

学 号:	0122015710114
------	---------------

武汉理工大学

课 程 设 计

题 目	简单电子邮件客户端的设计与实现
学 院	信息工程学院
专 业	信息工程
班 级	信息 2001
姓 名	胡 珊
指导教师	阮幼林

2023 年 12 月 21 日

课程设计任务书

学生姓名： 胡 珊 专业班级： 信息 2001

指导教师： 阮幼林 工作单位： 信息工程学院

题 目： 简单电子邮件客户端的设计与实现

初始条件：

1、JAVA 或 C++;

要求完成的主要任务：（包括课程设计工作量及其技术要求，以及说明书撰写等具体要求）

- 1、设计并实现一个简单的电子邮件客户端的设计与实现；
- 2、能完成电子邮件的收取，发送；
- 3、撰写新邮件；
- 4、按《武汉理工大学课程设计工作规范》要求撰写课程设计说明书。

时间安排：

2023 年 12 月 16 号—— 17 号

18 号—— 20 号

2023 年 12 月 21 号

22 号

选题及调研

系统设计制作

系统调试

撰写课程设说明书及答辩

指导教师签名：

年 月 日

系主任（或责任教师）签名：

年 月 日

简单电子邮件客户端的设计与实现

摘 要

本课程设计以设计与实现简单电子邮件客户端为主题，旨在探索电子邮件客户端的基本功能与界面设计。通过使用 Java 编程语言及相关的 Swing 和 JavaMail API，成功实现了电子邮件客户端的基本功能，包括登录邮箱、发送邮件、获取邮件列表以及显示邮件内容等。报告详细介绍了该电子邮件客户端的设计思路、技术选型、界面设计以及功能实现过程。在实现过程中，对技术难点进行了分析，并提出了可能的改进方向，展望了未来对该客户端的进一步优化与扩展。

关键词：电子邮件客户端；Java Swing；JavaMail API

Abstract

This course design focuses on the design and implementation of a simple email client, aiming to explore the basic functionalities and interface design of an email client. By utilizing Java programming language along with Swing and JavaMail API, the basic functionalities of the email client, including login, fetching mail listings, and displaying email content, were successfully implemented. The report provides a detailed insight into the design concepts, technological choices, interface design, and the process of functionality implementation of the email client. Throughout the implementation, technical challenges were analyzed, potential improvements were proposed, and future directions for optimizing and expanding the client were discussed.

Key Words: Email client; Java Swing; JavaMail API

目 录

第 1 章 绪论.....	1
1.1 背景介绍.....	1
1.2 预备知识.....	1
1.2.1 简单邮件传输协议 SMTP	1
1.2.1 邮局通讯协议 POP3	1
1.2.1 交互式邮件存取协议 IMAP.....	1
第 2 章 设计目标与需求	2
2.1 登录注册功能	2
2.2 邮件发送功能	2
2.3 邮件收取功能	2
第 3 章 技术选型与工具	2
3.1 Java Mail	2
3.1.1 JavaMail API 核心类	2
3.1.2 JavaMail 环境配置	2
3.2 Java Swing.....	3
3.3 IDEA 开发工具	3
第 4 章 设计与实现	4
4.1 登录注册功能与界面	4
4.2 邮件发送功能与界面	5
4.3 邮件收取功能与界面	5
第 5 章 测试与结果分析	6
5.1 注册登录.....	7
5.2 收取邮件.....	8
5.3 发送邮件.....	8
第 6 章 总结与展望	9
参考文献.....	10

第 1 章 绪论

1.1 背景知识

电子邮件作为当今信息传递和沟通的重要方式，在个人和商业领域都扮演着不可或缺的角色。通过电子邮件，人们可以快速高效地发送和接收消息、文件和数据，促进了全球范围内的信息交流和业务合作。电子邮件客户端作为连接用户和邮件服务器的桥梁，承载了许多功能，如账户管理、邮件筛选、信息展示等，影响着用户体验和工作效率。

