一、需求（作业要求）

需求来自于作业，但我进行了一些调整。整体上并未改变作业的具体要求。

1、用户注册

- 用户注册信息写入文件中存储。存储格式为：账号:加密后的密码:加密使用的salt
- 注册时输入的用户名不得为空，否则报错并结束程序运行。
- 用户需输入两次相同密码才能完成成功注册。
- 用户名不得重复。
- 用户密码加盐后加密存储。

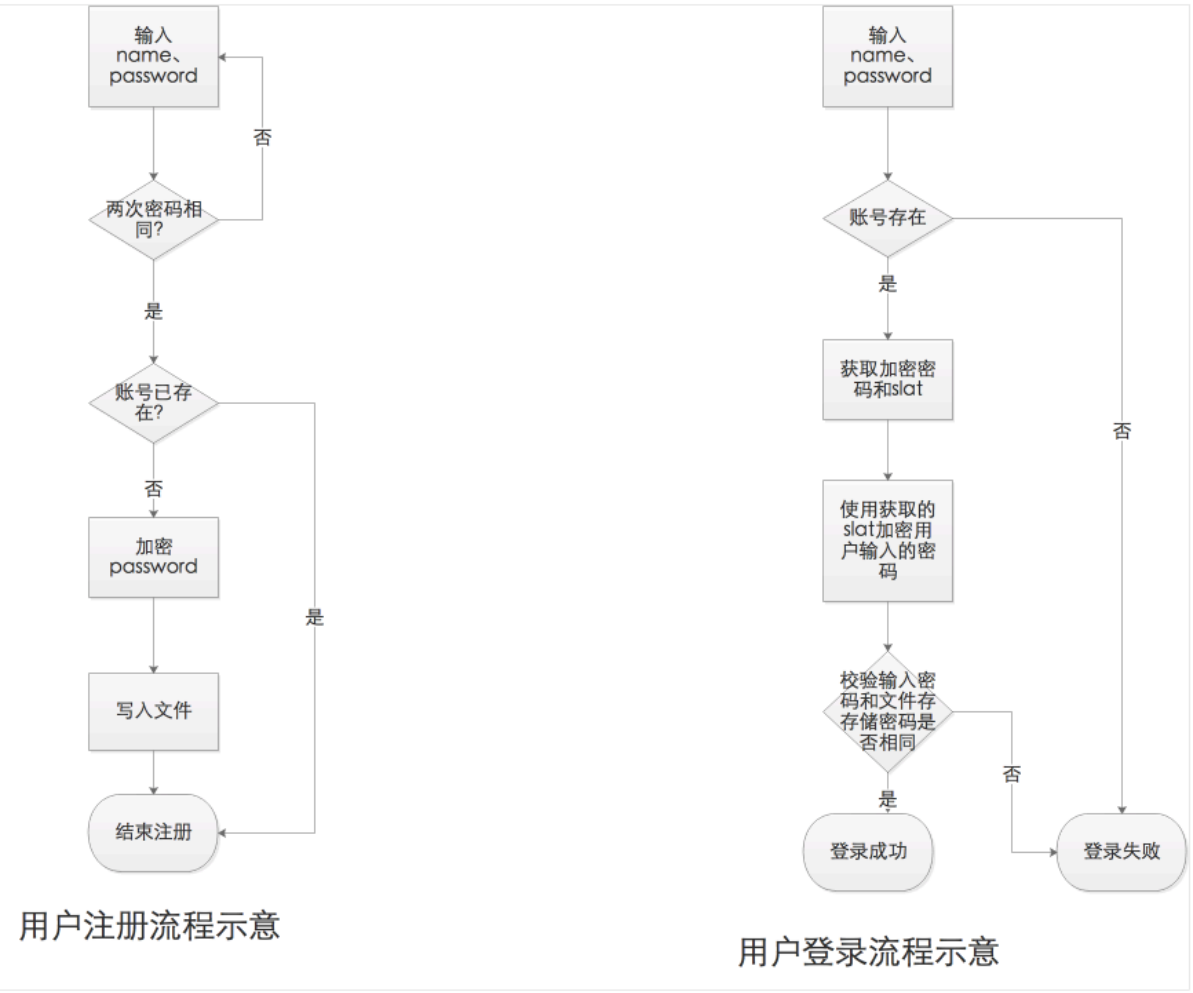
2、用户登录

- 用户输入的账号、密码均不得为空，否则报错并结束程序运行。
- 用户名、密码正确方能成功登录。
- 不得提醒用户用户名是否存在，是否仅是密码错误。

二、技术分析

- 涉及到文件读写操作。需要注意：写操作是内容追加，而不是覆盖。
- 判断用户名是否存在时，需要遍历所有用户名才能确定结果。
- 加密存储密码信息，密文不可逆。因此，用户登录时需要对密码进行加密，比对加密密文是否相同，密文相同则认为密码正确。
- 登录失败的提醒一律设为：您的用户名或密码错误！

三、流程示意



四、代码实现

1、用户注册

```
#encoding=utf-8

import random
import  hashlib
import time

# 作者：杨公旺
# 日期：2017-12-18
# 功能：用户注册

# 思路：
#   1、接收用户输入的账号、密码等信息，写入到文件中保存。其中密码需要输入两次，且两次密码相同
#   2、不允许账号重复
```

```
#  3、密码加密存储
#  4、用户输入字母q则退出程序

# 判断用户名是否存在的函数
# 函数接收两个参数：密码存储的文件名和用户名
# 如果用户名已经存在，则返回值是1。该返回值用于判断文件名是否存在
def is_user_password(filename,user_name):
    _count = 0
    with open(filename,'r') as f:
        for line in f.readlines():
            temp_strings = line.strip().split(':')[0]
            if temp_strings == user_name:
                return 1

# 用户名、账号输入函数
# 函数不接受参数
# 正常执行时，函数返回值是用户输入的账号和密码
def input_user_name_passwd():
    input_user_name = raw_input("请输入用户名，仅输入单个小写字母q则结束程序：")
    if input_user_name == 'q' or input_user_name == '':
        print "您输入了退出字符q或者未输入任何字符，程序终止！"
        exit()
    while True:
        input_user_password1 = raw_input("请输入密码，仅输入单个小写字母q则结束程
序：")
        if input_user_password1 == 'q' or input_user_password1 == '':
            print "您输入了退出字符q或者未输入任何字符，程序终止！"
            exit()
        input_user_password2 = raw_input("请再次输入密码，仅输入单个小写字母q则结束程
序：")
        if input_user_password2 == 'q' or input_user_password2 == '':
