

MicroMailClient 开发文档

一. 主要功能

- 使用 MVC 设计模式开发的邮件管理客户端(QT + Material Design, 设计参考 Win10 UWP Mail)
- 用户可以在客户端内使用常用的邮件系统(通过 SMTP/POP3/IMAP)
- 可以进行邮件及附件的收发, 查看等

二. 主要模块

1. View(视图)(2 人):
 - 登陆
 - 收件箱
 - 发送邮件
 - 查看邮件
 - 上传, 下载附件
2. Model(模型)(2 人): 客户端登录及收发邮件, 查看邮件, 附件下载等的程序实现
3. 整体架构及接口设计(1 人)
4. 邮件接受/发送功能实现(2 人)
5. 软件开发文档(4 人)

三. 开发平台及工具

1. 使用 git 作为版本管理工具
2. 全局使用 C++11 新特性开发, 使用智能指针, 尽量遵循 [Google 开源项目的 C++ 风格](#).

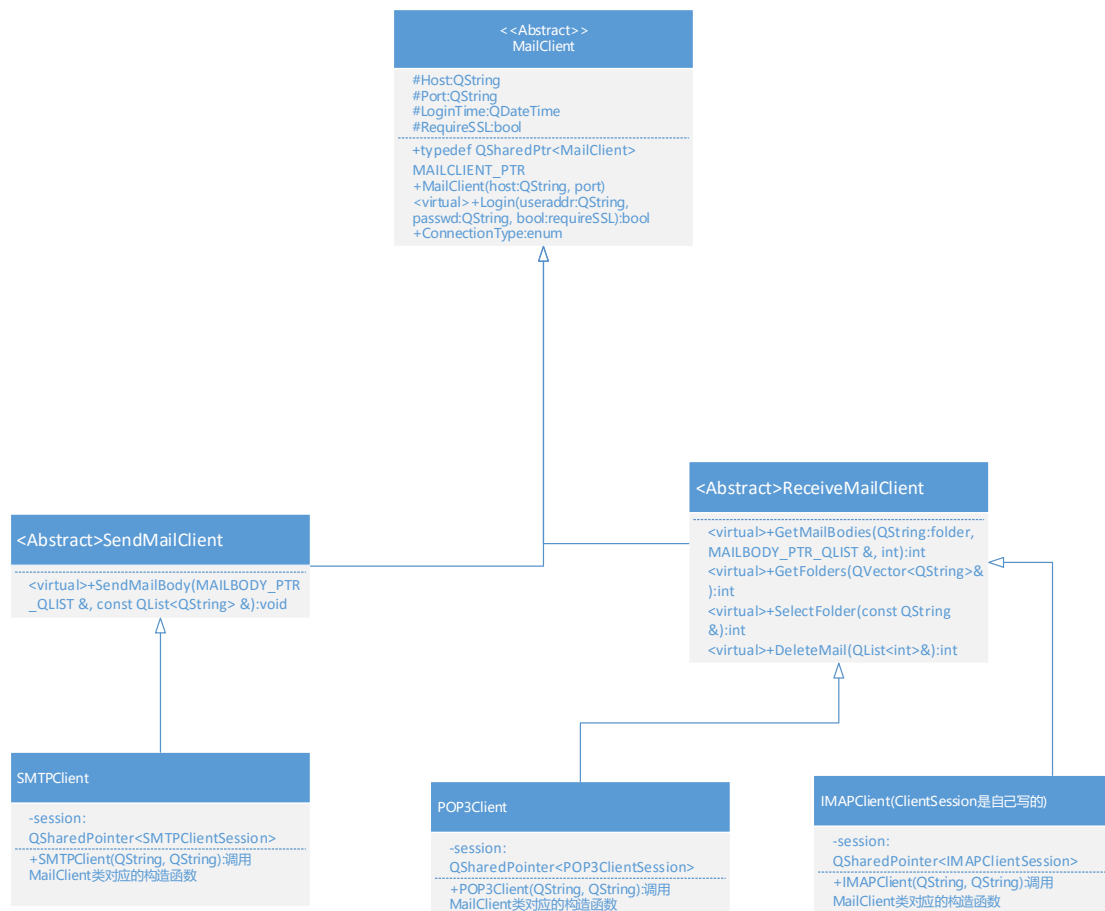
3. 开发环境和框架:Qt Creator + Qt 5.6(MinGW4.9)
4. 网络编程主要使用 POCO C++ Library 实现, [官方文档](#)
5. 前端主要使用 qml-material 扩展模块实现, [官方文档](#)

