

2018



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

Neusoft - mail 邮件系统

概要设计说明书

李林峰
北京理工大学计算机学院
2018 年 9 月 5 日



概要设计说明书

项目名称	Linux 系统下 Neusoft - mail 邮件系统
项目组长	李林峰
团队成员	马伯乐、沈云汉、卢茜君、尚楠、 陈华琳、覃靖文
单位	北京理工大学计算机学院软件工程系
版本号	2.4.15
发布日期	2018 年 9 月 5 日

二〇一八年·北京

版本记录

编号	修改内容	文档编号	修改日期	是否评审	评审日期
1	创建文档	0.0.0	2018/9/4	否	
2	初步撰写文档	1.0.0	2018/9/4	否	
3	修改用例图	1.1.1	2018/9/4	否	
4	完善模块结构图	1.1.2	2018/9/4	否	
5	完善模块流程图	1.1.3	2018/9/4	否	
6	完善模块时序	1.1.4	2018/9/4	否	
7	完善模块软件接口	1.2.5	2018/9/4	是	2018/9/4
8	修订模块设计	2.1.6	2018/9/5	否	
9	完善数据库表单设计	2.2.9	2018/9/5	是	2018/9/5
10	界面设计图	2.3.10	2018/9/5	否	
11	界面描述	2.3.11	2018/9/5	否	
12	最终发布	2.4.15	2018/9/5	是	2018/9/5

目录

§1. 文档的目标及范围	1
1.1. 文档概述.....	1
1.2. 名词解释.....	1
1.3. 关联文档.....	1
1.4. 参考文献.....	1
§2. 设计概述.....	2
2.1. 设计目标.....	2
2.2. 设计说明.....	2
§3. 体系结构设计.....	3
3.1. 系统用例图及其描述.....	3
3.2. 系统结构图.....	4
3.3. 模块划分.....	6
3.4. 状态转化图.....	6
§4. 模块设计.....	7
4.1. 注册模块设计.....	7
4.1.1. 模块功能点列表.....	7
4.1.2. 模块整体结构设计	7
4.1.3. 模块流程设计	8
4.1.4. 模块时序设计	9
4.1.5. 软件接口设计	10
4.1.6. 硬件接口设计	11
4.2. 登录模块设计.....	11
4.2.1. 模块功能点列表.....	11
4.2.2. 模块整体结构设计	11
4.2.3. 模块流程设计	12
4.2.4. 模块时序设计	13
4.2.5. 软件接口设计	14
4.2.6. 硬件接口设计	15
4.3. 发送邮件模块设计.....	15
4.3.1. 模块功能点列表.....	15

4.3.2. 模块整体结构设计	16
4.3.3. 模块流程设计	17
4.3.4. 模块时序设计	18
4.3.5. 软件接口设计	19
4.3.6. 硬件接口设计	20
4.4. 接收邮件模块设计	20
4.4.1. 模块功能点列表	20
4.4.2. 模块整体结构设计	20
4.4.3. 模块流程设计	21
4.4.4. 模块时序设计	22
4.4.5. 软件接口设计	23
4.4.6. 硬件接口设计	24
4.5. 管理邮件模块设计	24
4.5.1. 模块功能点列表	24
4.5.2. 模块整体结构设计	24
4.5.3. 模块流程设计	25
4.5.4. 模块时序设计	26
4.5.5. 软件接口设计	27
4.5.6. 硬件接口设计	29
4.6. 草稿箱模块设计	30
4.6.1. 模块功能点列表	30
5.6.2. 模块整体结构设计	30
5.6.3. 模块流程设计	31
5.6.4. 模块时序设计	32
4.6.5. 软件接口设计	33
4.6.6. 硬件接口设计	35
4.7. 通讯录模块设计	35
4.7.1. 模块功能点列表	35
4.7.2. 模块整体结构设计	35
4.7.3. 模块流程设计	36
4.7.4. 模块时序设计	37
4.7.5. 软件接口设计	38
4.7.6. 硬件接口设计	39
§5. 数据设计	40

5.1. 数据库设计说明.....	40
5.2. 数据描述与实体关系模型.....	40
5.3. 数据文件设计.....	41
5.3.1. UserInfo 表.....	41
5.3.2. Mail 表.....	41
5.3.3. Attachment 表.....	42
5.3.4. Draft 表.....	42
5.3.5. AddressBook 表.....	42
§6. 界面设计.....	43
6.1. 登录界面.....	43
6.2. 注册界面.....	44
6.3. 邮件系统主页.....	45
6.4. 邮件编写界面.....	46
6.5. 收件箱.....	47
6.6. 通讯录.....	48
6.7. 发件箱.....	49
6.8. 邮件查看.....	50
6.9. 草稿箱.....	51
6.10. 回收站.....	52

§1. 文档的目标及范围

1.1. 文档概述

本文档定义了该邮件系统使用的领域和必须满足的约束条件，确定了系统各个部分之间的接口和通信，说明了该系统的体系结构设计、数据设计、界面设计，从而对该产品的结构、方法等进行了详细的描述。

编写该文档便于设计人员进行沟通和理解，便于项目管理人员指导该系统的开发工作，有效控制系统开发的进程。该文档是设计结果的详细描述，它也是程序员编写程序的依据。

1.2. 名词解释

表 1. 术语及缩略语的解释说明

序号	术语	说明
1	C/S 模式	客户端/服务器模式(一种分布式体系结构模型)
2	API	应用程序编程接口
3	用户界面层	模式结构中对用户展示部分
4	业务逻辑层	模式结构中对用户输入处理部分
5	应用逻辑层	模式结构中对来自客户端输入处理部分
6	数据访问层	模式结构实现数据访问的部分

1.3. 关联文档

[1] Linux 系统下 Neusoft - mail 邮件系统项目要求书. V1.0.

[2] Neusoft - mail 邮件系统需求跟踪矩阵.

1.4. 参考文献

[1] 张黎. C/S 模式下的电子邮件系统设计与实现[J]. 科技资讯, 2012(02):45-49.

[2] 廖常武. 分布式邮件系统设计[J]. 计算机与现代化, 2005(07):90-91.

§2. 设计概述

2.1. 设计目标

本概要设计的目标是：完成系统设计，明确模块的划分和接口定义。明确功能分解后模块实现点。完成结构设计，包括确立合理的模块结构和构成，对重要数据结构相关的部分做概要设计，完成内部各组成部分的功能和划分。完全确定本模块的内外部接口。并对外针对系统设计中涉及本模块提出的需求和资源进行相关设计。最终得到一个高内聚、低耦合，具有高可靠性、高可维护性、高可理解性和高效的系统模型，为提高系统实现的质量提供坚实的基础。

2.2. 设计说明

表 2. 开发平台与计算机资源使用情况

No	项目	描述		备注
1	操作系统	Centos Linux 操作系统		虚拟机
2	技术规范	设计规范	部门最新设计模板	-
		编码规范	部门最新编码规范	-
		单体测试规范	暂无	-
		维护规范	暂无	-
3	开发与调试环境	计算机型号	Lenovo 拯救者 Y7000 - 1060	-
		编程语言	C++	-
		编译器	g++ 4.4.6	-
		调试器	GNU gdb (GDB) Red Hat Enterprise Linux (7.2-56.el6)	-
		硬件调试环境	CPU: i7 - 8750H 9MB 缓存 内存: 8G	-
4	技术难点说明	(1). 网络编程中 socket “套接字” 的使用; (2). 数据库 MySQL 的使用。		-

§3. 体系结构设计

3.1. 系统用例图及其描述

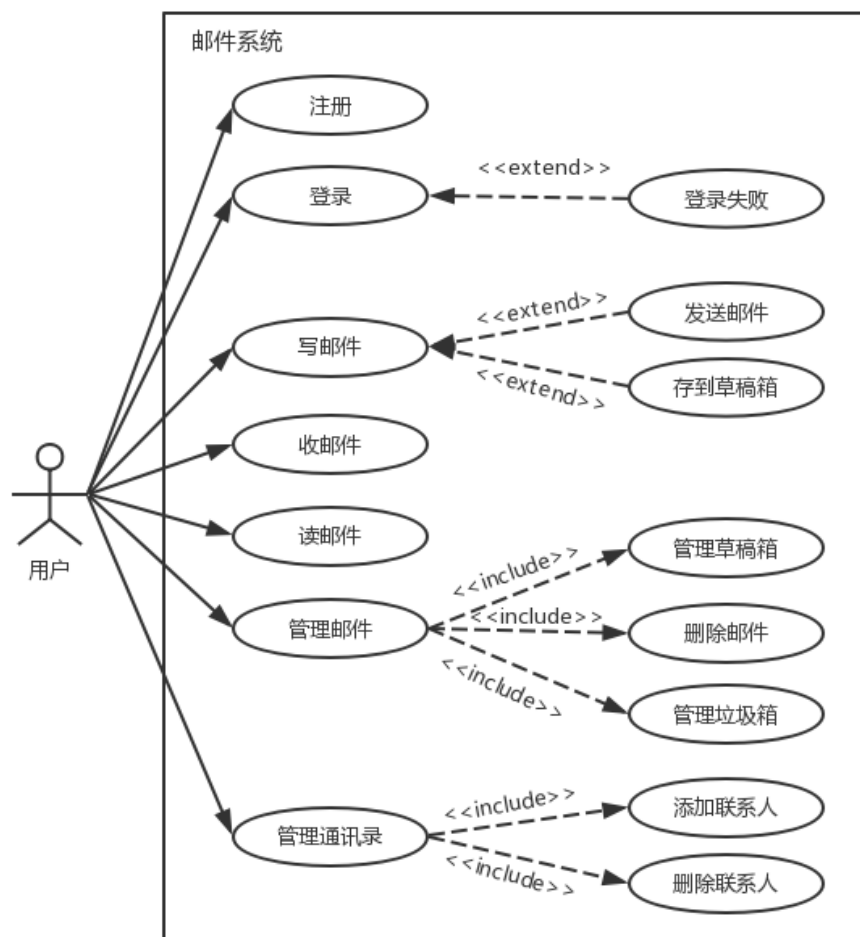


图 1. 邮件系统用例图

用例图综述：“邮件系统”通过用户完成系统功能。用户可以注册 / 登录该系统，进入系统后，可以进行邮件的收发和管理，编写邮件时可以直接将邮件发送，也可以将邮件存到草稿箱。管理邮件包括对垃圾箱和草稿箱的管理，以及删除邮件。还可以对通讯录进行管理，添加和删除联系人。

参与者：用户。

用例名称：注册、登录、写邮件、收邮件、读邮件、管理邮件、管理通讯录。

基本事件流：用户填写相关信息登录 / 注册进入系统，对邮件进行查看、收发和管理，管理联系人。

扩展事件流：无。

关系描述:

“登录”用例与“登录失败”之间存在“扩展关系”;

“写邮件”用例与“发送邮件”和“存到草稿箱”用例存在“扩展关系”;

“管理邮件”用例“包含”了“管理草稿箱”、“删除邮件”、“管理垃圾箱”用例;

“管理通讯里”用例“包含”了“添加联系人”和“删除联系人”两个用例。

前置条件: 用户先注册、登录进入邮件系统。

后置条件: 无。

异常: 无。

限制: 注册用户名符合要求, 账号无重复。

3.2. 系统结构图

本系统采用客户端/服务器模式(Client Server Model)的分布结构。系统运行时, 采用“请求—响应—结果”模式来实现。

本邮件系统使用三层网络设计模式: 用户界面、逻辑应用、数据访问。其中, 逻辑应用分为两个部分。与客户端计算的有关逻辑放在客户端完成, 需要频繁访问数据的逻辑部署在服务器端。

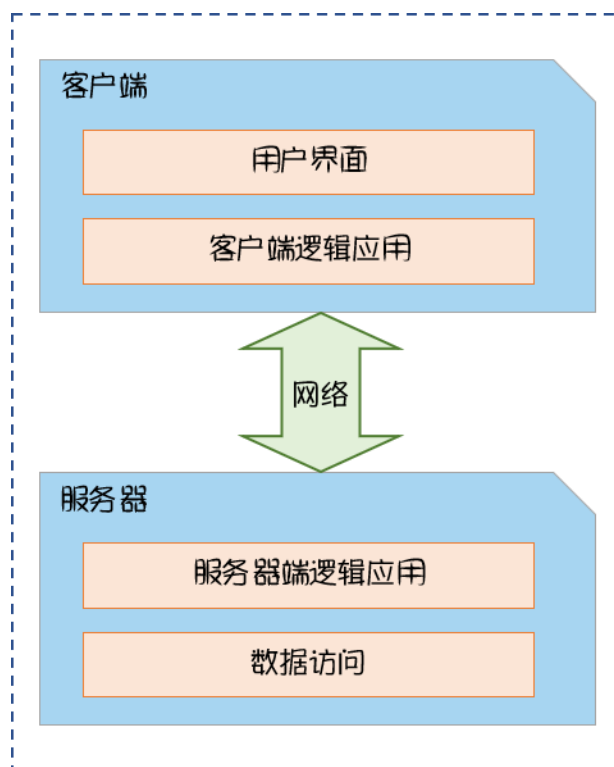


图 2. 邮件系统的分布式结构

客户端和服务端之间通信协议采用 TCP 协议，根据此协议传送数据包，数据包数据的传输格式统一为 Json 格式。实现客户端和服务器的数据交互。一次完整的数据交互过程如下。

客户端又分为用户界面层和业务逻辑层。在用户界面，用户可以根据提示进行注册和登录，登录之后可以写邮件、收邮件、查看通讯录、查看已发邮件、删除和恢复邮件、查看回收站等。业务逻辑层将用户在界面的输入进行处理，将其打包成统一格式的数据包。通过网络，将访问请求和数据包发送给服务器。服务器应用逻辑层一旦接受到访问请求，接收数据包，在应用逻辑层进行处理，将数据包解析成具体指令，然后根据指令访问数据库。数据库根据具体指令，查询数据文件并返回相关消息。服务器应用逻辑层接收到数据库传回的消息后，将其打包成统一格式的数据包，并向客户端发送响应及具体数据包。客户端业务逻辑层接收到数据包后，将其解析成具体数据，反馈给界面。

