

# 邮件管理系统项目设计说明

沈鼎涛 田泽睿 王若鹏 周凯文

## 1. 需求分析

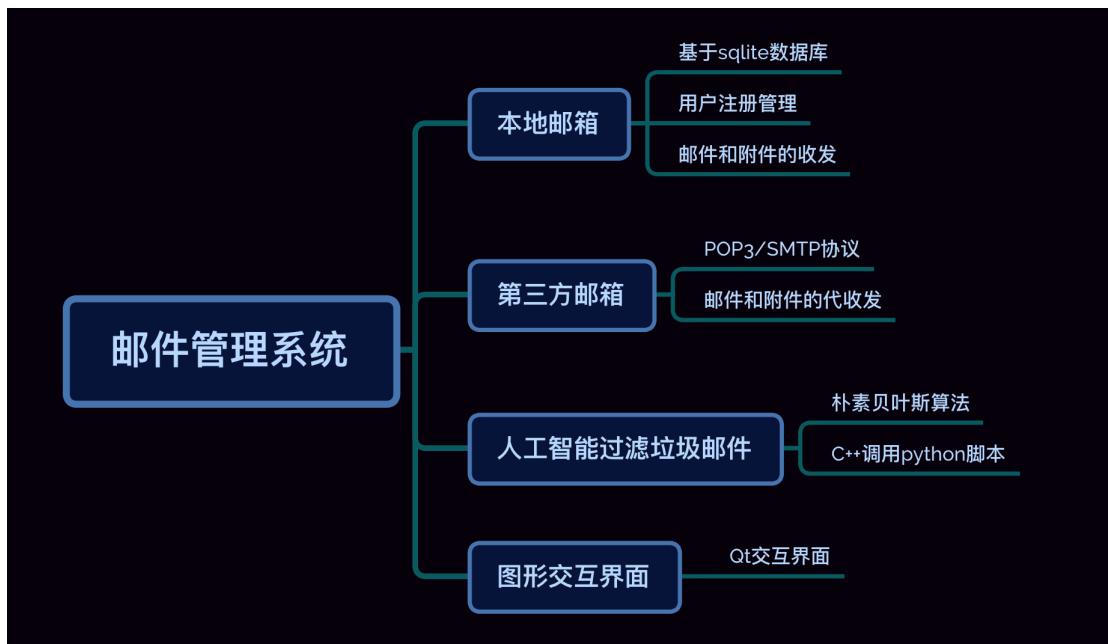
邮件管理系统首先要做到本地邮件的收发，用户具有如下的需求：本地邮箱的注册与登录，本地邮件的收发与管理，邮件和账户信息需要存储在数据库中。

除了本地邮件收发以外，邮件管理系统最重要的功能就是实现第三方邮件的收发。在我们日常生活中，接触过不少第三方邮箱管理的应用程序，如手机上的“邮件”app，网易邮箱大师等等。这样用户就可以只通过一个客户端，管理自己的所有邮箱，大大提高了工作效率。

优秀的邮件管理系统一定具备过滤垃圾邮件的功能。生活中我们常常会烦忧于各种各样的垃圾邮件，如果没有一个垃圾邮件过滤系统，那么用户的体验一定十分不好。所以我们建立了垃圾邮件过滤器，当鉴别出邮件为垃圾邮件时，会自动提示，并将其放进垃圾箱，优化用户体验。

除了功能设计，良好的应用程序还需要更友好的图形界面。

## 2. 总体设计



我们将本地邮箱账户信息和邮件信息储存在 Sqlite 数据库中，并利用 Sqlite 数据库操作实现了邮件账户的注册与登录，以及邮件的收发和管理。邮件的管理中包括对于收件箱邮件的删除、标记已读等操作。

第三方邮件收发功能旨在为邮件管理系统提供处理第三方邮件的能力，从而大大扩展

该系统的受众群体。我们通过 POP3 和 SMTP 协议，实现了邮件的收取和发送。同时在数据传输过程中使用 base64 编码并融入了 SSL 加密，使得邮件传输的安全性大大增强。

垃圾邮件的识别与过滤采用了朴素贝叶斯算法，首先利用训练数据进行了训练，得到朴素贝叶斯分类器，之后调用该分类器即可识别垃圾邮件。

图形界面的版本使用基于 c++ 的 Qt 图形用户界面应用程序开发框架，在图形界面的基础上实现了上述设计的部分功能。

### 3. 系统模块说明

#### 3.1 基于 sqlite 数据库的用户注册管理

Sqlite 创建 EmailSys 数据库，并创建 EmailAccount 表存储邮件账户信息，EmailLib 表存储所有邮件信息，两个表的结构设计如下：

EmailAccount			
列名	数据类型	意义	约束
ADDRESS	CHAR(50)	邮箱地址	PRIMARY KEY NOT NULL
PASSWORD	CHAR(50)	密码	NOT NULL