四. 框架设计

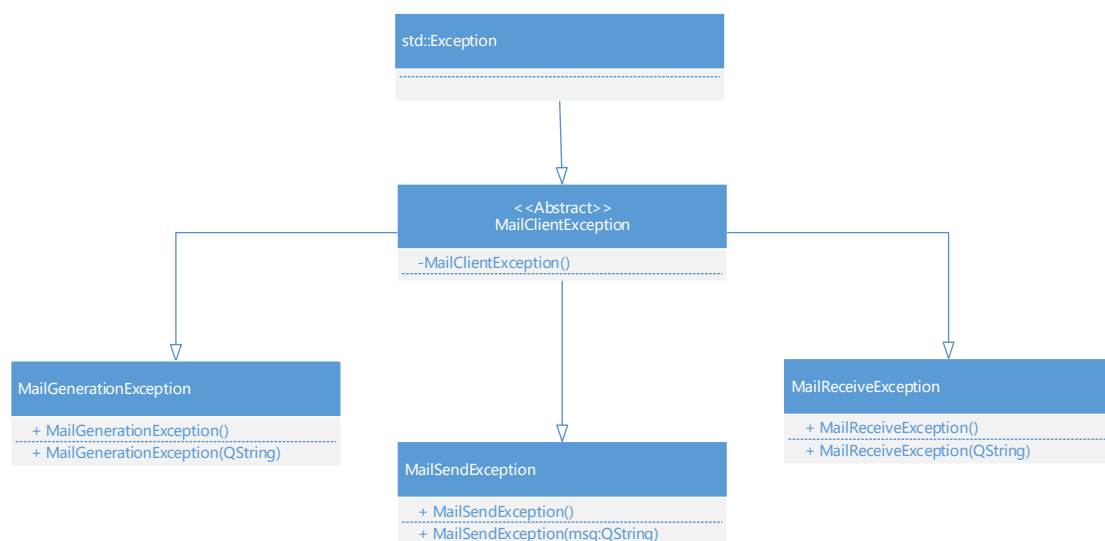
1. Model 部分主要实现:



2. 邮件收发:



3. 自定义异常:



五. 编码规范

- 私有成员(变量, 函数)名用_+小写字母开头, 比如 UserName 应为 _userName
- 若无特别注明, 私有变量(如果是继承过来的, 就重写基类的 get 函数)均要实现一个 get 函数, 函数名为 GetXXX(), 如 GetUserName()
- 公有成员(变量, 函数)名字用大写字母开头, 多个单词组合的每个单词首字母大写(即直接按照 UML 图中的命名)
- 自定义的类型(包括类, 结构, 枚举和 typedef 等)名首字母大写, 多个单词组合的每个单词首字母大写
- 如果不是特别注明的, 均需要实现一个公有的无参构造函数+拷贝构造函数(参数类型为 const T &)(除了继承自 QObject 的类之外)
- 为了方便查看, 先统一把所有类内函数的实现都放在类的定义(头文件)中, 最后再分开
- 尽量多地使用 const: 如果一个函数不修改调用该函数的对象的任何成员, 则把该函数设为 const; 如果一个参数或者局部变量不需要或不能被修改, 则把该变量或参数设为 const 和 const&(对于非基本类型的参数)
 - 抛出异常和处理异常: 所有用到网络的代码段都需要 catch 异常并作相应处理, 如果 catch 的地方无法处理则继续 throw; 注意调用 Poco 库函数可能会抛出的异常类型先看文档再写
- Account, MailBody 类需要能够在 QML 中访问, 必须要继承 QObject 类并且用 QT 的方式来声明成员, 具体参考 <http://doc.qt.io/qt-5/qtqml-cppintegration-definetypes.html#registering-an-instantiable-object-type>
- 容器主要用 QT 自带的容器:
 - Qlist: 存放 QObject 的派生类的 shared_ptr
 - QVector: 当做一般的数组用
- 如果需要用某个类的指针, 就在类中(public 部分)声明一个智能指针类型(主要是 QSharedPointer)以便使用, 比如

```
public:
    typedef QSharedPointer<MailBody>
    MAILBODY_PTR;
```

- 在 qml 中需要调用的 Model 类的函数必须在函数声明的前面加上宏 Q_INVOKABLE, 如 Q_INVOKABLE void appendRowFromQml(QObject *item);
- 参数表中的非基本类型(即非 int, double 等类型)如果是不需要改的, 最好设置为 const T & 类型, 尤其是所有的 QT 内置类型
- 公有成员(变量,函数)名字用小写字母开头, 多个单词组合的后面每个单词首字母大写

六. 组内分工

1. View(视图)(2 人):

- 登陆, 发送邮件: **管智超**
- 主界面, 查看邮件, 下载附件: **庞博**

2. Model(模型)(1 人): **马啸远**

3. 整体架构及接口设计(1 人): **庞博**

4. 邮件接受/发送功能实现(2 人)