详细结构图如下。

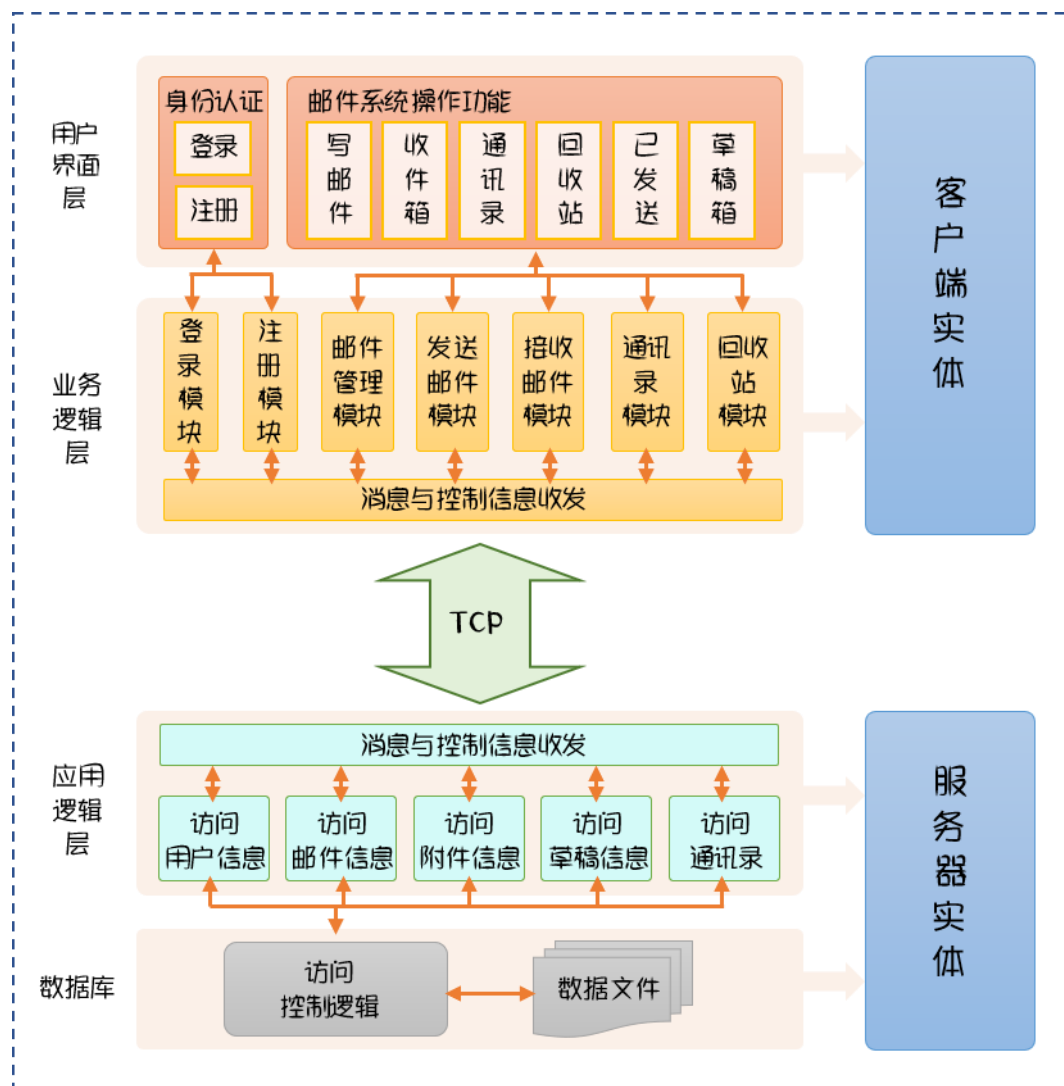


图 3. 邮件系统详细架构

3.3. 模块划分

表 3. 邮件系统模块划分

编号	模块名称	功能说明	备注
1	注册模块	用户输入注册信息注册账号。	-
2	登录模块	用户输入用户名和密码进入系统。	-
3	发送邮件模块	添加联系人，编写邮件，可以添加附件。将编写完成的邮件发送或者是存到草稿箱。	-
4	接收邮件模块	接收新的邮件，查看邮件内容。	-
5	管理邮件模块	对邮件进行管理，分为管理回收站、删除邮件两个模块。回收站可以对已删邮件进行恢复。	-
6	草稿箱模块	存储草稿邮件，以便再次编辑。	-
7	通讯录模块	显示已有联系人，可以对已有联系人进行修改或删除，也可以添加新的联系人。	-

3.4. 状态转化图

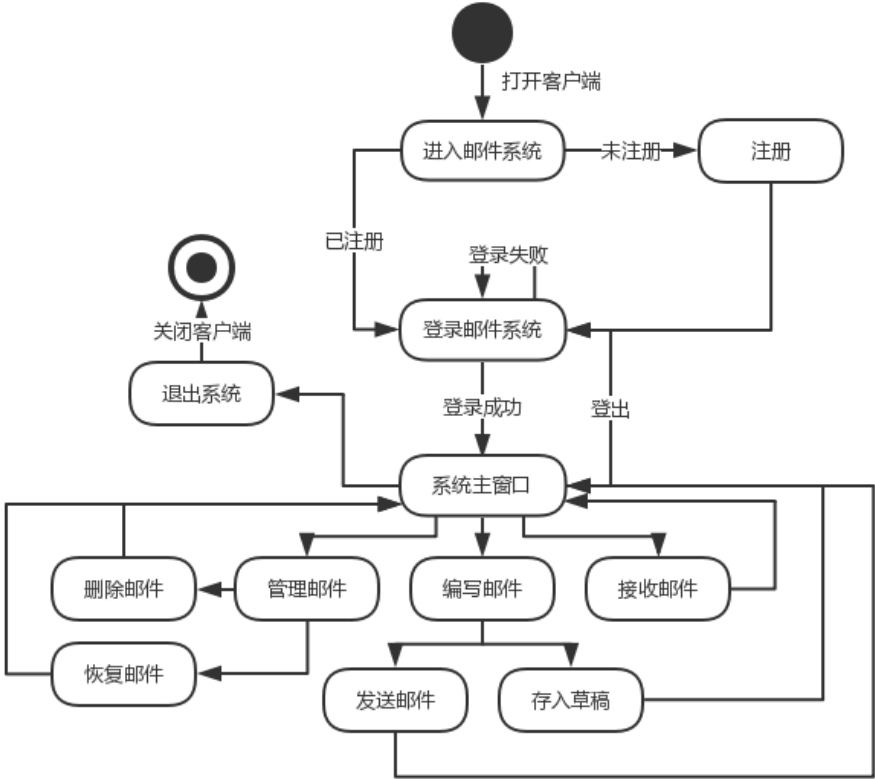


图 4. 邮件系统状态转化图

§4. 模块设计

4.1. 注册模块设计

4.1.1. 模块功能点列表

表 4. 注册模块功能点列表

序号	功能点	详细说明
1	输入	用户输入用户名和密码进入系统。
2	发送访问请求	界面向服务器发送请求。
3	服务器响应	向数据库查询、追加数据文件。返回信息

4.1.2. 模块整体结构设计

➤ 外部关系

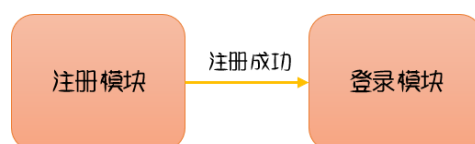


图 5. 注册模块外部关系示意图

➤ 内部结构

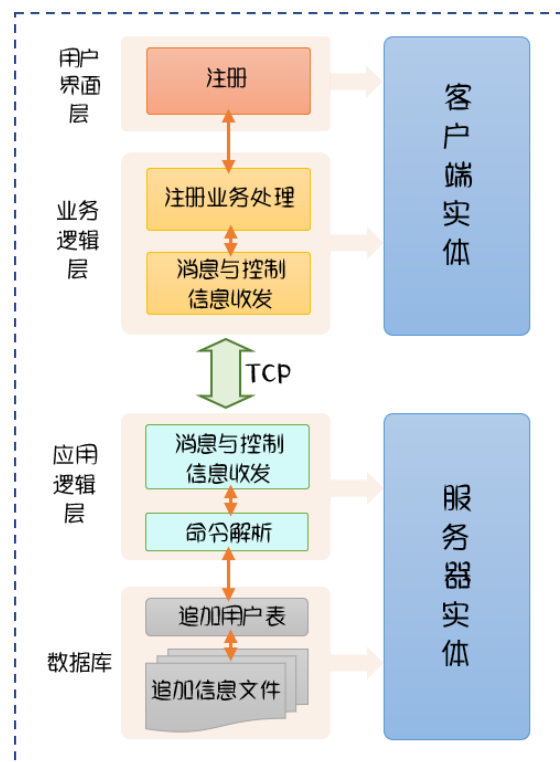


图 6. 注册模块内部结构示意图

4.1.3. 模块流程设计

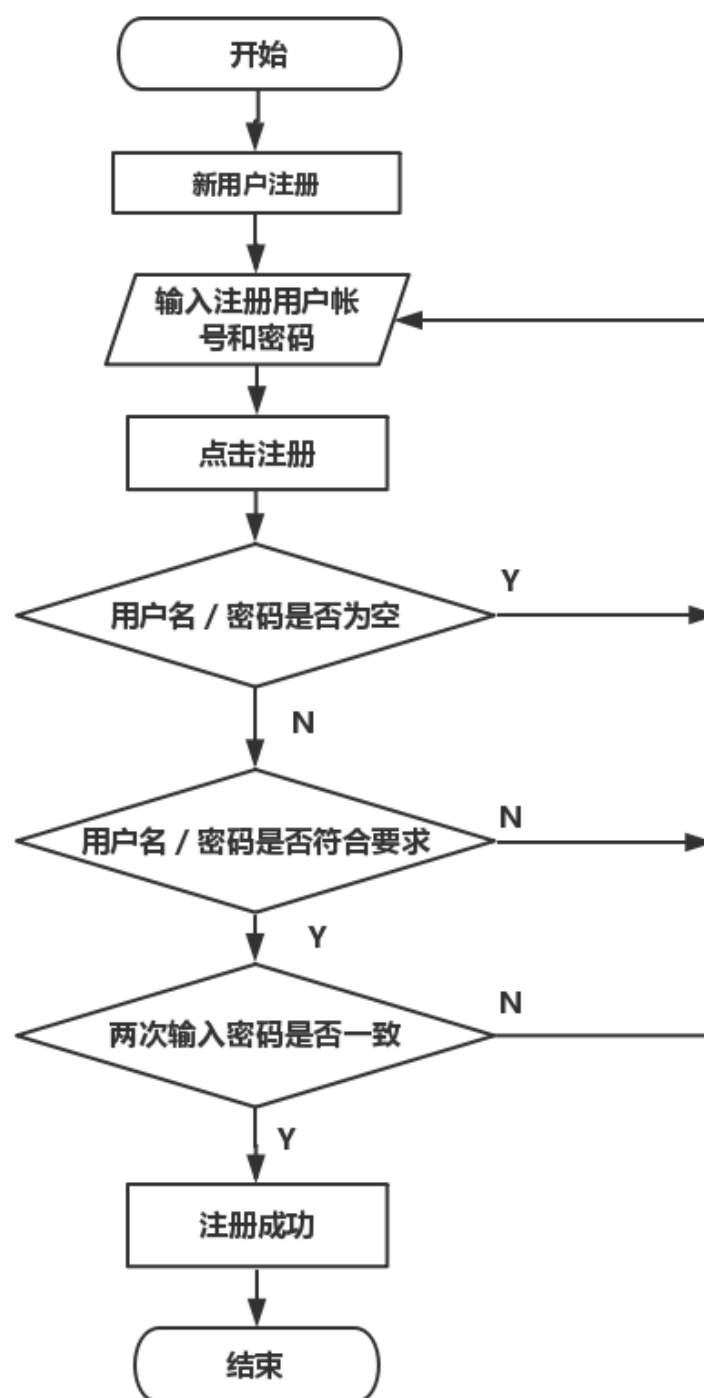


图 7. 注册模块流程图

4.1.4. 模块时序设计

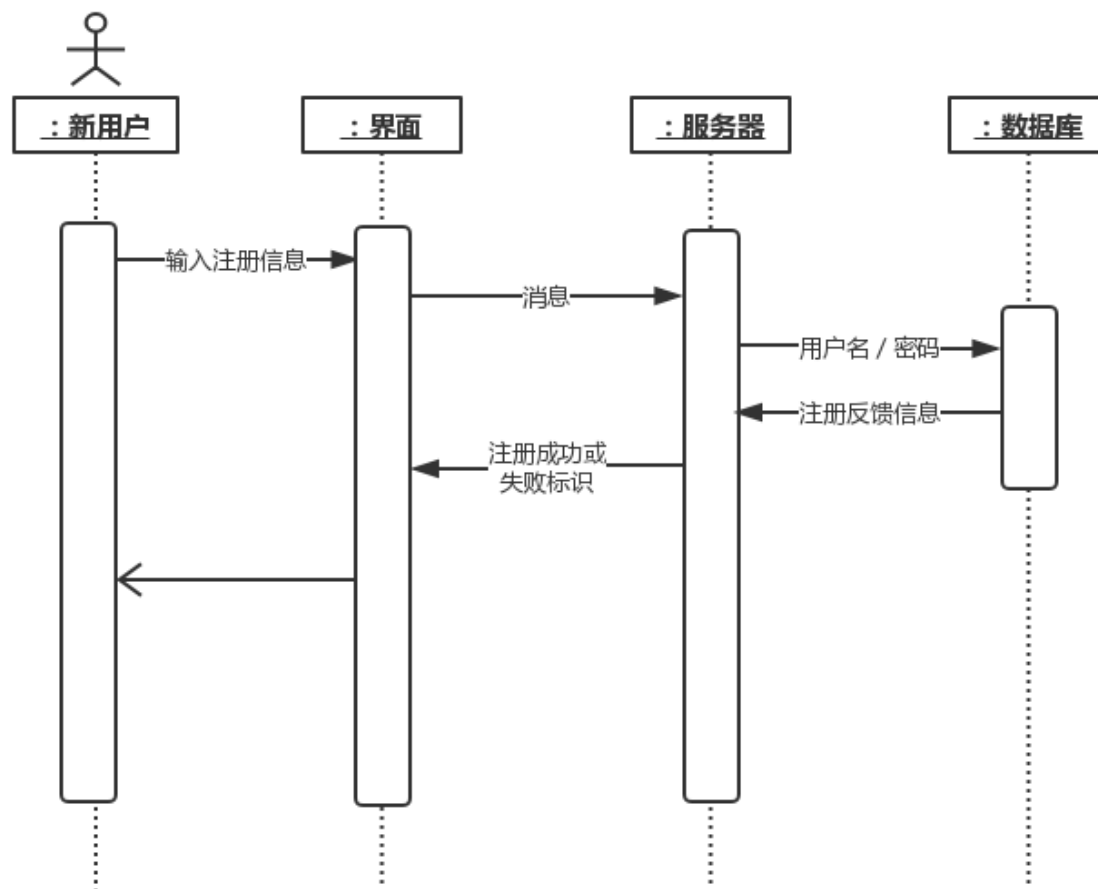


图 8. 注册模块时序图

顺序图综述：上图描述了“注册”用例的顺序图，涉及新用户、界面、服务器、数据库 4 个对象。

参与者对象描述：“新用户”是参与者，“界面”、“服务器”、“数据库”是三个对象。界面负责与用户和服务器进行交互，服务器负责接收来自界面的消息，查询数据库并对消息进行回复。数据库负责存储数据和对来自服务器的数据进行判断同时返回信息。

消息描述：“注册”的顺序是通过消息发送的前后关系得到体现。通过用户发送“用户名”和“密码”消息到界面，然后界面传递“消息”给服务器，服务器开始响应，将“用户名 / 密码”消息传递给数据库，并且等待数据库的响应，数据库接到消息后开始查询判断，将数据存储，之后返回查询数据给服务器，服务器根据数据库返回的消息向界面发送“注册成功或者失败的标识”消息，同时，界面将消息反馈给用户。

其他描述：无。

4.1.5. 软件接口设计

表 5. 注册模块 Register 接口说明

函数名	Register		
文件名	/Server/server.cpp		
功能概要	服务器处理用户注册		
参数			
类型	变量名	I/O	说明
QTcpSocket*	tcpSocket		客户端套接字
QString	username		用户名
QString	password		密码
返回值	类型	void	说明
	值		
详细说明			
客户端发出注册请求时调用，将用户名和密码传送给数据库进行查询，得到结果后通知客户端。			
使用注意事项			

表 6. 注册模块 AddUser 接口说明

函数名	AddUser			
文件名	/Server/database.cpp			
功能概要	数据库处理用户注册			
参数				
类型	变量名		I/O	说明
User	user			用户类，包含用户名和密码
返回值	类型	bool	说明	
	值	true	注册成功	
		false	注册失败	
详细说明				
服务器收到客户端发出的注册请求后调用，接收用户名和密码，在数据库中查询是否存在重名用户，若存在则返回注册失败，否则录入用户信息并返回成功。				
使用注意事项				