EmailLib			
列名	数据类型	意义	约束
ID	INTEGER	邮件唯一识别符	PRIMARY KEY AUTOINCREMENT NOT NULL
RECEIVER	CHAR(50)	收件人	NOT NULL
SENDER	CHAR(50)	发件人	NOT NULL
TIME	DATETIME	发件时间	DEFAULT CURRENT_TIMESTAMP
TITLE	CHAR(30)	标题	NOT NULL
CONTENT	TEXT	内容	NOT NULL
ATTACHPATH	CHAR(100)	附件路径	DEFAULT NULL
DELETE_STATUS	INT	删除状态	DEFAULT 0
READ_SIGN	CHAR(10)	阅读标记	DEFAULT '[Unread]'

Email 类定义如下：

```
class Email{  
    char* address;  
    char* password;  
public:  
    Email() {};  
    Email(char* Address, char* Password) {  
        address = Address; password = Password;  
    }  
    ~Email() {};  
    void Register();  
    void Login();  
};
```

其中成员函数 Email::Register() 和 Email::Login() 分别实现邮箱账户的登录与注册。

(1) 登录: 输入用户名和密码, 通过 Login\_Check\_Account 函数连接数据库查询 EmailAccount 表中是否有用户名和密码的匹配记录, 若有, 则登陆成功, 调用 Email\_Home 函数显示邮箱主页, 否则提示“用户名或密码错误”。

```
请输入邮箱地址: lijian@oop.com
请输入密码: 1998_love_suki
>>> Welcome, lijian@oop.com <<<
>>> 退出登录: T
>>> 写信: W | 收件箱: Q
>>>
```

(2) 注册：输入注册的用户名（加@oop.com后缀），通过 Check\_Account 函数连接数据库查询 EmailAccount 表中是否存在输入的用户名，若已存在，则提示“邮箱已存在，请更换用户名！”；否则提示输入密码，通过 Record\_Insert 函数连接数据库插入记录，完成注册。

```
>>>>> Welcome to the EmailSystem.<<<<<  
    选项          按键  
    本地邮箱登录  L1  
    第三方邮箱登录 L2  
    注册          R  
    退出          Q  
>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<  
请选择操作: R  
请输入用户名(加@oop.com后缀) lijian@oop.com  
邮箱已存在,请更换注册名!
```

```
请输入用户名（加@oop.com后缀）register@oop.com
请输入密码：registerTEST
是否确认注册？[确认：Y / 退出：N]
>>> Y
注册成功
```

### 3.2 本地邮件及附件的收发

### (1) 邮件读取和管理

操作流程：进入邮箱主页后，选择收件箱进入，可以看到所有收取邮件目录，显示的邮件信息包括邮件 ID，标题，读取状态（[Unread/Hasread]），发件人，附件地址，发件时间。后面列出返回或选择 ID 读取详细内容。在详情界面，可以选择对邮件的状态管理，选择删除或标记已读。

实现方式：邮件目录显示和内容详情显示是通过登陆邮箱名在数据库中以 select 语句进行条件查找，如目录显示 EmailLib 表中收件人(Receiver)是登陆邮箱名且删除状态(Delete\_Status)为 0 的记录。删除和标记已读操作则是利用 update 语句将对应邮件记录的删除状态(Delete\_Status)和已读标记(Read\_Sign)的值进行更新。

```
>>>           Welcome, lijian@oop.com      <<<
>>> 退出登录: T
>>> 写信: W | 收件箱: Q
>>> Q
-----
>>> >>> >>> >>> InBox   <<< <<< <<< <<<
-----
ID      状态    标题          发件人        附件地址      发送时间
1 [HasRead] test fuckoop@oop.com test/test.test 2020-07-04 11:07:44
>>> 返回: B / 输入ID阅读
>>> 1
-----
>>>标题: test
>>>发件人: fuckoop@oop.com
>>>正文:
test
-----
>>> 返回: B   删除: D  标为已读: F
>>> -
```

## (2) 邮件的发送

操作流程：通过邮箱主页选择进入写信，按照提示输入收件人、标题、正文、选择并输入附件的路径，并决定是否发送。如果收件人邮箱无记录，则提示“收件人不存在”。

实现方式：邮件发送再数据库中的操作实际是一个插入（INSERT）操作，将输入信息整合

成一条记录插入 EmailLib 中。

```
>>>          Welcome, lijian@oop.com      <<<
>>> 退出登录: T
>>> 写信: W | 收件箱: Q
>>> W
>>> >>> >>> Write Email    <<< <<< <<<
>>> 返回: B
>>> 收件人: ti@oop.com
收件人地址不存在!
>>> 返回: B
>>> 收件人: -
```

```
>>>          Welcome, lijian@oop.com      <<<
>>> 退出登录: T
>>> 写信: W | 收件箱: Q
>>> W
>>> >>> >>> Write Email    <<< <<< <<<
>>> 返回: B
>>> 收件人: register@oop.com
>>> 标题: 这是一个测试
>>> 正文:
哈喽世界，这是一个测试邮件，不知道是否有八哥。
是否添加附件? [Y/N]
Y
请输入附件路径: lujing/test.jpg
确认发送: [Y/N]
Y
发送成功!
```