SMTP/POP3 协议: **丁晨炜**

IMAP 协议: **庞博**

5. 软件开发文档(4 人): **丁晨炜, 马啸远, 管智超, 庞博**

七. 界面设计

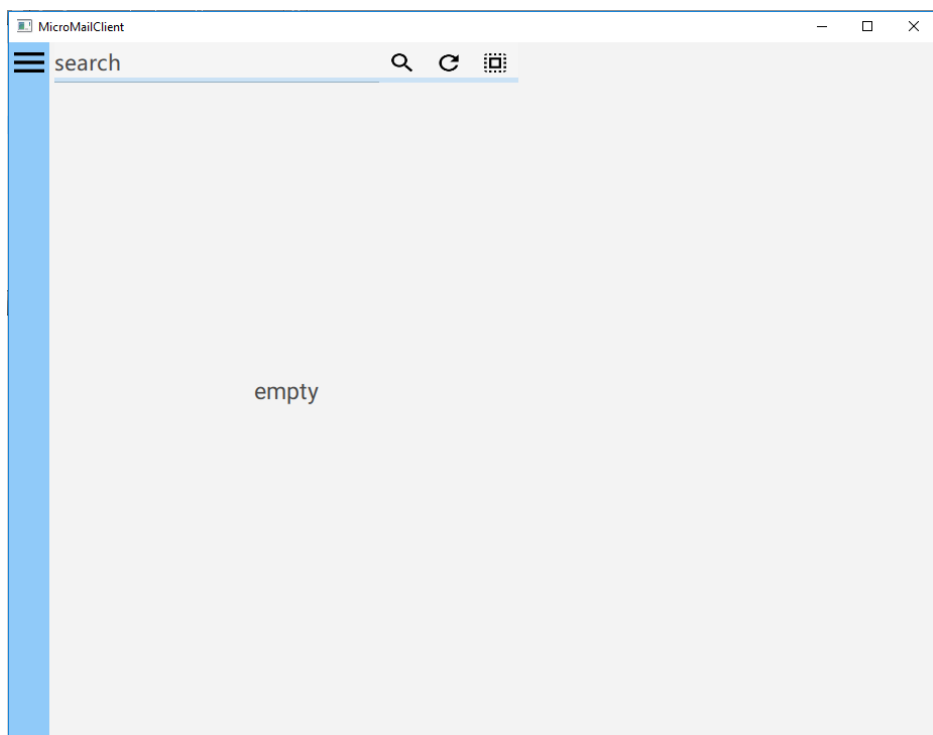


图 1 程序打开界面(未初始化账户)

