

面向对象程序设计课程大程

## *QsMail* 邮件系统设计报告

指导老师：陈奇

组 长：赵冰骞 3090103420

组 员：谭洪林 3090101605

王道远 3090104579

何宁栩 3090000227



## 目录

系统的功能设计 .....	3
类体系与主要数据结构.....	3
关键的技术问题研究 .....	5
开发过程中的问题和解决.....	9
程序存在的问题与改进展望 .....	11
项目合作总结.....	12
系统开发日志.....	12
程序使用说明.....	14

## 系统的功能设计

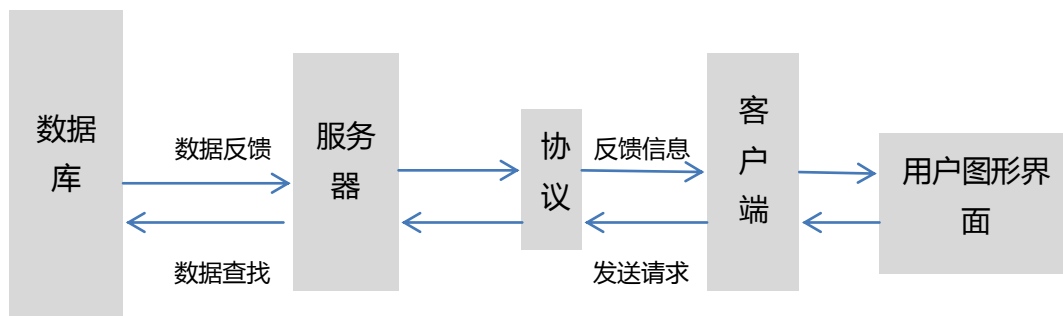
我们的邮件系统功能分两部分实现，一部分是客户端，一部分是服务器。总体实现的功能包括最基本的注册、登陆、收发邮件、收发附件等功能，随后添加了修改密码、删除邮件、多收件人等功能。具体功能如下表所示：

基本的用户注册和管理	<ul style="list-style-type: none"> <li>注册时需给出需要的唯一用户名和密码</li> <li>登陆后可以修改密码</li> </ul>
基本的邮件收发和管理功能	<ul style="list-style-type: none"> <li>邮件收发仅限本系统内联系人</li> <li>邮件收发支持多收件人</li> <li>邮件分别放入收件箱、发件箱、垃圾箱</li> <li>删除邮件将邮件转移到垃圾箱</li> <li>彻底删除将邮件本人邮件中删掉，不影响其他收件人或发件人</li> </ul>
附件的收发	<ul style="list-style-type: none"> <li>每封邮件可以添加一个任意文件格式的附件（大小&lt;10KB）</li> </ul>

其中服务器实现数据存放，客户端则实现了用户对本系统各项功能的使用。

## 类体系与主要数据结构

项目的总体架构如下图：



### ➤ 类体系

前面已经提到，我们将这个系统分为服务器端和客户端两部分进行分别实现。而这两部分都建立在网络传输的基础上，所以类体系和主要数据结构也分这三部分进行设计。

### 1. 网络传输：

**MySocket 类：**实现对 winsock2 库的调用，封装了发送字符串和接收字符串两个成员函数。

### 2. 服务器：

**Server 类：**是 MySocket 类的派生类，包含整个服务器的一个类，类里面的函数能完成所有客户端的功能。各成员函数实现对各项请求的处理与回应，功能与客户端差不多相反，后面以客户端为例说明。

### 3. 客户端：

**Client 类：**是 MySocket 类的派生类，包含整个客户端的一个类，类里面的函数能完成所有客户端的功能。客户端的功能包括：注册，登陆，收发邮件，发附件，下载附件，删除邮件，更改用户信息。成员函数有：log(), reg(), sendMail(), check(), sendAttach(), down(), dele(), update()。

这些函数的主要任务就是把用户输入的指令转换成一种标准的格式，让服务器也能读懂，然后再把从服务器反馈过来的按格式编辑好的字符串转换成用户能读懂的信息。因此客户端的任务本质上就是处理字符串。下面以 log()为例说明。