4.1.6. 硬件接口设计

无。

4.2. 登录模块设计

4.2.1. 模块功能点列表

表 7. 登录模块功能点列表

序号	功能点	详细说明
1	输入	用户输入用户名和密码进入系统。
2	发送访问请求	界面向服务器发送请求。
3	数据库响应	数据库查询数据文件。
4	跳转页面	跳转至主页。

4.2.2. 模块整体结构设计

➤ 外部关系

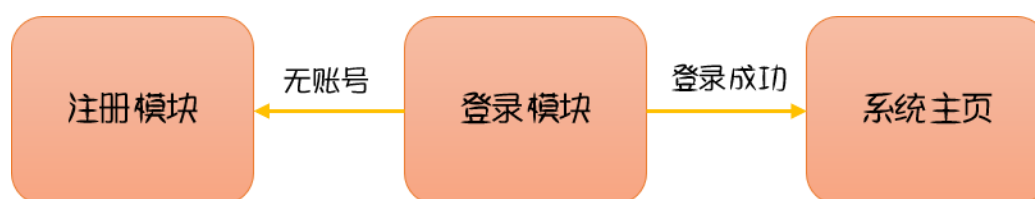


图 9. 登录模块外部关系示意图

➤ 内部结构

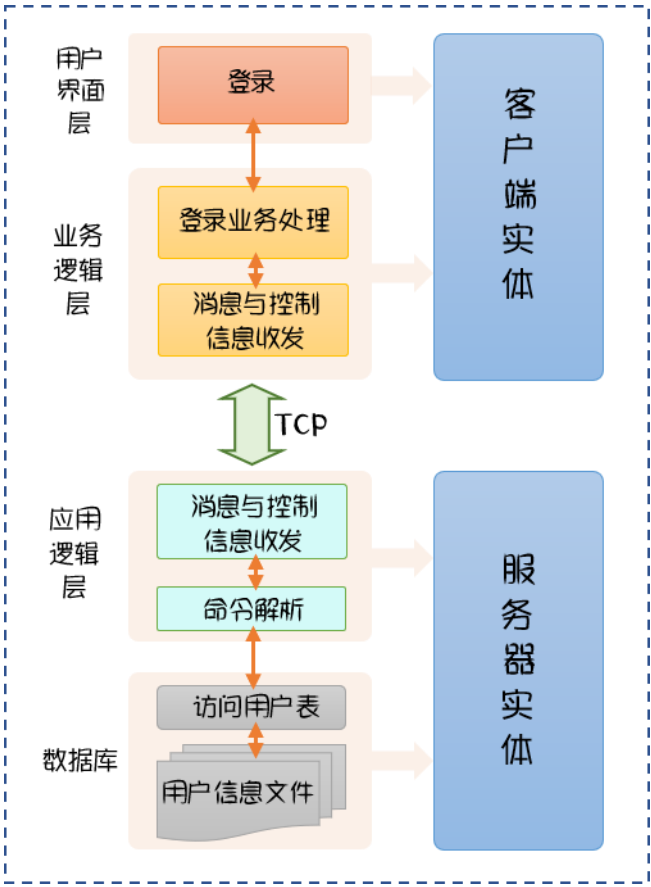


图 10. 登录模块内部结构示意图

4.2.3. 模块流程设计

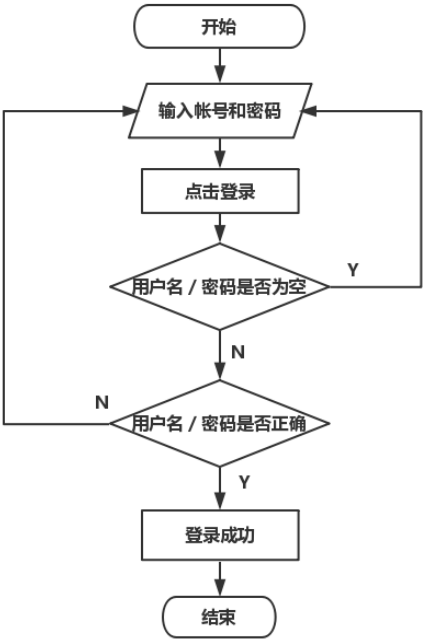


图 11. 登录模块流程图

4.2.4. 模块时序设计

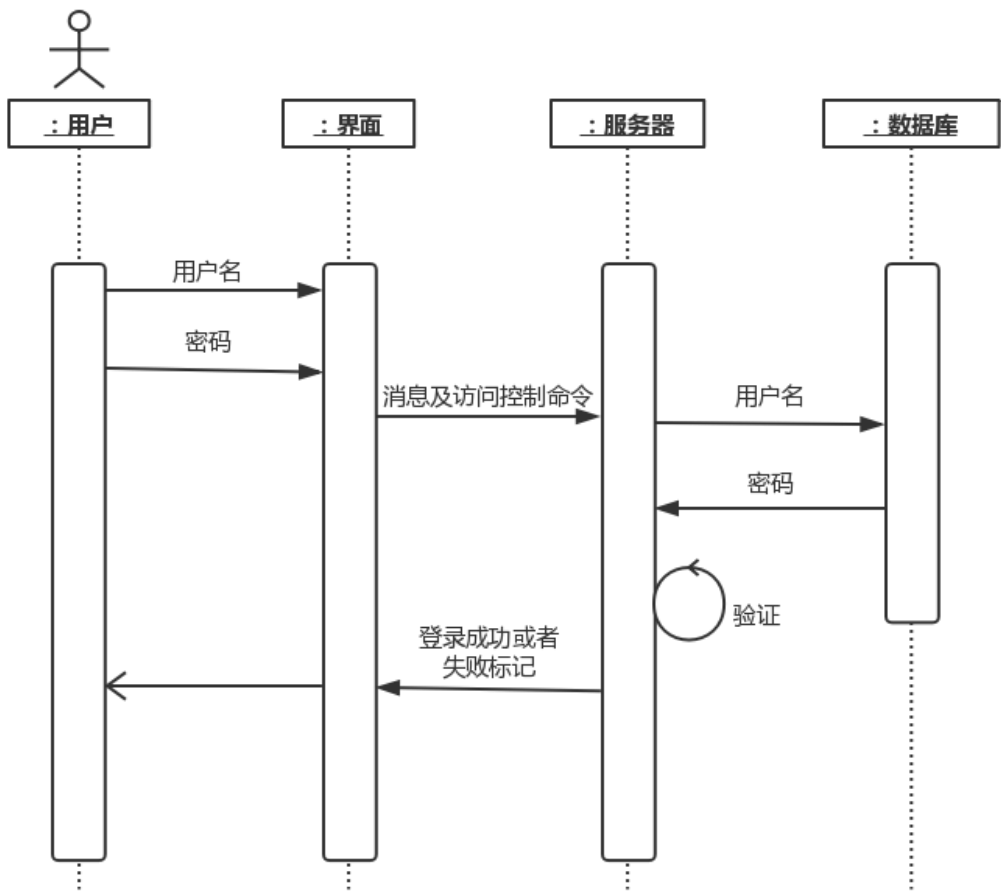


图 12. 登录模块时序图

顺序图综述：图描述了“登录”用例，涉及用户、界面、服务器和数据库 4 个对象。

参与者对象描述：“用户”是参与者，“界面”、“服务器”和“数据库”是三个对象。界面负责与用户进行交互，服务器负责接收来自界面的消息及访问控制命令，并对消息进行验证。数据库负责存储数据和对来自服务器的数据进行查询同时返回信息。

消息描述：“登录”的顺序是通过消息发送的前后关系得到体现。通过用户发送“用户名”和“密码”消息到界面，然后界面传递“消息及访问控制命令”给服务器，服务器开始响应，将“用户名”消息传递给数据库，并且等待数据库的响应，数据库接到消息后开始查询，之后返回查询数据给服务器，服务器根据数据库返回的“密码”消息进行用户名和密码的验证，向界面发送“登录成功或者失败的标记”消息，同时，界面将消息反馈给用户。

其他描述：无。

4.2.5. 软件接口设计

表 8. 登录模块 Login 接口说明

函数名	Login		
文件名	/Server/server.cpp		
功能概要	服务器处理用户登录		
参数			
类型	变量名	I/O	说明
QTcpSocket*	tcpSocket		客户端套接字
QString	username		用户名
QString	password		密码
返回值	类型	void	说明
	值		
详细说明			
客户端发出登录请求时调用，将用户名和密码传送给数据库进行查询，得到结果后通知客户端。			
使用注意事项			

表 9. 登录模块 FindUser 接口说明

函数名	FindUser			
文件名	/Server/database.cpp			
功能概要	数据库查询用户			
参数				
类型	变量名		I/O	说明
User	user			用户类，包含用户名和密码
返回值	类型	uint32_t	说明	
	值	userId		
详细说明				
需要查询用户信息时调用，接收用户名和密码，在数据库中查询是否存在该用户及密码是否匹配，若合法则返回用户 ID，否则返回 0。				
使用注意事项				

表 10. 登录模块 Logout 接口说明

函数名	Logout			
文件名	/Server/server.cpp			
功能概要	服务器处理用户登出			
参数				
类型	变量名		I/O	说明
QTcpSocket*	tcpSocket			客户端套接字
返回值	类型	void	说明	
	值			
详细说明				
客户端发出登出请求和客户端关闭时调用，抹除在线用户集合中此客户端套接字信息。				
使用注意事项				

4.2.6. 硬件接口设计

无。

4.3. 发送邮件模块设计

4.3.1. 模块功能点列表

表 11. 发送邮件模块功能点列表

序号	功能点	详细说明
1	编写邮件	进入“写信”界面，对信件内容进行编辑。
2	添加附件	点击“添加附件”，可以添加附件。
3	将邮件存到草稿箱	将写好的邮件保存到草稿箱。
4	将邮件发送	将写好的邮件发送。
5	数据库写入	数据库将邮件信息写入。