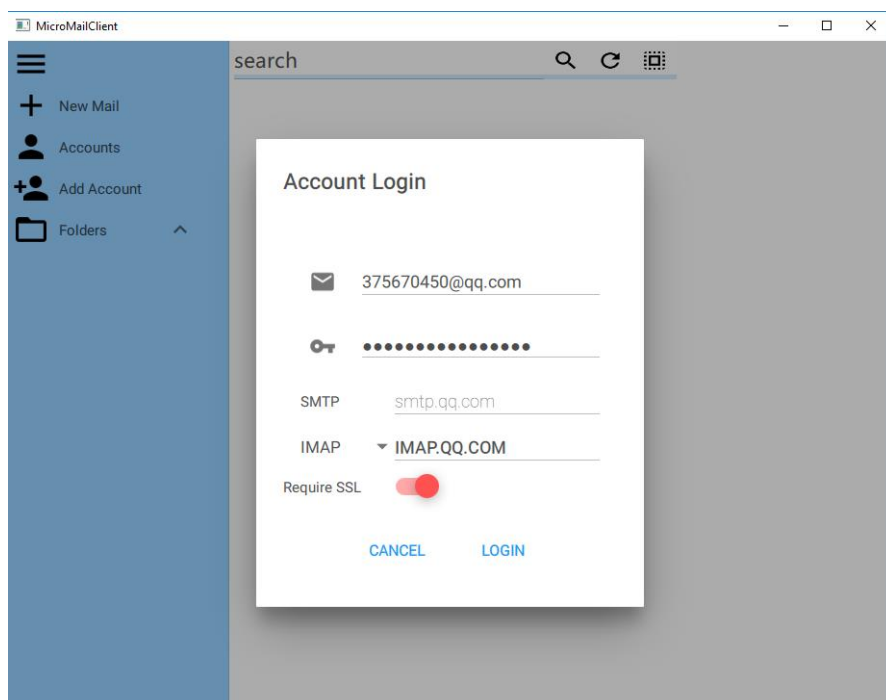


图 2 登陆账户

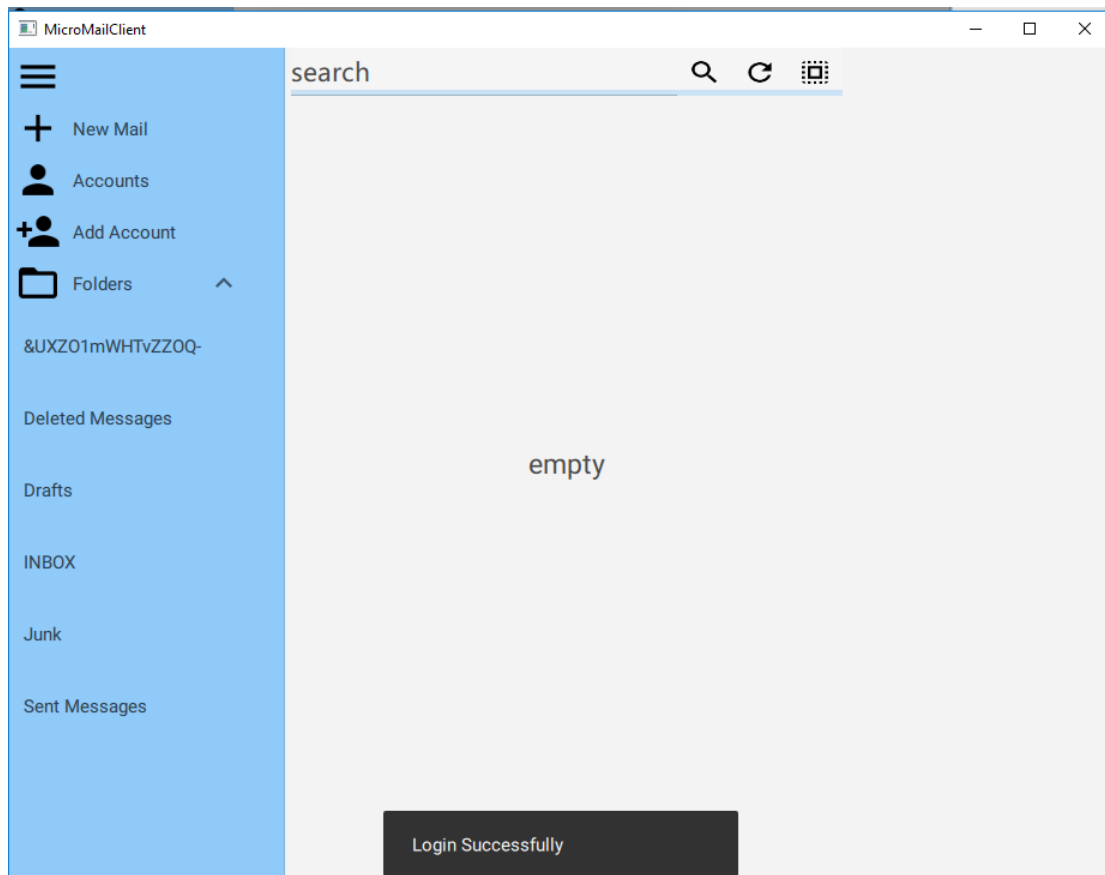


图 3 成功登录账户(及给出提示)