用户输入用户名 aaa，密码 bbb。然后程序生成字符串“LOGIN aaa bbb”，发送到服务器。服务器返回邮件列表信息，接着 log()把这些信息切开，再按类 mail()保存下来，输出给用户。

其他函数的内容基本上相似，除了收发附件。由于附件的上传跟发邮件不是同时进行的，先上传附件，发邮件时再把服务器反馈回来的附件代码加上，这样来实现附件与邮件的连接。发附件是把附件的文件名以及该文件转换成 ASCII 码的字符串。然后再收到服务器反馈的附件代码。

**Mail 类：**邮件的类，一个 mail()相当于一个邮件，有用户名，密码，发件人，收件人，主题，邮件的正文，附件，时间日期，标志等类成员变量，能完全反映一个邮件的所有信息，便于处理。没有成员函数，相当于一个结构体变量。

**Qt 界面中的类：**Qt 中每一个对象都是一个类。首先，每个对话框、主窗口是一个类；在这个大的类里面，包含了多个界面元素的小类；这些小类通过一些界面布局的类组织起来，填充在窗口界面中。这个界面元素布局的设计，都可以通过 Qt Designer 来完成。然后，通过 Qt 的信号与槽机制，处理各用户请求，通过调用各元素类的函数来实现相应的功能。

## ➤ 数据结构

由于不能使用数据库，数据以文件的形式存储在服务器的硬盘上。

- 一个 ini 文件记录用户名和密码信息，每行以“用户名 密码”的形式存储。
- 每封邮件以 ini 格式存储，记录发件人，时间，主题，附件真实文件名，附件存储文件名，正文。通过发件人和发件时间一起作为文件名。
- 每个用户都在服务器端有一个 txt 文件记录自己每个信箱的信息，包括邮件发件人，时间，附件，主题，所属邮件箱。
- 附件直接以无格式文件形式存储在服务器上，文件名为十位从 0000000000 开始的自动编号。
- 一个附件列表用 ini 存储在服务器上，记录服务器上存在的附件编号以及附件当前引用次数。

## 关键的技术问题研究

### 1、网络基础

- 1) 首先尝试建立基于 TCP/IP 协议的网络连接。该连接使用由 windows 提供给应用程序编程一个 API：winsocket. 总共完成客户端基类 CLsocket 类和客户端基类 Mysoket 类。客户端通过套接口和服务器相连，将请求打包成数据包，发送给服务器。服务器处理请求，并将反馈信息打包，通过同一个套接字反馈给客户端。在一次“发送请求——处理请求——反馈信息”的过程中，套接字的两端是不变的，因此能保证服务器将信息反馈给正确的客户端。

但是将邮件发送至外网邮箱（Gmail 之类），单单基于 TCP/IP 协议是不够的。在 TCP/IP 协议簇下，有专门的电子邮件协议，SMTP、POP3 等等，在其之上指定一条消息的一个或多个接收者（在大多数情况下被确定是存

在的)，消息文本就可以被传输。但在此情况下，服务器的搭建将会是一件异常轻松的事情——通过专门软件设置即可。于是本项目剩下的内容就只是写关于客户端的 API 而已，这显然不是我们想要的。所以本项目虽然只实现了基于 TCP/IP 的数据传输，不能发邮件至外网信箱，但却完成了一个完整的邮件系统的设计。

## 2) 多线程 or 单线程, it's a question.

服务器端处理请求时，有一个问题是必须考虑的。即如果存在多个用户，如何处理他们的请求？容易想到的是，设置一个用户上线 MAX\_CLIENT，开一个容量为用户上线的套接口堆栈，每连接一个用户，就保持该用户和服务器之间的套接字。这样，服务器端可以同时和多个用户保持连接，逐一处理他们的请求。