            print "您输入了退出字符q或者未输入任何字符，程序终止！"
            exit()
        if input_user_password1 == input_user_password2:
            break
        else:
            print "两次输入密码不一致，请重新输入！"
    return (input_user_name,input_user_password1)

# 加盐函数
# 接收一个参数，参数有默认值
# 函数返回值是一个字符串
# 默认salt长度是6个字符，可以在函数调用时指定slat的长度
# salt的内容来自于一个长字符串，可以根据需要灵活调整该字符串。为增加字符串的随机
性，字符串包含当前的时间信息。
```

```
# 每次随机从字符串中取出一个字符作为salt字符串的一部分
def create_salt(length = 6):
    my_time = time_start = time.time()
    _salt = ''
    chars =
'AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz0123456789' +
str(my_time)
    len_chars = len(chars) - 1
    for i in xrange(length):
        # 每次从chars中随机取一位
        _salt += chars[random.randint(0, len_chars)]
    return _salt


# 加密密码函数
# 接收两个参数，无参数默认值
# 返回值是加密后的密码密文
# 被加密的字符串是用户输入的密码和加盐函数随机生成的slat
def jiami_password(user_password,salt):
    return hashlib.sha512(user_password + salt).hexdigest()


# 账号、密码写入文件函数
# 接收三个参数用户账号、密码密文、salt
def file_write(filename,pwd,salt):
    with open(filename,'aw') as f:
        f.write(user_name+':'+pwd+':'+salt+'\n')


###################### 以上是函数定义，以下是函数调用 ######################
filename = 'user.txt'

# 调用用户输入函数
user_name_password = input_user_name_passwd()

# 将用户输入函数的返回值分别提取出来：账号、密码
user_name = user_name_password[0]
user_password = user_name_password[1]

# 调用判断用户名是否存在的函数
if is_user_password(filename,user_name) == 1:
    print "用户名已经存在！"
    exit()

# 调用创建salt的函数，创建长度为9位字符的salt
salt = create_salt(9)
```

```
# 调用加密函数，将加密后的密码赋值给pwd
pwd = jiami_password(user_password,salt)

# 调用文件写入函数
# 需要传入文件名、账号和密码三个参数
file_write(filename,pwd,salt)
```