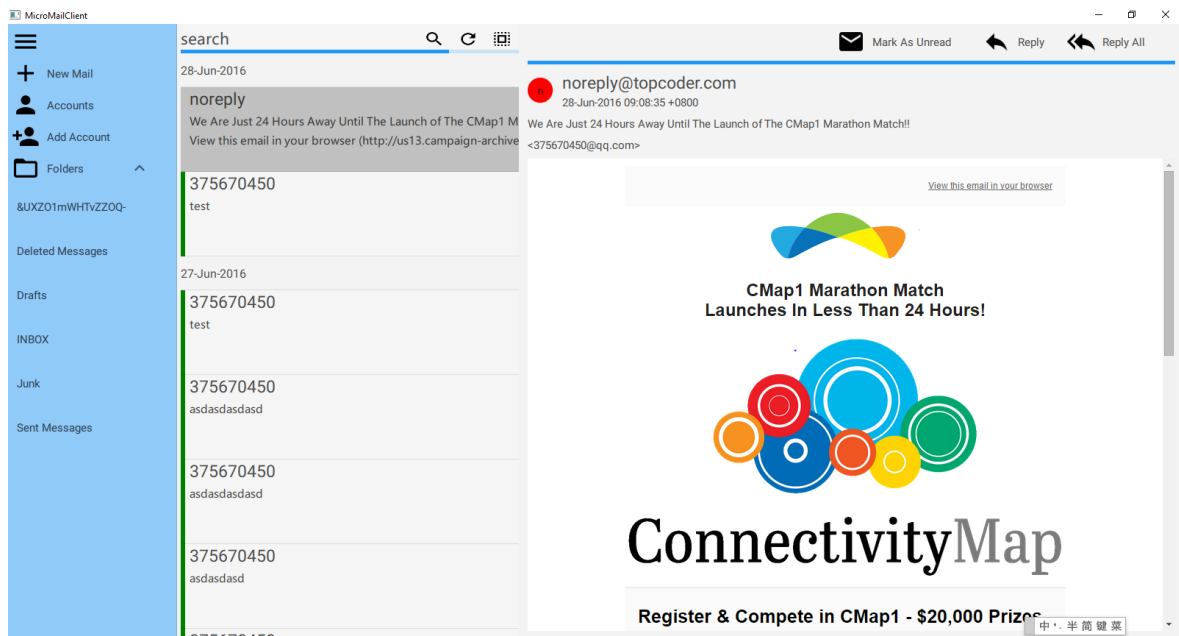


图 4 收件箱(正在接受邮件)

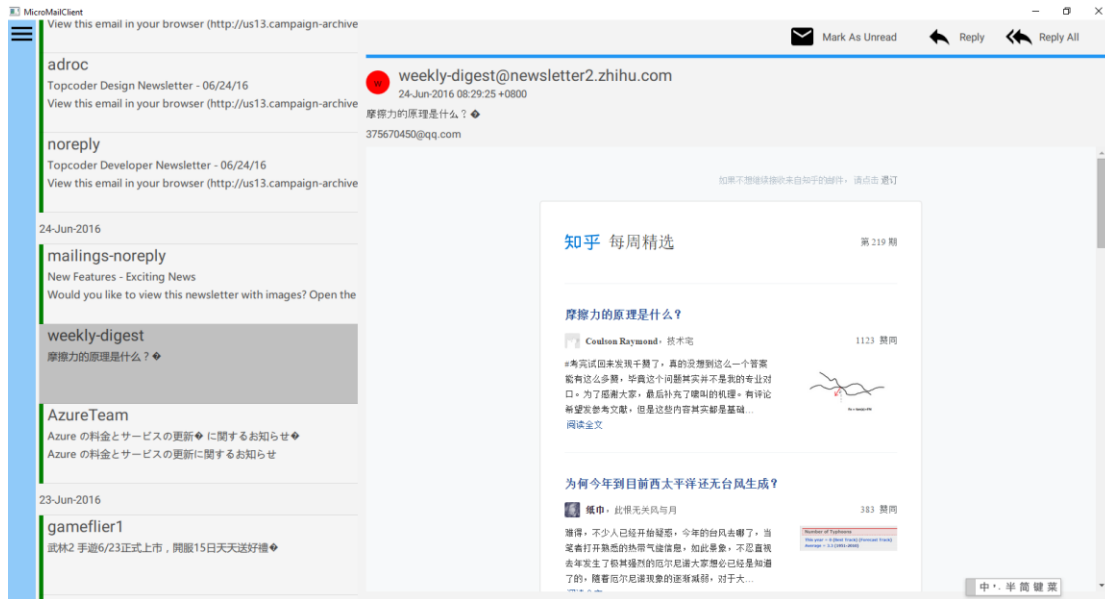


图 5 显示收件箱中的邮件(收回侧边栏)