1.2 预备知识

1.2.1 SMTP 协议介绍

SMTP（Simple Mail Transfer Protocol）是用于在网络中传输电子邮件的标准协议。它定义了邮件的发送方式，通过 SMTP 服务器将邮件从发件人发送到收件人的邮箱服务器。SMTP 协议属于 TCP/IP 协议簇，它是一组用于从源地址到目的地址传输邮件的规范，通过它来控制邮件的中转方式，帮助每台计算机在发送或中转信件时找到下一个目的地。SMTP 是一个“推”的协议，它不允许根据需要从远程服务器上“拉”来消息。SMTP 服务器就是遵循 SMTP 协议的发送邮件服务器。

1.2.2 POP3 协议介绍

POP3（Post Office Protocol version 3）是接收电子邮件的协议，它允许用户从邮件服务器上下载电子邮件到本地计算机，并在下载后删除服务器上的邮件。POP3 是 TCP/IP 协议族中的一员（默认端口是 110）。本协议主要用于支持使用客户端远程管理在服务器上的电子邮件。它规定怎样将个人计算机连接到 Internet 的邮件服务器和下载电子邮件的电子协议。它是因特网电子邮件的第一个离线协议标准，POP3 允许用户从服务器上把邮件存储到本地主机（即自己的计算机）上，同时删除保存在邮件服务器上的邮件，而 POP3 服务器则是遵循 POP3 协议的接收邮件服务器，用来接收电子邮件的。

1.2.3 IMAP 协议介绍

IMAP（Internet Message Access Protocol）是与 POP3 类似的邮件访问协议，但更加强大和灵活。IMAP 允许用户在邮件服务器上管理邮件，包括创建文件夹、标记邮件、在线阅读邮件等。它是一个应用层协议（端口是 143）。用来从本地邮件客户端访问远程服务器上的邮件。它是跟 POP3 类似邮件访问标准协议之一。不同的是，开启了 IMAP 后，在电子邮件客户端收取的邮件仍然保留在服务器上，同时在客户端上的操作都会反馈到服务器上，如：删除邮件，标记已读等，服务器上的邮件也会做相应的动作。所以无论从浏览器登录邮箱或者客户端软件登录邮箱，看到的邮件以及状态都是一致的。

第 2 章 设计目标与需求

2.1 登录注册功能

实现用户可以安全、便捷地登录和注册邮箱账户。提供用户友好的登录界面，包括输入用户名、密码等信息。实现注册新用户的功能，包括输入必要的注册信息并完成账户注册过程。使用安全的身份验证机制来保护用户账户，如加密存储密码、防止恶意登录尝试等。

2.2 邮件发送功能

邮件发送界面包括一封完整邮件的常用字段表单输入，如收件人、主题、正文等。页面底部包含发送和附件两个按钮，发送按钮用于发送邮件，附件按钮用于上传附件到服务器。

2.3 邮件收取功能

接收邮件为系统的主要模块，用户经过注册后，登录系统。通过系统接收未读邮件，并根据发件人昵称、发件人邮箱、主题、收件时间、邮件正文内容列表显示。

第 3 章 技术选型与工具

3.1 Java Mail 简介

Email 是互联网最重要和最广泛的应用之一，因此 E-mail 通信在 Java 服务器编程中占有非常重要的地位。对一般的电子邮件使用者来说，Email 是非常简单地执行邮件传输，但是对于编程人员来讲 Email 的打包、传输、发送和接受是很复杂的。这要涉及到很多协议，主要有发送电子邮件的 SMTP 协议、用来接收电子邮件的 POP3 协议和消息访问协议 IMAP 以及多用途网络邮件扩充协议 MIME。JavaMail 是 SUN 发布的用来处理这些问题的统一的 API，使用它编程人员就可以不用自己去处理那些复杂的协议了。

电子邮件的接收和发送功能的实现涉及到一个服务器和客户端的问题。要使用电子邮件服务就必须有电子邮件服务器，JavaMail API 主要是封装了电子邮件的基本协议，它可以使用在客户端也可以使用在服务器端，本文以邮件的客户端为主。