相伴而来的问题是，通过何种方式逐一获取用户的请求？单线程方法显然不可行，如果用一个循环逐一查找各个套接口是否有请求传入，当某一个接口没有请求传入时，服务器会一直等候该接口的请求，下一个接口的请求必须要在上一个接口请求处理后才能被处理。就好比银行只有一个工作人员，她要从 1 号窗口服务至 n 号窗口，第 k+1 个客户必须要在第 k 个客户被服务之后才能被服务。

当套接口不多时，多线程编程方法可以解决该问题。可以理解为人不多时，该工作人员可以随机出现在某一个窗口停留一小段时间，等候请求。但是客户太多时，工作人员就会因为处理请求而无暇顾及其他客户的请求。并且线程太多程序也很难实现。

保持和客户端的套接字，可以方便服务器处理该客户端发送的连续请求。并且服务器端可以自行对请求归属于哪个客户端加以识别。但这个构想很难实现。并且如果客户端占用某个套接口，却一直没有请求发送，会造成资源浪费。

## 3) 服务器：我不知道你是谁，但我只处理你一个请求.

最终的服务器模型设计为：一次只处理一个请求，在请求当中，客户端必须对自己的身份进行标记。服务器端不保持和任何客户端的套接字，即无法连续处理客户端的请求。每次处理完请求后，立即关闭套接口，等待下一个请求。

这种服务器模型的好处是，只用开放一个套接口，只要用户请求不是绝对的同时到达服务器，就可以“同时”处理多个用户的请求。请求到达的时间绝对相等，在现实中概率是极低的。

#### 4) 数据传输格式：base64 编码

数据在发送过程中，不能使用“明文”，而需要一定加密。否则邮件被截获，就没有任何保护可言。本项目中数据传输全使用 base64 编码。

base64 不是安全领域下的加密解密算法。其实 base64 只能算是一个编码算法，对数据内容进行编码来适合传输。Base64 所造成数据冗余不是很严重，Base64 是当今比较流行的编码方法，因为它编起来速度快而且简单。

#### 5) 查找文件的问题：

开始使用的数据库模型是所有用户邮件列表放在一个文件中，所有正文放在一个文件中。基于这种模式设计完成了邮件的收发请求，登陆注册请求。但在小组讨论中，大家一致认为这种数据存储模式很浪费查找时间而且查找不方便。

于是我们把数据存储方式修改为每一个用户用一个文件存储邮件列表，每一封邮件正文存放在一个文件中。经过较大的改动，我们完成了在新的数据模型下的邮件收发、登陆注册请求。

#### 6) 处理删除邮件的问题

为了彻底删除一封邮件，必须把它对应的文件从服务器上删掉。但是为了防止服务器上其他用户对此邮件的引用仍然存在，我们必须记录邮件引用次数。当引用次数仍大于 1 时仅把它从当前用户邮件列表中删除，并将引用次数减 1。仅当引用次数减至 0 时，删除对应正文的文件。

附件的处理方式与邮件大致相同。

## 2、数据库基础

服务器端需要设计数据库用以存储以下信息（通过 primary key 查找到该文件）：

## 1) 用户信息

使用.ini 格式存储，唯一。内容如下：

用户名	密码
-----	----

## 2) 用户邮件列表

使用.txt 格式存储，多个文件。内容如下：

发件人	发件时间	附件信息	所属信箱	主题	正文
-----	------	------	------	----	----

Primary key：用户名（收件人）

## 3) 附件信息

使用.ini 格式存储，唯一。内容如下：

附件编号	引用次数
------	------

## 4) 正文内容

使用.ini 格式存储，多个文件。内容如下：

引用次数	正文
------	----

Primary key：发件人+发件时间

## 5) 附件内容

无格式，多个文件。内容如下：

附件编号	附件内容
------	------

Primary key：附件编号



## 开发过程中的问题和解决

**Bug:** 户登入后，无法自动显示邮件列表

**分析：**

原实现方式是：在主窗口构造函数里调用登入函数，登入成功后，调用切换到“收件箱”函数，显示“收件箱”邮件列表。但由于主窗口未创建，邮件列表无法显示，只有空白的原始界面。