(3) 附件的收发：附件储存在本地文件夹中，将存储路径作为一个字符串随邮件存储在 EmailLib 表中，读取时取出该路径，并利用该类路径读取附件。如果有图形化界面，可以通过附件的图片地址读取并显示图片。

### 3.3 第三方邮件及附件的收发

#### 3.3.1 原理

第三方邮件收发的功能主要基于 SMTP 协议和 POP3 协议来实现，数据先通过 base64 进行加密，为了提高安全性在传输层中使用 SSL 加密技术。

SMTP 协议即简单邮件传输协议（Simple Mail Transfer Protocol）是一个在互联网上传输电子邮件的标准。主要用于系统之间的邮件信息传递，用于第三方邮件发送。

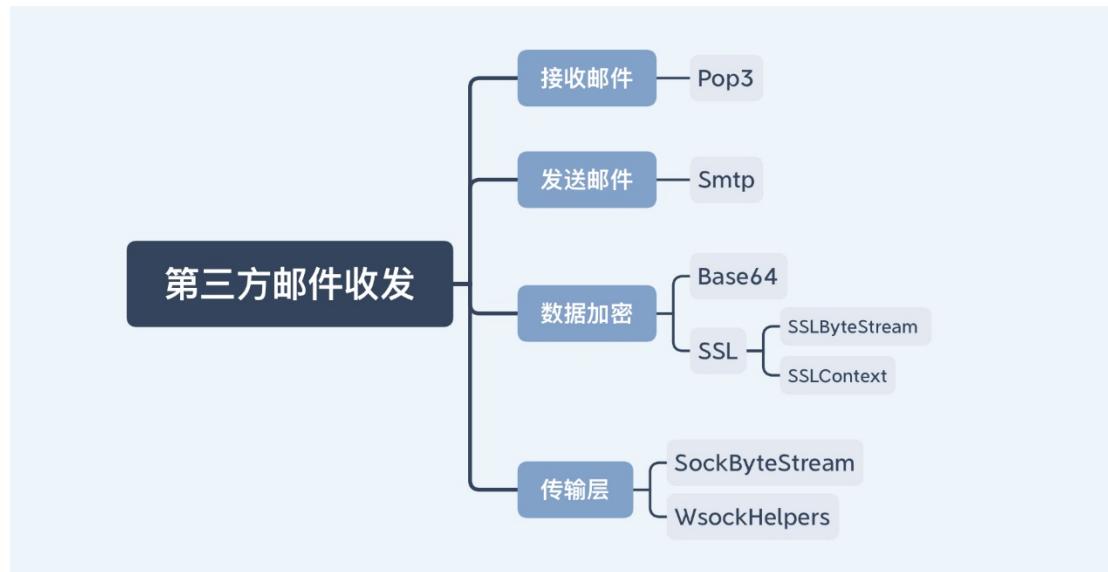
POP3 协议即邮局协议（Post Office Protocol），支持第三方邮件接收。其具体过程是：邮件发送到服务器上，电子邮件客户端调用邮件客户机程序以连接服务器，并下载所有未阅读的电子邮件。

Base64 是一种基于 64 个可打印字符来表示二进制数据的表示方法。每 6 个比特为一个单元，对应某个可打印字符。3 个字节相当于 24 个比特，对应于 4 个 Base64 单元，即 3 个字节可由 4 个可打印字符来表示。多用来作为电子邮件的传输编码。

SSL 全称安全套接层，是一种安全协议，目的是为互联网通信提供安全及数据完整性保障。通过各种加密和身份认证算法，保证两个应用间通信的保密性和可靠性，使客户与服务器应用之间的通信不被攻击者窃听。能在邮件收发过程中，能极大地保护用户的隐私。

### 3.3.2 实现过程

第三方邮件收发功能的结构如下所示：



他们的功能描述如下：

源文件名称	功能描述
<b>Base64Codec</b>	Base64 加密解密
<b>Pop3</b>	POP3 协议接收邮件
<b>Smtp</b>	SMTP 协议发送邮件
<b>SockByteStream</b>	Socket 比特流控制
<b>SSLSocketStream</b>	SSL 比特流控制
<b>SSLContext</b>	SSL 数据加密
<b>WsockHelpers</b>	WINSOCK 协助

