****

Southwest University of Science and Technology

本科毕业设计（论文）

题目名称：邮件收发软件的开发与实现

|  |  |
| --- | --- |
| 学院名称 |  |
| 专业名称 |  |
| 学生姓名 |  |
| 学号 |  |
| 指导教师 |  |

二〇二二年六月

**西南科技大学**

**本科毕业设计（论文）学术诚信声明**

本人郑重声明：所呈交的毕业设计（论文），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日期： 年 月 日

**西南科技大学**

**本科毕业设计（论文）版权使用授权书**

本毕业设计（论文）作者同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权西南科技大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业设计（论文）。

**保密**□，在 年解密后适用本授权书。

本论文属于

**不保密**□。

（请在以上方框内打“**√**”）

作者签名： 指导教师签名：

日期： 年 月 日 日期： 年 月 日

基于JavaMail的邮件收发系统

摘要：在当今社会中，电子邮件已经成为了日常通信的关键工具，其使用人数迅速增长。本文的目的是深入探讨基于Javamail的邮件收发系统的开发过程和实现技术。同时，我们评估了各种在国内外广泛使用的电子邮件系统，并对其优点和缺点进行了全面的比较和总结。

我们对Javamail技术进行了深度探讨，并构建了一个基于Javamail的邮件收发系统。这个系统凭借SMTP和POP协议实现底层开发，并使用JavaMail API来提供与协议无关的邮件接收和发送接口。在此上下文中，本研究详尽地阐述了SMTP（负责邮件发送）的角色。

本文内容分为五个主要部分。首先，我们介绍了Javamail邮件系统的重要性及其当前状况。接下来，我们更详细地讨论了相关协议和Javamail API的特性。第三部分主要阐述了系统的设计理念以及可行性分析。在第四部分，我们详细论述了Javamail邮件接受与发送系统的实现过程。最后，我们通过仔细的测试，验证了系统的可用性和稳定性。

关键词：JavaMail SMTP POP3 MIME 邮件收发软件

**MAIL SYSTEM BASED ON JAVAMAIL**

**Abstract**：In today's society, email has become a key tool for daily communication, and its usage is growing rapidly. The purpose of this article is to delve into the development process and implementation technology of a Javamail-based mail sending and receiving system. At the same time, we evaluate various email systems widely used at home and abroad, and conduct a comprehensive comparison and summary of their advantages and disadvantages.

We conducted an in-depth discussion of Javamail technology and built a Javamail-based mail sending and receiving system. This system implements low-level development with SMTP and POP protocols, and uses the JavaMail API to provide protocol-independent mail receiving and sending interfaces. In this context, this study elaborates on the role of SMTP (responsible for mail sending).

The content of this article is divided into five main parts. First, we introduce the importance of the Javamail mail system and its current status. Next, we discuss the relevant protocols and the characteristics of the Javamail API in more detail. The third part mainly expounds the design concept and feasibility analysis of the system. In the fourth part, we discuss in detail the implementation process of the Javamail mail receiving and sending system. Finally, we verify the usability and stability of the system through careful testing.

**Key words:** JavaMail SMTP POP3 MIME Mail Sending And Receiving Software

目 录

[第1章 绪论 6](#_Toc144287036)

[1.1 研究背景 6](#_Toc144287037)

[1.2 研究意义 6](#_Toc144287038)

[1.3 国内外研究现状 7](#_Toc144287039)

[1.3.1 国外研究现状 7](#_Toc144287040)

[1.3.2 国内研究现状 8](#_Toc144287041)

[1.4 研究内容及目标 9](#_Toc144287042)

[1.4.1 研究内容 9](#_Toc144287043)

[1.4.2 研究目标 9](#_Toc144287044)

[1.5 论文组织结构 9](#_Toc144287045)

[1.6 本章小节 10](#_Toc144287046)

[第2章 相关理论基础和技术简介 11](#_Toc144287047)

[2.1 相关邮件协议 11](#_Toc144287048)

[2.2 JavaMail API 11](#_Toc144287049)

[2.2.1 JavaMail API简介 11](#_Toc144287050)

[2.2.2 JavaMail API核心类 12](#_Toc144287051)

[2.2.3 JavaMail API的应用 15](#_Toc144287052)

[2.3 本章小节 16](#_Toc144287053)

[第3章 JavaMail系统分析 17](#_Toc144287054)

[3.1 系统需求分析 17](#_Toc144287055)

[3.1.1 功能性需求分析 17](#_Toc144287056)

[3.1.2 非功能性需求分析 17](#_Toc144287057)

[3.2 可行性研究 18](#_Toc144287058)

[3.2.1 技术可行性 18](#_Toc144287059)

[3.2.2 法律可行性 18](#_Toc144287060)

[3.3 本章小结 19](#_Toc144287061)

[第4章 JavaMail系统总体设计 20](#_Toc144287062)

[4.1 JavaMail系统结构设计 20](#_Toc144287063)

[4.2 功能模块设计 20](#_Toc144287064)

[4.2.1 发送邮件模块设计 21](#_Toc144287065)

[4.2.2 接收邮件模块设计 21](#_Toc144287066)

[4.2.3 附件处理模块设计 22](#_Toc144287067)

[4.2.4 界面设计 22](#_Toc144287068)

[4.3 数据库设计 22](#_Toc144287069)

[4.3.1 用户信息表设计 23](#_Toc144287070)

[4.3.2 邮件信息表设计 23](#_Toc144287071)

[4.4 本章小结 24](#_Toc144287072)

[第5章 JavaMail系统实现 25](#_Toc144287073)

[5.1 JavaMail系统实现 25](#_Toc144287074)

[5.1.1 JavaMail的实现 25](#_Toc144287075)

[5.1.2 SMTP协议的实现 25](#_Toc144287076)

[5.1.3 POP3协议实现 26](#_Toc144287077)

[5.1.4 邮件收发模块 27](#_Toc144287078)

[5.1.5 邮件删除模块 30](#_Toc144287079)

[5.1.6 邮件端口模块 31](#_Toc144287080)

[5.2 界面设计 32](#_Toc144287081)

[5.3 本章小结 34](#_Toc144287082)

[第6章 系统测试 35](#_Toc144287083)

[6.1 系统总体测试 35](#_Toc144287084)

[6.2 系统功能测试 36](#_Toc144287085)

[6.2.1 邮件发送功能测试 36](#_Toc144287086)

[6.2.2 邮件接收功能测试 37](#_Toc144287087)

[6.2.3 邮件删除功能测试 37](#_Toc144287088)

[6.2.4 附件发送功能测试 38](#_Toc144287089)

[6.2.5 附件接受功能测试 38](#_Toc144287090)

[6.3 本章小结 39](#_Toc144287091)

[第7章 结论 40](#_Toc144287092)

[致谢 41](#_Toc144287093)

[参考文献 42](#_Toc144287094)

第1章 绪论

## 1.1 研究背景

电子邮件(E-mail)是一种基于电子技术的信息交换方式，广泛应用于全球各类网络中。通过连接国际的互联网，可以实现各类信号源的转发、接收和存储，达到全球的各个角落。

电子邮件系统采取了“存储-转发”模式，为我们提供了一种异步的通信手段。这意味着，发送者可以在任何时候发送邮件，而无需等待接收者处于在线状态。邮件将被安全地保存在接收者的电子邮箱中，使得接收者能在任何适合的时间阅读邮件，而无需受到地点或时间的限制。

电子邮件系统是通讯科学与计算机科学融合的结果，其邮件的传递依托于简称为SMTP的简易邮件传递协议。每个互联网使用者都有属于自己的电子邮箱，可以通过密码访问自己的邮箱，并进行发送、阅读、编辑、重定向以及归档等各种操作。

总的来说，电子邮件作为一种高效且方便的信息传递手段，不止能够传输文本内容，还可以传递各种类型的文件，为我们的生活和工作带来了极大的便利。

## 1.2 研究意义

随着信息社会和知识经济的发展，电子邮件已经成为了日常交流的重要工具，其简洁、快捷、低成本和全球接入的优势深受人们喜爱。因此，对电子邮件系统进行深入研究具有很大的意义。

JavaMail电子邮件系统作为研究的重点，吸引了大量网络用户。其优势在于不需要特定的邮件客户端软件，用户可以使用任何的网络浏览器接收和发送电子邮件，无需担心操作系统的兼容性问题。

此外，JavaMail邮件系统的配置数据是储存在邮件服务器中的，无论用户调换设备或者更改操作环境，邮件的传输方式都将保持不变。这种设计方式为用户带来了极大的便利性和独特的使用体验。

JavaMail邮件系统的另一大优点是它几乎不需要考虑硬件配置问题，同时也提供了用户邮件的安全保障。虽然它在动态页面支持和数据库处理功能上存在一些限制，但这些问题已经得到了一定程度的解决。

电子邮件技术的发展趋势体现在以下几个方面：

1. Web邮件技术：为用户提供更方便的移动办公环境，用户可以在任何地方登录邮件服务器收发邮件。
2. 多域邮件服务：一台物理服务器为多个独立注册的Internet域名提供服务，提供了更强大的横向邮件服务能力。
3. Linux邮件服务器：结合Sendmail、MySQL等开源软件，降低了系统价格，同时保证了稳定性和可靠性。
4. 安全防护：在数据身份认证、传输加密、垃圾邮件和邮件病毒过滤等方面的安全技术得到了更好的应用。
5. 多语言支持：支持多种字符集，保证了全球的沟通无障碍。
6. 远程监控和性能调整：通过Web方式进行远程邮件监控和性能调整，提高了邮件处理的效率和速度。

综上所述，随着计算机和Internet的不断发展，JavaMail邮件系统因其快捷、方便等优点，将会在各个领域得到广泛的应用，进一步提升电子邮件在日常通讯中的重要地位。

## 1.3 国内外研究现状

电子邮件在现代社会中是最主要的通讯工具之一，它在我们的日常工作和学习中起着极为关键的作用。对电子邮件的研究已经数见不鲜，大多数基于JavaMail的邮件系统的构建和操作都是通过研究和使用Sun公司提供的JavaMail API来实现邮件的发送和接收功能。

