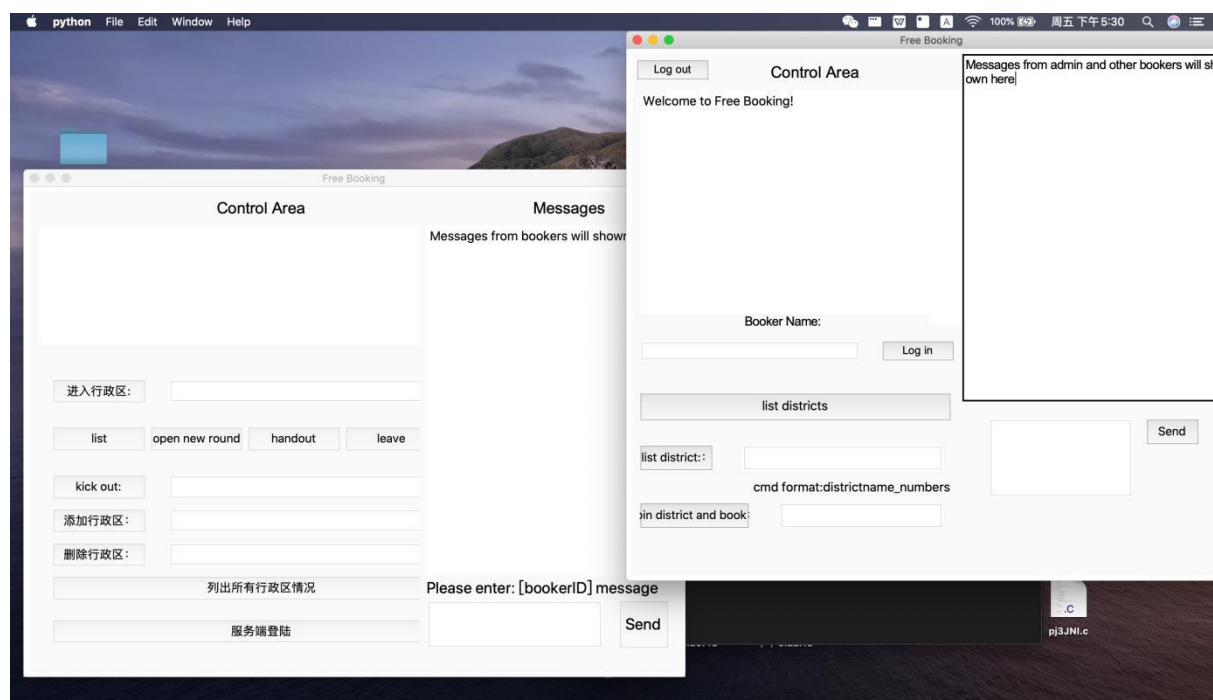


二. 协议设计

按照报告指导要求，要求应用场景可以用于相对分组丢失率非常小的网络环境，同时可以在internet 上应用；考虑到大多数情况下分组很少丢失，采用基于 UDP 的实现策略，并且用python+socket 编程实现。

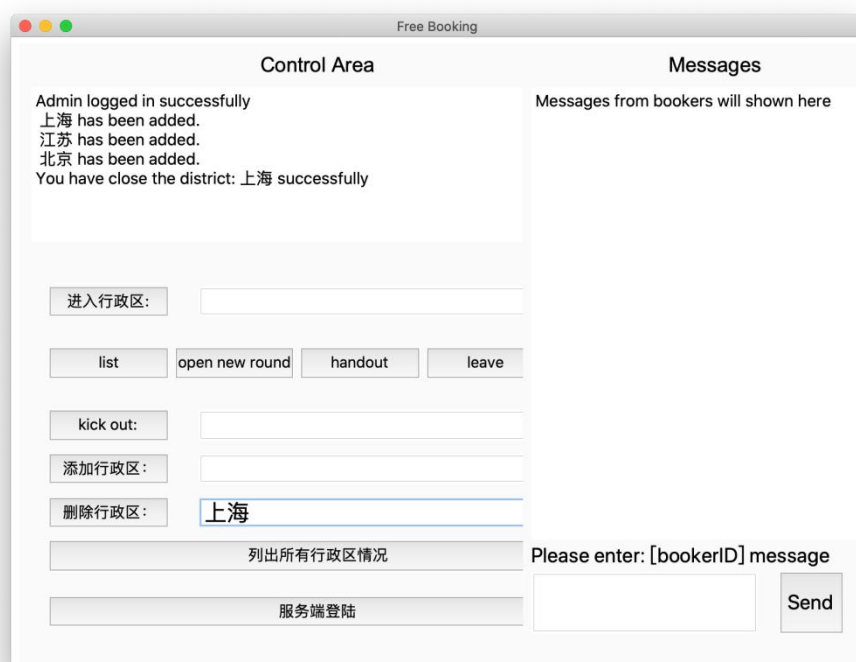
三. 功能介绍&用户手册

用户界面效果：客户+服务端



SERVER 端使用手册:

