NAJI----定格情绪瞬间

项目开发人员

姚東序（学号：10185101174 软件工程学院）

引言摘要

1. 项目背景 ： GUI程序大作业
2. 项目基础 ： 基于python语言 、tkinter库的启动端GUI
3. 项目介绍

项目名称： NAJI----定格情绪瞬间

项目简介： 将用户的时刻情绪记录在数据库中，通过六个标签定格住用户此刻的心情，通过文本框编辑记录让用户心情起伏的生活小事情！

项目操作： 用户只需要注册登录之后，即可时时刻刻写下当前情绪爆炸的瞬间，也能够时刻查询往日的情绪瞬间

项目特色： 由于情绪记录涉及到用户较为隐私的部分，故做好隐私安全防护是十分必要的。在这个项目中，特定引入了手机号注册的方式。通过阿里云的短信服务，实现注册的时候验证手机号码有效与否的操作。

1. 功能介绍
2. 密码登录：采用了账号密码登录，防止信息丢失和隐私被窃取。
3. 短信注册：采用了短信验证的方式，有效的验证注册号码的有效性。
4. 情绪编辑：
   1. 设置了一个计时器，记录编写情绪的当下时间；
   2. 设置了“开心”、“愤怒”、“悲伤”、“恐惧”、“焦虑”、“感动”的六个情绪标签

让用户得以迅速定位此刻的情绪。

c） 设置了编辑文本框，用户可以在此写下当下发生的点点滴滴的事情，当做对生活的记录。文本框文字提供了多种颜色，用户可以自行选择

④ 查询情绪历史：用户编辑好的情绪记录会存到txt文件下，当用户点击“查询”时候，可以随时看到以往的情绪状态。

1. 核心算法技术
2. 使用阿里云短信服务

通过在阿里云端申请短信服务，把接口接入到该项目中，即可通过真实手机号码注册账号。

代码如下：

#进行短信验证开始

req1 = CommonRequest()

req1.set\_accept\_format('json')

req1.set\_domain('dysmsapi.aliyuncs.com')

req1.set\_method('POST')

req1.set\_protocol\_type('https') # https | http

req1.set\_version('2017-05-25')

req1.set\_action\_name('SendSms')

req1.add\_query\_param('RegionId', "cn-hangzhou")

req1.add\_query\_param('PhoneNumbers', phone\_number)

req1.add\_query\_param('SignName', "NAJI纳己")

req1.add\_query\_param('TemplateCode', "SMS\_190727303")

req1.add\_query\_param('TemplateParam', "{\"code\":\""+cha\_num+"\"}")

#进行短信验证结束

req1.add\_query\_param('RegionId', "cn-hangzhou")

response = client.do\_action(req1)

1. 写入写出文件

将用户的账号密码包括情绪条目记录到本地文件之中，用写入文件和写出文件操作完成该功能

代码如下：

//写出文件

def saveAccounts(d):

print(d)

with open("test.txt","w") as f:

for name,item in d.items():

username = item['username']

password = item['password']

f.write(f'{name}\t{username}\t{password}\n')

//写入文件

def readAccouts():

d = {}

with open("test.txt") as f:

for line in f:

record = line.split()

tmp\_dict = {}

tmp\_dict["username"] = record[1]

tmp\_dict["password"] = record[2]

if(len(record) == 3):

tmp\_dict["text"] = []

# print('yes')

else:

# tmp\_dict["text"] = json.loads(record[3])

new\_str=""

new\_record = line.split()[3:]

for a in new\_record:

new\_str += a + ' '

print(new\_str)

print(type(new\_str))

tmp\_dict["text"] = json.loads(new\_str)

d[record[0]] = tmp\_dict

print(d)

return d

1. 注册登录页面的各种错误排查

注册登录涉及到很多不规范的输入，需要一个个的排查，进行相应的错误输出提示操作。

代码如下：

def sign\_to\_database(cha\_num):

n\_pwd = new\_pwd.get()

nr\_pwd = new\_CHA.get()

n\_name = new\_name.get()

cha\_num = cha\_num

# 读取本地用户信息

global exists\_user\_info

exists\_user\_info = readAccouts()

if(len(n\_name) != 11):

tk.messagebox.showerror(title='error', message='手机号错误，请重试！')

pass

else:

# 判断是两处密码是否一致

if (cha\_num != nr\_pwd):

tk.messagebox.showerror(title='error', message='验证码错误，请重试！')

pass

else:

# 判断该用户是否存在数据库

if (n\_name in exists\_user\_info):

tk.messagebox.showerror(title='error', message='该用户已存在！')

else:

# 更新写入本地数据

tmp = {}

tmp["username"] = n\_name

tmp["password"] = n\_pwd

exists\_user\_info[n\_name] = tmp

# exists\_user\_info[n\_name].username = n\_name

# exists\_user\_info[n\_name].password = n\_pwd

print(exists\_user\_info)

# with open("user\_info.pickle", 'wb') as f:

# pickle.dump(exists\_user\_info, f)

saveAccounts(exists\_user\_info)

# 注册成功

tk.messagebox.showinfo(title='welocme', message='注册成功！')

# 关闭窗口

window\_signup.destroy()

1. 查询情绪记录操作

用户点击查询时候，需要将用户的数据取出来，并且排版好布局到页面上。

代码如下：

def search():

window\_signup = tk.Toplevel()

window\_signup.title('welocme to sign up')

window\_signup.geometry('600x500')

label = tk.Label(window\_signup, text="情绪历史", font=LARGE\_FONT)

label.pack(pady=10, padx=10)

global exists\_user\_info

exists\_user\_info = readAccouts()

print('-=-=-=-=-=-=-=-=-=-=')

# print(exists\_user\_info)

print(len(exists\_user\_info[user\_name]["text"]))

print('-=-=-=-=-=-=-=-=-=-=')

length = len(exists\_user\_info[user\_name]["text"])

print(type(length))

# print(len(exists\_user\_info["text"]))

for i in range(length):

# print(i)

label=tk.Label(window\_signup, text=exists\_user\_info[user\_name]["text"][i][0], font=LARGE\_FONT)

label.place(x=40, y=70+80\*i, anchor='nw')

label=tk.Label(window\_signup, text=exists\_user\_info[user\_name]["text"][i][2], font=LARGE\_FONT ,fg="green")

label.place(x=400, y=70+80\*i, anchor='nw')

label=tk.Label(window\_signup, text=exists\_user\_info[user\_name]["text"][i][1], font=LARGE\_FONT ,fg="skyblue")

label.place(x=60, y=100+80\*i, anchor='nw')

1. 亮点
2. 接入了短信服务接口，让情绪记录软件更有可靠性。
3. 注册登录都进行了输入错误排查，完成了基本的准确性检测。
4. 运用了丰富的控件，例如文本框，单选框，按钮，计时器，下拉菜单等等……
5. 运用了丰富的情绪标签，让用户可以精确选择自己当前的情绪。
6. 用户可以随时记录情绪，并且无数量限制。
7. 不足处与将来可能工作
8. 页面可视化：目前页面还单纯只有文字，后续需要添上一些好看的表情图片等，让用户真正有情绪记录的感觉。
9. 智能化有待加强： 只是单纯的记录，没有对数据进行预测，分析，可视化成图表等功能，后续需要完善加强。

六、运行截图