2、用户登录

```
#encoding=utf-8

import  hashlib

# 作者：杨公旺
# 日期：2017-12-18
# 功能：用户登录

# 思路：
#   1、接收用户输入的账号、密码等信息
#   2、判断账号是否存在，如果不存在即返回错误
#   3、对密码信息进行加密后比对，如果一致，则登录成功。因密码是不可逆加密，所以比对加密后的密文内容


# 判断用户名是否存在的函数
# 需要传入两个参数：文件名和账号名
# 将每行的第一个元素（用户名）取出来，放到一个list中，判断用户输入的账号是否在这个list中
def is_user_password(filename,user_name):
    with open(filename,'r') as f:
        user_list = []
        for line in f.readlines():
            temp_strings = line.strip().split(':')[0]
            user_list.append(temp_strings)
        if user_name in user_list:
            return line.strip().split(':')[2]
        else:
            return 1

# 取用户密码和salt的函数
# 需要传入两个参数：文件名和用户名
def get_password(filename,user_name):
    with open(filename,'r') as f:
        for line in f.readlines():
            temp_strings = line.strip().split(':')[0]
            if temp_strings == user_name:
```

```
            return (line.strip().split(':')[1],line.strip().split(':')[2])

# 用户名登录的输入函数
def input_user_name_passwd():
    input_user_name = raw_input("请输入用户名，仅输入单个小写字母q则结束程序：")
    if input_user_name == 'q' or input_user_name == '':
        print "您输入了退出字符q或者未输入任何字符，程序终止！"
        exit()
    input_user_password = raw_input("请输入密码，仅输入单个小写字母q则结束程序：")
    if input_user_password == 'q' or input_user_password == '':
        print "您输入了退出字符q或者未输入任何字符，程序终止！"
        exit()
    return (input_user_name,input_user_password)

# 加密函数
def jiami_password(user_password,salt):
    return hashlib.sha512(user_password + salt).hexdigest()



###################### 以上是函数定义，以下是函数调用 ######################

filename = 'user.txt'

# 调用用户输入函数，并获得用户输入的账号和密码
user_name_password = input_user_name_passwd()
user_name = user_name_password[0]
user_password = user_name_password[1]

# 判断用户输入的账号是否存在
if is_user_password(filename,user_name) == 1:
    print "您输入的用户名或密码错误，请核对后重新登录！"
    exit()

# 取出用户名对应的salt字符串，对用户输入的密码进行加密
salt = get_password(filename,user_name)[1]
pwd = get_password(filename,user_name)[0]
tmp_pwd = jiami_password(user_password,salt)

# 比对用户输入密码的密文与文件中存储的密文是否相同
if pwd == tmp_pwd:
    print "登录成功！"
else:
    print "您输入的用户名或密码错误，请核对后重新登录！"
    exit()
```