- 1.请点击左下角服务端登陆按钮，激活服务端；
- 2.在按钮“添加行政区”后的输入框内输入行政区名称并且点击按钮，可以添加行政区；
- 3.在按钮“删除行政区”后的输入框内输入行政区名称并且点击按钮，可以添删除政区；



- 4.点击按钮“列出所有行政区情况”，可以查看所有行政区的状态

The screenshot shows a window titled "Free Booking" with a "Control Area" section. The text in the Control Area reads: "please leave it first. You have left districts successfully You have enter the district: 上海 successfully 当前行政区已在预约中 All districts are following: 上海 状态: 预约中 北京 状态: 非预约中 江苏 状态: 非预约中". Below the text is a button labeled "进入行政区:" followed by a text input field containing "上海". At the bottom are four buttons: "list", "open new round", "handout", and "leave".

5.在按钮“进入行政区”后的输入框内输入行政区名称并且点击按钮，可以进入该行政区（功能 6--10 需要在进入某一行政区后完成）；

6.点击 list 按钮，可以列出该行政区的具体预约情况；

7.点击 open new round 按钮，可以开始这个行政区的新一轮预约；

8.点击 handout 按钮，可以向该行政区中预约口罩的用户分发口罩，并且通知用户

The screenshot shows a window titled "Free Booking" with two main sections: "Control Area" and "Messages". The "Control Area" section contains the following text: "You have left districts successfully You have enter the district: 上海 successfully 当前行政区不在预约中 你已经开启该行政区新一轮预约 您已成功向该地区预约者分发口罩 You have left districts successfully". Below the text are several buttons and input fields: "进入行政区:" followed by a text input field containing "上海", a row of buttons "list", "open new round", "handout", and "leave", a "kick out:" button followed by a text input field, a "添加行政区:" button followed by a text input field, a "删除行政区:" button followed by a text input field, a button "列出所有行政区情况", and a button "服务端登陆". The "Messages" section contains the text "s from bookers will shown here". At the bottom right, there is a text input field with the placeholder "Please enter: [bookerID] message" and a "Send" button.

Free Booking

Log out

Control Area

Welcome to Free Booking!
You have logged in successfully.
All districts are as following:
江苏 状态: 非预约中
北京 状态: 非预约中
上海 状态: 预约中
All districts have been shown
You have joined the district and booked successfully. May you have a good time!
上海预约总人数: 1 预约口罩总数:30
恭喜您, 您预约的口罩已经成功发货

Messages from admin and other bookers will shown here
attention:booker:黛安has booked 40facemasks in上海

Booker Name:

汤姆

Log in

list districts

list district: 上海

cmd format:districtname_numbers

join district and book:

Send

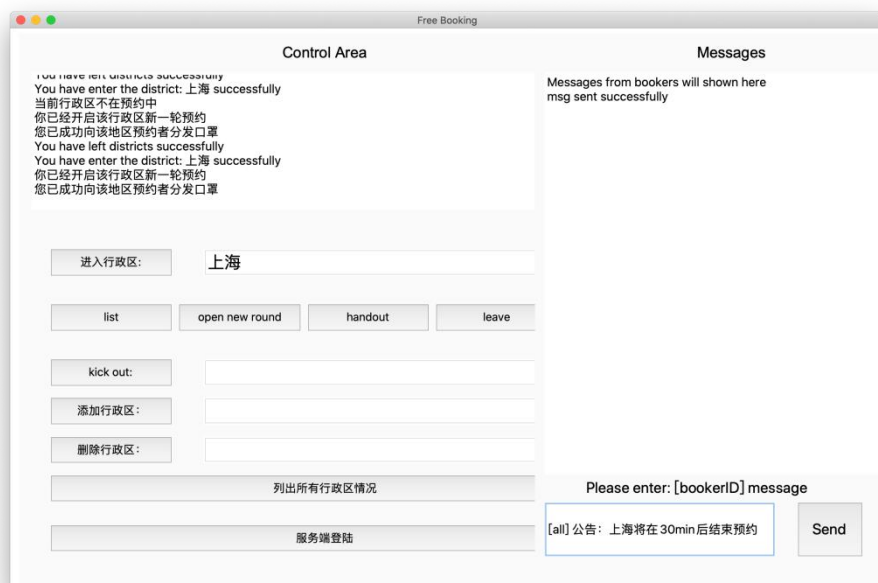
9.在按钮“kick out”后的输入框内输入 **booker** 名称并且点击按钮, 可以踢出该 **booker**;