**解决方法：**

取消软件打开后，登入窗口自动弹出功能，改为软件一开始默认为未登入状态，点击“登入”按钮，在登入窗口登入。

**Bug:** 换用户后，邮件列表信息显示混乱

**分析：**

主窗口有保存当然用户数据的变量，邮件列表：mailList。但在切换用户后将其清空，问题仍存在。由于 mailList 的数据都是从 recvbuf 中读取，因此 recvbuf 也应该清空。

**解决方法：**

在每次向服务器请求操作结束后，清空 recvbuf；在每次用户登出后，清空用户数据。

**Bug:** 户登出后，界面仍显示登出前的信息

**分析：**

虽然用户数据已经清空，但界面上的数据未清空。

**解决方法：**

自动刷新界面，切换界面到“收件箱”界面。

**Bug:** 界面切换后，邮件列表栏有多余的空行，即有框架，无内容

**分析：**

界面内容虽已清空，但界面框架数据未回复原状。

**解决方法：**

网上搜索解决方法后，可以使用QTableWidget类中的setRowCount(0)函数将列表的列维度回复到

**Bug:** 邮件发送后，在发件人的“发件箱”中存在，但收件人的“收件箱”中没有

**分析：**

初步判断为服务器端bug，与服务器端开发者联系后，得知是邮件发送请求格式错误，服务器端无法正常处理邮件发送请求。

**解决方法：**

按照邮件发送协议修改邮件发送请求。

**Bug:** 查看邮件时，邮件正文最后出现一个奇怪的字符

**分析：**

对查看邮件函数进行调试，发现从服务器端接收的正文数据最后含有'-1'，即EOF。与服务器端开发者联系后，确认将文件结束符EOF连同邮件正文数据传输到客户端上。

**解决方法：**

修改服务器端查看邮件函数，不将文件结束符EOF返回。

**Bug:** 中文显示异常

**分析：**

Qt用字符串自身的字符串类，对每个字符使用的是16位Unicode编码。将含有中文的Qt字符串与C风格的字符进行转化时，存在编码转化的问题。由于转化编码选择不当，导致中文显示异常。

**解决方法：**

经网络搜索，可以使用 toLocal8Bit()函数转化编码。

## 程序存在的问题与改进展望

由于时间有限，我们的程序中还存在一些问题。经过我们的讨论，程序的不足之处大致如下：

- 没有黑名单功能
- 每封邮件暂时只能添加一个附件
- 无邮件回复、转发功能
- 邮件列表查看不够人性化
- 无垃圾箱中邮件恢复功能
- 不能与普通的 E-mail 直连

对于黑名单功能，我们需要给每个用户的用户信息文件中添加一个序列，将序列中的用户发来的邮件直接标记为垃圾邮件放入垃圾箱。要实现这个改进难度不大。

对于附件个数的限制也牵扯到自定义邮件协议的修改，总的来说实现起来不困难。

邮件的转发、回复，只需在现有邮件的基础上，进行一些修改，再重新发送出去，较容易实现。

现邮件列表只有单页显示。若邮件数量足够大时，查看邮件显得不够方便。可以使用分页显示邮件列表。这在 Qt 中应该有相应的功能，不难实现。

垃圾邮件的回复是邮件删除的逆操作，只需进行相对应的修改便可。

直连问题是最难解决的，因为要实现直连就必须按统一的标准来实现邮件传输，而我们在这个实验中选择了自己设计邮件协议。如果要直接添加这个功能的话，程序几乎要面临重写。我们经过讨论认为，通过把跨系统邮件在进入系统时进行常规邮件协议与自定义邮件协议的转化比较切合实际。这也需要建立在了解了常用邮件协议的基础之上。这同时带来了内部邮件更安全的特性。

另外，传输内容的加密方法也可以进一步改进。

## 项目合作总结

作为面向对象程序设计的课程设计，我们选择了邮件系统来作为我们小组的选题。选这个题目是因为我们觉得这个题目比较有意思，可以在实验的过程中学习到网络传输相关的知识。将我们在课本上学到非常理论化的东西转化成为更有现实意义的一个实用程序，这本身就非常有意义。