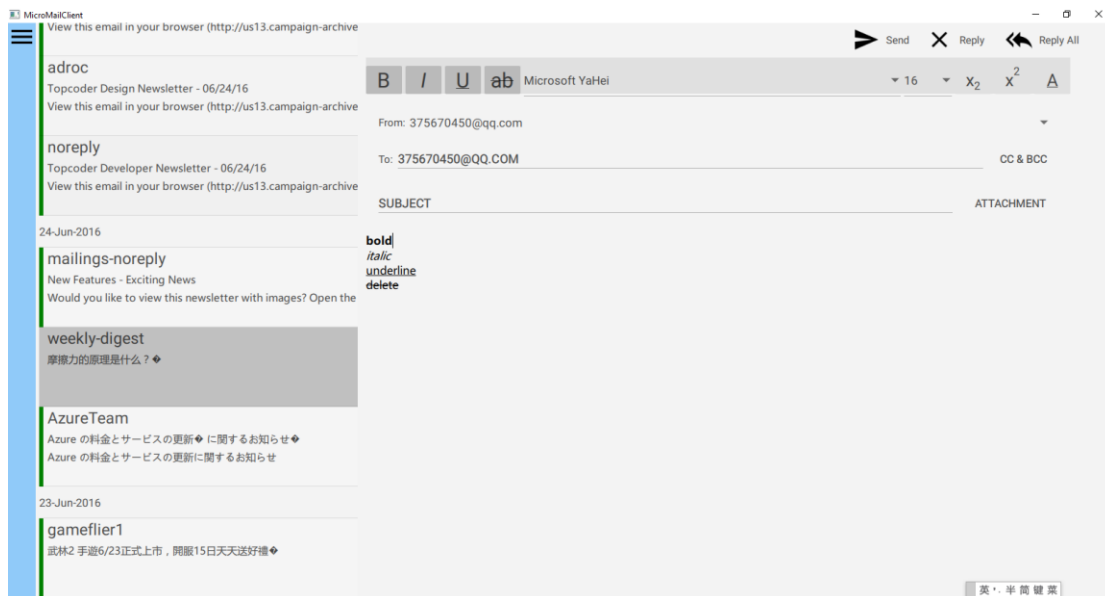


图 6 新写邮件及文本修饰

八. 接口详细定义

1. 头文件-MailBody.h

描述：

定义了一封邮件的构成方式，包括内容、附件、收发人等等信息。

继承：

继承自抽象类 QObject。

成员函数列表：

```
getSubject, setSubject, getContent, getHTMLContent, setHTMLContent,  
setContent, getSender, setSender, getDateTime, setDateTime,  
getIsread, setIsread, getRecipients, addRecipient
```

类型定义

```
MAILBODY_PTR
```

```
typedef QSharedPointer<MailBody> MAILBODY_PTR;
```

```
MAILBODY_PTR_QLIST
```

```
typedef QList<MAILBODY_PTR> MAILBODY_PTR_QLIST;
```

构造函数：

```
MailBody(QString _s = "subject", QString _c = "content", bool _r =  
false ){  
    _subject = _s;  
    _content = _c;  
    _isread = _r;  
}
```

传入主题、内容、是否读过的信息。

成员函数介绍：

函数名	函数类型及参数表	功能
getSubject	QString getSubject();	获取 Subject
setSubject	void setSubject(const QString&);	设置 Subject
getContent	QString getContent();	获取 Content
setContent	void setContent(const QString&);	设置 Content
getHTMLContent	QString getHTMLContent();	获取 HTMLContent
setHTMLContent	void setHTMLContent(const QString&);	设置 HTMLContent
getSender	QString getSender();	获取 Sender
setSender	void setSender(const QString&);	设置 Sender
getDateTime	QString getDateTime();	获取 DateTime
setDateTime	void setDateTime(const QString&);	设置 DateTime
getIsread	bool getIsread();	获取 isread
setIsread	void setIsread(bool);	设置 isread
getRecipients	QList<QString> getRecipients()	获取好友列表
addRecipient	void addRecipient(const QString&)	添加好友

2. 头文件：MailClientException.h

描述：

定义了邮件管理系统所有异常的基类。是一个抽象类。

继承：

继承自C++标准库的抽象类exception。

成员函数列表：

what

构造函数：

```
MailClientException();
```

无参构造函数。

```
MailClientException(const string& exc): exception(exc.c_str());
```

带有异常信息的构造函数。

```
MailClientException(const MailClientException& mce): exception(mce);
```

拷贝构造函数。

析构函数：

```
virtual ~MailClientException() noexcept = 0
```

成员函数介绍：

what

```
virtual const char* what() = 0;
```

获取异常信息的纯虚函数，将由继承这个类的其他类实现。

在 MailGenerationException.h 和 MailReceiveException.h 和 MailSendException.h 三个继承类中，都对 what 函数进行了重载并实现。

以上三个类由于和 MailClientException 相近，仅仅是重载了纯虚函数，这里不展开。

3. 头文件-Attachment.h

描述：

定义了邮件客户端与本地的连接。

继承：

继承自抽象类 Poco::Net::PartSource。

成员函数列表：

_satisfyFuzzy, _myCompare, ListContacts, GB2312ToUTF8, trimchar, decodeRFC2047, decoder, tokenize

类型定义

Check

```
struct check {  
    bool _satisfy;  
    int _degree;  
    QString _string;  
};
```

为模糊查找定义的一个结构体，是否满足模糊查找条件、满足的程度、字符串本身。

MediaType

```
typedef Poco::Net::MediaType MediaType;
```

定义了附件类型，直接使用 Poco 库提供的类型。

ProtocolType

```
enum ProtocolType{  
    SMTP, POP3, IMAP  
};
```

三种连接协议

成员函数介绍：

_satisfyFuzzy

```
static check _satisfyFuzzy(const QString& key, const QString& src);
```