10.点击 **leave** 按钮, 退出该行政区。

11.右侧 **Messages** 区域, 会显示 **booker** 发送的消息;

12.右下角输入信息并且点击 **send**, 可以向用户发送信息。注意消息格式:

[all] 消息内容 (如果发给所有用户) **/[该用户 bookername]** 消息内容 (如果发给特定用户)



Client 端使用手册:

- 1.输入 bookername 并且点击 log in 按钮后, 可以成功登陆;
- 2.点击按钮“list districts”, 可以查看所有行政区的状态
- 3.在按钮“list:”后的输入框内输入某一行政区名称并且点击按钮, 可以查看该行政区具体预约情况
- 4.在按钮“join district and book:”后的输入框内输入并且点击按钮,可以进入行政区并且预约口罩, 并且会通知同一行政区的其他用户

消息格式: 行政区名_口罩数量

注意: 一位用户一次最多可以预定口罩数量为: 50

18307130089 吴嘉琪

Free Booking

Log out

Control Area

Welcome to Free Booking!
You have logged in successfully.
You have joined the district and booked successfully. May you have a good time!
sorry, the maximum number you can book once is 50

Booker Name:

黛安

Log in

list districts

list district:.

cmd format:districtname_numbers

join district and book:.

上海_60

5.在右下角输入信息并且点击 **send**，可以发送信息给管理员和同一行政区内的 **booker**

Free Booking

Log out

Control Area

Welcome to Free Booking!
You have logged in successfully.
You have joined the district and booked successfully. May you have a good time!

Booker Name:

汤姆

Log in

list districts

list district:.

cmd format:districtname_numbers

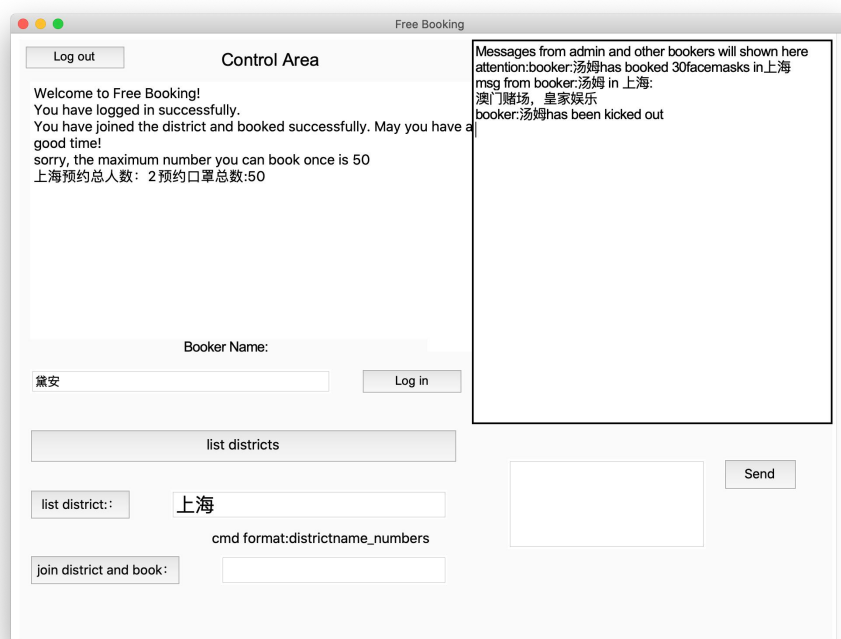
join district and book:.