JavaMail API 的开发是 SUN 为 Java 开发者提供公用 API 框架的持续努力的良好例证。提倡公用框架，反对受限于供应商的解决方案，充分预示着一个日益开放的开发环境的建立。

3.1.1 JavaMail API 核心类

- `Javax.mail.Session:Session` 类定义了一个基本邮件会话 (session)，是 Java Mail API 最高层入口类。所有其它类都是经由这个 session 才得以生效。Session 对象用 `Java.util.Properties` 对象获取信息，如邮件服务器、用户名、密码及整个应用程序中共享的其它信息。

- **Javax.mail.Message**:一旦获得 **Session** 对象, 就可以继续创建要发送的消息。这由 **Message** 类来完成。因为 **Message** 是个抽象类, 必需用一个子类, 多数情况下为 **Javax.mail.internet.MimeMessage**。**MimeMessage** 是个能理解 **MIME** 类型和头的电子邮件消息, 正如不同 **RFC** 中所定义的。虽然在某些头部域非 **ASCII** 字符也能被译码, 但 **Message** 头只能被限制为用 **US-ASCII** 字符。

- **Javax.mail.Address**:一旦您创建了 **Session** 和 **Message**, 并将内容填入消息后, 就可以用 **Address** 确定信件地址了。和 **Message** 一样, **Address** 也是个抽象类。您用的是 **Javax.mail.internet.InternetAddress** 类。

- **Javax.mail.Authenticator**:与 **Java.net** 类一样, **JavaMailAPI** 也可以利用 **Authenticator** 通过用户名和密码访问受保护的资源。对于 **JavaMail API** 来说, 这些资源就是邮件服务器。**JavaMail Authenticator** 在 **Javax.mail** 包中, 而且它和 **Java.net** 中同名的类 **Authenticator** 不同。两者并不共享同一个 **Authenticator**, 因为 **JavaMail API** 用于 **Java 1.1**, 它没有 **Java.net** 类别。要使用 **Authenticator**, 先创建一个抽象类的子类, 并从 **getPasswordAuthentication()** 方法中返回 **PasswordAuthentication** 实例。创建完成后, 必需向 **session** 注册 **Authenticator**。然后, 在需要认证的时候, 就会通知 **Authenticator**。您可以弹出窗口, 也可以从配置文件中 (虽然没有加密是不安全的) 读取用户名和密码, 将它们作为 **PasswordAuthentication** 对象返回给调用程序。

- **Javax.mail.Transport**:消息发送的最后一部分是使用 **Transport** 类。这个类用协议指定的语言发送消息 (通常是 **SMTP**)。它是抽象类, 它的工作方式与 **Session** 有些类似。仅调用静态 **send()** 方法, 就能使用类的缺省版本:**Transport.send(message)**;或者, 读者也可以从针对自己的协议的会话中获得一个特定的实例, 传递用户名和密码 (如果不必要就不传), 发送消息, 然后关闭连接。

- **Javax.mail.Store**:**Store** 类实现特定邮件协议上的读、写、监视、查找等操作。通过 **Javax.mail.Store** 类可以访问 **Javax.mail.Folder** 类。

- **Javax.mail.Folder**:**Folder** 类用于分级组织邮件, 并提供照 **Javax.mail.Message** 格式访问 email 的能力。

3.1.2 JavaMail 环境配置

首先, 需要安装 **JavaMail API**。**JavaMail API** 最新版本是 **1.3**, 要使用 **JavaMail 1.3 API**, 请下载 **JavaMail 1.3** 压缩包, 解开 **Javamail-1__3.zip** 文件, 并将 **mail.jar** 文件添加到 **lib** 中。

其次是 **JavaBeans Activation Framework** 的安装。**JavaMail API** 的所有版本都需要 **JavaBeansActivation Framework** 来支持任意数据块的输入及相应处理。功能似乎不多, 但目前许多浏览器和邮件工具中都能找到这种基本的 **MIME** 型支持。下载完后, 解开 **jaf1__0__2.zip** 文件, 并将 **activation.jar** 文件添加到 **lib** 中。