其中 POP3 和 SMTP 的核心类及其主要成员如下：

类	成员权限	成员类型及名称	功能
POP3	private	int port	目的端口号, 995 (使用 SSL)
		string Pop3Server	POP3 服务器地址, 如 pop.163.com
		string Pop3User	邮箱用户名, 如 cs_test2020@163.com
		string Pop3Pass	登录密码或授权码
	public	POP3()	POP3 类的构造函数
		bool CreateSocket()	创建 socket 链接, 成功则返回 1
SMTP	private	int port	目的端口号, 995 (使用 SSL)
		string SmtpServer	SMTP 服务器地址, 如 smtp.163.com
		string SmtpUser	邮箱用户名, 如 cs_test2020@163.com
		string SmtpPass	登录密码或授权码
		string target	邮件发送的目的邮箱

public	string title	邮件的标题
	string content	邮件的正文内容
	SMTP()	SMTP 类的构造函数
	bool CreateSocket()	创建 socket 链接，成功则返回 1
	void AddFile()	为邮件添加附件
	int SendMail	发送邮件

### 3.3.3 实现结果

- (1) 实现了 SMTP 和 POP3 协议，成功进行第三方邮箱的邮件收发。
- (2) 第三方邮箱代发送：以 cs\_test2020@163.com 的身份向 wrp1999@qq.com 发送邮件，标题为”OOP TEST MAIL”，附件为一张图片 1.jpg，效果如下：



- (3) 第三方邮箱代接收：从 VS 上登录 cs\_test2020@163.com，获取最近接收到的邮件，效果如下。可以看到收件箱里一共有 2 封邮件，Message #1 标记为第一封邮件，为网易 service 发送的，Message #2 标记为第二封邮件，是来自 qq 邮箱发送的。

```

选择Microsoft Visual Studio 调试控制台
Connected to: pop.163.com port:995
Server greeting: +OK Welcome to coremail Mail Pop3 Server (163coms[10774b260cc7a37d71b52404dcf5cs])
User name: cs_test2020@163.com
Total messages: 2

Message #1 headers:
Received: from =>?UTF-8?B?572R5piT6YKu5Lu25Lit5b+D?= (unknown [192.168.194.38])
    by mfast9 (Coremail) with SMTP id uMCowAAnFt+p+sle9t6GDQ--.35329S2;
    Wed, 27 May 2020 13:29:13 +0800 (CST)
From: =>?UTF-8?B?572R5piT6YKu5Lu25Lit5b+D?= <mail@service.netease.com>
Sender: mail@service.netease.com
To: cs_test2020@163.com
Message-ID: <1567417957.6368547.1590557354123.JavaMail.mail@service.netease.com>
Subject: =>?UTF-8?B?572R5piT6YKu566x77yM6YKj5Lqb5L2g?=
=?UTF-8?B?55+16YGT52KM5LiN5+16YGT55qE5LqL?=
MIME-Version: 1.0
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable
X-CM-TRANSID:uMCowAAnFt+p+sle9t6GDQ--.35329S2
X-Coremail-Antispam: 1uf129KBjdUh29KB7ZKAUJUUjj529EdanIXcx71UUUUU7v73
    VFW2AGmfu7bjvjm3AaLaJ3ubIYCTnIWievJa73eTB2vfuFKfJI43ZExa7xR_UUUUUUUU=
    =
X-Originating-IP: [192.168.194.38]
Date: Wed, 27 May 2020 13:29:13 +0800 (CST)

Message #2 headers:
Received: from qq.com (unknown [59.36.132.57])
    by mx20 (Coremail) with SMTP id RsCowAD3cpA3NNNeLJOACA--.16682S3;
    Sun, 31 May 2020 12:36:08 +0800 (CST)
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=qq.com; s=s201512;
    t=1590899767; bh=bDWes5v7f0P49.jPAIM1CaHoT40xEn1VfnVDrvrYKq0PE=;
    h=From;To;Subject;Mime-Version;Date;Message-ID;
    b=NcNaCjKhgEVYF76ocwZoJPtu3PpFcxyh2SqNpu/ln2HoaTxJZdybSHQFukTtf.jAn

```

## 3.4 基于人工智能的垃圾邮件过滤

### 3.4.1 原理

我们系统采用的算法为朴素贝叶斯分类器来进行垃圾邮件的分类。对于分类任务，贝叶斯决策论一般会考虑如何基于概率和误判损失来选择最有利的判别分类结果，假设有 $N$ 种类别 $\{c_1, c_2, \dots, c_N\}$ ， $\lambda_{ij}$ 是将一个本属于 $c_j$ 的样本误分类为 $c_i$ 的损失，则可以得到基于样本把其分类为 $c_i$ 的期望损失：

$$R(c_i | \mathbf{x}) = \sum_{j=1}^N \lambda_{ij} P(c_i | \mathbf{x})$$

我们期望可以寻找一个判别准则，以最小化总体风险：

$$R(h) = \mathbb{E}_{\mathbf{x}}[R(h(\mathbf{x}) | \mathbf{x})]$$

这就产生了贝叶斯判别准则：为了最小化总体风险，只需在每个样本上选择能使条件风险最小的类别标记，即

$$h^*(\mathbf{x}) = \arg \min_{c \in \mathcal{Y}} R(c | \mathbf{x})$$

此时 $h^*$ 称为贝叶斯最优分类器，一般可将误判损失写作：

$$\lambda_{ij} = \begin{cases} 0, & \text{if } i = j \\ 1, & \text{其他} \end{cases}$$