1.3.1 国外研究现状

在国际上，关于"邮件传输"的研究包含了更广泛的技术和实用环境，比如：人工智能、区块链、大数据等，以期实现更加智能、安全和高效的邮件传递服务。

例如，Google公司在其邮件服务Gmail中使用了人工智能技术，通过机器学习等算法对用户的邮件进行智能分类和管理，提供了更加智能化的邮件收发服务。同时，Gmail还使用了DMARC、SPF和DKIM等安全协议，保障了邮件传输的安全性。

此外，英国的研究人员也在其论文中提出了一种基于区块链的安全邮件系统，通过将邮件内容和元数据加密存储在区块链上，确保邮件的机密性和不可篡改性。该系统使用了PGP和S/MIME等安全协议，提供了更加安全的邮件传输和存储服务。

另外，美国的研究人员在其论文中提出了一种基于大数据和机器学习的邮件反垃圾方案，通过对邮件元数据和内容进行分析和分类，减少垃圾邮件对用户的干扰和影响。

综上所述，国外的研究人员在“邮件收发”方面，主要关注于技术创新和应用场景拓展，使用人工智能、区块链、大数据等技术，提供更加智能化、安全化和高效化的邮件收发服务，以满足用户的不同需求。

1.3.2 国内研究现状

在“邮件收发”方面，国内研究人员主要关注于邮件客户端的开发和实现，以提供更加便捷、安全和高效的邮件收发服务。

例如，中国科学院计算技术研究所的研究人员在其论文中提出了一种基于Web的邮件客户端的设计和实现方法。该方法使用HTML、CSS和JavaScript等Web技术，将邮件客户端实现为一个网页应用程序，用户可以通过浏览器访问该网页，完成邮件收发的操作。该方法相较于传统的邮件客户端具有更加灵活的操作方式，可以随时随地进行邮件收发操作，具有较好的可移植性和跨平台性。

此外，南京大学的研究人员也在其论文中提出了一种基于移动终端的邮件客户端设计方案，该方案主要针对移动设备用户的邮件收发需求，提供了一些特色功能，例如支持离线邮件查看和回复、自动邮件分类等。该方案还使用了SSL和TLS等安全协议，加强了邮件传输的安全性。

综上所述，国内的研究人员在“邮件收发”方面，主要关注于邮件客户端的开发和实现，使用Web技术和移动技术等，提供更加便捷、安全和高效的邮件收发服务，以满足用户的不同需求。

## 1.4 研究内容及目标

本研究的主要内容和目标是设计和实现一个基于JavaMail的电子邮件系统，重点关注其可扩展性、安全性、和稳定性。具体包含以下几个方面：

1.4.1 研究内容

邮件发送和接收功能的实现：研究和使用Sun公司提供的JavaMail API，实现邮件发送和接收功能，包括文本邮件和附件邮件的发送和接收。

邮件存储和检索：研究如何有效存储和检索邮件，以提高邮件系统的效率和用户体验。

邮件系统的安全性：研究如何提高邮件系统的安全性，防止邮件被恶意攻击和滥用。

邮件系统的扩展性：研究如何提高邮件系统的扩展性，使邮件系统可以适应更大的用户群和更复杂的应用环境。

邮件系统的稳定性：研究如何提高邮件系统的稳定性，使邮件系统在各种网络环境下都能稳定运行。

1.4.2 研究目标

实现基于JavaMail的电子邮件系统：利用JavaMail API，完成电子邮件系统的设计和实现，满足用户的基本邮件发送和接收需求。

1. 提高邮件系统的安全性：设计和实现有效的安全机制，防止邮件系统被恶意攻击和滥用。
2. 提高邮件系统的扩展性：设计和实现有效的扩展机制，使邮件系统可以适应更大的用户群和更复杂的应用环境。
3. 提高邮件系统的稳定性：通过优化邮件存储和处理机制，提高邮件系统的稳定性，使邮件系统在各种网络环境下都能稳定运行。
4. 提供优秀的用户体验：通过优化邮件检索和显示机制，提供优秀的用户体验，使用户可以方便快捷地使用邮件系统。

## 1.5 论文组织结构

本文的结构安排如下：

首章，短序。本章主要阐释了研究的背景和含义，设定了研究的目的和任务，对研究的主题和方法进行了归纳，并对全篇的结构布局进行了解释。

第二章,技术概览。本章主要概述了实现邮件传输应用所需的相关技术和知识，包括JavaMail、SMTP、POP3、MIME等通讯协议。

第三章，需求分析和功能设计。本章主要对邮件收发软件的需求进行了详细的分析，并在此基础上设计了软件的主要功能模块。

第四章，系统设计。本章主要介绍了邮件收发软件的系统设计，包括系统的总体设计和各个模块的详细设计。

第五章，系统实现和界面设计。本章主要介绍了邮件收发软件的实现过程，包括系统实现和界面设计，并详细解析了各个功能模块的实现方式。

第六章，系统评估与结果解析。本章主要对邮件传输应用进行了深入的评估，并对测试结果进行了解析，以验证应用的功能和性能。

第七章，总结与前瞻。本章对全篇的研究主题进行了概括，并对未来的研究方向进行了预期。

以上就是本文的组织构，每一章的内容都是为了实现邮件收发软件这一目标而展开的，各章之间关系紧密，层次分明。

## 1.6 本章小节

本章主要介绍了本研究的背景和目标，以及电子邮件系统的研究现状。

首先，我们详细阐述了电子邮件在信息化时代中的重要性，以及基于JavaMail的电子邮件系统设计与实现的必要性。我们讨论了现有电子邮件系统的一些问题，例如安全性、稳定性和可扩展性等。

接下来，我们对全球的电子邮件系统研究状况进行了深入的探讨。在国际上，我们主要关注了Unix平台的Sendmail和Windows平台的Microsoft Exchange的研究。在国内，我们主要探索了电子邮件系统的稳定性、可扩展性和安全性等问题。

最后，我们明确了本研究的内容和目标，包括实现基于JavaMail的电子邮件系统，提高邮件系统的安全性、扩展性和稳定性，以及提供优秀的用户体验。

第2章 相关理论基础和技术简介

## 2.1 相关邮件协议

邮件传递的实现依赖于特定的通信协议。这些主要协议包括SMTP、POP3、MIME、IMAP以及NNTP和其它的第三方协议，每个协议都有其独特的功能和实用环境。

SMTP：简单邮件传输协议（SMTP）是一组用于控制邮件从起始点到终点的传递规则。SMTP协议是TCP/IP协议群的一部分，它协助每台计算机在发送或转发邮件时找到下一个目标。然而，SMTP最初的设计并未考虑在互联网中传递重要和敏感的信息，因此其安全性相对较低。

POP3：邮局协议（POP），其当前版本为3，因此通常被称为POP3。POP是一种让用户能从邮件服务器接收和发送邮件的协议。POP3是当前最为广泛使用的电子邮件服务协议。用户可以通过POP3客户端程序登录到邮件服务器，然后发送邮件和附件；邮件服务器将这些邮件从服务器上简略节略，并将用户提交的邮件转发到运行SMTP协议的计算机中。

MIME：多用途网际邮件扩展（MIME）是一种扩展方案，用于在各种邮件系统中交流信息。MIME格式具有高度的灵活性，使得人们能够在互联网上共享各种类型的文件，如音频文件、视频文件、图像文件、APP和其他各类文件。

IMAP：互联网邮件访问协议（IMAP）是一个较高级的邮件接收协议，当前的版本为IMAP4。相较于POP，IMAP提供了更为高级的功能，如在服务器上为每个互联网使用者网站维护多个文件夹，并允许这些文件夹在多个联网使用者之间共享。

NNTP和其他第三方协议：在设计JavaMail API时，开发者考虑到了与第三方协议实现提供者的解耦，因此可以轻易地添加一些第三方协议，比如新闻组协议（NNTP）和安全的MIME协议（S/MIME）。

## 2.2 JavaMail API

2.2.1 JavaMail API简介

JavaMail API是SUN提供的一个电子邮件处理扩展，为Java开发者提供了编辑、读取和发送电子新闻的工具（标准扩展）。这个API为电子邮件管理提供了一个统一的应用编程接口，能够处理各种电子邮件格式（如IMAP、POP、SMTP以及MIME）以及其他与Internet相关的通信协议。

利用这个包，开发者能够构建邮件用户代理（Mail User Agent，MUA）类型的软件，这类应用与Foxmail、Eudora以及Microsoft Outlook等邮件应用有许多相似之处。然而，这些软件的主要职能并不是进行邮件传输、发送和转发新闻等操作，这些任务通常由邮件传输代理（Mail Transfer Agent，MTA）类型的软件来完成。换言之，用户可以和MUA类型的软件交互，阅读和编辑电子邮件，而MUA则依赖MTA来处理实际的发送任务。

2.2.2 JavaMail API核心类

（1）Session：Session类在JavaMail API中扮演着重要的角色，它定义了邮件交互所需的基础环境。它和Http会话有许多相似的地方，我们的邮件收发工作都在这样的环境中进行。Session对象通过使用java.util.Properties对象获取邮件服务器、用户名、密码以及可能会在整个应用程序中使用的其他共享信息。

由于Session类的构造方法被设置为私有，我们无法直接创建其实例。但我们可以使用Session类提供的getDefaultInstance()静态方法获得一个默认的Session对象。

Properties props = new Properties();

// fill props with any information

Session session = Session.getDefaultInstance(props, null);

或者使用getInstance()这个静态工厂方法[8]获得自定义的Session:

Properties props = new Properties();

// fill props with any information

Session session = Session.getInstance(props, null);

观察以上两个示例，我们可以看到getDefaultInstance()和getInstance()方法的第二个参数均为null。这是由于这些示例中没有涉及到邮件授权的使用。在访问邮件服务器的过程中，使用共享的Session已经足够满足需求，即便在处理多个用户邮箱的情况下也同样适用。