在几次简单的小组讨论之后我们为了减小难度决定把用户邮件限制在系统内部以避免对正式邮件协议的学习。王道远确定了数据安排的形式以及自定义邮件协议的雏形后，我们在此基础上进行了分工，谭洪林封装了用来实现数据传输的类，并且和王道远负责服务器端的编写，这部分包含了程序的主要处理部分。客户端则有何宁栩和赵冰霁一同编写，虽然没有什么复杂的数据结构但是对我们这些没有开发经验的人来说，使用好 Qt 作为我们的界面开发工具也比较有难度。在实现过程中我们又根据实际情况对自定义的邮件协议进行了及时的修改，最终成功的实现了设计功能。

过程不算轻松吧，也有通宵奋斗。但是收获确实不是写一个工资管理系统之流所能比拟的。越是不容易，成就感也越强啊！当然也有一些关于合作的经验，这也是很重要的收获。

## 系统开发日志

5/28:开始进行网络基础设计的自学。

6/1:实现了简单的服务器和客户端搭建。

6/3:尝试在自己搭建的服务器上实现向外网发送邮件，尝试失败。由此决定限制在系统内部发送邮件。

6/6:将服务器和客户端的网络基础各自封装成基类，以待服务器类和客户端类继承。

6/7:实现 64 位加密传输。

6/7:根据网上教程实现 Qt 与 VS 的连接。

6/8:自定义邮件协议制定完成

6/12:服务器第一版基本完成。

6/13:为添加多收件人而引入引用次数。同时方便删除的实现。

6/13:服务器实现删除处理。

6/15:客户端功能实现。

6/16:增加附件收发功能。

6/18:增加垃圾箱。

6/19:服务器与客户端对接调试。

6/20:图形界面完成。

6/20:调试最终完成。

## 程序使用说明

### 1. 环境配置：

在Demo文件夹中提供的客户端，设置的是本地电脑的ip地址，所以需要将服务器端和客户端放在同一台计算机上。

若要实现不同计算机间的邮件通信，需要将客户端中的数据发送目标ip改为当前服务器端计算机的ip，重新编译客户端便可以实现。

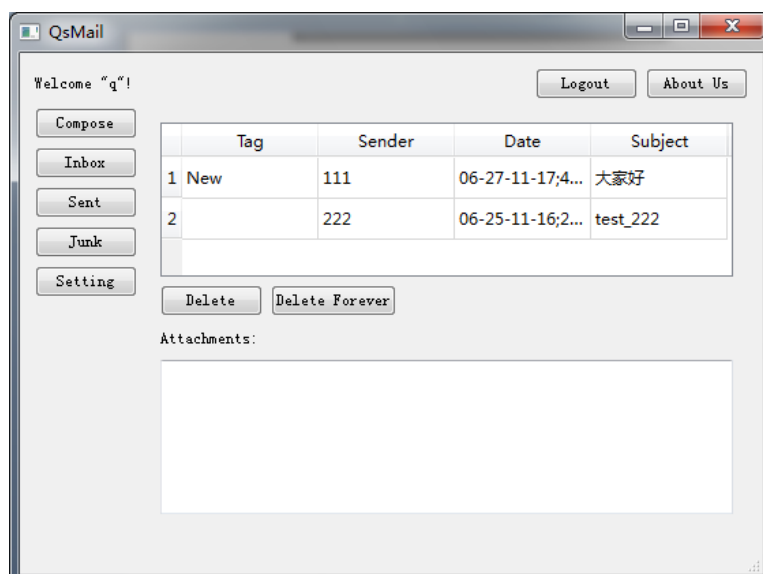
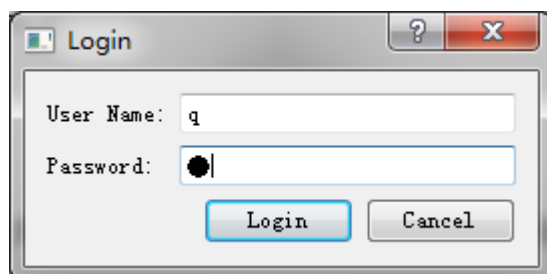
使用客户端时，须将QtCore4.dll、QtGui4.dll放在客户端同目录下。