此时条件风险为：

$$R(c | \mathbf{x}) = 1 - P(c | \mathbf{x})$$

所以此时的贝叶斯最优分类器为：

$$h^*(\mathbf{x}) = \arg \max P(c | \mathbf{x})$$

可以看出，贝叶斯判定准则关键就在于求得后验概率，由贝叶斯公式：

$$P(c | \mathbf{x}) = \frac{P(c) P(\mathbf{x} | c)}{P(\mathbf{x})}$$

类先验概率 $P(c)$ 是样本空间中各类样本所占的比例，当训练集包含充分多样本时，由大数定理，其可根据各类样本出现频率来估计，而 $P(x | c)$ 则难以根据有限样本直接进行估计。

朴素贝叶斯分类器采用了这个假设：对已知的类比，假设所有属性互相独立，可就将公式写为：

$$P(c | \mathbf{x}) = \frac{P(c) P(\mathbf{x} | c)}{P(\mathbf{x})} = \frac{P(c)}{P(\mathbf{x})} \prod_{i=1}^d P(x_i | c)$$

则贝叶斯判别准则为：

$$h_{nb}(\mathbf{x}) = \arg \max_{c \in \mathcal{Y}} P(c) \prod_{i=1}^d P(x_i | c)$$

朴素贝叶斯分类器的训练过程就是估计先验概率  $P(c)$  以及每个属性的条件概率  $P(x_i | c)$ 。令  $D_c$  表示训练集中  $c$  类样本的集合，则可估计出先验概率：

$$P(c) = \frac{|D_c|}{|D|}$$

对于离散取值的属性，令  $D_{c,x_i}$  表示  $D_c$  在第  $i$  个属性上取值为  $x_i$  的样本组成的集合，则条件概率可估计为：

$$P(x_i | c) = \frac{|D_{c,x_i}|}{|D_c|}$$

对于连续取值的属性，可假设其服从正态分布，即

$$p(x_i | c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}\right)$$

有时为了避免其他属性被未出现的属性值抹去，还需要用到拉普拉斯平滑技术。

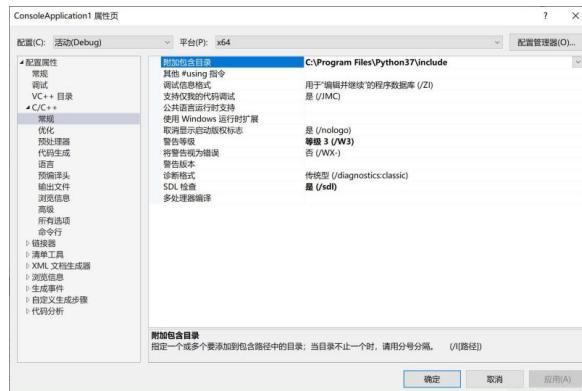
朴素贝叶斯算法的流程为：

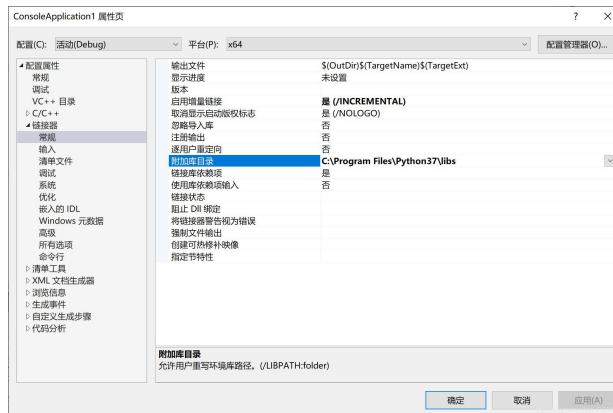
1. 计算先验概率
2. 计算特征的条件概率
3. 根据贝叶斯判别准则进行判别归类

### 3.4.2 实现过程

我们采用的方法为利用 python 脚本实现朴素贝叶斯分类器的搭建与训练，并利用 Visual Studio 对该脚本进行调用，进行垃圾邮件的甄别。

首先我们需要修改属性页：





当收到邮件时会调用 python 脚本，由训练好的朴素贝叶斯分类器对邮件进行鉴别，并输出提示。

### 3.4.3 实现结果

接受到邮件后会显示邮件鉴别系统，做到垃圾邮件过滤

```
Microsoft Visual Studio 调试控制台
mail:'this is a normal mail.' is not a spam
D:\junior\OOP\proj\ConsoleApplication2\x64\Debug\ConsoleApplication2.exe (进程 8764) 已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口. . .

Microsoft Visual Studio 调试控制台
mail:'watch this' is a spam
D:\junior\OOP\proj\ConsoleApplication2\x64\Debug\ConsoleApplication2.exe (进程 15616) 已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口. . .
```