（2）Message：一旦我们得到了Session对象，就可以开始创建要发送的消息。SUN为此提供了Message类型的类来协助开发者。Message是一个抽象类，我们常常使用它的子类javax.mail.internet.MimeMessage，这个类主要用于处理MIME类型和头的电子邮件消息。

消息头只能使用US-ASCII字符集，非ASCII字符需要经过编码转换为ASCII字符后才能使用。要创建一个MimeMessage对象，我们需要将Session对象作为MimeMessage构造函数的参数传入：

MimeMessage message = new MimeMessage(session);

需要注意的是，MimeMessage类有多种构造函数，例如，也可以使用输入流作为参数。

创建了MimeMessage对象后，我们需要设定它的各个部分，这些部分对应于MimePart接口。最基本的消息内容设定方式是通过调用setContent()方法并传入表示消息内容和MIME类型的参数：

message.setContent("Hello", "text/plain");

然而，如果我们的MimeMessage的消息内容是文本，我们可以直接使用setText()方法来方便地设定文本内容：

message.setText("Hello");

对于文本信息，后者方法更为简便。对于其他类型的消息，如HTML消息，我们需要使用前者。我们还可以通过setSubject()方法设定邮件的主题：

message.setSubject("First");

（3）Address：我们已经构造了Session和Message对象。和Message的情况类似，Address也是一个抽象类，因此我们会使用其子类javax.mail.internet.InternetAddress。

我们可以通过传入表示电子邮件地址的字符串来创建一个电子邮件地址实例：

Address address = new InternetAddress("president@whitehouse.gov");

如果要在件地址后面增加名字的话，可以通过传递两个参数：代表邮件地址和名字的字符串来建立一个具有邮件地址和名字的邮件地址类：

Address address = new InternetAddress("president@whitehouse.gov", "George Bush");

这里创建的InternetAddress对象主要用于设置电子邮件消息的发件人和收件人。创建了InternetAddress对象后，我们可以通过Message的setFrom()和setReplyTo()方法设置电子邮件的发件人：

message.setFrom(address);

message.setReplyTo(address);

如果电子邮件有多个发件人地址，我们可以使用addFrom()方法添加更多的发件人：Address address[] = ...;

message.addFrom(address);

要设置收件人，我们使用addRecipient()方法添加收件人，这个方法需要使用Message.RecipientType的常量来区分收件人的类型：

message.addRecipient(type, address)

下面是Message.RecipientType的三个常量:

Message.RecipientType.TO

Message.RecipientType.CC

Message.RecipientType.BCC

如果我们想给其他人发送电子邮件，并且抄送给另一个人，我们可以使用如下的方法：

Address toAddress = new InternetAddress("vice.president@whitehouse.gov");

Address ccAddress = new InternetAddress("first.lady@whitehouse.gov");

message.addRecipient(Message.RecipientType.TO, toAddress);

message.addRecipient(Message.RecipientType.CC, ccAddress);

需要注意，JavaMail API并没有提供检查电子邮件地址有效性的机制。我们可以自己完成这个功能，例如验证电子邮件地址的字符是否符合RFC822规定的格式，或者通过DNS服务器上的MX记录进行验证等。

（4）Authenticator：JavaMail API也可以配合Authenticator使用，通过用户名和密码来访问受保护的资源。对于JavaMail API而言，这些资源主要是邮件服务器。JavaMail的Authenticator位于javax.mail包中。

要使用Authenticator，首先需要创建一个其子类，并在getPasswordAuthentication()方法中返回PasswordAuthentication实例。创建完成后，必须将Authenticator注册到Session中。然后，每当需要进行身份验证时，系统就会通知Authenticator。

Authenticator authenticator = new Authenticator() {

protected PasswordAuthentication getPasswordAuthentication() {

return new PasswordAuthentication("username", "password");

}

};

Session session = Session.getInstance(props, authenticator);

在上述代码中，我们创建了一个Authenticator的匿名子类，并重写了getPasswordAuthentication()方法，返回了一个PasswordAuthentication实例，该实例中包含了用户名和密码。然后，我们将这个Authenticator实例注册到了Session中。当Session需要进行身份验证时，就会调用我们提供的Authenticator实例。

（5）Transport：JavaMail API通过使用Authenticator类（类似于java.net包中的类），利用用户名和密码访问受保护的资源，这里的"资源"特指邮件服务器。JavaMail的Authenticator类可以在javax.mail包中找到。

在使用Authenticator这个抽象类时，我们需要创建一个继承自Authenticator的子类，并且这个子类需要实现getPasswordAuthentication()方法，该方法需要返回一个PasswordAuthentication对象（用于存储认证过程中所需的用户名和密码）。然后，我们需要将这个Authenticator子类注册到Session实例中，使得Session知道在身份认证过程中需要使用哪个类。

下面代码片断中的MyAuthenticator就是一个Authenticator的子类。

Properties props = new Properties();

// fill props with any information

Authenticator auth = new MyAuthenticator();

Session session = Session.getDefaultInstance(props, auth);

（6）Folder和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。

Folder类用于分级组织邮件，并提供javax.mail.Message格式访问e-mail的能力。

2.2.3 JavaMail API的应用

（1）阅读电子邮件

对于阅读邮件来说，首先要获取一个会话，然后获取并连接到一个相应的用于您的收件箱的存储上，接着打开相应的文件夹，再获取消息。同时，不要忘记了操作完成后关闭连接。

步骤如下：

创建一个Properties 对象，Session 对象通过Properties 对象来接受选项，在这种情况下，没有选项，但仍然需要Properties 对象。

创建Session 对象，并不直接对Session 实例化，而是通过调用getDefaultInstance

使用Session.getStore 来获取服务器的POP3存取，也可以指定IMAP。

使用store.connect 来创建一个从store 对象到实际服务器上的活连接。

用store.getFolder 找到INBOX（收件箱）文件夹（POP3 存储里唯一的文件夹），对IMAP 服务器而言，可以访问不同的文件夹。用folder.open 以只读方式打开文件夹。

（2）发送电子邮件

发送电子邮件消息涉及到获取会话、创建和填充消息并发送消息这些操作。在获取Session 时，通过为要传递的Properties 对象设置mail.smtp.host属性来指定所用的SMTP 服务器。

（3）转发电子邮件

转发消息涉及的内容要稍微多一点，没有一个专门用于转发消息的方法，可以通过处理组成消息的各个部分来创建要转发的消息。一条邮件消息可由多个部分组成，每一部分是一个BodyPart（报文部分），或更特殊一点，在操作MIME 消息时则是MimeBodyPart [11]。不同的报文部分组合到一个称为Multipart 的容器中，或者更特殊一点，是一个MimeMultipart 容器。要转发消息，用户需要创建一个用于消息文本的部分，和用于要转发的消息的第二个部分，并将这两个部分组合成一个Multipart（多个部分）。然后， 可以把这个Multipart添加到一个合适的注明地址的消息中并发送它。

这就是转发消息的过程。要把一条消息的内容复制给另一条消息，只需通过它的ataHandler 类复制即可，它是出自于JavaBeans Activation Framework的一个类。

上面着重分析和探讨基于JavaMail API 的邮件服务系统的设计和实现的相关技术。为系统实际开发过程中充分利用Java 编程语言的开放性、平台无关性等优点，极大地提高了系统的效率和性能。

## 2.3 本章小节

在本章中，我们深入探讨了JavaMail API的基本概念和用法，同时深入分析了基于JavaMail API的邮件服务系统的设计和实现技术。

首先，我们介绍了JavaMail API的基本架构和主要组件，包括Session、Message、Address、Authenticator、Transport、Folder和Store等。我们讲解了如何通过这些组件来创建、发送和接收电子邮件。

紧接着，我们详细探讨了JavaMail API的应用实例，包括查看、发送以及转发电子邮件的方法。我们展示了如何创建一个Properties对象，如何建立与邮件服务器的连接，如何打开邮件文件夹，以及如何获取和处理消息。同时，我们也介绍了如何使用SMTP服务器发送电子邮件，以及如何处理和转发消息的各个部分。

最后，我们重点分析了基于JavaMail API的邮件服务系统的设计和实现。我们讨论了如何充分利用Java编程语言的开放性和平台无关性，以提高系统的效率和性能。

# 通过本章的学习，读者应能理解和掌握JavaMail API的基本用法，并能够设计和实现基于JavaMail API的邮件服务系统。在实际开发中，读者可以根据自己的需求，灵活运用JavaMail API，实现各种复杂的邮件服务功能。第3章 JavaMail系统分析

3.1 系统需求分析

在本章节中，我们将对邮件收发软件的需求进行详细的分析，包括功能需求、非功能性需求、可靠性需求和用户体验需求。

3.1.1 功能性需求分析

邮件收发软件的主要功能包括邮件的发送软件、接收和管理。用户需要能够方便地发送邮件，并能够在软件中接收、查看和管理已发送和已接收的邮件。此外，用户还需要能够对邮件进行删除、下载附件等操作，以提高邮件处理的效率。具体如图3.1所示[17]。

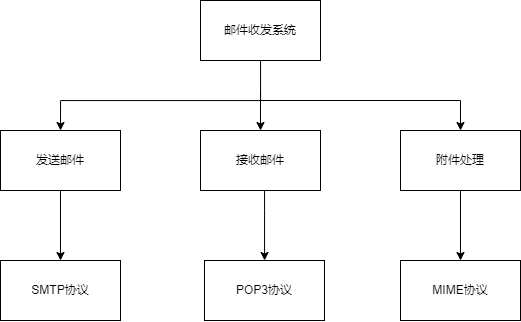


图3.1 系统功能模块组织图

### 3.1.2 非功能性需求分析

非功能性需求并非直接关联至系统的具体功能，但对系统的运行和使用起着至关重要的作用。在电子邮件系统中，这类需求主要涉及到安全性、稳定性和用户体验等方面。

首先，我们来探讨安全性需求。为了确保我们设计的电子邮件系统在运行时具有良好的安全性，我们需要满足以下要求：

安全的邮件传输协议[18]：电子邮件系统应支持如SMTPS、POP3S等安全的邮件传输协议，以确保邮件在传输过程中的数据安全。

数据加密：电子邮件系统需要支持SSL和TLS协议，这样可以对邮件传输过程中的数据进行加密，以防止数据在传输过程中的泄露或被恶意拦截，从而避免数据的丢失或篡改。同时，系统应支持对邮件内容进行加密，以防止邮件内容的丢失或篡改。