私有函数，用来确定一个字符串是否满足给定字符串的模糊查找条件。

_myCompare

```
static bool _myCompare(const check & p, const check & q);
```

私有函数，比较两个满足模糊查找条件的字符串的优先级，主要用来排序。

GB2312ToUTF8

```
static std::string GB2312ToUTF8(const char* lpszGb32Text)
```

将 GB2312 转码为 UTF-8 编码

trimchar

```
template <class S=std::string>
static S trimchar(const S& str, const char ch)
```

将字符串 str 开头和末尾的 ch 字符删除

decodeRFC2047

```
static void decodeRFC2047 (const std::string & ins, std::string &
outs, const std::string & charset_to = "UTF-8" )
```

解码 RFC2047 协议规定格式的邮件

decoder

```
static std::string decoder (const std::string & _text, const
std::string& charset = "UTF-8")
```

检测 text 的中的编码信息并采取不同的解码方式解码后返回原文

tokenize

```
template < class T = std::string>
static void tokenize (const T& str, std::vector<T>& tokens,
const T& delimiters, const T& parenthesis, bool trimParenthesis =
false)
```

将字符串按照不同的字符或括号拆分

4. 头文件-Account.h

描述：

定义了一个账户，包含了用户信息，邮箱地址，协议等等信息。

继承：

继承自抽象类 QObject。

成员函数列表：

```
getUserName, setUsername, getPassword, setPassword, getMailHost,  
setMailHost, getSMTPHost, setSMTPHost, getPOP3Host, setPOP3Host,  
getIMAPHost, setIMAPHost ,  
  
getRequireSSL, setRequireSSL, getContacts
```

类型定义

ACCOUNT

```
typedef QSharedPointer<Account> ACCOUNT_PTR;
```

账户类指针类型

ACCOUNT_PTR_QLIST

```
typedef QList<ACCOUNT_PTR> ACCOUNT_PTR_QLIST;
```

一个存放账户的 list

构造函数：

```
Account() {}
```

无参构造函数

```
Account(QString un, QString pw) {  
    _userName = un;  
    _passWord = pw;  
}
```

传入用户姓名、密码。

成员函数介绍：

getUserName

```
QString getUserName();
```

获取 UserName

setUserName

```
void setUserName(const QString&);
```

设置 UserName

getPassWord

```
QString getPassWord();
```

获取 PassWord

setPassWord

```
void setPassWord(const QString&);
```

设置 PassWord

getMailHost

```
QString getMailHost();
```

获取 MailHost

setMailHost

```
void setMailHost(const QString&);
```

设置 MailHost

getSMTPHost

```
QString getSMTPHost();
```

获取 SMTPHost

setSMTPHost

```
void setSMTPHost(const QString&);
```

设置 SMTPHost

getPOP3Host

```
QString getPOP3Host();
```

获取 POP3Host

setPOP3Host

```
void setPOP3Host(const QString&);
```

设置 POP3Host

getIMAPHost

```
QString getIMAPHost();
```

获取 IMAPHost

setIMAPHost

```
void setIMAPHost(const QString&);
```

设置 IMAPHost

getRequireSSL

```
bool getRequireSSL() const;
```

获取是否需要 SSL 加密

setRequireSSL


```
void setRequireSSL(bool i);
```

设置是否需要 SSL 加密

getContacts

```
QStringList getContacts() const;
```

获取联系人列表

5. 头文件-Attachment.h

描述：

定义了邮件客户端与本地的连接。

继承：

继承自抽象类Poco::Net::PartSource。

成员函数列表：

```
getFileSize, setFileSize, getFileSize, setFileSize, getFileName,  
setFileName, getFilePath, setFilePath, getAccessCommand,  
setAccessCommand, getAttachmentType, Download, Upload
```

类型定义

```
downloadState
```

```
typedef enum{unDownloaded, isDownloading, isDownloaded}  
downloadState;
```

代表了附件的几种状态，没有下载、正在下载、已经下载。

构造函数：

```
Attachment(const std::string& mediaType):  
    _attachmentType(mediaType);
```

```
Attachment(const std::string& type, const std::string& subType):
```

```
_attachmentType(type, subType) ;
```

```
Attachment(const Utils::MediaType& mediaType):
```

```
_attachmentType(mediaType) ;
```