本系统支持多客户端与服务器端连接通信。

### 2. 登入：

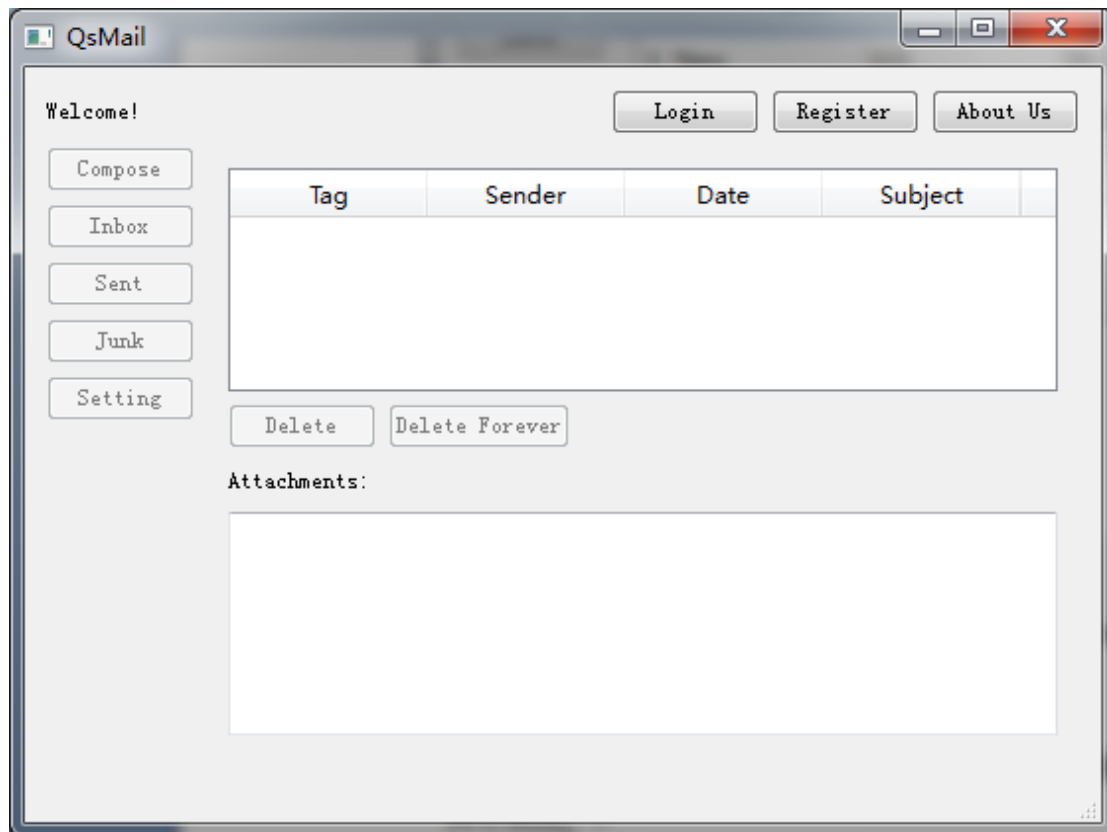
打开QsMail后，软件默认为登入状态，显示为空的收件箱界面，且无法进行界面切换。

点击右上角“Login”进入登入窗口。输入用户名、密码登入，若登入信息错误，弹出错误信息窗口；否则，登入成功，界面的左上角将显示当前登入用户信息，并自动切换到登入者的“收件箱”界面，显示“收件箱”邮件列表。