身份验证机制：软件应该支持邮件收发过程中的身份验证机制，以确保只有合法用户可以收发邮件，防止邮件被恶意使用。

用户隐私保护：软件应该提供用户隐私保护功能，例如用户可以选择不公开自己的邮件地址，以保护用户的隐私安全。

其次，我们分析可靠性需求。邮件是用户进行通信和交流的重要工具，因此邮件收发软件需要保障其可靠性，避免邮件在传输过程中丢失或丢失部分内容。因此，软件需要使用可靠的邮件传输协议和机制，例如SMTP的重传机制和POP3的删除标记机制，以确保邮件能够可靠地传输和接收[19]。

最后，我们分析用户体验需求。用户体验是邮件收发软件的重要方面。软件需要提供用户友好的界面和操作体验，使用户能够轻松地发送、接收和管理邮件，同时也需要保障软件的稳定性和性能，以提高用户的满意度。

3.2 可行性研究

在启动邮件收发软件的开发项目之前，我们必须对其可行性进行全面的评估。这个过程包括对技术实现的可能性、法律合规性等因素的考量，确保我们的项目在后续的开发阶段不会遇到不可预见的难题，为项目的成功提供一个坚实的基础。

3.2.1 技术可行性

邮件收发软件的实现核心依托于SMTP、POP3、MIME等关键通信协议和技术。因此，这个毕业设计项目的成功，需要我具有足够的理论知识和实际操作能力来处理这些协议。

通过自我评估和导师的指导，我发现我已经掌握了这些协议的基本工作原理，而且我能够成功地将这些协议应用到我的毕业设计项目中。此外，我也已经具备了解决可能出现的技术问题和挑战的能力，这对于保证软件的稳定性和性能至关重要。

总的来说，通过对自身的技术理解和实践能力的全面评估，我相信我可以成功地完成这个毕业设计项目，实现邮件收发软件的核心功能，证明这个项目在技术上是完全可行的。

3.2.2 法律可行性

在开发邮件收发软件的过程中，我们需要遵守所有相关的法律法规，包括但不限于网络安全法、个人信息保护法等。这些法律法规为我们的软件设定了一系列的法律框架，我们必须在这个框架内进行开发和实现。

因此，我们需要深入了解这些法律法规，理解它们对我们的软件可能产生的影响，并在软件的设计和实现过程中尽可能地遵守这些规定。这不仅可以确保我们的软件在法律上的合规性，也可以帮助我们避免可能的法律风险。

综上，通过对相关法律法规和政策要求的深入理解和分析，我们可以确定邮件收发软件在法律上的可行性。

3.3 本章小结

本章主要介绍了邮件收发系统的系统分析。首先进行了需求分析，包括功能性需求分析和非功能性需求分析并在此基础上进行了可行性分析，包括技术可行性和法律可行性。

在功能需求方面，系统需要支持邮件的发送和接收，同时需要支持邮件附件的添加和处理，并支持多种邮件格式以及安全传输协议。在非功能性需求分析方面，系统需要保障邮件的机密性和完整性，并保护用户的隐私。在可靠性需求方面，系统需要保证邮件的可靠性和稳定性，避免邮件丢失和延迟等问题。在用户体验需求方面，系统需要提供简洁、直观、易用的界面，方便用户操作和管理邮件。

在可行性分析方面，本章主要从技术可行性和法律可行性两个方面进行了探讨。在技术可行性方面，系统基于JavaMail，并使用SMTP、POP3和MIME等通信协议，具有良好的可扩展性和兼容性，可以满足系统的需求。在法律可行性方面，系统需要遵循相关的法律法规和规范，保护用户的合法权益，避免违反相关法规而导致的风险和损失[21]。

第4章 JavaMail系统总体设计

## 4.1 JavaMail系统结构设计

本系统采用三层架构模式具体架构如下：

表示层：提供用户操作界面，实现与用户的交互式操作。采用Java Swing技术实现，包括主窗口、邮件列表窗口、邮件编辑窗口等。

业务逻辑层：处理业务逻辑和系统控制，负责数据处理和状态管理。主要包括邮件发送、接收、解析、显示等功能，采用JavaMail、SMTP、POP3和MIME协议实现。

数据访问层：负责数据的存储和访问，采用文件系统或数据库存储邮件数据，提供数据操作的接口。

系统结构设计图如图4.1所示[22]。

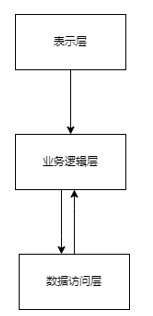


图4.1 系统结构设计框图

4.2 功能模块设计

本系统主要包括以下功能模块[23]：

邮件发送模块，邮件接收模块，邮件解析模块，邮件显示模块，邮件设置模块。

上述功能单元是通过MVC（Model-View-Controller）架构模式完成的。在这个模式中，Model层负责数据的管理和处理，View层负责展示用户界面并进行交互，而Controller负责处理和控制业务逻辑。通过这样的层次化设计和模块化的开发方式，我们提高了系统的扩展性和维护性。

总体结构图如下：

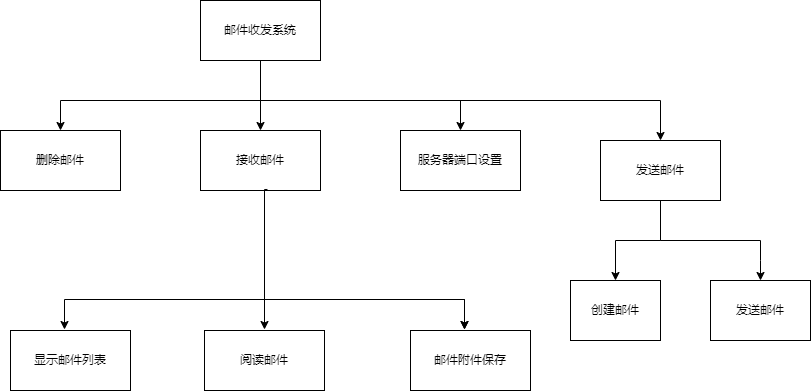


图4.2 功能框图

通过以上设计，实现了邮件客户端程序的基本功能，同时保障了邮件在传输过程中的安全性，提高了用户的使用体验。未来可以在此基础上继续完善功能，提升软件的性能和稳定性。

4.2.1 发送邮件模块设计

发送邮件模块是整个系统的核心功能之一，需要支持多种邮件格式（如纯文本邮件、HTML邮件等）和邮件安全传输协议（如SMTPS、SSL、TLS等），并能够添加和处理邮件附件。该模块主要包括以下功能模块[24]：

邮件发送器：负责发送邮件，包括设置邮件内容、添加附件、设置收件人、主题、发送时间等信息。

SMTP客户端：负责与SMTP服务器建立连接，并通过SMTP协议发送邮件消息。

MIME编码器：负责将邮件内容转换为HTML或纯文本格式，以及对附件进行编码。

邮件安全传输器：负责保障邮件的安全传输，如使用SMTPS、SSL、TLS等协议。

4.2.2 接收邮件模块设计

接收邮件模块是系统的另一个核心功能，需要支持多种邮件格式（如纯文本邮件、HTML邮件等）和邮件安全传输协议（如POP3S、SSL、TLS等），并能够处理邮件附件。该模块主要包括以下功能模块：

邮件接收器：负责接收邮件，包括设置邮件内容、保存附件、解析邮件头部信息。

POP3客户端：负责与POP3服务器建立连接，并通过POP3协议接收邮件消息。

邮件安全传输器：负责保障邮件的安全传输，如使用POP3S、SSL、TLS等协议。

以上功能模块都需要与JavaMail API进行交互，利用JavaMail API提供的邮件收发相关类实现具体功能。同时，还需要使用Java的多线程技术，使得邮件发送和接收功能可以在后台异步运行，提高系统的响应速度和性能[25]。

4.2.3 附件处理模块设计

附件处理模块是邮件收发系统中的一个重要模块，负责对邮件附件的添加、处理和下载。系统需要支持各种类型的附件，包括文本文件、图片、视频等。在实现该模块时，需要考虑到安全性、可靠性和用户体验等方面[26]。

该模块的具体设计如下：

添加附件功能：用户可以通过系统界面选择本地文件并添加到待发送邮件中，也可以通过拖拽操作将文件添加到待发送邮件中。

附件处理功能：当邮件被接收后，系统需要对附件进行解析和处理。对于文本文件和图片等常见类型的附件，系统需要提供在线预览和下载功能；对于视频等大型文件，系统需要提供下载链接或者引导用户通过云存储等方式进行下载。

安全性保障：在邮件附件的传输和处理过程中，系统需要采用加密技术和安全传输协议，确保数据的安全性和保密性。

4.2.4 界面设计

用户界面设计是邮件系统中的重要组成部分，直接影响邮件系统使用者的使用体验。系统需要提供清晰、简洁、易用的界面，方便用户进行邮件的发送、接收和管理。在设计界面时，需要考虑到以下因素：

功能布局：界面需要将不同的功能模块进行合理布局，使用户可以快速找到需要的功能，同时避免界面过于复杂和拥挤。

操作便捷性：系统需要提供简单易懂的操作指引，让用户能够快速上手并熟练使用系统。

界面美观度：系统需要采用统一的UI设计风格，保证界面的美观度和协调性。

用户体验：界面需要考虑到用户的使用习惯和心理需求，提供更加贴近用户需求的设计。例如，可以提供主题换肤功能、快捷键等用户体验优化功能。

在总体设计阶段，需要综合考虑系统的需求和目标，对各个功能模块进行细致的设计和规划，为后续的系统开发和实现提供有力支持。

## 4.3 数据库设计

数据库设计是整个系统设计的重要部分，它决定了数据的存储、查询、更新和管理如何进行。邮件收发系统需要存储用户信息、邮件信息和附件信息等数据，因此需要设计一个高效、安全、可扩展的数据库。