4.3.2. 模块整体结构设计

外部关系

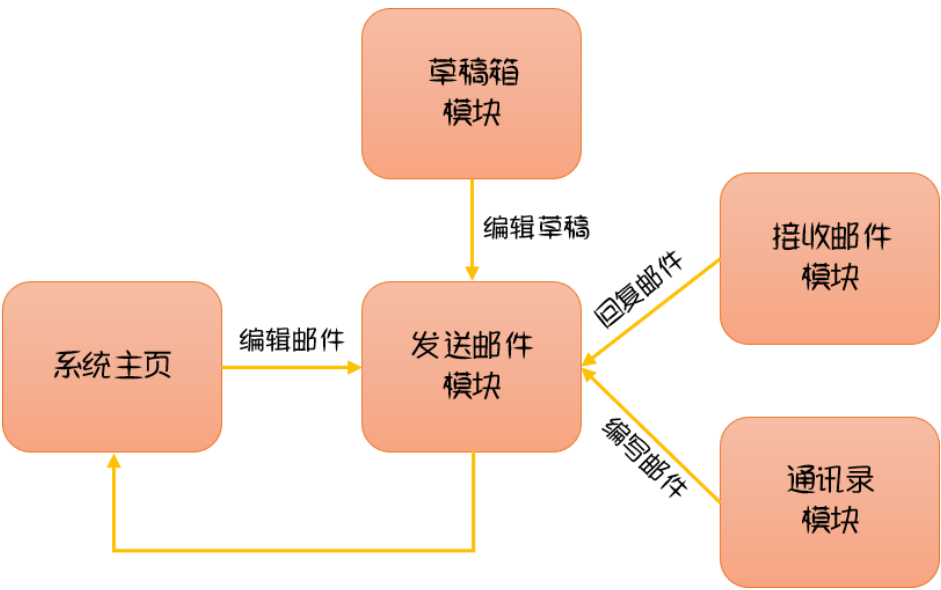


图 13. 发送邮件模块外部关系示意图

内部结构

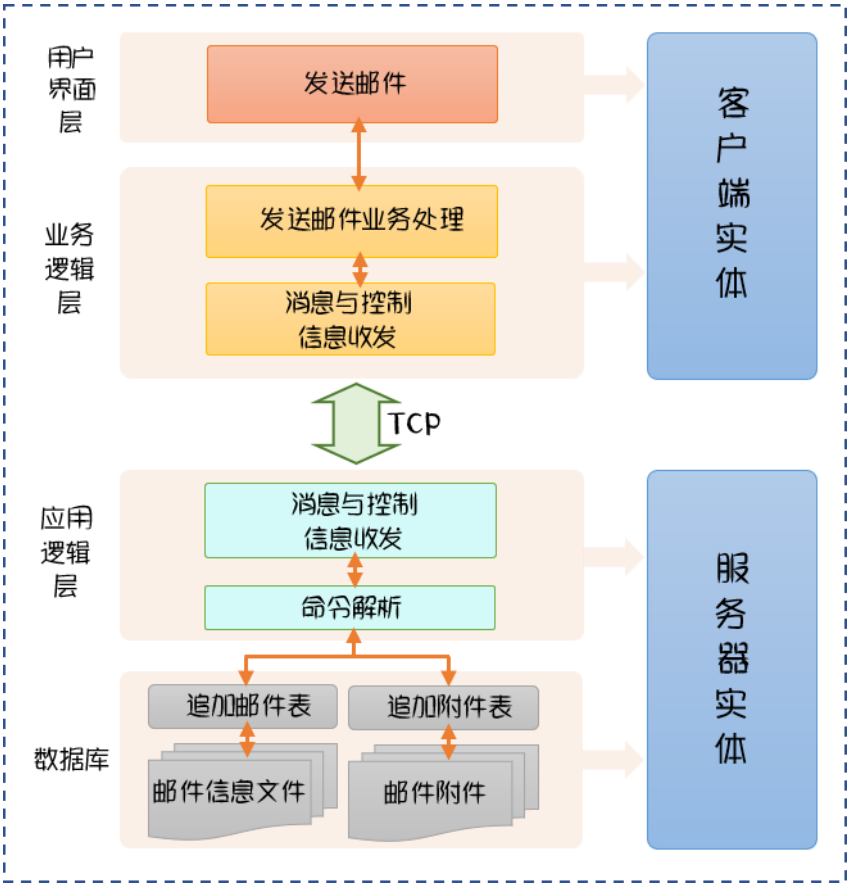


图 14. 发送邮件模块内部结构示意图

4.3.3. 模块流程设计

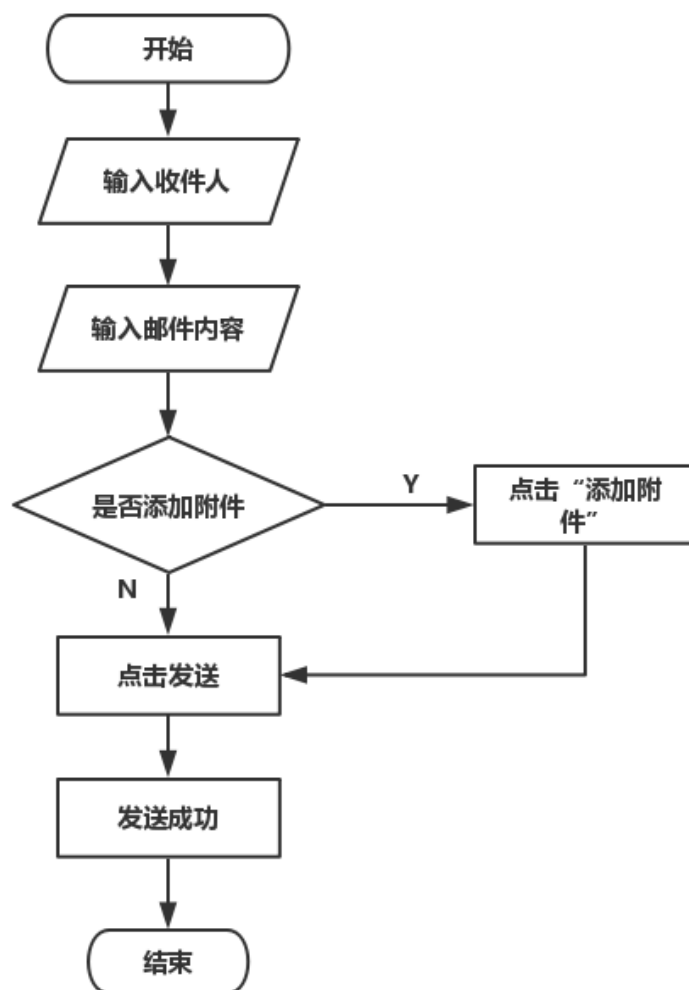


图 15. 发送邮件模块流程图

4.3.4. 模块时序设计

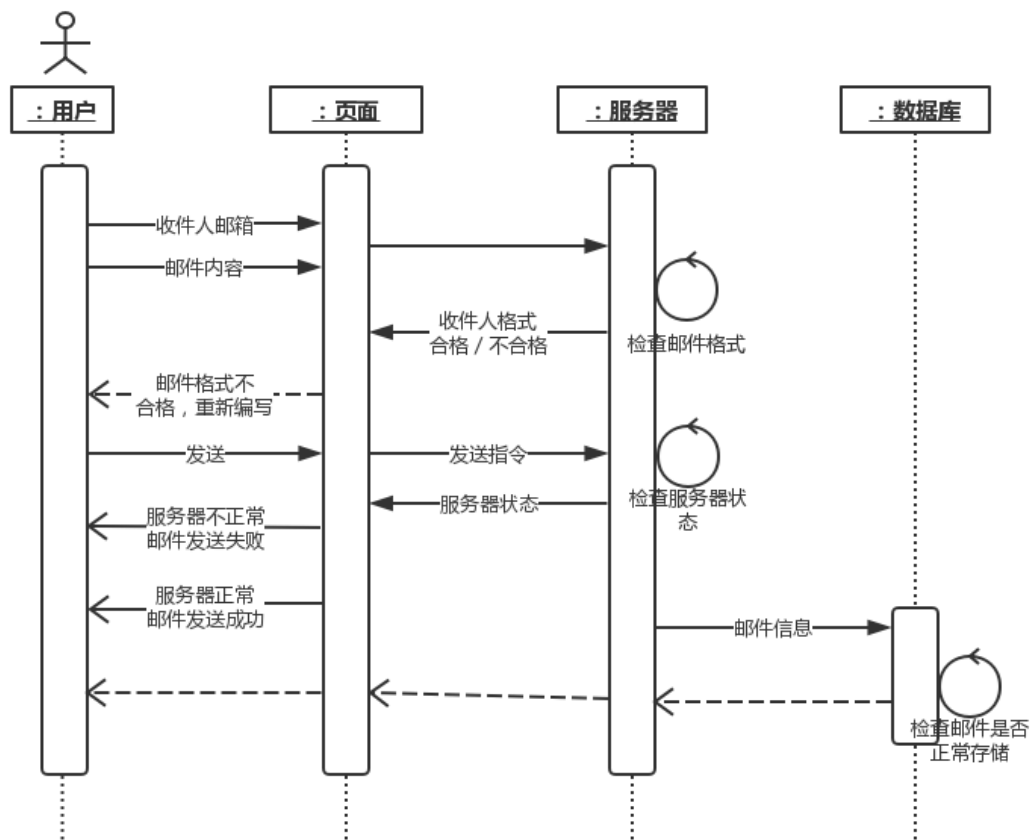


图 16. 发送邮件模块时序图

顺序图综述：图描述了“发送邮件”用例，涉及用户、界面、服务器和数据库 4 个对象。

参与者对象描述：“用户”是参与者，“界面”、“服务器”、“数据库”是三个对象。界面负责与用户和服务器进行交互，服务器负责接收来自界面的消息，将数据传给数据库并对消息进行回复。数据库负责存储数据同时返回信息。

消息描述：“发送邮件”的顺序是通过消息发送的前后关系得到体现。通过用户发送“收件人邮箱”和“邮件内容”消息到界面，然后界面传递消息给服务器，服务器开始响应，对邮件的收件人邮箱格式进行检查，格式不合格，界面向用户发送“邮件格式不正确，重新编写”消息，邮件格式合格用户向界面发出“发送”消息，界面对服务器传递该消息，服务器检查状态，返回“服务器状态”消息给界面，界面向用户反馈“服务器不正常，邮件发送失败”或者“服务器正常，邮件发送成功”消息。服务器将“邮件信息”消息传递给数

据库，数据库接到消息后，将数据存储，之后返回查询数据给服务器。

其他描述：无。

4.3.5. 软件接口设计

表 12. 发送邮件模块 SendEmail 接口说明

函数名	SendEmail		
文件名	/Server/sever.cpp		
功能概要	服务器处理发送邮件		
参数			
类型	变量名	I/O	说明
QTcpSocket*	tcpSocket		客户端套接字
uint32_t	sendId		发件人 ID
uint32_t	recId		收件人 ID
QString	content		邮件内容
返回值	类型	void	说明
	值		
详细说明			
客户端发送邮件时调用，将发件人 ID，收件人 ID 和邮件内容传入数据库，通知客户端邮件发送成功。			
使用注意事项			

表 13. 发送邮件模块 AddMail 接口说明

函数名	AddEmail			
文件名	/Server/database.cpp			
功能概要	数据库增加邮件信息			
参数				
类型	变量名		I/O	说明
Mail	mail			Mail 类，包含发件人 ID、收件人 ID 和邮件内容
返回值	类型	void	说明	
	值			
详细说明				

服务器收到客户端发出的邮件后调用，接收发件人 ID、收件人 ID 和邮件内容，存入数据库。
使用注意事项

4.3.6. 硬件接口设计

无。

4.4. 接收邮件模块设计

4.4.1. 模块功能点列表

表 14. 接收邮件模块功能点列表

序号	功能点	详细说明
1	服务器口令	服务器发送指令
2	界面显示新邮件	从数据库读入新邮件信息。

4.4.2. 模块整体结构设计

➤ 外部关系

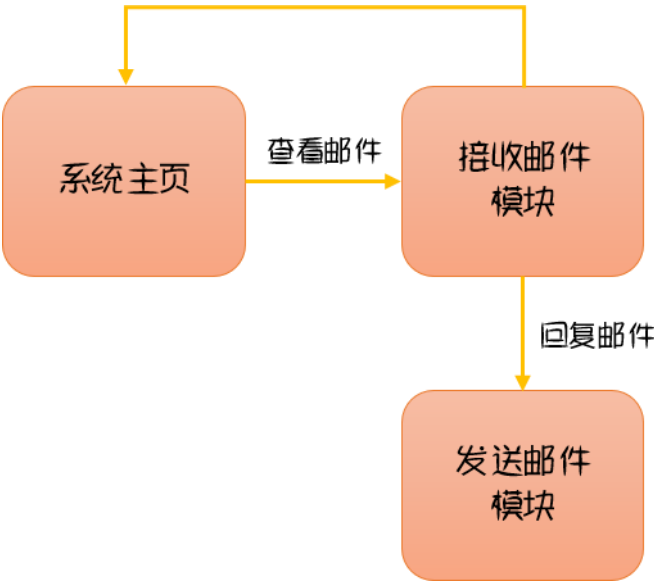


图 17. 接收邮件模块外部关系示意图

➤ 内部结构

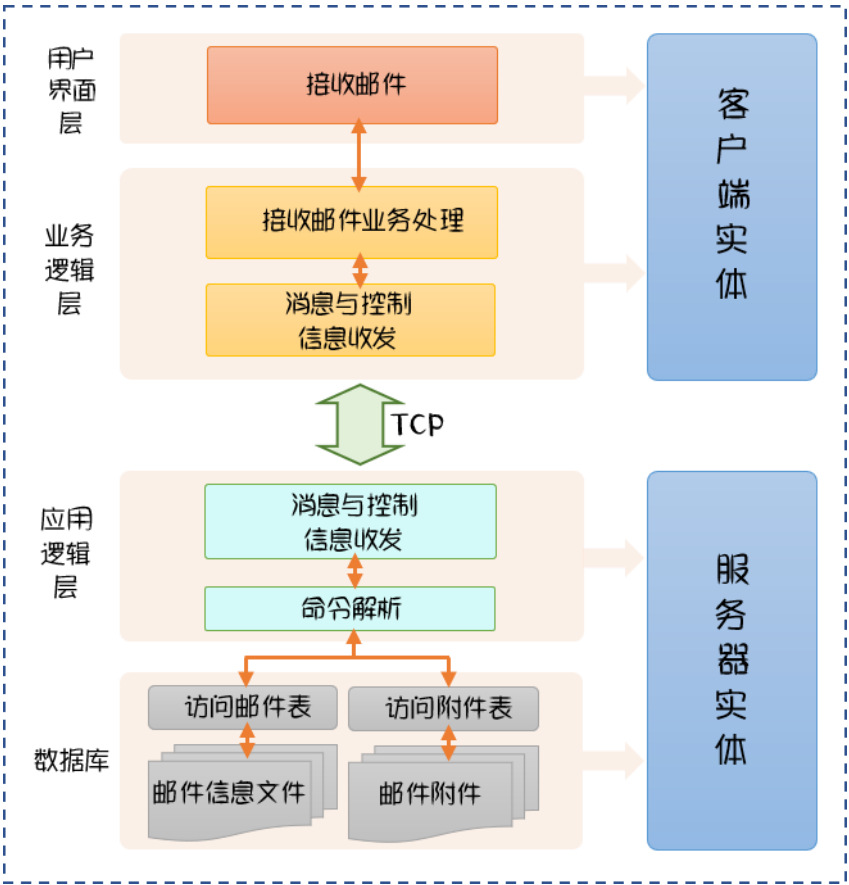


图 18. 接收邮件模块内部结构示意图

4.4.3. 模块流程设计

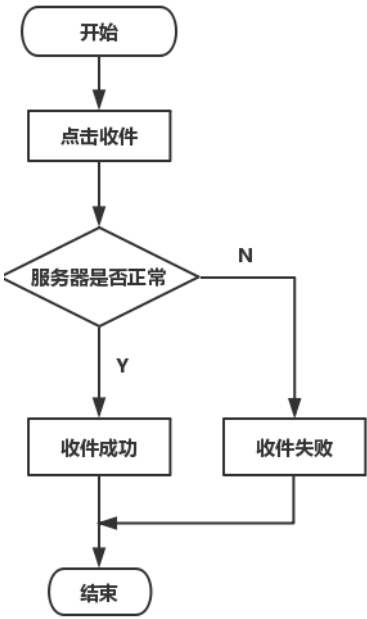


图 19. 接收邮件模块流程图

4.4.4. 模块时序设计

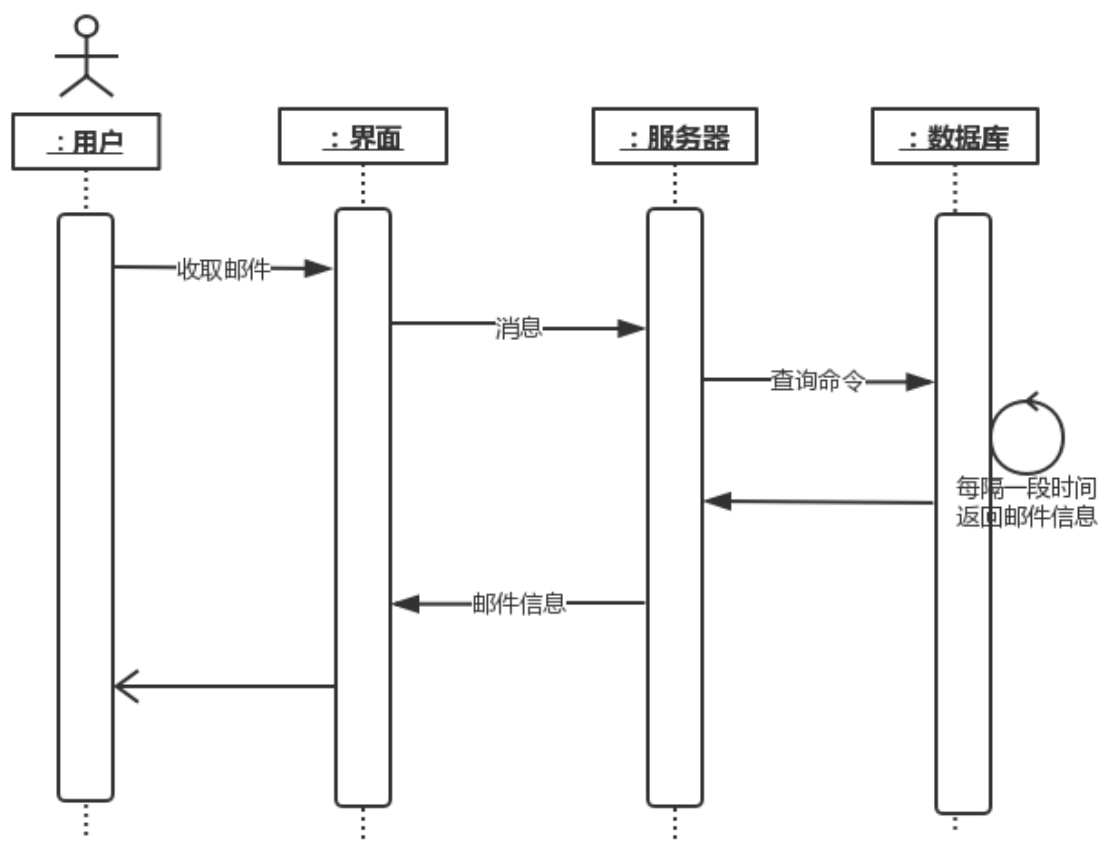


图 20. 接收邮件模块时序图

顺序图综述：图描述了“接收邮件”用例，涉及用户、界面、服务器、数据库、邮件 5 个对象。

参与者对象描述：“用户”是参与者，“界面”、“服务器”、“数据库”、“邮件”是四个对象。界面负责与用户和服务器进行交互，服务器负责接收来自界面的消息，将消息传给数据库并对消息进行回复。数据库负责查找信息、存储数据同时返回信息。

消息描述：“接收邮件”的顺序是通过消息发送的前后关系得到体现。通过用户发送“收取邮件”消息到界面，然后界面传递“消息”给服务器，服务器开始响应，向数据库发送“查询命令”消息，数据库接到消息后，开始在数据库内部查找相应的邮件信息，并返回查询数据给服务器，数据库每隔一段时间就会自动返回邮件信息。服务器将“邮件信息”消息传送给界面，界面表达给用户，用户通过“查看邮件内容”消息查看邮件。

其他描述：无。

4.4.5. 软件接口设计

表 15. 接收邮件模块 ReceiveEmail 接口说明

函数名	ReceiveEmail		
文件名	/Server/sever.cpp		
功能概要	服务器处理接收邮件		
参数			
类型	变量名	I/O	说明
QTcpSocket*	tcpSocket		客户端套接字
uint32_t	recId		收件人 ID
返回值	类型	void	说明
	值		
详细说明			
客户端接收邮件时调用，根据收件人 ID 在数据库中查询发件人 ID 和邮件内容，将结果传送给客户端。			
使用注意事项			