3.2 Java Swing 简介

Java Swing 技术主要是用于构建跨平台桌面窗口程序，它是在 AWT(abstract window toolkit) 基础上采用 Java 语言编写的一种图形组件库技术。Java Swing 组件除了是轻量级的，即在运行时不需要产生相应的“对等 (Peer)”组件，实际由自身外观类支持，没有本地代码，不依赖操作系统，而且它的外观是可插拔的(plug-gable look-and-feel)。Swing 组件包括各种容器和基本组件，容器主要用来容纳和布置基本组件，也可以包含相应的容器。常见的容器有 JFrame、JDialog、JPanel 等；常见的基本组件包括按钮 JButton、标签 JLabel、单选按钮 JRadioButton。组件之间的关系如图所示。

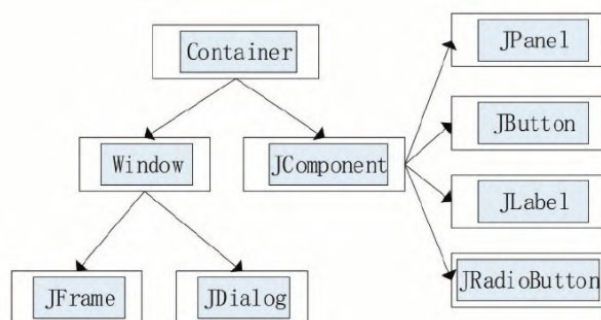


图 3.1 组件关系图

3.3 IDEA 开发工具

IntelliJ IDEA 的每个方面都专门设计用于最大限度地提高开发人员的工强大的静态代码分析和符合人体工程学的设计使开发不仅具有高效性，而且还具有令人愉悦的体验。在 IntelliJ IDEA 为您的源代码编制索引之后，它通过在每个上下文中提供相关建议提供了快速而智能的体验：即时和巧妙的代码完成，动态代码分析和可靠的重构工具。

第 4 章 设计与实现

4.1 登录注册功能

登录功能：用户输入邮箱地址和密码。完成身份验证，验证用户的合法性。提供友好的登录成功或失败提示。注册功能：用户输入注册所需信息，如邮箱地址、用户名、密码等。验证信息的合法性和唯一性。完成注册过程并创建新用户。

代码结构如下图所示，其中 `User` 类作为用户实体类，包含用户账户、密码成员属性，以及设置或获取用户输入的 `Set` 和 `get` 方法。`LoginUI` 类和 `RegistUI` 类为使用 `Java Swing` 构建的用户登录和注册的界面，`LoginInput` 类和 `RegistInput` 类主要负责接收用户的登录和注册时的账号和密码输入。

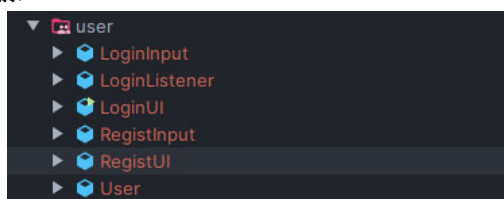


图 4.1 User 包代码结构



图 4.2 注册界面设计



图 4.3 登录界面设计

4.2 邮件发送功能

编写邮件：提供界面以便用户填写收件人、主题、正文等信息。支持添加附件常见功能。发送邮件用户点击发送按钮后，将填写好的邮件发送给指定的收件人。

界面设计：邮件编写界面输入框：收件人、主题、正文、附件等。按钮：发送、取消发送等。发送成功/失败提示，显示发送成功或失败的提示信息。

使用 Swing 创建用户界面元素，包括文本框、按钮、标签等。使用 JavaMail API 将用户编写的邮件发送到指定的收件人邮箱。分别输入收件人 Email 地址，邮件主题，邮件正文内容，以及选择是否添加附件。