4.3.1 用户信息表设计

用户信息表用于存储用户的基本信息，包括用户名、密码、邮箱地址等。下表是用户信息表的设计：

表4.3.1 用户信息表

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| UserID | INT | 用户ID，主键，自增长 |
| UserName | VARCHAR(50) | 用户名 |
| Password | VARCHAR(50) | 用户密码，需加密存储 |
| Email | VARCHAR(50) | 用户的邮箱地址 |

4.3.2 邮件信息表设计

邮件信息表用于存储邮件的基本信息，包括邮件的发送者、接收者、主题、发送时间等。下表是邮件信息表的设计：

表4.3.2 邮件信息表

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| MailID | INT | 邮件ID，主键，自增长 |
| Sender | VARCHAR(50) | 发送者邮箱地址 |
| Receiver | VARCHAR(50) | 接收者邮箱地址 |
| Subject | VARCHAR(100) | 邮件主题 |
| SendTime | DATETIME | 邮件发送时间 |

4.3.3 附件信息表设计

附件信息表用于存储与邮件相关的附件信息，包括附件的名称、类型、大小等。下表是附件信息表的设计：

表4.3.3 附件信息表

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 说明 |
| AttachID | INT | 附件ID，主键，自增长 |
| MailID | INT | 对应的邮件ID，外键 |
| FileName | VARCHAR(50) | 附件名称 |
| FileType | VARCHAR(20) | 附件类型 |
| FileSize | INT | 附件大小 |

4.4 本章小结

本章介绍了邮件收发系统的总体设计。在系统结构设计中，提出了基于MVC模式的系统结构，明确了模型层、视图层和控制层的作用和职责。在功能模块设计中，分别对发送邮件模块、接收邮件模块、附件处理模块和界面设计进行了详细的设计和说明。其中，发送邮件模块主要包括邮件信息的构建、邮件内容的格式化和发送邮件的实现；接收邮件模块主要包括邮件服务器的配置、连接和接收邮件的实现；附件处理模块主要负责添加和处理邮件附件；界面设计主要通过Java Swing实现，提供友好的用户交互界面。

通过本章的设计，系统具备了完整的邮件收发功能和附件处理功能，并且提供了良好的用户交互界面，满足了需求分析中的功能需求和用户体验需求。在后续的实现和测试中，将会按照本章设计的要求进行具体实现和测试，以达到系统设计目标。

第5章 JavaMail系统实现

5.1 JavaMail系统实现

5.1.1 JavaMail的实现

导入JavaMail库

JavaMail是一个独立的库，需要在项目中引入该库才能使用JavaMail API。从JavaMail官网下载JavaMail库，或者使用Maven或Gradle等构建工具进行依赖管理。

创建Session对象

Session是JavaMail API的核心对象，表示一个邮件会话，包含了与SMTP服务器的连接信息、身份验证信息等。可以使用javax.mail.Session.getDefaultInstance()方法创建默认的邮件会话，或者使用javax.mail.Session.getInstance(Properties props, Authenticator authenticator)方法创建一个指定的邮件会话。

创建Message对象

Message是JavaMail API中表示邮件的对象，可以设置邮件的发送者、接收者、主题、内容、附件等信息。可以使用javax.mail.internet.MimeMessage类创建Message对象，也可以使用javax.mail.Message子类创建不同类型的邮件对象。

使用Transport.send(Message message)方法发送邮件，该方法会自动将Message对象转换为符合SMTP协议的数据，并通过SMTP服务器发送邮件。发送邮件前需要设置SMTP服务器的信息和身份验证信息。

5.1.2 SMTP协议的实现

JavaMail提供了SMTP协议的实现方法。在JavaMail中，可以使用javax.mail包中的SMTPTransport类来实现SMTP协议。使用SMTPTransport类可以连接SMTP服务器、发送邮件和接收响应信息等操作。以下是JavaMail中实现SMTP协议的基本步骤：

创建Session对象：使用Session.getDefaultInstance()方法创建Session对象，设置SMTP服务器地址、端口号等属性。

创建Message对象：创建MimeMessage对象，设置邮件的发件人、收件人、主题、内容等信息。

创建Transport对象：使用Session对象创建SMTPTransport对象。

连接SMTP服务器：使用SMTPTransport对象的connect()方法连接SMTP服务器。

发送邮件：使用SMTPTransport对象的sendMessage()方法发送邮件。

关闭连接：使用SMTPTransport对象的close()方法关闭连接。

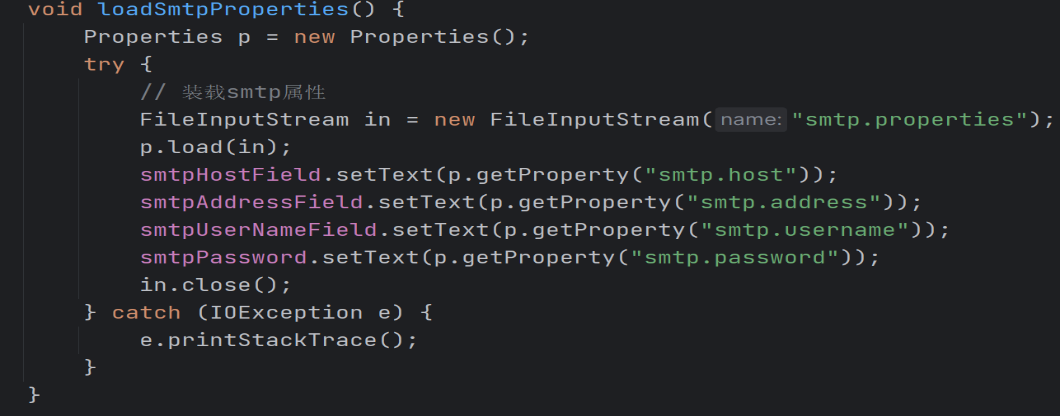


图5.1.2 SMTP加载关键代码

5.1.3 POP3协议实现

服务器端的实现

POP3服务器的实现通常由邮件服务器提供商完成。服务器需要提供基本的POP3功能，如接受来自客户端的连接请求、进行身份验证、检查并下载邮件、删除邮件等。POP3服务器通常会使用TCP协议来与客户端进行通信，并且需要一个独立的端口（默认是110）。

客户端的实现

POP3客户端的实现通常是邮件客户端程序。客户端程序需要提供基本的POP3功能，如连接服务器、进行身份验证、获取邮件、删除邮件等。JavaMail API提供了POP3客户端的实现，使得开发者可以使用Java语言来开发邮件客户端应用程序。

首先需要设置POP3服务器的地址、用户名和密码等参数。然后创建一个Properties对象来存储这些参数。接下来，通过Session.getInstance()方法创建一个会话对象，该方法需要传递Properties对象和一个Authenticator对象，用于进行身份验证。在会话对象中，使用getStore()方法获取存储对象，并调用connect()方法连接服务器。

通过存储对象获取邮件列表，并遍历每一封邮件，并打印邮件的主题、发送人、内容等信息。最后，关闭文件夹和存储对象，释放资源。



图5.1.3 POP3加载关键代码

5.1.4 邮件收发模块

邮件收发模块的实现主要分为两个部分：邮件发送和邮件接收。下面分别介绍这两部分的实现。

邮件发送

邮件发送模块的实现包括两个部分：SMTP协议的通信和邮件消息的构建。下面分别介绍这两部分的实现。

SMTP协议的通信：

使用JavaMail API实现SMTP协议的通信需要以下步骤：

创建一个Properties对象，用于设置SMTP服务器的地址和端口号、是否需要身份验证等信息；

创建一个Session对象，将Properties对象传入，用于与SMTP服务器建立连接；

创建一个MimeMessage对象，用于构建邮件消息；

使用Transport对象的send方法将MimeMessage发送给SMTP服务器。

邮件消息的构建：

具体实现步骤如下：

创建一个MimeMessage对象；

设置邮件的发件人地址、收件人地址、主题、发送时间等信息；

创建一个MimeMultipart对象，用于存储邮件的文本和附件内容；

创建一个MimeBodyPart对象，设置邮件的文本内容；

将MimeBodyPart对象添加到MimeMultipart对象中；

遍历邮件的附件列表，为每个附件创建一个MimeBodyPart对象，并将其添加到MimeMultipart对象中；

将MimeMultipart对象设置为MimeMessage的内容；

调用Transport对象的send方法将MimeMessage对象发送给SMTP服务器。

以上是JavaMail API实现邮件发送的一般步骤，具体实现还需根据具体的业务需求进行调整。

邮件接收

邮件接收的实现需要使用POP3协议，通过JavaMail提供的API可以方便地实现邮件接收。邮件接收的步骤如下：

创建JavaMail的Session对象；

使用Store对象连接到POP3服务器；

打开Folder对象，并读取邮件；

遍历邮件列表，获取每个邮件的主题、发件人、收件人、时间等信息；

根据需求获取邮件正文和附件，并进行解码；

关闭Folder和Store对象。

需要注意的是，在进行邮件接收时，要先连接到邮件服务器并打开相应的邮件文件夹，才能读取其中的邮件。在获取邮件的正文和附件时，需要使用MIME协议进行解码，以得到可读的邮件内容和附件文件。