表 16. 接收邮件模块 Getmail 接口说明

函数名	Getmail			
文件名	/Server/database.cpp			
功能概要	数据库查询邮件信息			
参数				
类型	变量名		I/O	说明
uint32_t	recId			收件人 ID
返回值	类型	vector<Mail>	说明	
	值	mails	Mail 类容器，包含发件人 ID、收件人 ID 和邮件内容	
详细说明				
服务器收到客户端接收邮件命令后调用，接收收件人 ID，在数据库中查询相关 ID 的未读邮件，存入容器中返回给服务器。				
使用注意事项				

4.4.6. 硬件接口设计

无。

4.5. 管理邮件模块设计

4.5.1. 模块功能点列表

表 17. 管理邮件模块功能点列表

序号	功能点	详细说明
1	管理垃圾箱	管理垃圾箱内的邮件，可以将其彻底删除，也可以将其恢复。
2	删除邮件	将邮件放入垃圾箱。

4.5.2. 模块整体结构设计

➤ 外部关系

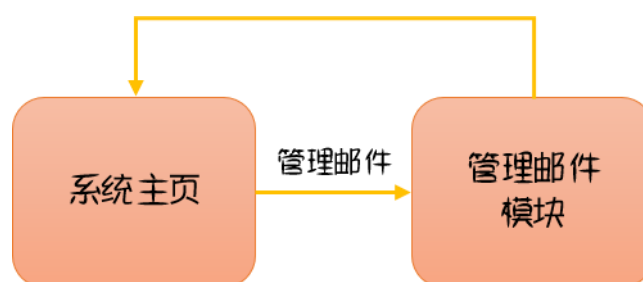


图 21. 管理邮件模块外部关系示意图

➤ 内部结构

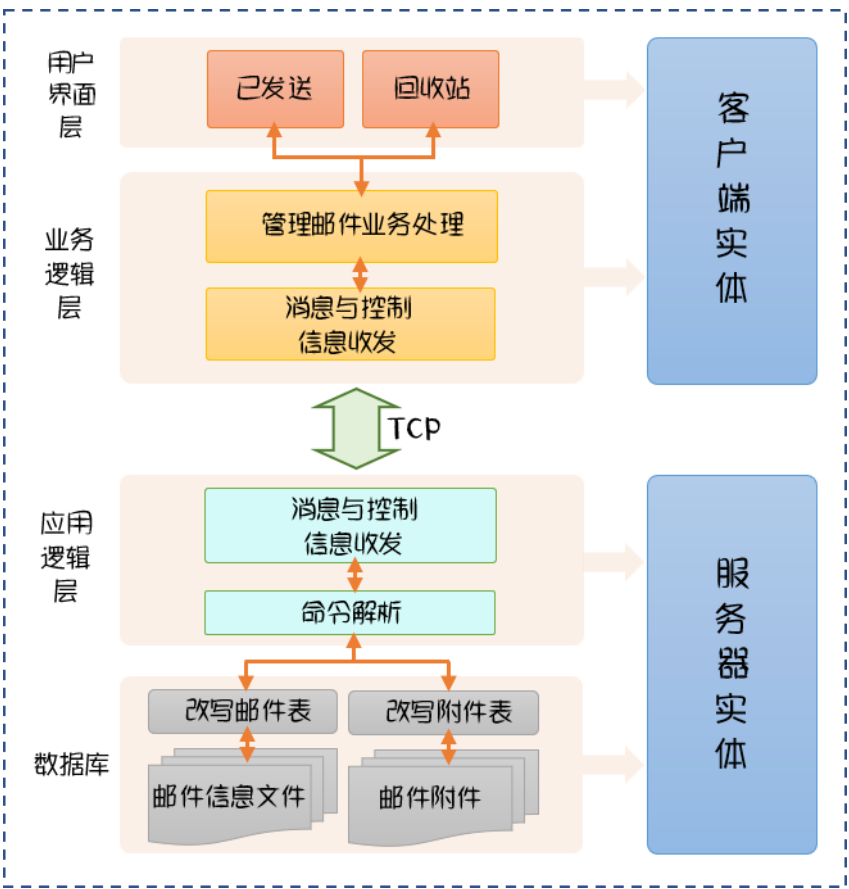


图 22. 管理邮件模块内部结构示意图

4.5.3. 模块流程设计

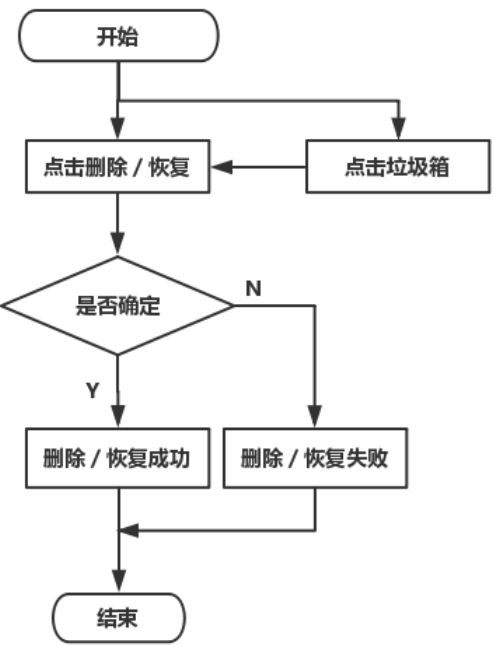


图 23. 管理邮件模块流程图

4.5.4. 模块时序设计

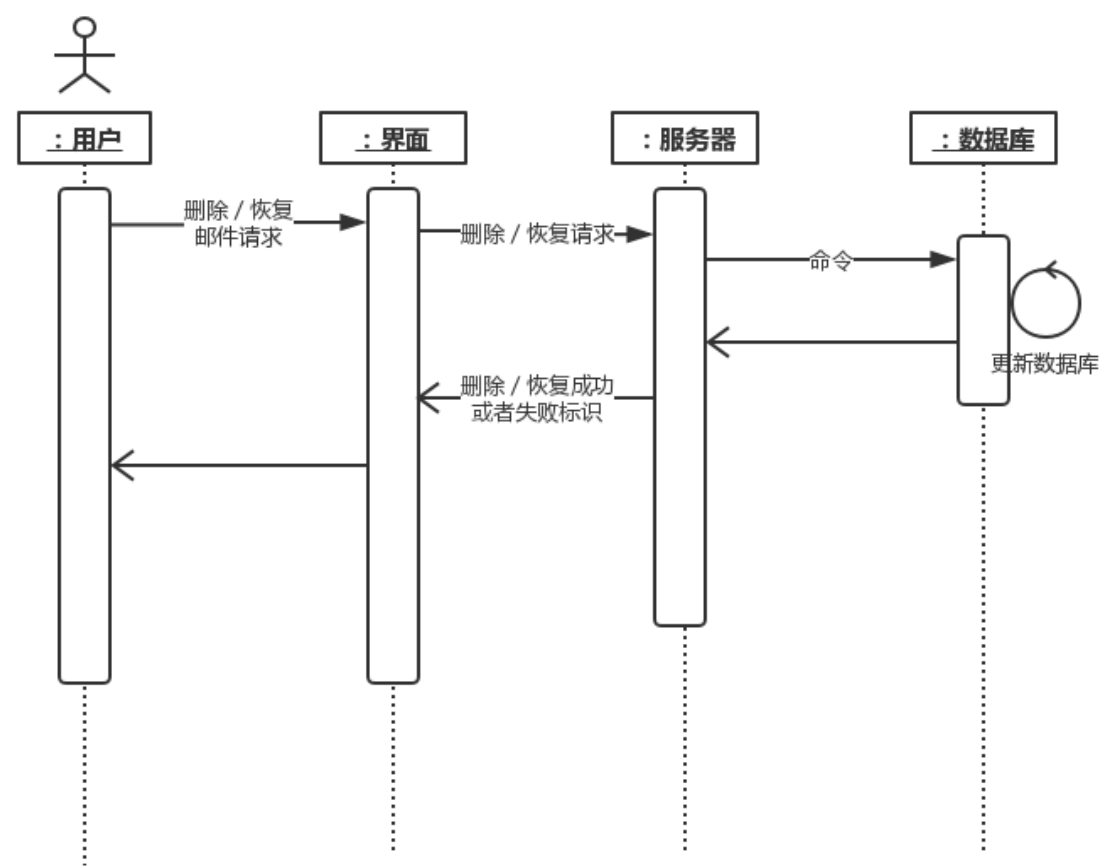


图 24. 管理邮件模块时序图

顺序图综述：图描述了“邮件管理”用例，涉及用户、界面、服务器、垃圾箱 4 个对象。

参与者对象描述：“用户”是参与者，“页面”、“服务器”、“垃圾箱”是三个对象。界面负责与用户和服务器进行交互，服务器负责接收来自界面的消息，并对消息进行回复。垃圾箱负责接收服务器的命令，执行并返回信息。

消息描述：“邮件管理”的顺序是通过消息发送的前后关系得到体现。通过用户发送“删除 / 恢复邮件请求”消息到界面，然后界面传递“删除 / 恢复请求”给服务器，服务器开始响应，向数据库发送“查询命令”消息，数据库接到消息后，开始在数据库内部查找相应的邮件信息，服务器检查服务器状态，并向界面发送“服务器不正常，删除 / 恢复失败”或“服务器正常，删除 / 恢复成果”消息给界面，界面表达给用户。同时服务器发送“删除 / 恢复命令”消息给垃圾箱，垃圾箱相应后将“删除 / 恢复信息”消息返回给服务器，服务检查服务器状态。

其他描述：无。

4.5.5. 软件接口设计

表 18. 管理邮件模块 DeleteEmail 接口说明

函数名	DeleteEmail		
文件名	/Server/sever.cpp		
功能概要	服务器处理删除邮件		
参数			
类型	变量名	I/O	说明
QTcpSocket*	tcpSocket		客户端套接字
uint32_t	mailId		邮件编号
返回值	类型	void	说明
	值		
详细说明			
客户端删除邮件时调用，将邮件编号传入数据库，通知客户端删除邮件成功。			
使用注意事项			

表 19. 管理邮件模块 DelMail 接口说明

函数名	DelMail			
文件名	/Server/database.cpp			
功能概要	数据库处理删除邮件			
参数				
类型	变量名		I/O	说明
uint32_t	mailId			邮件编号
返回值	类型	void	说明	
	值			
详细说明				
服务器收到客户端删除邮件命令后调用，接收邮件编号，在数据库中查询相关邮件并将其删除标记更改为 1。				
使用注意事项				

表 20. 管理邮件模块 RecoverEmail 接口说明

函数名	RecoverEmail		
文件名	/Server/sever.cpp		
功能概要	服务器处理恢复邮件		
参数			
类型	变量名	I/O	说明
QTcpSocket*	tcpSocket		客户端套接字
uint32_t	mailId		邮件编号
返回值	类型	void	说明
	值		
详细说明			
客户端恢复邮件时调用，将邮件编号传入数据库，通知客户端恢复邮件成功。			
使用注意事项			

表 21. 管理邮件模块 RecMail 接口说明

函数名	RecMail			
文件名	/Server/database.cpp			
功能概要	数据库处理恢复邮件			
参数				
类型	变量名		I/O	说明
uint32_t	mailId			邮件编号
返回值	类型	void	说明	
	值			
详细说明				
服务器收到客户端恢复邮件命令后调用，接收邮件编号，在数据库中查询相关邮件并将其删除标记更改为 0。				
使用注意事项				

表 22. 管理邮件模块 EmptyEmail 接口说明

函数名	EmptyEmail			
文件名	/Server/sever.cpp			
功能概要	服务器处理清空回收站邮件			
参数				
类型	变量名		I/O	说明
QTcpSocket*	tcpSocket			客户端套接字
uint32_t	recId			收件人 ID
返回值	类型	void	说明	
	值			
详细说明				
客户端清空回收站时调用，将收件人 ID 传入数据库，通知客户端清空回收站成功。				
使用注意事项				

表 23. 管理邮件模块 EmpMail 接口说明

函数名	EmpMail			
文件名	/Server/database.cpp			
功能概要	数据库处理清空回收站			
参数				
类型	变量名		I/O	说明
uint32_t	recId			收件人 ID
返回值	类型	void	说明	
	值			
详细说明				
服务器收到客户端清空回收站命令后调用，接收收件人 ID，在数据库中查询相关收件人所有邮件，将删除标记为 1 的邮件从数据库中删除。				
使用注意事项				

4.5.6. 硬件接口设计

无。

4.6. 草稿箱模块设计

4.6.1. 模块功能点列表

表 24. 草稿箱模块功能点列表

序号	功能点	详细说明
1	添加新联系人	点击添加，输入新联系人的信息，可在联系人列表中添加新的联系人。
2	删除已有联系人	找到具体的联系人，点击删除可将联系人从列表中删除。