### 3. 登出：

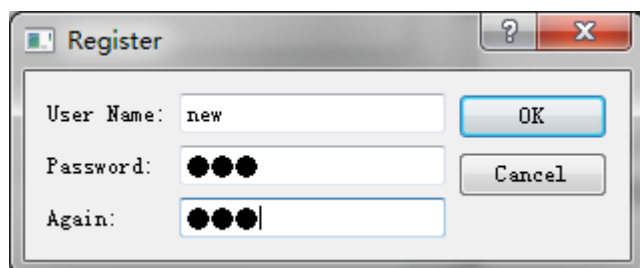
在登入后，点击右上角的“Logout”登出系统，界面将恢复到登入前状态，但窗口并不关闭。



### 4. 注册：

在未登入状态，点击右上角的“Register”进入用户注册界面。

输入用户名，用户名不支持中文；输入密码两次，判断密码无误、用户名有效后，注册成功；否则，提示相关错误。



## 5. 写邮件：

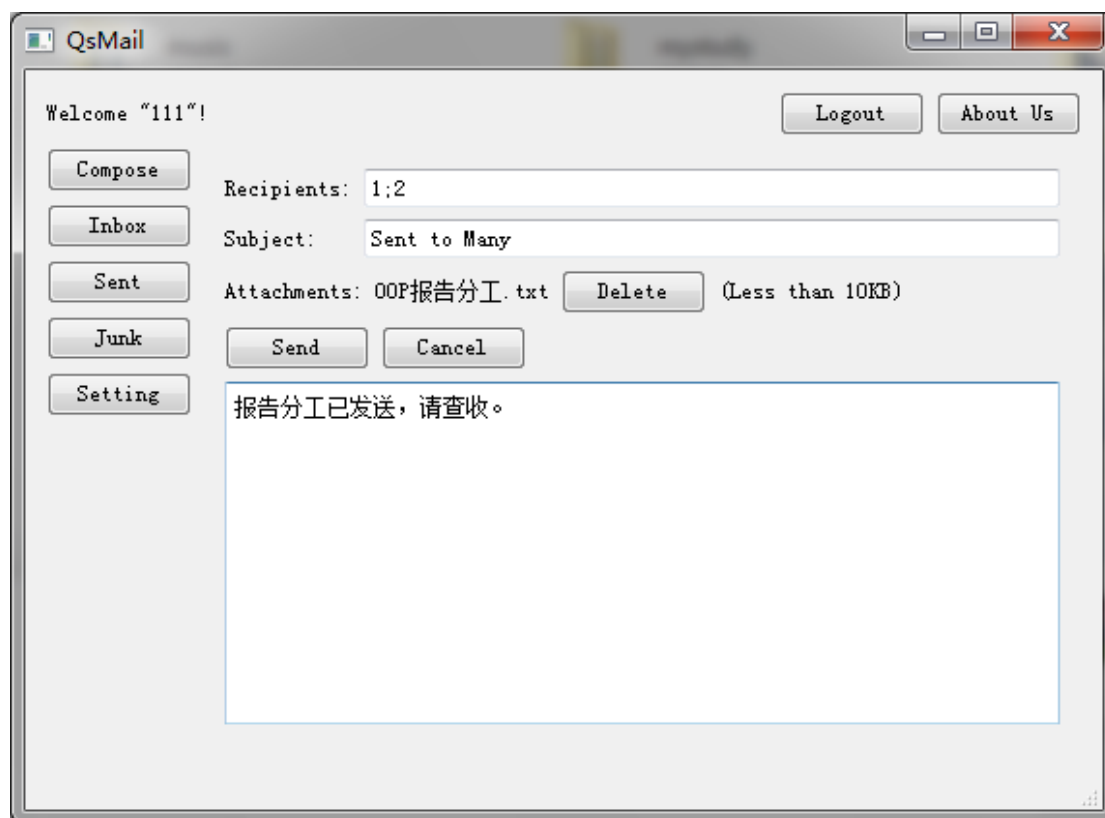
在“收件人”栏中填入收件人用户名，支持多对象邮件发送，各收件人用“;”隔开。

支持添加附件发送。点击“Upload”上传附件，附件大小需小于10KB。若上传成功后，提示成功信息，并显示当前添加的附件名。若重新上传附件，可以点击“Delete”删除已上传附件，重新上传。