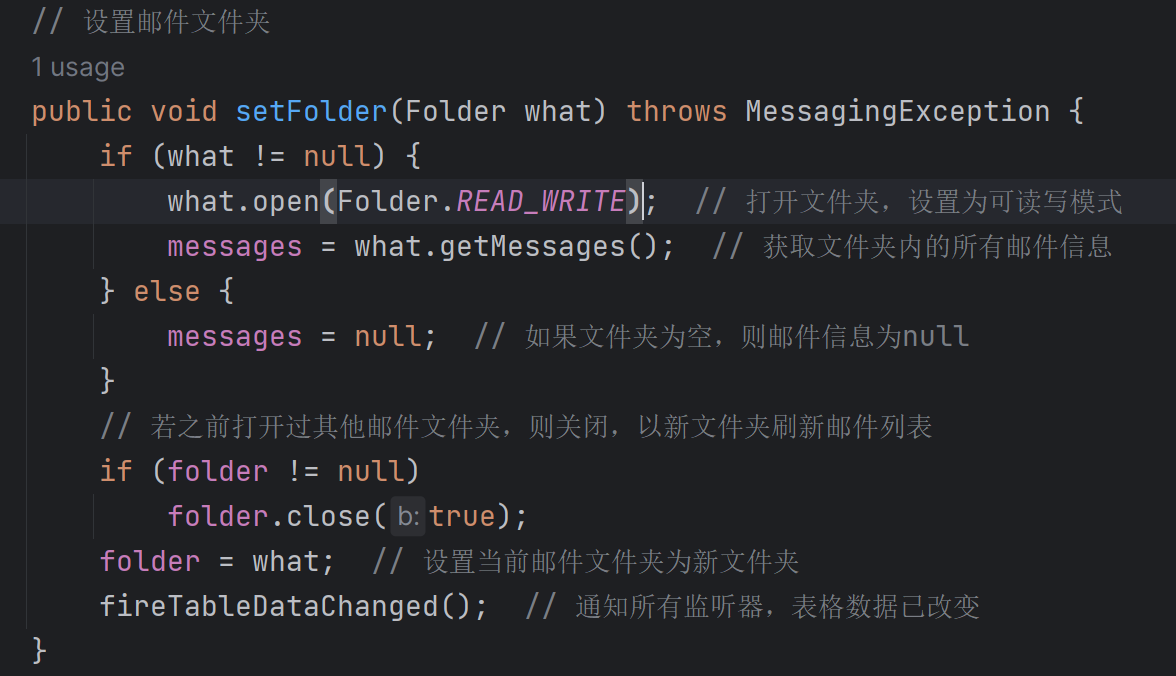


图5.1.4 邮件接受模块关键代码

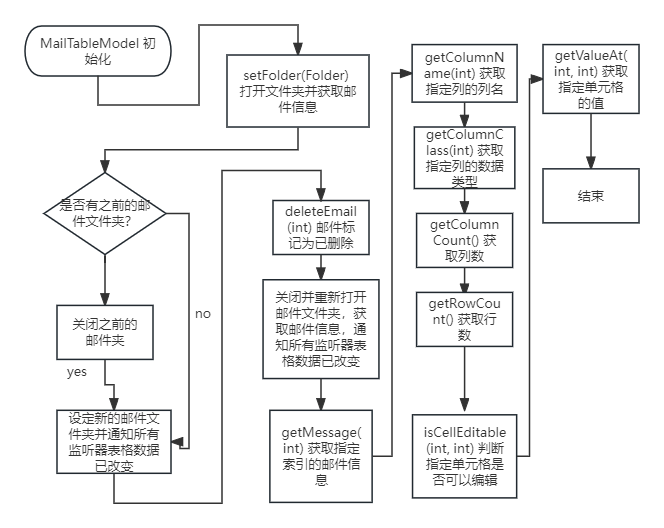


图5.1.4 接受邮件流程图

图5.1.4 邮件发送模块关键代码

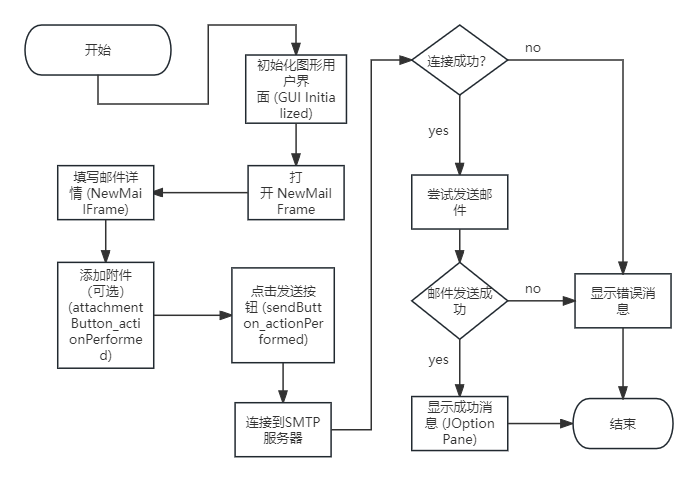


图5.1.4 发送邮件流程图

5.1.5 邮件删除模块

邮件删除模块的流程和工作原理如下：

用户在系统主窗体中选择一个或多个邮件，并点击删除按钮。

系统获取选中邮件的ID或其他唯一标识符。

邮件删除模块通过调用邮件管理模块提供的API，向邮件服务器发送删除请求，并传递选中邮件的ID或其他唯一标识符。

邮件服务器接收到请求后，将选中的邮件从服务器上删除。

邮件删除模块通过调用邮件管理模块提供的API，将选中邮件从本地邮件数据库中删除。

邮件删除模块刷新邮件显示表格，更新显示的邮件列表。

用户可以在表格中看到已删除的邮件已经消失。

需要注意的是，在删除邮件之前，用户需要对删除操作进行确认，以免误删重要邮件。



图5.1.5 邮件删除模块关键代码

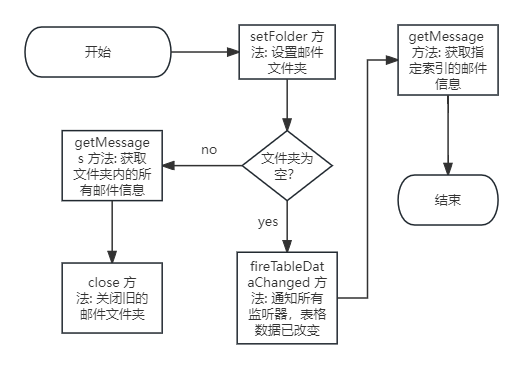


图5.1.5 邮件删除模块流程图

5.1.6 邮件端口模块

系统提供了一个属性文件smtpPop3.properties来存储这些信息。该属性文件包含以下几个属性：

pop3host：POP3邮件服务器地址，如pop3.sina.com.cn；

smtphost：SMTP邮件服务器地址，如smtp.sina.com.cn；

address：收发邮件帐户，如key@sina.com；

username：账户用户名，如key；

password：账户密码，如password。

系统可以通过Java的Properties类来读取该属性文件，并使用setProperty(), getProperty()方法来进行存取属性内容。在获取用户设置的邮件服务器和账户信息后，系统可以使用该信息来实现邮件的发送和接收功能。

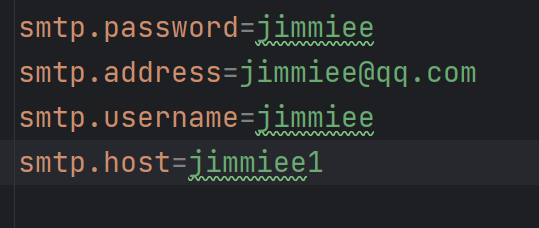


图5.1.6 SMTP配置文件

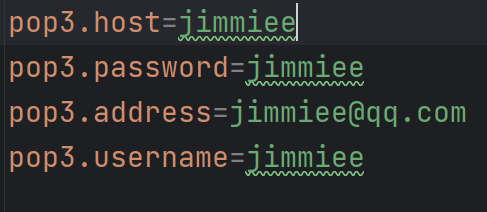


图5.1.6 POP3配置文件

## 5.2 界面设计

在邮件客户端的界面设计中，主要包括了邮件列表区域、邮件显示区域以及功能按钮等几个关键部分。

邮件列表区域是一个JTable，它用于展示邮件的主题、发件人、收件日期等信息。用户可以通过在列表中点击某一行来选中并查看某一封邮件的具体内容。

邮件显示区域是一个JPanel，它用于展示用户在邮件列表中选中的邮件的详细内容。当用户在邮件列表中选中一封邮件时，该邮件的内容就会在此区域中展示。

除了邮件的展示区域，界面上还设置了一系列的功能按钮，具体如下：

创建邮件按钮：用户点击后，将打开一个新的窗口，在这个窗口中，用户可以编写新的邮件并发送。

接收邮件按钮：用户点击后，将从邮件服务器上接收新的邮件并刷新邮件列表。

删除邮件按钮：用户点击后，将删除邮件列表中被选中的邮件。

POP3/SMTP设置按钮：用户点击后，将打开一个新的窗口，在这个窗口中，用户可以设置POP3/SMTP的相关参数。

当用户点击这些功能按钮时，会触发对应的事件处理函数。例如，当点击删除邮件按钮时，就会触发deleteEmailButton\_actionPerformed函数。这个函数会获取用户在邮件列表中选中的邮件的索引，然后调用MailTableModel类中的删除方法，将邮件从邮件列表中删除。类似地，点击其他的功能按钮也会触发对应的事件处理函数，完成相应的操作。

为了处理这些按钮的点击事件，我们定义了四个适配器类，分别是MainFrame\_receiveEmailButton\_actionAdapter，MainFrame\_propertiesSetButton\_actionAdapter、MainFrame\_createEmailButton\_actionAdapter和MainFrame\_deleteEmailButton\_actionAdapter。

这些适配器类实现了java.awt.event.ActionListener接口，并在actionPerformed方法中调用了主界面类的相应方法，从而完成了对按钮点击事件的处理。