## 3.5 交互界面

Qt 交互界面的实现主要根据功能模块和操作步骤，定义了多个窗口类，其中主窗口以 Qt 中已定义的 QMainWindow 类为基类，其余窗口以 QDialog 为基类。QMainWindow, QDialog 为 Qt 封装好的图形界面类，定义好了图形界面的基本部件和操作。

图形界面的设计首先在 QtCreator 上设计好窗口需要的图形部件，形成.ui 文件：



在 QtCreator 编译项目后，会根据.ui 文件自动生成相应的 ui\_xxx.h 头文件(xxx 为定义的窗口类的名称)，其中定义了 ui\_xxx 类，类中包括了.ui 文件中图形部件的声明，且定义了命名空间 Ui，在该命名空间下，ui\_xxx 为 xxx 类的父类，那么 xxx 类在 Ui 命名空间下就继承了 ui\_xxx 类的成员，由此图形部件能成功在窗口上展示。

设计好图形部件后第二步是设计图形部件对应的功能，此处主要利用了 Qt 的槽与信

号。槽与信号用于 Qt 对象之间的通信，以本次设计的主窗口为例：

```
class mail : public QMainWindow
{
    .....;(省略)

private slots:
    void slotBtn();//for log in
    void slotBtn2();// for register
};

mail::mail(QWidget *parent)
: QMainWindow(parent)
, ui(new Ui::mail)
{
    ....;
    connect(ui->pushButton_2, SIGNAL(clicked()), this, SLOT(slotBtn()));
    connect(ui->pushButton_3, SIGNAL(clicked()), this, SLOT(slotBtn2()));

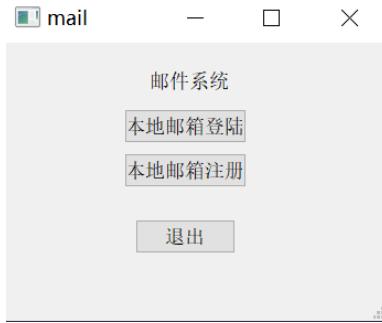
}
```

以上为类和构造函数的定义，connect 语句实现了按钮在被点击时触发 slotBtn() 函数。  
以上述为设计原理，本次开发设计了六个窗口分别实现登陆后界面，收件箱界面，写信界面，登陆界面，主界面，注册界面的功能：

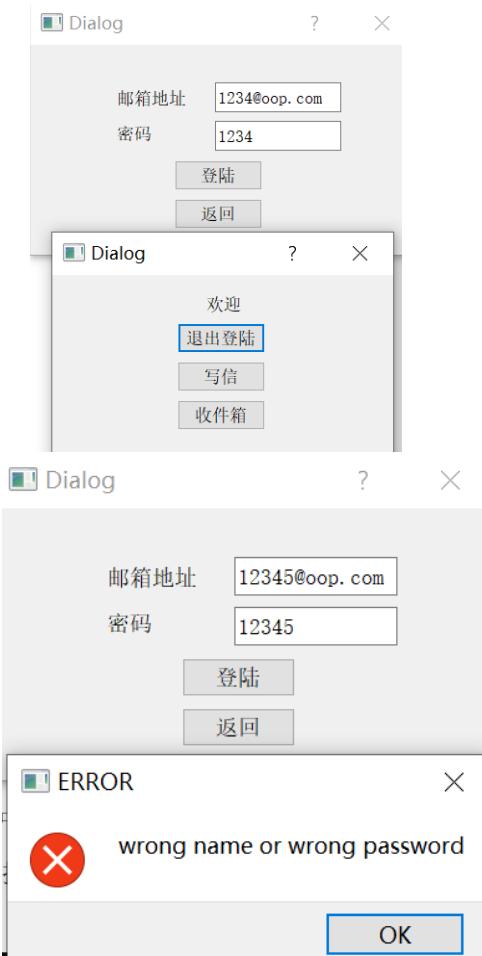
- ▷  ui\_Email.h
- ▷  ui\_Email\_read.h
- ▷  ui\_Email\_write.h
- ▷  ui\_Login.h
- ▷  ui\_mail.h
- ▷  ui\_Register.h