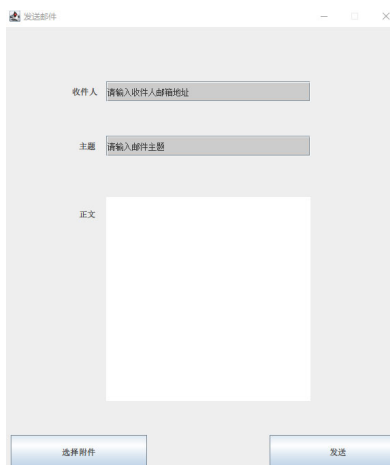


图 4.4 发送邮件界面设计

4.3 邮件收取功能

功能设计：获取邮件列表连接邮箱服务器并认证用户身份。从服务器获取邮件列表，并提取发件人、主题、日期等信息。查看邮件内容，用户选择特定邮件并请求查看其详细内容。解析邮件内容并在界面显示发件人、主题、内容等信息。

界面设计：邮件列表界面列表显示：发件人、主题、日期等邮件基本信息。按钮或菜单：选择邮件、刷新列表等功能。邮件内容界面显示：发件人、主题、正文等详细信息。按钮或选项：返回邮件列表、下载附件等功能。

实现方法：使用 JavaMail API 连接邮箱服务器，使用 IMAP 或 POP3 协议获取邮件信息。解析邮件内容并将其展示在 Swing 界面上。

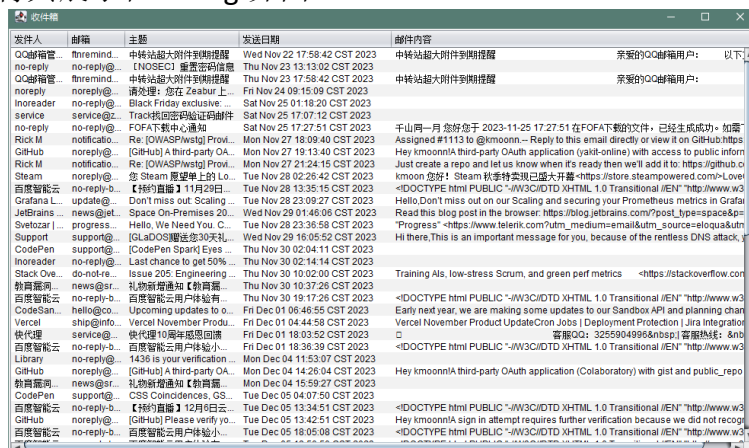


图 4.5 收取邮件界面设计

第 5 章 测试与结果分析

5.1 登录注册

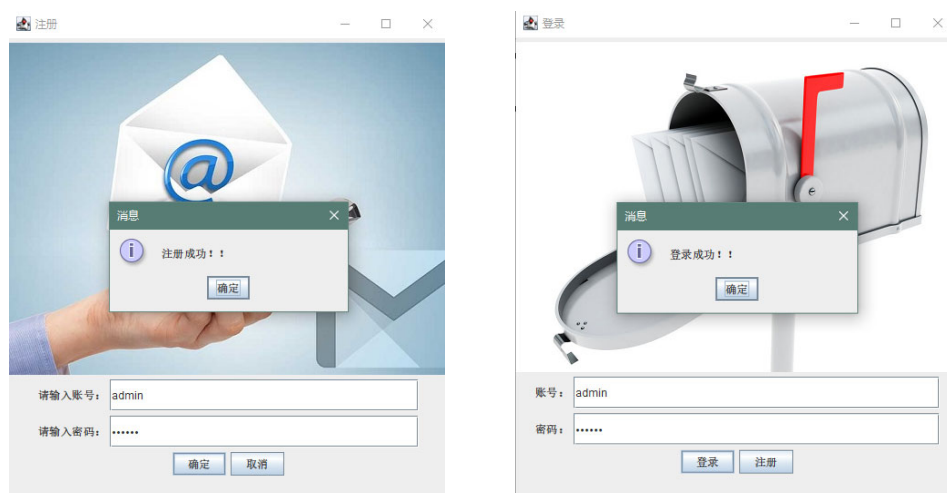


图 5.1 注册与登录



图 5.2 进入首页界面

5.2 发送邮件



图 5.3 发送邮件

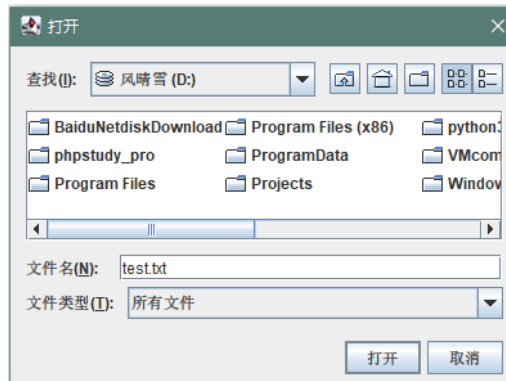


图 5.4 选择附件



图 5.5 发送成功并接收

5.3 收取邮件

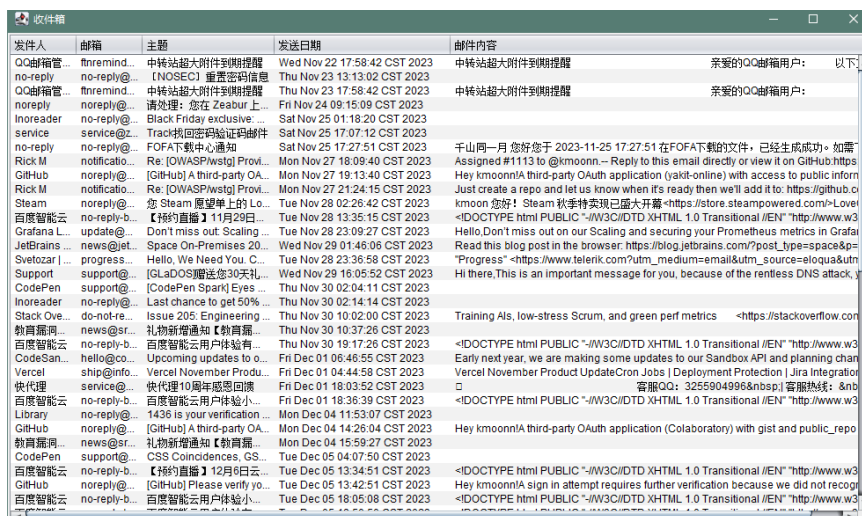


图 5.6 收取邮件

第 6 章 总结与展望