收件人、邮件标题为必填内容。邮件内容、标题、附件支持中文。

邮件写好后，可以点击“Send”发送；若重新填写，点击“Cancel”清空当前内容。

注意：在写邮件过程中，若切换到“收件箱”、“发件箱”等其他界面，将丢失已编辑邮件内容。



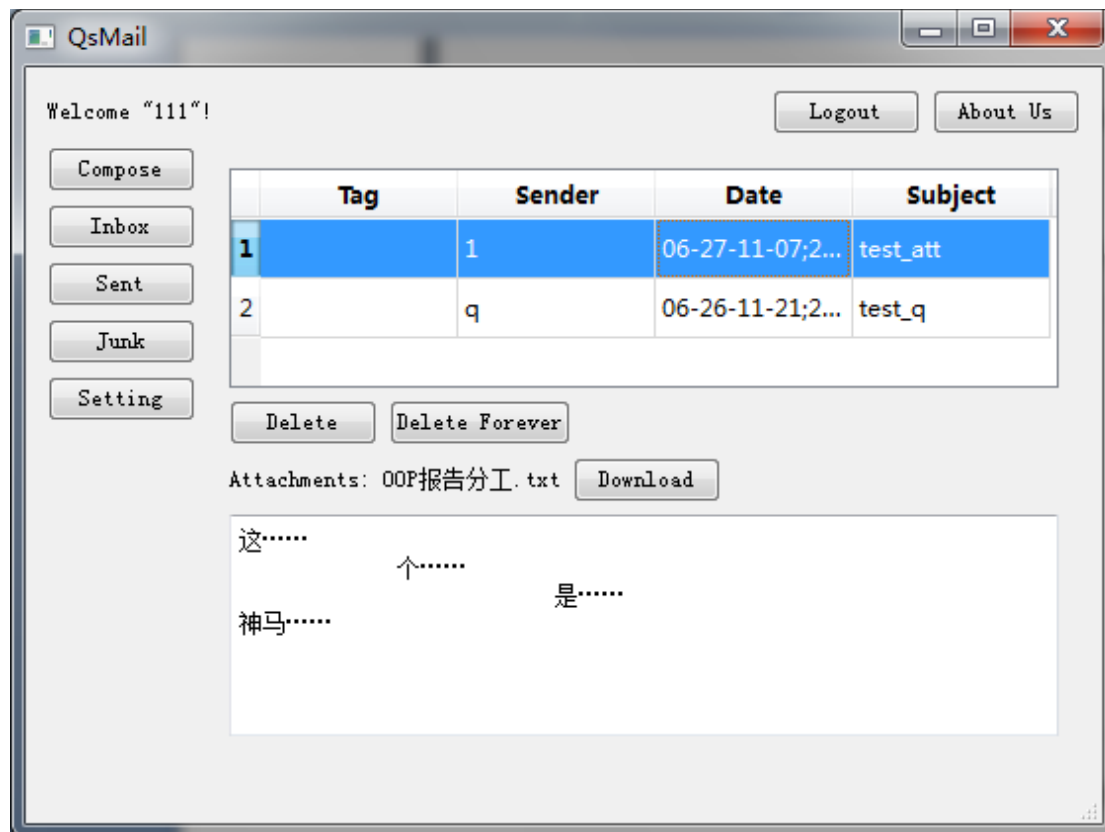


## 6. 查看邮件：

系统共有三个邮箱：“收件箱”、“发件箱”、“回收站”。点击相应邮箱，显示其邮件列表，单击具体邮件，可以进行删除、永久删除等操作。

若在“Tag”栏有“New”标记的邮件，是新收到的邮件。

双击邮件打开，在界面下方显示邮件正文；若有附件，显示附件名，可以点击“Download”下载。



## 7. 修改密码：

用户登入后，点击“Sentting”进入设置界面，可以进行密码修改操作。若原密码输入正确，新密码输入无误，将更新用户密码。

The screenshot shows a 'Modify Password' dialog box with the following fields and buttons:

- Old Password: [password field]
- New Password: [password field]
- Again: [password field]
- Update button