下面为功能的实现：

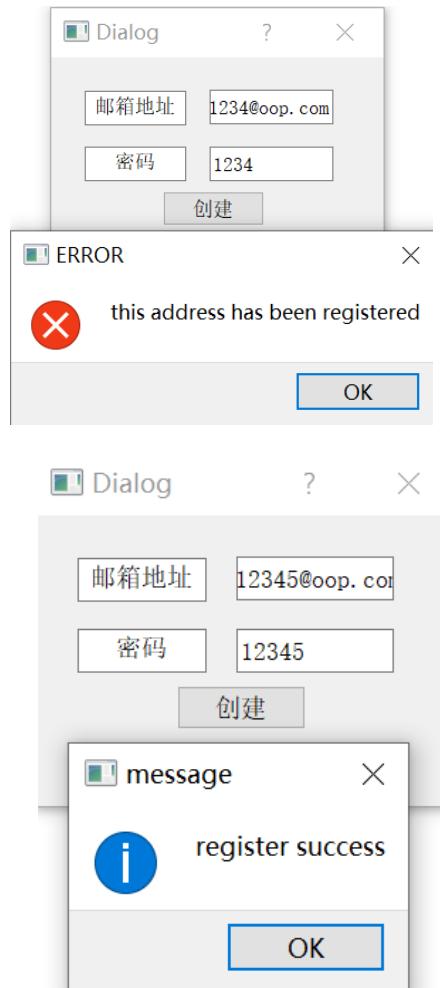
(1) 主界面：



(2) 登录界面：输入用户名和密码，查询 EmailAccount 表中是否有用户名和密码的匹配记录，若有，则登陆成功，弹出登录后界面，否则提示错误信息。



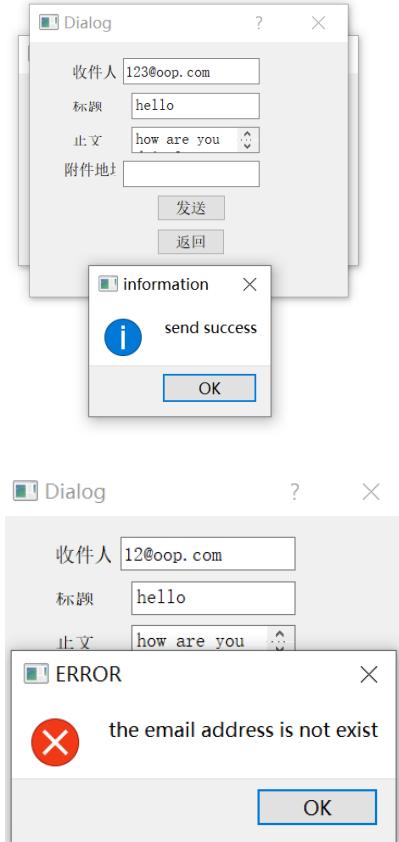
(3) 注册：主界面点击本地邮箱注册，输入注册的用户名，密码，连接数据库查询 EmailAccount 表中是否存在输入的用户名，若已存在，则提示错误信息；否则提示注册成功。



(4) 邮件读取：在登录后界面点击收件箱，会显示邮件列表，输入邮件 ID 点击查找，显示对应邮件的内容：



(5) 邮件的发送：登录后界面选择写信，编辑收件人，标题，正文，附件后点击发送，若收件人存在，则发送成功，否则显示错误信息：



#### 4. 系统设计难点及其解决

(1) 邮件的删除：考虑到邮件从收件人邮箱删除后，发件人的发件记录不受影响。因此考虑对邮件增加一个删除标记的属性，删除操作即在数据库中将该字段的值进行更新，在读取邮件时只读取删除标记符合要求的邮件，已读状态的改变与此类似。

(2) 附件的收发：如上叙述，将附件储存在本地，存取其路径。

(3) 第三方邮件收发模块的难点主要在于编写网络协议，包括传输层协议和应用层协议。传输层方面，需要编写 socket 链接；应用层方面，主要是 POP3 协议和 SMTP 协议。由于以前没有过网络编程的经验，所以只能借鉴一下既有的代码并加以修改。经过一系列 debug 之后，效果还是非常好的。以下是记录的两个问题及其解决方式：

编写 socket 需要用到库函数 winsock2.h，然而 mac 电脑不自带这个库，只能迁移至 Windows 系统下运行；

登录第三方邮箱时，以原密码登录是无法成功的。查阅资料后发现，必须使用授权码代替原密码才能成功登录。授权码可以在网页登录相应的邮箱后在“邮箱设置”菜单栏里获得。

(4) 在一开始设计人工智能算法时，我考虑利用 C++设计算法，但是这样会比较麻烦，整个算法底层的实现过程需要耗费大量的时间，所以我们考虑利用 Python 这种比较适合应用于人工智能领域的语言来编写算法，然后再利用 C++对其进行调用。我们认为系统不必局限于某种特定的语言，只要能提高其效率可以考虑其他语言进行功能实现，只需要利用