澳门赌场, 皇家娱乐

Send

Messages from admin and other bookers will shown here
msg sent successfully
msg from booker:汤姆 in 上海:
澳门赌场, 皇家娱乐

6.如果您存在上图这样发送广告、骚扰信息等不符合规范的行为，服务端管理员将会把您踢出行政区，并且通知您原来所在行政区的所有用户。



7.点击左上角 log out 按钮，可以退出登陆，并且您的 booker 名称将被回收

四. 关键代码说明

server 端:

```
booker = {}
bookerids = {}
bkerstat = {}
bookerips = {}
districts = []
status={}
booknum={}
bkernum={}
```

使用这些数据结构来保存信息。booker 保存 (bookername,所在行政区) 二元组;

bookerids 保存唯一的 bookerid; bookerips 保存 (booker IP 地址, bookername) 二元组; districts

保存所有的行政区, status 保存每个行政区是否在预约中; booknum 和 bkernum 分别保存每个行政区的预约口罩总数和预约总人数。

AdLogin()和 startListen () 函数, 开启线程调用主体函数: MainContact, 用来接受 client 端的命令,

解析，并且发出响应。

可以看到，**MainContact** 函数，本质上是通过 **Client** 端发送的字符串格式的服务请求，通过分割、解析头部，得到服务类型以及 **booker** 的具体信息；随后便可以通过访问上述数据结构，进行具体操作，最后通过 **bookerips** 来得到该 **booker** 的 **ip** 地址，发送回复消息。


```

def MainContact(cmd, ipaddr):
    result = parsedata(cmd)
    #登录
    if result == 0:
        if cmd in bookerips.values():
            reString = "Sorry,bookername have been used.\n"
            s.sendto(reString.encode('utf-8'), ipaddr)
        else:
            reString = "You have logged in successfully.\n"
            s.sendto(reString.encode('utf-8'), ipaddr)
            bookerips[ipaddr] = cmd
            #刚刚登陆, 暂时无预约
            bkerstat[cmd] = None
            #保证生成唯一的bookerid
            bookerid = random.randint(0, 4096)
            while bookerid in bookerids.values():
                bookerid = random.randint(0, 4096)
            bookerids[cmd] = bookerid

    elif result[2][0] == "/":
        bookerCommand = parsecmd(result[2])
        # list all districts
        if bookerCommand[0] == "/listdis":
            if bookerCommand[2] != "": # check arguments' amount
                reString = "cmd wrong format\n"
                s.sendto(reString.encode('utf-8'), ipaddr)
                return
            else:
                reString = "All districts are as following:\n"
                s.sendto(reString.encode('utf-8'), ipaddr)
                print("list all dis\n")
                for i in districts:
                    s.sendto(i.encode('utf-8'), ipaddr)
                    s1=" "
                    s.sendto(s1.encode('utf-8'), ipaddr)
                    s2="状态: "
                    s.sendto(s2.encode('utf-8'), ipaddr)
                    String = status[i]
                    s.sendto(String.encode('utf-8'), ipaddr)
                    s3="\n"
                    s.sendto(s3.encode('utf-8'), ipaddr)
                reString = "All districts have been shown\n"
                s.sendto(reString.encode('utf-8'), ipaddr)
                return
            #用户加入行政区并预约口罩
        elif bookerCommand[0] == "/join":
            print("see join\n")
            args = bookerCommand[2].partition("_")
            print(args)
            if args[2] == "":
                # print("error in")
                reString = "cmd wrong format\n"
                s.sendto(reString.encode('utf-8'), ipaddr)
                return
            else:
                if districts.count(args[0]) == 0:
                    print("dis error sent\n")

```

Client 端:

client 端的实现比较简单，主要功能都是通过特定的函数，生成命令字符串，再向 server 端发送。

接收到 server 端的信息后，用该函数进行解析：

```
#处理来自server的信息
def analyse(data):
    String =data
    print("analyze:" + String)
    if data.startswith("notice from administrator: "):
        noticeMsg.insert(END,data)
        noticeMsg.insert(END,"\n")

    elif data.startswith("msg from booker:"):
        noticeMsg.insert(END,data)

    elif data.startswith("booker:"):
        noticeMsg.insert(END, data)

    elif data.startswith("attention:"):
        noticeMsg.insert(END, data)

    else:
        CtrlMsg.insert(END,data)

def recv():
    while True:
        data = s.recvfrom(1024)
        analyse(data[0].decode('utf-8'))
```

从而实现不同的类型的消息的发送。

GUI 设计:

因为是第一次尝试 python 编程，所以上网查询资料，采用了比较简单的 tkinter 库，来实现 GUI 界面设计，学习使用了 text, label, button 还有 scrollbar 等多种元素，主要以简洁明了为主要思路。

关于更多代码实现的细节，请参考 py 源代码中的注释部分。