成员函数介绍：

stream

```
std::istream& stream() override;
```

获取一个文件流

getFileSize

```
QString getFileSize();
```

获取 FileSize

getFileSize

```
size_t getFileSize() const;
```

获取附件的大小

setFileSize

```
void setFileSize(size_t i);
```

设置附件大小

setFileSize

```
void setFileSize(const QString&);
```

设置 FileSize

getFileName

```
QString getFileName();
```

获取 FileName

setFileName

```
void setFileName(const QString&);
```

设置 FileName

getFilePath

```
QString getFilePath();
```

获取 FilePath

setFilePath

```
void setFilePath(const QString&);
```

设置 FilePath

getAccessCommand

```
QString getAccessCommand();
```

获取 AccessCommand

setAccessCommand

```
void setAccessCommand(const QString&);
```

设置 AccessCommand

Download

```
bool Download(const QByteArray&);
```

下载一个附件，传入的是一个字节串。

Upload

```
bool Upload(QByteArray& data);
```

上传一个附件，将要传的数据存放在 data 里

6. 头文件-MailListModel.h

描述：

最主要的 Model 类(List)，视图仅需要调用该类提供的函数来执行后端任务。

继承：

继承自抽象类QAbstractListModel

成员函数列表：

UpdateMailBox, ToList, Get, BuildMailList, AppendRow, Delete, Send, GetSubject, GetContent, GetSender, GetDateTime, GetRecipients, GetAttachment

构造函数：

```
MailListModel(QObject * parent = 0)
```

成员函数介绍：

```
QVariant(MailBody) Get(int)
```

获取MailList中的一封邮件，参数为邮件在List中的下标

```
bool BuildMailList(int)
```

获取一个文件夹中的所有邮件并放入List中，参数为文件夹下标

```
void AppendRow(MAILBODY_PTR)
```

```
void AppendRows(MAILBODY_PTR_QLIST)
```

`bool Delete(int)`

`bool Delete(QList<int>)`

上述四个函数增删List

`bool Send(MailBody)`

发送一封邮件, 参数即为邮件体

`QString GetSubject(int)`

获取指定邮件的标题

`QString GetContent(int)`

获取指定邮件的内容

`QString GetSender(int)`

获取指定邮件的发件人

`QString GetDateTime(int)`

获取指定邮件的发件时间

`QString GetRecipients(int)`

获取指定邮件的收件人

7. 头文件-SMTPClient.h

描述：

利用 SMTP 协议(SMTP, [RFC 2821](#))实现发送邮件 (包括附件) 的功能。

继承：

继承自抽象类 SendMailClient。

成员函数列表：

`login` , `sendMailBodies` , `setTimeout` , `getTimeout`。

类型定义

`SESSION_PTR`

```
typedef QSharedPointer<Poco::Net::SMTPClientSession> SESSION_PTR。
```

构造函数：

```
SMTPClient(  
    QString host ,  
    QString port = QString::number (Poco::Net::SMTPClientSession::SMTP_PORT)  
);
```

构造一个利用 SMTP 协议实现的发件客户端对象。

析构函数：

```
~SMTPClient();
```

销毁该发件客户端对象。

成员函数介绍：

login

```
bool login(QString _user, QString _passwd,bool requireSSL) override
```

由用户指定的账号和密码实现登录功能，返回登录是否成功（True or False）。

sendMailBodies

```
bool sendMailBodies(const MAILBODY_PTR & MailBody) override
```

核心函数，传入要发送的邮件内容的指针为参数，构造一封符合标准格式的邮件，并发送。

setTimeout

```
void setTimeout(int val) override
```

设置响应超时时间。

getTimeout

```
int getTimeout () override
```

获取响应超时时间。

8. 头文件-POP3Client.h

描述：

利用 POP3 协议(POP3, [RFC 1939](#))实现接受邮件的功能。

继承：

继承自抽象类 ReceiveMailClient。

成员函数列表：

```
login , getFolders , selectFolder , getMailBodies , DeleteMail ,  
setTimeout , getTimeout
```

类型定义：

```
SESSION_PTR
```

```
typedef QSharedPointer<Poco::Net::POP3ClientSession> SESSION_PTR。
```

构造函数：

```
POP3Client(  
    QString host,  
    QString port = QString::number  
    (Poco::Net::POP3ClientSession::POP3_PORT)  
);
```

构造一个利用 POP3 协议实现的收件客户端对象。

析构函数：

```
~ POP3Client ();
```

销毁该收件客户端对象。

成员函数介绍：

login

```
bool login(QString _user, QString _passwd,bool requireSSL) override
```

由用户指定的账号和密码实现登录功能，返回登录是否成功（True or False）。

getFolders

```
int getFolders(QList<QString> & _folders) override
```

获取收件文件夹（形如收件箱、发件箱、垃圾邮件等文件夹）。

selectFolder

```
int selectFolder(const QString & folder) override
```

选中一个文件夹打开。

getMailBodies

```
int getMailBodies(QList<MAILBODY_PTR> & result,int count)
```

获取收到的邮件主体，得到邮件标题、内容、发件人

DeleteMail

```
int DeleteMail (const QList<int> & ids)override
```

删除一封邮件。

setTimeout

```
void setTimeout(int val) override
```

设置响应超时时间。

getTimeout

```
int getTimeout () override
```

获取响应超时时间。