电子邮件作为当今信息传递和沟通的重要方式，在个人和商业领域都扮演着不可或缺的角色。通过电子邮件，人们可以快速高效地发送和接收消息、文件和数据，促进了全球范围内的信息交流和业务合作。电子邮件客户端作为连接用户和邮件服务器的桥梁，承载了许多功能，如账户管理、邮件筛选、信息展示等，影响着用户体验和工作效率。

电子邮件客户端选取了 IDEA 开发平台，利用 Swing 可视化窗体开发工具进行开发，使得图形界面设计更加绚丽，图形界面的开发过程更加简洁。采用了 SMTP 协议和 IMAP 协议，实现电子邮件的接收和发送。并且运用 Java Mail API，有效避免了编写客户端与邮件服务器交互的详细过程。在此基础上，还实现了客户端的注册与登录功能，相信随着科技的发展，网络的广泛运用，邮件客户端的设计与实现将愈加完善。

参考文献

- [1] 刘娅,张鹏博. 基于 SMTP 和 POP3 的电子邮件客户端[J]. 计算机光盘软件与应用, 2012(08):83-84.
- [2] 于新波. 基于 Web 的在线邮件系统的设计与研究[D]. 吉林大学, 2011.
- [3] 钱诚慎. SMTP 电子邮件客户端与服务器的设计与实现[D]. 大连理工大学, 2007.
- [4] 杨宁. 使用 Java Mail API 实现电子邮件客户端[J]. 晋东南师范专科学校学报, 2004(02):38-40.
- [5] 张翔. 基于 Java 的邮件收发系统设计与实现[D]. 吉林大学, 2018.
- [6] 宋佳颖. 基于 Java 的邮件接收系统分析[J]. 无线互联科技, 2015(24):12-13.

本科生课程设计成绩评定表

[illegible]