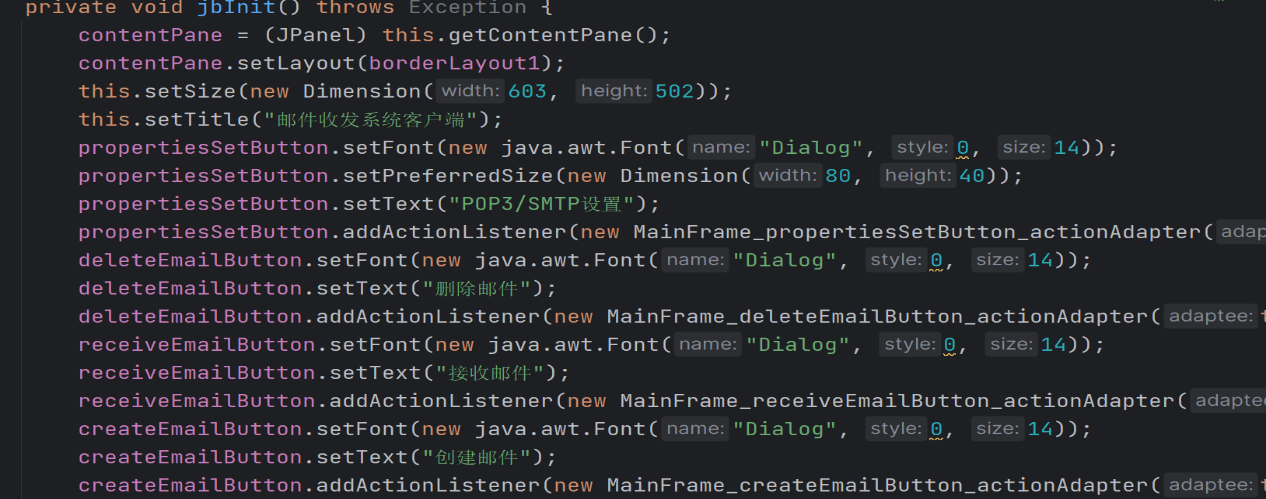


图5.2 界面设计实现关键代码

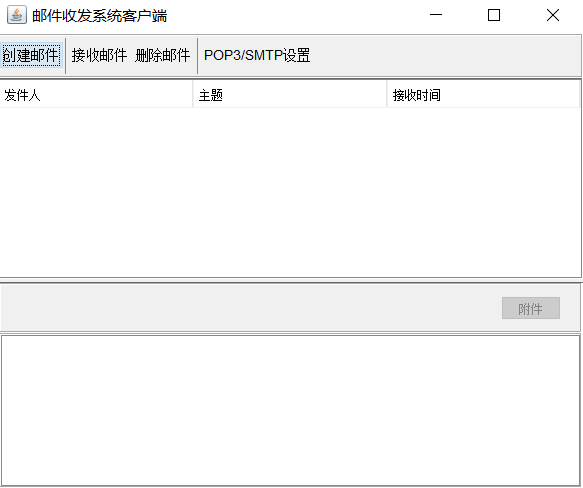


图5.2 主界面

5.3 本章小结

本章主要介绍了邮件收发软件的实现过程，包括系统设计和系统实现两个方面。在系统设计方面，采用了JavaMail和SMTP、POP3、MIME等通信协议，设计了系统的功能模块，表示了各模块之间的关系。在系统实现方面，详细介绍了JavaMail、SMTP、POP3和MIME协议的实现过程，并重点介绍了邮件收发、邮件删除和邮件端口等功能模块的实现。通过本章的实现过程，成功实现了一个基于JavaMail的邮件收发软件，并实现了其基本的功能模块。

第6章 系统测试

6.1 系统总体测试

当系统进行部署之后，整个系统运行正常，如图6-1所示，各个交互界面运行正常：



图6.1 系统部署运行

在配置好服务器的属性值之后，用户可点击“创建邮件”按钮开始编辑并准备发送邮件。在打开的新窗口中，用户有可能输入收件人、抄送人、邮件主题和邮件正文等信息，同时，如果有需要发送的附件，也可以在此添加。邮件编辑完成后，点击“发送”按钮，程序会根据用户输入的邮件信息生成邮件，并将其发送出去。邮件成功发送后，会出现一个通知框，向用户确认邮件已被成功发送，如图6.1所示。



图6.1POP3设置



**图6.1 SMTP设置**

6.2 系统功能测试

邮件系统是一款复杂的软件，为了保证其正常运行和提供良好的用户体验，需要进行系统功能测试。

其中包括邮件接收功能测试，包括单封邮件和批量接收邮件功能。

邮件删除功能测试，用于验证邮件删除功能是否正常，包括单封邮件和批量删除邮件功能。

附件发送功能测试，包括单个附件和多个附件发送功能。

附件接收功能测试，包括单个附件和多个附件接收功能。

通过这些测试，可以有效地发现和解决邮件系统中存在的问题，保证其稳定性和安全性，提高用户满意度。

6.2.1 邮件发送功能测试

在进行邮件发送的功能测试时，首先需要正确配置邮件服务器和相关参数，然后借助邮件客户端进行实际测试。在测试阶段，用户需点击“创建邮件”按钮进入邮件编辑界面，填写收件人、抄送人、主题和正文等信息，同时也可以添加附件。编辑完成后按下“发送”按钮，系统就会根据输入的内容生成邮件，并将之发送出去。在整个发送过程中，我们可以观察邮件的发送状态，以验证邮件是否已成功发送，如图所示：



图6.2.1 邮件发送功能测试截图

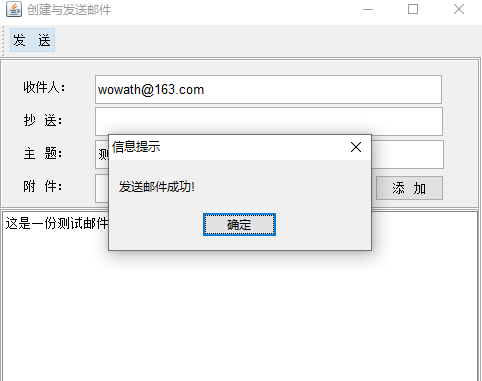


图6.2.1 邮件发送成功截图

6.2.2 邮件接收功能测试

在进行邮件接收功能测试时，需要先设置好邮件服务器和相关参数，并使用不同的邮件客户端进行测试。可以分别测试单封邮件和批量接收邮件功能，检查是否能够正常接收邮件内容、主题、附件等信息。可以观察邮件客户端收到的邮件列表，验证邮件是否被正确接收并显示出来。测试人员可以通过多次测试，确保邮件接收功能的可靠性和稳定性,如图所示：

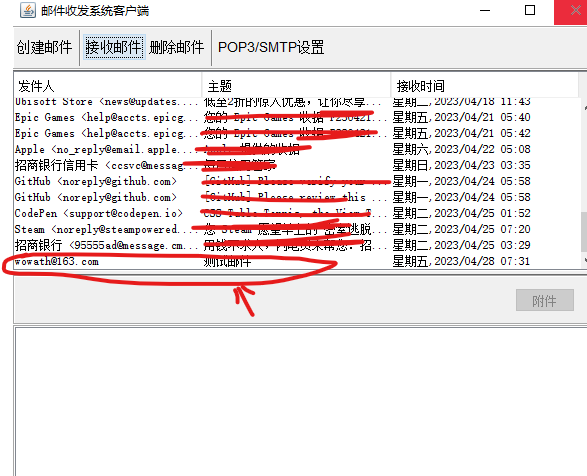


图6.2.2 邮件接收功能测试截图

6.2.3 邮件删除功能测试

当用户收到一些不需要保存的邮件时，可以先选中要删除的邮件，然后在系统主窗口中点击“删除邮件”按钮。系统程序会将当前被选中的邮件从邮件列表中删除，并更新显示邮件列表，以方便用户查看更新后的邮件列表。如图所示：

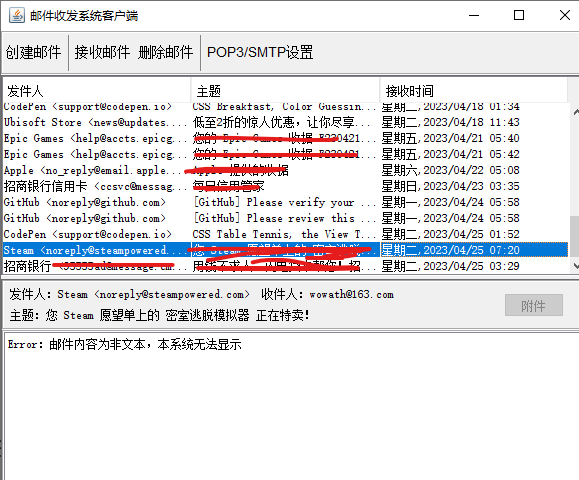


图6.2.3 邮件删除功能测试截图

6.2.4 附件发送功能测试

在测试附件发送功能时，我们需要先设置好邮件内容并添加要发送的附件。然后，使用不同的邮件客户端进行测试，看看单个附件和多个附件的发送功能是否正常。我们需要确认附件能够被正确地发送并且能够被正确地接收，同时需要检查附件的大小和类型是否符合要求。测试的过程中可以通过图示来查看测试结果。

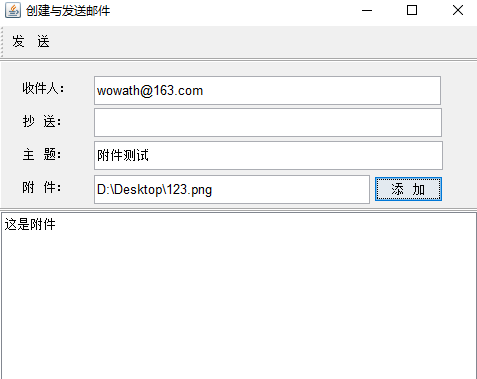


图6.2.4 附件发送功能测试截图

6.2.5 附件接受功能测试

在进行附件接收功能测试时，可以设置邮件内容并添加附件，对于邮件列表中一些邮件自带的附件，用户可以点击“附件”按钮，弹出“邮件附加处理”窗口，选中需要保存的附件并点击“保存”按钮，就可以将附件保存在自己的电脑上了。在保存附件时，需要确认保存的路径和文件名是否正确，避免出现错误。如图所示:

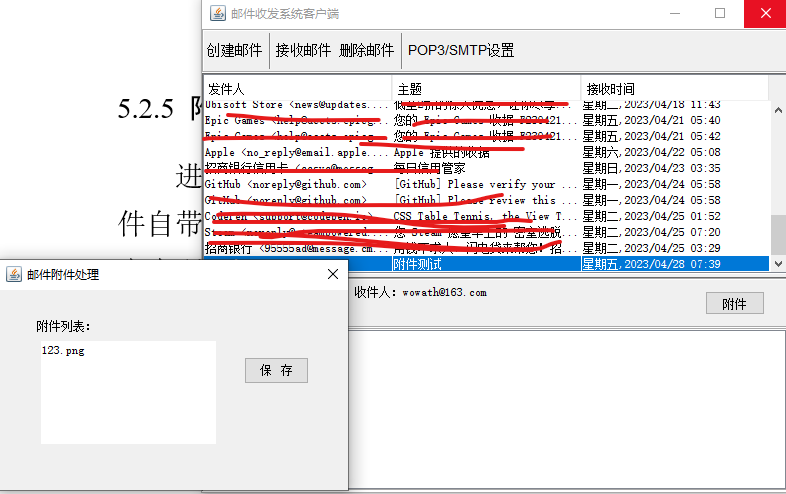


图6.2.5 附件接受功能测试截图

总的来说，使用JavaMail实现了一些基本的邮件收发功能，比如发送邮件、接收邮件、删除邮件，以及发送和接收附件。在实现这些功能之前，先需要了解SMTP、POP3和MIME等协议的基本原理和实现方法。然后使用JavaMail API来实现邮件收发模块，并进行了功能测试，确保软件能够正常运行。

6.3 本章小结

本章主要集中于系统的测试工作，其中包括了全面的系统测试以及各项功能测试。在功能测试部分，我们对邮件的发送、接收、删除功能进行了详尽的测试，同时也对附件的发送和接收进行了深入的试验。测试的过程中，验证了系统的各项功能，并发现了一些可以改进和完善的地方。

第7章 结论

本论文主要阐述了一个基于JavaMail的邮件收发系统的设计与实现过程。我们深入研究了JavaMail、SMTP协议、POP3协议、MIME协议等相关技术，并对系统的功能需求、安全需求、可靠性需求以及用户体验需求进行了仔细的分析。在此基础上，我们详细地说明了邮件收发系统的总体设计，并对每个功能模块进行了具体的设计。

在实现阶段，我们依赖JavaMail API来实现邮件发送、接收以及删除等功能，同时对附件的发送和接收进行了特殊处理。在系统测试阶段，我们对邮件发送、接收、删除以及附件发送和接收的功能进行了详尽的测试。测试结果表明，系统运行稳健，各项功能均为稳定可靠。

然而，我们的工作中也存在一些不足。例如，对于一些特殊情况，比如网络中断、邮件发送失败等，我们的处理措施还不够完善。这部分需要我们进一步改进系统的处理方式，以提高系统的稳定性和可靠性。

在未来的改进方向上，我们计划优化用户交互方式，以提升用户体验。同时，我们也会对系统进行持续优化和升级，以应对各种可能出现的特殊情况，从而进一步提高系统的稳定性和可靠性。总的来说，我们的目标是提供一个功能齐全、稳定可靠、用户友好的邮件收发系统。。

致谢

非常感谢我的导师对我的悉心指导，您的专业知识和热情让我深受启发，使我在研究过程中取得了很多进展。在我的研究中，您为我提供了很多宝贵的建议和指导，帮助我深入了解邮件收发软件的开发和实现。我从您那里学到了很多有用的知识和技能，这将对我的未来发展产生深远的影响。

同时，我还要感谢我的家人和朋友们对我一直以来的支持和鼓励。在我研究的艰苦时期，他们始终是我的坚强后盾，为我提供了无私的支持和鼓励。没有他们的帮助和理解，我很难完成这个学位论文。

我还要感谢我的大学生活，它为我提供了学习和成长的机会。在大学里，我认识了很多优秀的同学和老师，他们在我的学术和职业生涯中起到了重要的推动作用。大学里的学习和实践经历也让我更好地了解了自己的兴趣和能力，为未来的发展奠定了更坚实的基础。

最后，我要感谢所有支持我的人，包括老师、同学和朋友们。我将继续珍惜这段宝贵的经历，为未来的发展奠定更坚实的基础，并回馈社会，为社会做出更多的贡献。

参考文献

1. 来学伟.基于移动端的邮件系统的设计与开发[J].电脑知识与技术,2019,15(31):47-48.DOI:10.14004/j.cnki.ckt.2019.3662.
2. 刘佳. 基于AES算法的安全电子邮件系统研究与实现[D].黑龙江大学,2018.
3. 韩晓翠,刘印锋,张百聪,王迪,张陆雨,张宸.校园电子邮箱管理策略[J].电脑编程技巧与维护,2022(12):87-90.DOI:10.16184/j.cnki.comprg.2022.12.042.
4. 王圣雯,胡爱群.基于SM9算法的邮件加密系统设计[J].信息网络安全,2022,22(06):53-60.
5. 梁利,蔡先勇,李文杰,张冠楠,王海诚,刘博.电子邮件系统安全技术要求标准应用[J].信息技术与标准化,2022(05):76-78.
6. 樊子钊,边禹帆,巴一.基于James的院内邮件管理系统的实现[J].承德医学院学报,2022,39(02):172-174.DOI:10.15921/j.cnki.cyxb.2022.02.005.
7. 宋瑶,邱小彬,陈红茜等. 高校电子邮件系统建设分析[C]//中国计算机用户协会网络应用分会.中国计算机用户协会网络应用分会2021年第二十五届网络新技术与应用年会论文集.《计算机科学》编辑部,2021:128-131.DOI:10.26914/c.cnkihy.2021.047809.
8. 李文婷,张格,刘冬.电子邮件网络安全风险分析及应对[J].保密科学技术,2021(09):69-71.
9. 王骏翔.基于用户行为识别的企业级邮件安全分析平台设计与实现[J].上海船舶运输科学研究所学报,2020,43(04):59-64.
10. 尚菁菁,朱宇佳,刘庆云.电子邮件安全扩展协议应用分析[J].网络与信息安全学报,2020,6(06):69-79.
11. 牛淑芬,谢亚亚,杨平平,王彩芬,杜小妮.加密邮件系统中基于身份的可搜索加密方案[J].电子与信息学报,2020,42(07):1803-1810.
12. 朱次平,周燕.政府公务邮箱系统建设实践——以贵阳市公务邮箱系统建设为例[J].信息系统工程,2020(05):18-19.
13. 胡菊霞.基于Java语言的电子邮件系统的研究[J].职大学报,2020(02):75-77.
14. 赵睿斌,禄凯,闫桂勋,陈月华.政务邮件系统安全第三方评价方法研究[J].保密科学技术,2020(01):40-44.
15. 吉小鹏.企业邮箱及反垃圾邮件系统的部署与实现[J].电子元器件与信息技术,2019,3(12):18-20.DOI:10.19772/j.cnki.2096-4455.2019.12.008.
16. 王露阳,罗国富.高校自建邮件系统的账户安全管理优化研究[J].计算机系统应用,2019,28(12):232-237.DOI:10.15888/j.cnki.csa.007204.
17. 张璐.电子邮件系统跨网络区域多防火墙分析[J].山西电力,2019(05):35-37.
18. 牛淑芬,杨平平,谢亚亚,王彩芬,杜小妮.电子邮件系统中指定服务器的关键字搜索加密方案[J].计算机工程,2020,46(10):137-142+150.DOI:10.19678/j.issn.1000-3428.0055654.
19. 蒲小英,付敏.电子邮件系统安全特性及防护方向[J].电脑知识与技术,2019,15(28):38-39.DOI:10.14004/j.cnki.ckt.2019.3539.
20. 周迪贵.企业邮件系统数据库切换故障分析与处理[J].现代工业经济和信息化,2019,9(02):73-76.DOI:10.16525/j.cnki.14-1362/n.2019.02.30.
21. 陈宏媛. 基于邮件系统的动态可搜索加密研究[D].华东师范大学,2019.
22. 宋德强,高建.基于Java语言的电子邮件系统的研究与实现[J].数字通信世界,2019(03):83.
23. 常晓洁,徐锋,郭晔,厉晓华.基于私有云的邮件传输处理优化算法[J].通信学报,2018,39(S1):195-199.
24. 张晔.基于Java Web的高校内部邮箱系统设计[J].电脑知识与技术,2018,14(23):129-130.DOI:10.14004/j.cnki.ckt.2018.2578.
25. Lee Mundy. NAVSUP Enterprise Mail System Boosting Efficiencies and Offering Significant Customer Improvements[J]. Newsletter - United States. Navy Supply Corps,2021.
26. Piedrahita Castillo Diego,Regidor Francisco Machío,Higuera Javier Bermejo,Higuera Juan Ramón Bermejo,Montalvo Juan Antonio Sicilia. A New Mail System for Secure Data Transmission in Cyber Physical Systems[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,2020,28(Supp02).
27. Zuhar Musliyana,Andita Ghaitsa Satira,Mahendar Dwipayana,Ayu Helinda. Integrated Email Management System Based Google Application Programming Interface Using OAuth 2.0 Authorization Protocol[J]. Elkawnie,2020,6(1).
28. Fang Lei. The Safety precautions and Design of Mail System[J]. IOP Conference Series: Materials Science and Engineering,2020,740.
29. Ismail Roayat,Abdelsalam Lamiaa,Ahmed Mahmoud. An efficient Secure Electronic Mail System based on Elliptic Curve Certificateless Signcryption[J]. International Journal of Computer Applications,2019,177(20).
30. Ajai Verma. Voice based Electronic Mail System for Visually Challenged Individuals[J]. International Journal of Innovative Technology and Exploring Engineering (IJITEE),2019,8(12s).
31. Emailing System with Elevated Safeguard Procedures[J]. International Journal of Innovative Technology and Exploring Engineering,2019,8(9S3).