C++调用即可。

(5) Qt 架构及其功能的学习：Qt 开发虽说基于 c++, 但是需要了解 Qt 开发项目的架构，封装的函数功能，以及信号和槽的使用，需要花一定时间熟悉。

## 5. 总结

### 5.1 开发经验

邮箱账户的注册与登录、邮件的收发主要依靠 sqlite 数据库操作实现。由于对于数据库技术的陌生，因此需要从头学习 sql 语言和 sqlite 的 c++接口 API 等。熟悉掌握了 sql 的基础命令，以及在 C++中连接和操作 sqlite 数据库，很多功能实现的框架是类似的，只要通过第一次实现的函数模板进行修改就可以打通。对于删除和邮件状态管理，充分利用数据库操作的特点将需求进行转化，以间接实现需求，如删除操作不必真正地删除，可以转变邮件显示的选取条件已到达相同的效果。

对于网络编程 Windows 系统比 macOS 系统更加友善。出现 bug 时，根据错误代码，可以很容易地在 CSDN、StackOverflow 等平台上找到解决方式。

对于人工智能算法，均可以利用 python 语言进行算法的实现，再利用 C++调用 python 脚本，这样可以大大提高系统的效率与开发难度。

了解了 Qt 开发的架构和原理，完成了 Qt 开发的入门。在项目开发过程中感受到，同一个业务逻辑和功能，Qt 架构和控制台架构的实现有较大的不同，主要是程序交互的部分实现了封装，但熟悉操作之后会发现 Qt 程序的框架较为清晰。

### 5.2 项目优点

(1) 利用了轻量级的 sqlite 数据库，操作框架简单明了，同时具有简单和易于使用的 API。SQLite 不需要配置，这意味着不需要安装或管理，SQLite 是自给自足的，这意味着不需要任何外部的依赖。同时充分利用数据库管理操作的特点，对邮件管理系统的需求进行了转化，直接或间接地达到了良好的实现效果。

(2) 能快速而准确地登录第三方邮箱，并且进行邮件收发的操作。

(3) 可以有效识别垃圾邮件，优化用户的体验

(4) 实现了交互界面，用户体验优于控制台界面，并且代码的组织架构更清晰明了。

### 5.3 项目缺点

(1) 由于对于 sqlite 数据库操作没有精通，没能实现利用高级操作完成复杂功能；数据库操作和表设计存在不严密之处，对于字段没有设置严格的约束，需要在前端交互时进行限制和引导，在该方面可以更简洁；对于附件的存取没有实现更加直接完美的效果；没能实现网络传输邮件，这需要网路协议等专业知识。

(2) 首次登录较为麻烦，需要手动获取授权码才可以。邮件发送内容的设置需要在 main 函数中实现。

数里进行，不太方便。

(3) 由于训练集是英文邮件内容，所以只能识别英文垃圾邮件，后续会加入中文训练集，进行对中文垃圾邮件的识别。

(4) 由于时间有限，图形界面仅实现了基础功能。

## 附录 1：程序使用说明

图形界面版本：由于需要安装 Qt 等相关配置，运行需求较高，安装 visual studio，同时安装 Qt <http://download.qt.io/>，在 visual studio 中配置好 Qt 界面，打开图形界面版本文件夹中 QtWidgetsApplication1\QtWidgetsApplication1.sln 文件，visual studio 中点击调试（或直接运行），弹出主界面之后可按照图形界面提示操作。

控制台界面版本：解压压缩包后打开文件夹中 EmailSys(3)\EmailSys\x64\Debug\EmailSys.exe 文件，之后安装 sqlite，即可运行程序，打开邮件管理系统后，按照命令行中交互式指示信息进行操作即可。

## 附录 2：系统开发日志

**6.1-6.7** 学习 sqlite 相关知识和 sql 语言，学习 sqlite 的 C++ 接口和相关 API，完成 VS 环境中 sqlite 的配置，在命令行中尝试数据库基本操作；查找朴素贝叶斯分类器的训练数据，搭建朴素贝叶斯分类器，并利用训练数据进行训练。

**6.8-6.14** 查询相关资料，学习 POP3 协议和 SMTP 协议的基本原理，下载并配置好 Qt，完成 hello world 图形界面；搭建 Visual Studio 运行 Python 脚本的环境；配置好相关环境。

**6.15-6.22** 利用 C++ 实现 sqlite 数据库操作，编写相关函数，创建数据库，实现表的创建、记录插入、删改等操作；根据项目功能需求设计表结构，并利用已有的函数模板创建相关数据库和表；编写 POP3 和 SMTP 协议的 C++ 代码并调试；利用 C++ 调用 Python 脚本文件，成功识别垃圾邮件。

**6.23-6.30** 申请测试邮箱 cs\_test2020@163.com，进行邮件收发测试；在初步控制台程序的基础上开始图形界面的开发，调整代码结构，构建好思路和框架；定义好各个窗口类，在 QtCreator 上设计好图形部件；将第三方邮件收发模块整合入邮件管理系统。

**7.1-7.5** 完善邮件管理系统框架，接入数据库并编写函数实现前端对数据库的操作和数据的回调等，完成交互界面的设计与实现；实现窗口类之间的交互，并配置好数据库，完成程序所有功能的实现；对图形界面进行优化。