5.6.2. 模块整体结构设计

➤ 外部关系

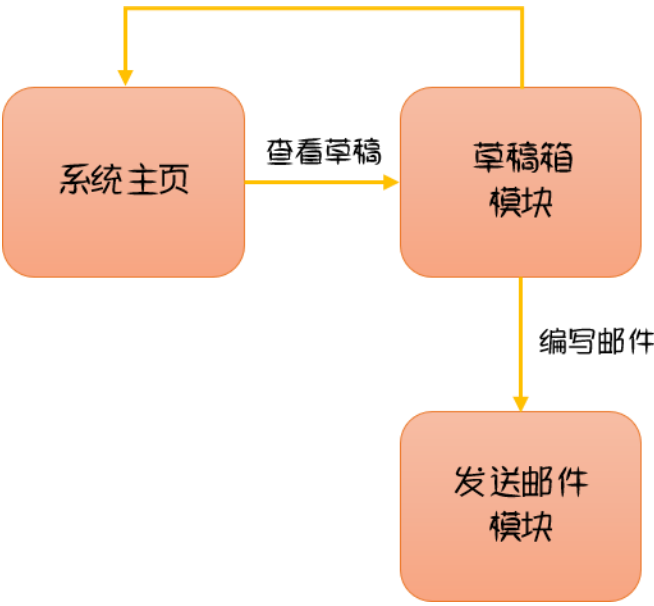


图 25. 草稿箱模块外部关系示意图

➤ 内部结构

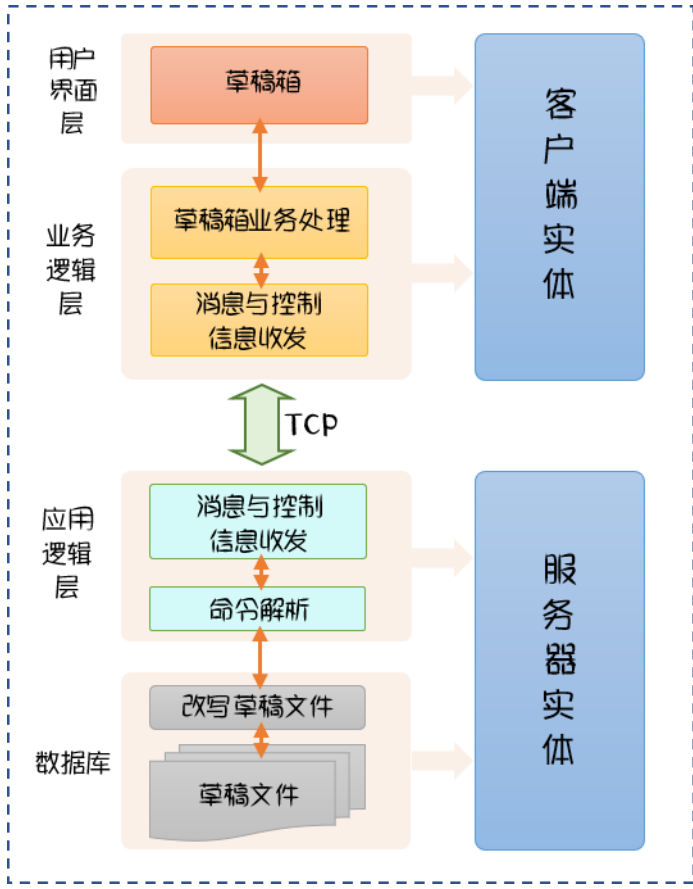


图 26. 草稿箱模块内部结构示意图

5.6.3. 模块流程设计

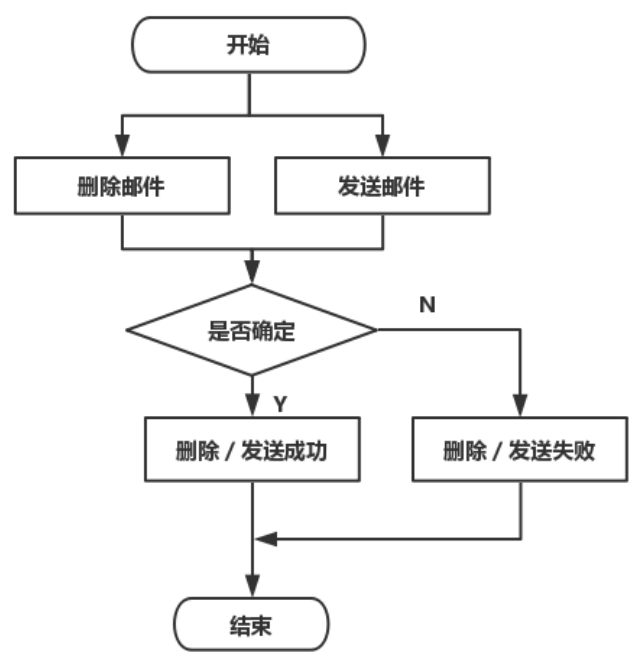


图 27. 草稿箱模块流程图

5.6.4. 模块时序设计

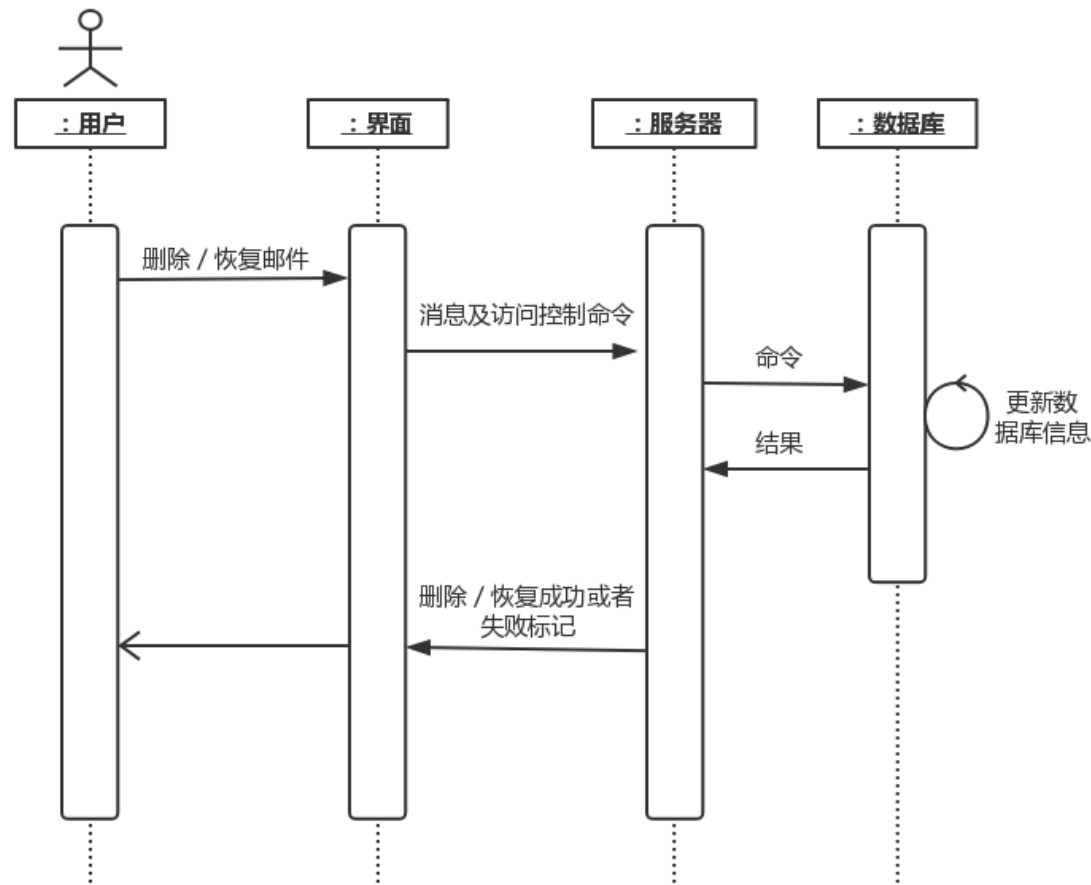


图 28. 草稿箱模块时序图

顺序图综述：图描述了“管理通讯录”用例，涉及用户、界面、服务器、数据库 4 个对象。

参与者对象描述：“用户”是参与者，“页面”、“服务器”、“服务器”是三个对象。界面负责与用户和服务器进行交互，服务器负责接收来自界面的消息，对数据库发送控制命令，返回信息给界面。数据库负责接收服务器的命令，执行并返回信息。

消息描述：“管理通讯录”的顺序是通过消息发送的前后关系得到体现。通过用户发送“删除 / 添加联系人”消息到界面，然后界面传递“消息及访问控制命令”给服务器，服务器开始响应，向数据库发送“命令”消息，数据库接到消息后，开始建立新的联系人或者删除已有联系人信息，将“结果”消息返回给服务器。服务器向界面发送“删除 / 添加成功或者失败标记”，界面返回结果给用户。

其他描述：无。

4.6.5. 软件接口设计

表 25. 草稿箱模块 SaveDraft 接口说明

函数名	SaveDraft			
文件名	/Server/server.cpp			
功能概要	服务器处理保存草稿			
参数				
类型	变量名		I/O	说明
QTcpSocket*	tcpSocket			客户端套接字
uint32_t	userId			用户 ID
QString	content			草稿内容
返回值	类型	void	说明	
	值			
详细说明				
客户端保存草稿时调用，将用户 ID、草稿内容传入数据库，通知客户端保存草稿成功。				
使用注意事项				

表 26. 草稿箱模块 AddDraft 接口说明

函数名	AddDraft			
文件名	/Server/database.cpp			
功能概要	数据库增加草稿信息			
参数				
类型	变量名		I/O	说明
Draft	draft			Draft 类，包含用户 ID 和草稿内容
返回值	类型	void	说明	
	值			
详细说明				
服务器收到客户端保存草稿请求后调用，接收用户 ID 和邮件内容，存入数据库。				
使用注意事项				

表 27. 草稿箱模块 DeleteDraft 接口说明

函数名	DeleteDraft		
文件名	/Server/sever.cpp		
功能概要	服务器处理删除草稿		
参数			
类型	变量名	I/O	说明
QTcpSocket*	tcpSocket		客户端套接字
uint32_t	draftId		草稿编号
返回值	类型	void	说明
	值		
详细说明			
客户端删除草稿时调用，将草稿编号传入数据库，通知客户端删除草稿成功。			
使用注意事项			

表 28. 草稿箱模块 DelDraft 接口说明

函数名	DelDraft			
文件名	/Server/database.cpp			
功能概要	数据库处理删除草稿			
参数				
类型	变量名		I/O	说明
uint32_t	draftId			草稿编号
返回值	类型	void	说明	
	值			
详细说明				
服务器收到客户端删除草稿命令后调用，接收草稿编号，在数据库删除相关草稿信息。				
使用注意事项				

4.6.6. 硬件接口设计

无。

4.7. 通讯录模块设计

4.7.1. 模块功能点列表

表 29. 通讯录模块功能点列表

序号	功能点	详细说明
1	彻底删除邮件	点击彻底删除键，将邮件彻底删除。
2	恢复邮件	恢复以删除邮件。
3		
4		

4.7.2. 模块整体结构设计

➤ 外部关系

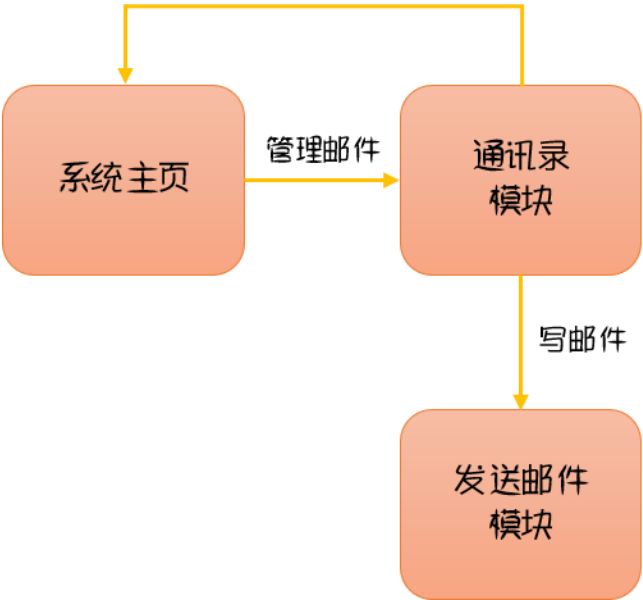


图 29. 通讯录模块外部关系示意图

➤ 内部结构

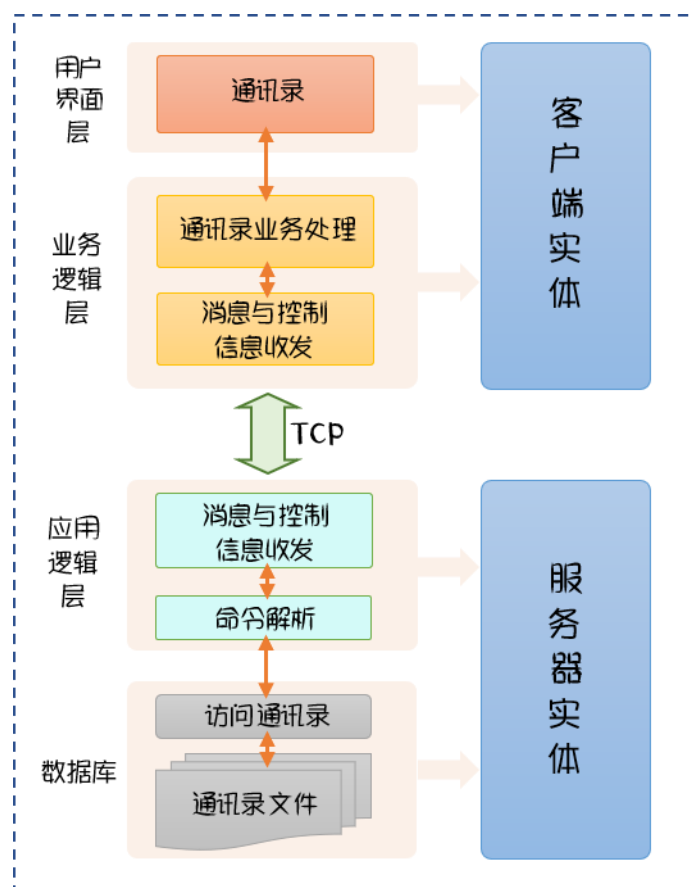


图 30. 通讯录模块内部结构示意图

4.7.3. 模块流程设计

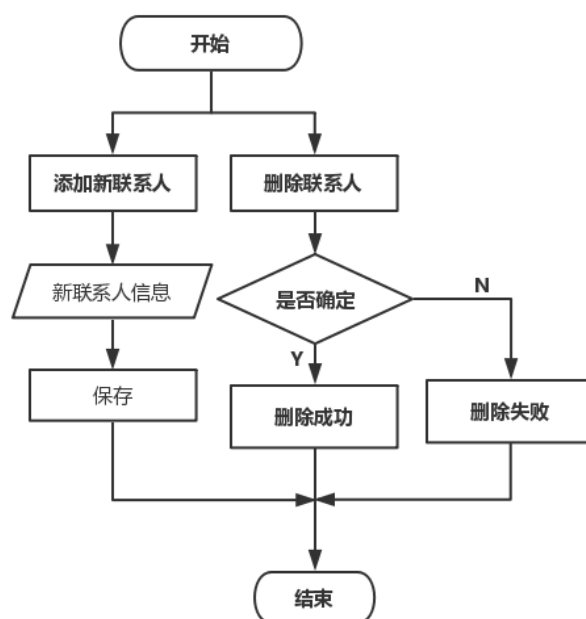


图 31. 通讯录模块流程图

4.7.4. 模块时序设计

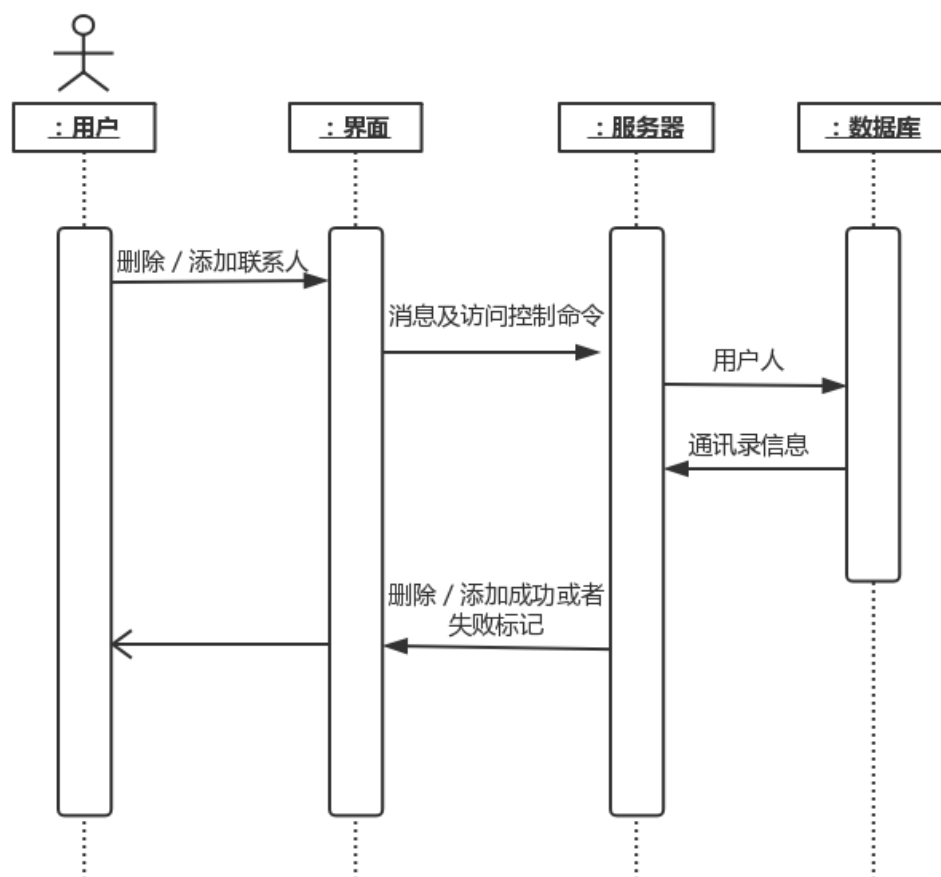


图 32. 通讯录模块时序图

顺序图综述：图描述了“草稿箱”用例，涉及用户、界面、服务器、数据库 4 个对象。

参与者对象描述：“用户”是参与者，“页面”、“服务器”、“服务器”是三个对象。界面负责与用户和服务器进行交互，服务器负责接收来自界面的消息，对数据库发送控制命令，返回信息给界面。数据库负责接收服务器的命令，执行并返回信息。

消息描述：“草稿箱”的顺序是通过消息发送的前后关系得到体现。通过用户发送“删除 / 恢复邮件”消息到界面，然后界面传递“消息及访问控制命令”给服务器，服务器开始响应，向数据库发送“命令”消息，数据库接到消息后，开始删除邮件或者将以删除邮件恢复，并更新数据库信息，将“结果”消息返回给服务器。服务器向界面发送“删除 / 恢复成功或者失败标记”，界面返回结果给用户。

其他描述：无。

4.7.5. 软件接口设计

表 30. 通讯录模块 SaveContact 接口说明

函数名	SaveContact			
文件名	/Server/server.cpp			
功能概要	服务器处理保存联系人			
参数				
类型	变量名		I/O	说明
QTcpSocket*	tcpSocket			客户端套接字
uint32_t	userId			用户 ID
QString	contactname			联系人用户名
返回值	类型	void	说明	
	值			
详细说明				
客户端保存联系人时调用，将用户 ID、联系人用户名传入数据库，通知客户端保存联系人成功。				
使用注意事项				

表 31. 通讯录模块 AddContact 接口说明

函数名	AddContact			
文件名	/Server/database.cpp			
功能概要	数据库增加联系人信息			
参数				
类型	变量名		I/O	说明
Contact	contact			Contact 类, 包含用户 ID, 联系人用户名
返回值	类型	void	说明	
	值			
详细说明				
服务器收到客户端保存联系人请求后调用, 接收用户 ID 和联系人用户名, 存入数据库。				
使用注意事项				

表 32. 通讯录模块 DeleteContact 接口说明

函数名	DeleteContact		
文件名	/Server/sever.cpp		
功能概要	服务器处理删除联系人		
参数			
类型	变量名	I/O	说明
QTcpSocket*	tcpSocket		客户端套接字
uint32_t	contactId		联系人编号
返回值	类型	void	说明
	值		
详细说明			
客户端删除联系人时调用，将联系人编号传入数据库，通知客户端删除联系人成功。			
使用注意事项			

表 33. 草稿箱模块 DelContact 接口说明

函数名	DelContact			
文件名	/Server/database.cpp			
功能概要	数据库处理删除联系人			
参数				
类型	变量名		I/O	说明
uint32_t	contactId			联系人编号
返回值	类型	void	说明	
	值			
详细说明				
服务器收到客户端删除联系人命令后调用，接收联系人编号，在数据库删除相关联系人信息。				
使用注意事项				

4.7.6. 硬件接口设计

无。

§5. 数据设计

5.1. 数据库设计说明

本邮件系统采用 SQLite 数据库进行开发。SQLite 是一个轻量级、跨平台的关系型数据库。数据库中的所有信息都包含在一个文件内，可移植性较高。SQLite 支持多种语言编程接口，本系统采用 C/C++ 接口进行实现。

5.2. 数据描述与实体关系模型

邮件系统有关实体及属性为：

- (1). 用户：用户 ID、用户名、账号密码；
- (2). 邮件：已读标记、删除标记、邮件编号、邮件主题、收件人 ID、收件人用户名、发件人 ID、发件人用户名、邮件内容、发送时间、附件编号；
- (3). 附件：附件名、附件编号、所属邮件编号、附件内容；
- (4). 草稿：草稿编号、草稿主题、草稿内容、编辑时间、编写人 ID、编写人用户名；
- (5). 通讯录：联系人 ID、联系人用户名、联系人备注。

实体关系模型如下：

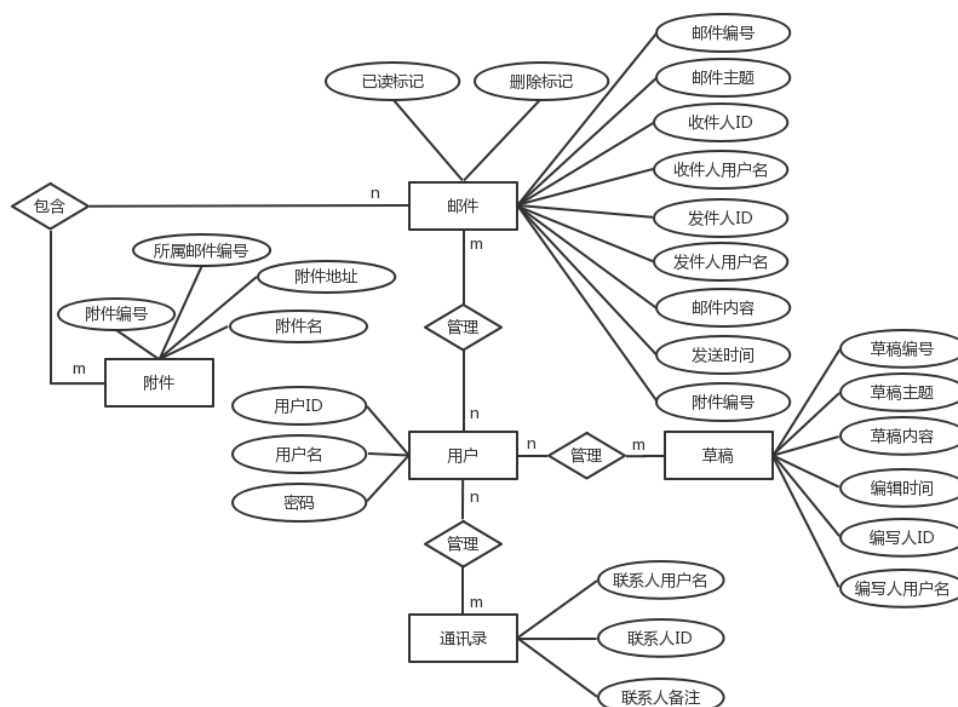


图 33. 实体关系模型图

5.3. 数据文件设计

数据库数据文件列表如下：

(1). UserInfo 表：存储用户信息，包括用户 ID、用户名和密码，用户登录或注册发送请求，需要经过数据库验证用户名和密码；

(2). Mail 表：存储邮件信息，包括邮件编号、邮件主题和收、发件人相关信息等，便于对邮件进行查找；

(3). Attachment 表：存储附件信息，包括附件编号、附件名称等，用户发送的附件首先上传到服务器中，再将存放地址导入数据库中；

(4). Draft 表：存储草稿信息，包括草稿编号、草稿主题和草稿内容等；

(5). AddressBook 表：存储通讯录信息。

5.3.1. UserInfo 表

表 34. 数据库 UserInfo 表单设计

序号	字段名	数据类型	是否主键	字段说明
1	用户 ID	string	是	用户唯一 ID
2	用户名	string	否	用户名称
3	账号密码	string	否	用户账号登录密码

5.3.2. Mail 表

表 35. 数据库 Mail 表单设计

序号	字段名	数据类型	是否主键	字段说明
1	邮件编号	string	是	邮件唯一编号
2	收件人 ID	string	否	邮件收件人 ID
3	收件人用户名	string	否	邮件收件人用户名
4	发件人 ID	string	否(外键)	邮件发件人 ID
5	发件人用户名	string	否	邮件发件人用户名
6	邮件主题	string	否	邮件主题名称
7	邮件内容	string	否	邮件具体内容
8	发送时间	string	否	邮件发送时间
9	附件编号	int	否	附件编号(0 表示无附件)
10	已读标记	bool	否	0 表示未读，1 表示已读
11	删除标记	bool	否	0 表示未删除，1 表示删除

5.3.3. Attachment 表

表 36. 数据库 Attachment 表单设计

序号	字段名	数据类型	是否主键	字段说明
1	附件编号	int	是	附件唯一编号
2	所属邮件编号	int	否(外键)	附件所在邮件编号
3	附件名	string	否	附件文件名称
4	附件地址	string	否	附件存放地址

5.3.4. Draft 表

表 37. 数据库 Draft 表单设计

序号	字段名	数据类型	是否主键	字段说明
1	草稿编号	int	是	草稿唯一编号
2	草稿主题	string	否	草稿主题名称
3	草稿内容	string	否	草稿具体内容
4	编辑时间	string	否	草稿上次修改时间
5	编写人 ID	int	否(外键)	编写草稿用户 ID
6	编写人用户名	string	否	编写草稿用户名称

5.3.5. AddressBook 表

表 38. 数据库 AddressBook 表单设计

序号	字段名	数据类型	是否主键	字段说明
1	持有者 ID	int	是	通讯录唯一编号
2	联系人 ID	int	否	联系人唯一编号
3	联系人用户名	string	否	联系人系统名字
4	联系人备注	string	否	联系人好友备注名字

§6. 界面设计

6.1. 登录界面

➤ 概述

左上角有邮件系统名称，在 Email 和 Password 框输入正确的邮箱和密码，完成登录进入主页面，如果没有账号，可以注册。

➤ 设计图



图 34. 登录界面设计图

➤ 链接

点击 Login 可以进入到主页面，点击 Register 跳转到注册界面。

6.2. 注册界面

➤ 概述

左上角有邮件系统名称，有 Email Address、Password 和 Verify Password 三项，输入格式正确的邮箱和密码，再次确认密码提交后可拥有账号。

➤ 设计图



The diagram shows a registration window titled "FengFeng Mail". Inside the window, the text "FengFeng Mail" is displayed in a stylized orange font. Below the text, there are three input fields: "Email Address", "Password", and "Verify Password". Each field is represented by a gray rectangular box. Below these fields is an orange "Submit" button.

图 35. 注册界面设计图

➤ 链接

点击 Submit 跳转到登录界面。

6.3. 邮件系统主页

➤ 概述

左上角有邮件系统名称，系统名称下面是用户个人的邮箱账号，左边是固定的导航栏，有 Write、Inbox、Contact、Send、Drafts、Dustbin。界面中间显示问候语，以及提示用户总共有几封未读信件。右上角为登出和帮助选项。

➤ 设计图



图 36. 主页界面设计图

➤ 链接

点击 Write 跳转到写邮件界面，点击 Inbox 跳转到收件箱，点击 Contact 跳转到通讯录界面，点击 Send 跳转到已发送界面，点击 Drafts 跳转到草稿箱，点击 Dustbin 跳转到垃圾箱界面，点击 logout 跳转到登录界面，点击 help 弹出帮助框。

6.4. 邮件编写界面

➤ 概述

左上角有邮件系统名称，系统名称下面是用户个人的邮箱账号，左边是固定的导航栏，有 Write、Inbox、Contact、Send、Drafts、Dustbin，右上角为登出和帮助选项。页面中间有 Receiver 栏，下面可以选择 BCC（密送）或者 Carbon Copy（抄送），下一栏是 Subject，填写邮件主题，Accessory 可以添加附件，Content 填写邮件正文。可以选择 close，Send，Save 邮件。

➤ 设计图

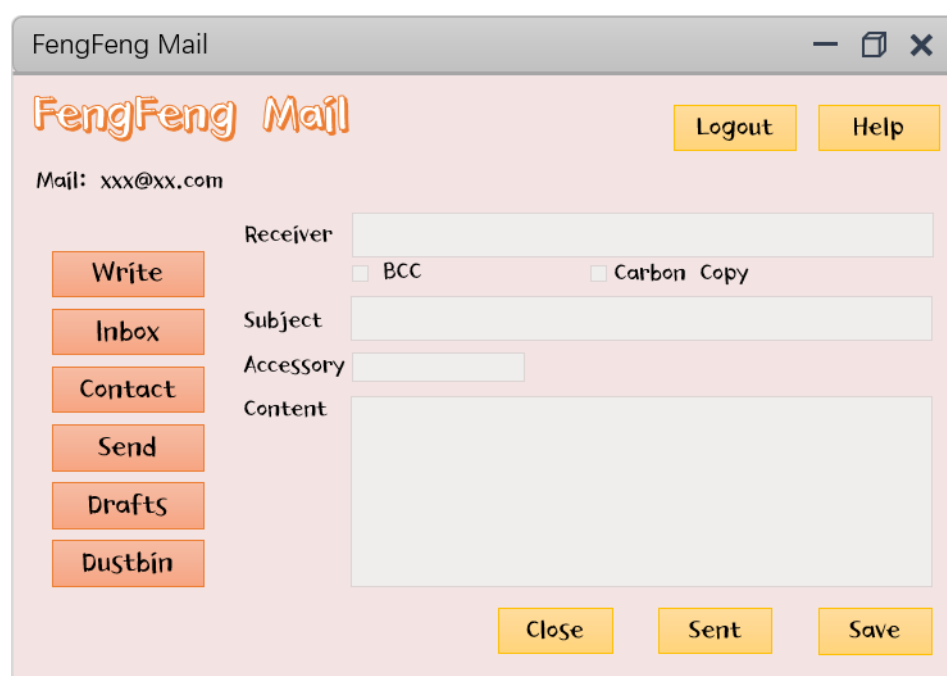


图 37. 邮件编写界面设计图

➤ 链接

点击 Write 跳转到写邮件界面，点击 Inbox 跳转到收件箱，点击 Contact 跳转到通讯录界面，点击 Send 跳转到已发送界面，点击 Drafts 跳转到草稿箱，点击 Dustbin 跳转到垃圾箱界面，点击 logout 跳转到登录界面，点击 help 弹出帮助框。点击 Close 跳转到主页，点击 Sent 留在本界面，提示邮件发送成功 / 失败，点击 Save 留在本界面，提示邮件保存成功 / 失败。

6.5. 收件箱

➤ 概述

左上角有邮件系统名称，系统名称下面是用户个人的邮箱账号，左边是固定的导航栏，有 Write、Inbox、Contact、Send、Drafts、Dustbin，右上角为登出和帮助选项。页面中间为收到的邮件列表，每一条包含发件人邮箱，主题，部分内容以及日期。

➤ 设计图

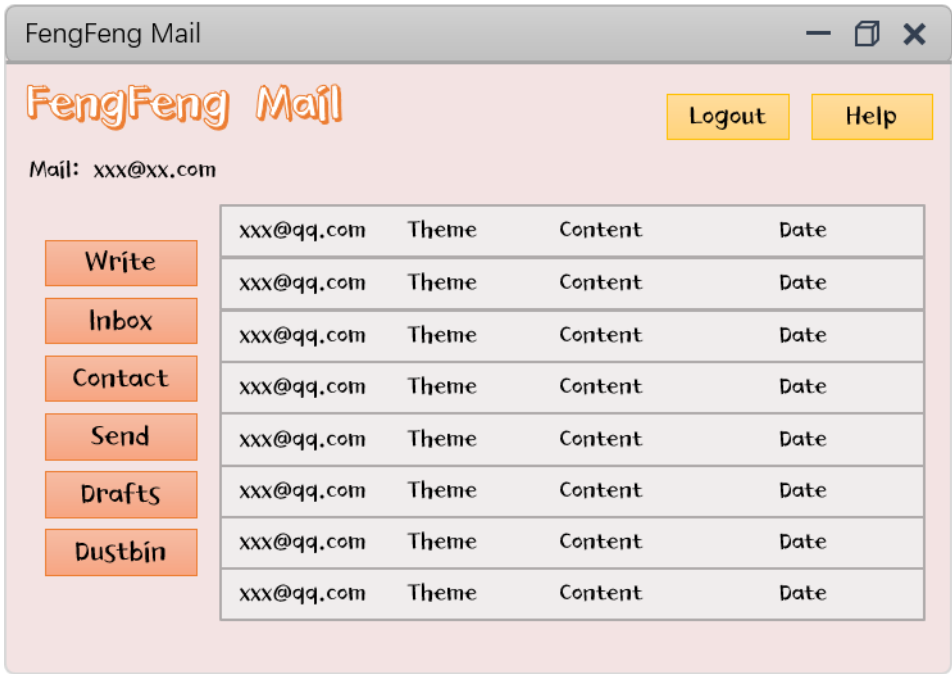


图 38. 收件箱界面设计图

➤ 链接

点击 Write 跳转到写邮件界面，点击 Inbox 跳转到收件箱，点击 Contact 跳转到通讯录界面，点击 Send 跳转到已发送界面，点击 Drafts 跳转到草稿箱，点击 Dustbin 跳转到垃圾箱界面，点击 logout 跳转到登录界面，点击 help 弹出帮助框。点击各条邮件消息跳转到阅读邮件界面。

6.6. 通讯录

➤ 概述

左上角有邮件系统名称，系统名称下面是用户个人的邮箱账号，左边是固定的导航栏，有 Write、Inbox、Contact、Send、Drafts、Dustbin，右上角为登出和帮助选项。界面中间为联系人列表，每一条包括联系人 Nick name、User name 和 Mail address。右下角可以 Add 或者 Delete 联系人。

➤ 设计图

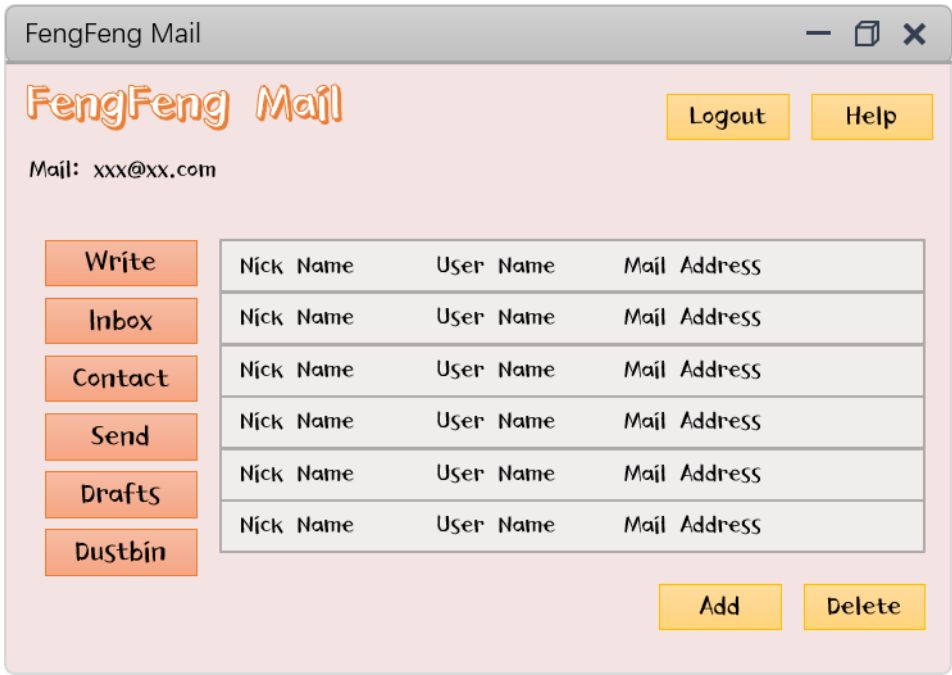


图 39. 通讯录界面设计图

➤ 链接

点击 Write 跳转到写邮件界面，点击 Inbox 跳转到收件箱，点击 Contact 跳转到通讯录界面，点击 Send 跳转到已发送界面，点击 Drafts 跳转到草稿箱，点击 Dustbin 跳转到垃圾箱界面，点击 logout 跳转到登录界面，点击 help 弹出帮助框。点击 Add 跳转到添加联系人界面，点击 Delete 删除选中联系人，留在本页面，提示删除成功 / 失败。

6.7. 发件箱

➤ 概述

左上角有邮件系统名称，系统名称下面是用户个人的邮箱账号，左边是固定的导航栏，有 Write、Inbox、Contact、Send、Drafts、Dustbin，右上角为登出和帮助选项。页面中间为已发送的邮件列表，每一条包含收件人邮箱，主题，部分内容以及日期。右下角可以选择删除选中邮件。

➤ 设计图

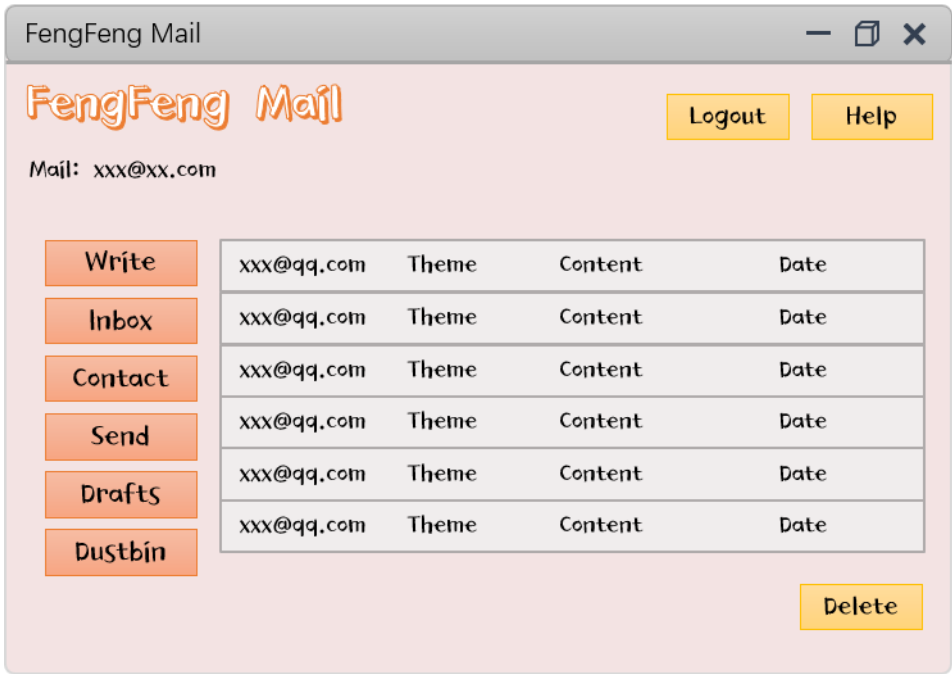


图 40. 发件箱界面设计图

➤ 链接

点击 Write 跳转到写邮件界面，点击 Inbox 跳转到收件箱，点击 Contact 跳转到通讯录界面，点击 Send 跳转到已发送界面，点击 Drafts 跳转到草稿箱，点击 Dustbin 跳转到垃圾箱界面，点击 logout 跳转到登录界面，点击 help 弹出帮助框。点击 Delete 删除选中邮件，留在本页面，提示删除成功 / 失败。

6.8. 邮件查看

➤ 概述

左上角有邮件系统名称，系统名称下面是用户个人的邮箱账号，左边是固定的导航栏，有 Write、Inbox、Contact、Send、Drafts、Dustbin，右上角为登出和帮助选项。页面中间显示邮件详细内容，发件人，主题和正文。右下角可以选择 forward、Answer、Delete 进行邮件转发、回复和删除。

➤ 设计图

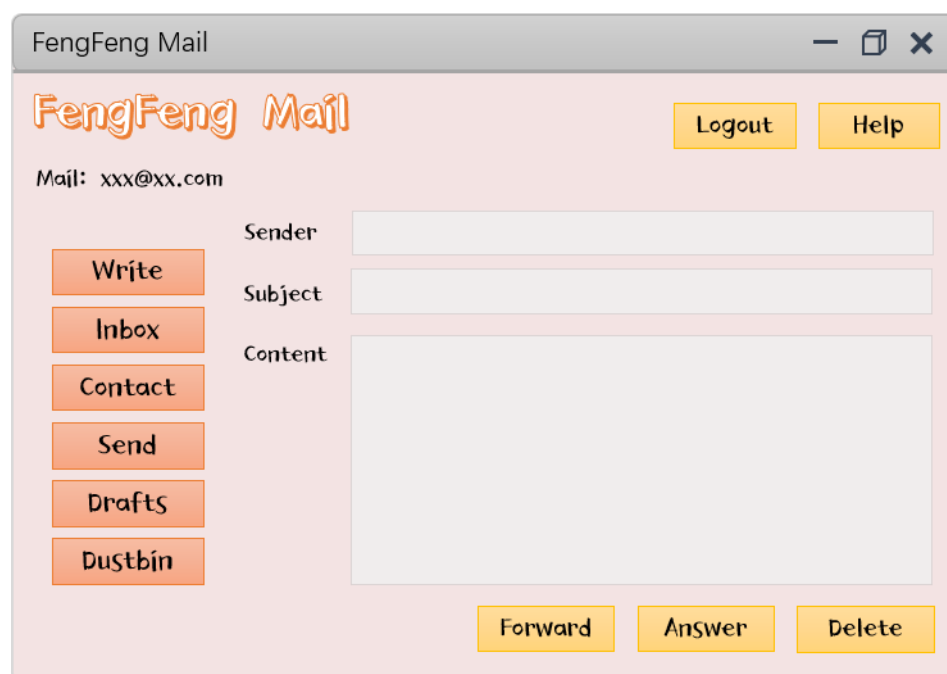


图 41. 邮件查看界面设计图

➤ 链接

点击 Write 跳转到写邮件界面，点击 Inbox 跳转到收件箱，点击 Contact 跳转到通讯录界面，点击 Send 跳转到已发送界面，点击 Drafts 跳转到草稿箱，点击 Dustbin 跳转到垃圾箱界面，点击 logout 跳转到登录界面，点击 help 弹出帮助框。点击 Forward 跳转到发送邮件界面，点击 Answer 跳转到写邮件界面，点击 Delete 删除选中邮件，留在本页面，提示删除成功 / 失败。

6.9. 草稿箱

➤ 概述

左上角有邮件系统名称，系统名称下面是用户个人的邮箱账号，左边是固定的导航栏，有 Write、Inbox、Contact、Send、Drafts、Dustbin，右上角为登出和帮助选项。页面中间为草稿列表，每一条包括主题，内容，日期。右下角 Redact 可以对草稿邮件进行重新编辑。

➤ 设计图

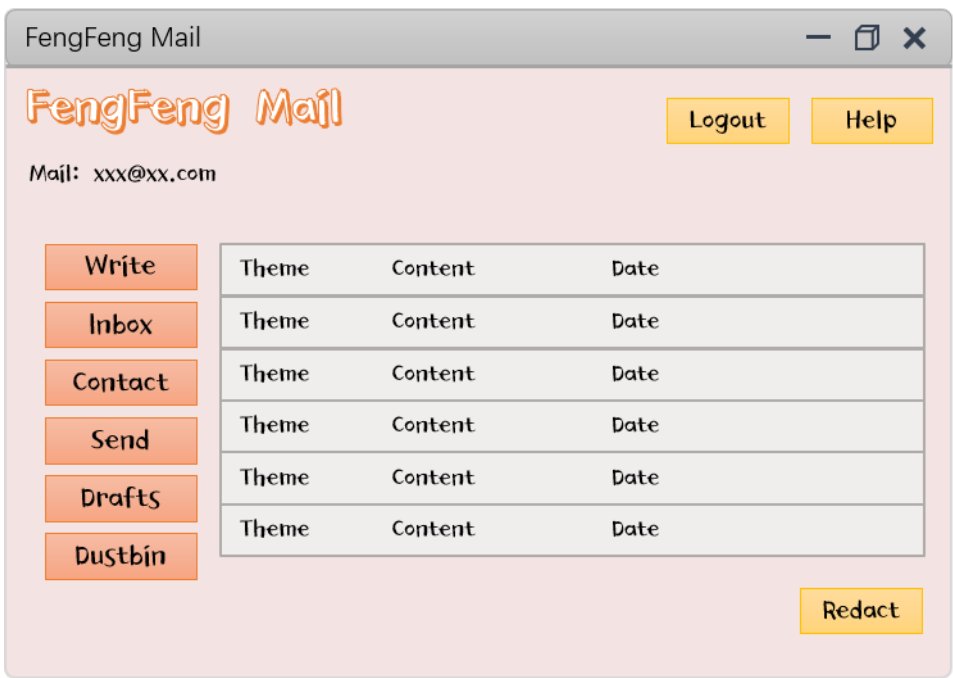


图 42. 草稿箱界面设计图

➤ 链接

点击 Write 跳转到写邮件界面，点击 Inbox 跳转到收件箱，点击 Contact 跳转到通讯录界面，点击 Send 跳转到已发送界面，点击 Drafts 跳转到草稿箱，点击 Dustbin 跳转到垃圾箱界面，点击 logout 跳转到登录界面，点击 help 弹出帮助框。点击 Redact 跳转到编辑邮件界面。

6.10. 回收站

➤ 概述

左上角有邮件系统名称，系统名称下面是用户个人的邮箱账号，左边是固定的导航栏，有 Write、Inbox、Contact、Send、Drafts、Dustbin，右上角为登出和帮助选项。页面中间显示已删除邮件详细内容，发件人，主题、正文和日期。右下角可以对选中的已删除邮件进行恢复。

➤ 设计图

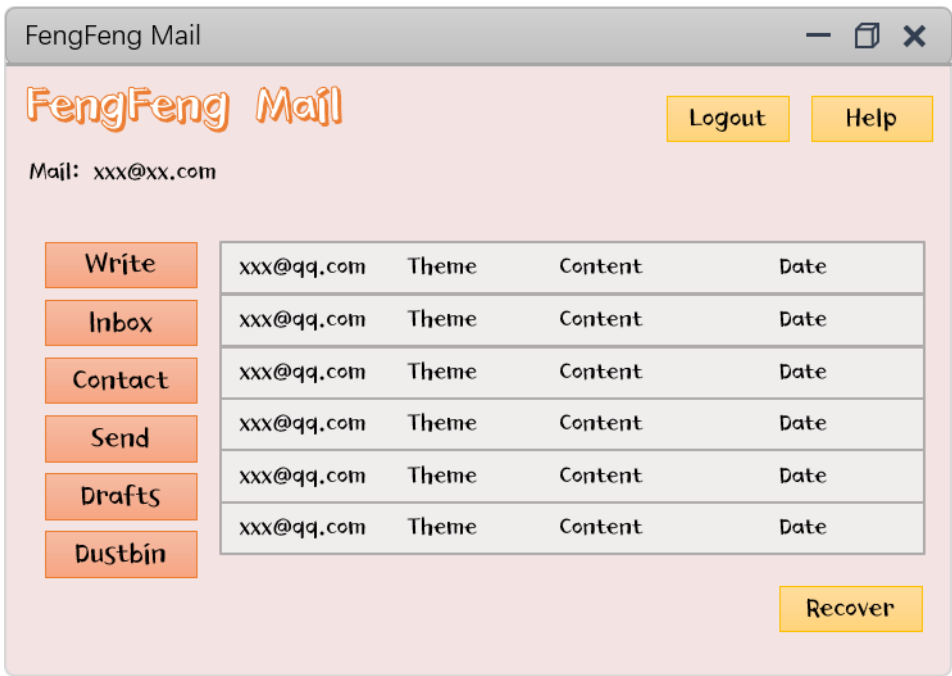


图 43. 回收站界面设计图

➤ 链接

点击 Write 跳转到写邮件界面，点击 Inbox 跳转到收件箱，点击 Contact 跳转到通讯录界面，点击 Send 跳转到已发送界面，点击 Drafts 跳转到草稿箱，点击 Dustbin 跳转到垃圾箱界面，点击 logout 跳转到登录界面，点击 help 弹出帮助框。点击 Recover 留在本页面，提示恢复成功 / 失败。