Software Requirements Specification

for

MailX

Version **1.0 Approved**

April 24, 2017

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Sachin, Shubham	17-03-17	initial draft	1.0 draft 1
Samarth	04-03-17	File Transfer Protocol	1.0 draft 2
Samarth	20-03-17	Interface Problems	1.0 approved

1. Introduction

1.1. Purpose

This SRS describes the software functional and nonfunctional requirements for release 1.0 of the MailX Software. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

1.2. Project Scope

The MailX will permit Process Impact employees to send mails and files from the company on-line to be delivered to specified locations.

1.3. References

1. Stack-Overflow (mainly for file transfer)
   http://stackoverflow.com

2. Github
   https://github.com/MiniProjectsX

3. java-mail-1.4.* - api references from Oracle
   http://www.oracle.com/technetwork/java/javamail

2. Overall Description

2.1. Product Perspective

The MailX Software is a new software that replaces old slow loading interfaces that took ages to load and showed lot of latency both in case of efficiency and performance. Mainly the reason was that the users had to configure the whole interface for specific user and the file sharing was not always a safe method, Here now we have a easy to use system that requires no pre-hand configuration or any type of delay for user, by its simple design and easy to use interface one may only need to enter his/her and the receivers email ID.

2.2. Product Features

Mainly the best feature is its easy to use interface including its ability to transfer mails as text as well as file including all type of formats (.png, .jpg, .rft et cetera). No user should know any configuration about the protocols used or the servers involved, just the senders and receivers email ID will be great.

2.3. User Classes and Characteristics

Patron (Sender)	A Patron is a sender at the corporate campus in Clackamas, Oregon, who wishes to send mail to be delivered from the company. There are about 600 potential Patrons, of which an estimated 400 are expected to use the MailX System an average of 4 times per week each. Patrons will sometimes send multiple mails for group events or guests. An estimated 90 percent of mails will be placed using the corporate Intranet, with 10 percent of mails being placed from home. All Patrons have Intranet access from their offices.
Receiver	The Server currently employs about 20 Staff, who will receive mail from the MailX System, send mails, send files and request delivery. Most of the Staff will need to be trained in the use of the computer, the Web browser, and the MailX System.
Server	The Server is a central hub, perhaps the manager, who is responsible for establishing and maintaining daily mails of the staff available from the mail and the times of day that each staff is available. Some mails may not be available (draft). The Server will also define the MailX requirements. The Server will need to edit the entries to maintain its database.

2.4. Operating Environment

OE-1: This System works with all Operating System, Mac OSX, Windows, Linux as far as *jre installed.
OE-2: The MailX System shall communicate with the following Web browsers: Microsoft Internet Explorer versions 5.0 and 6.0, Netscape Communicator version 4.7, and Netscape versions 6 and 7, Safari, Chrome, Firefox etc.
OE-3: The MailX System shall operate on a server running the Senders Mail Server.
OE-4: The MailX System shall permit user access from the corporate Intranet and, if a user is authorized for outside access through the corporate firewall, from an Internet connection at the user's home.

2.5. Design and Implementation Constraints

CO-1: The system shall use the current corporate standard Oracle database engine.
CO-2: All JAVA code shall conform to the JAVA 1.8 standard.
CO-3: All scripts shall be written in JAVA.

2.6. User Documentation

UD-1: The system shall provide Mailing ease and cross-linked help system in JAVA that describes and illustrates all system functions.

2.7. Assumptions and Dependencies

AS-1: The MailX System assumes that the java-1.8 standard in installed on the clients machine

DE-1: There should exist an account with respective sender and receiver mail IDs

(as shown in use-case diagram)

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

The following is a look at the user interface for first time

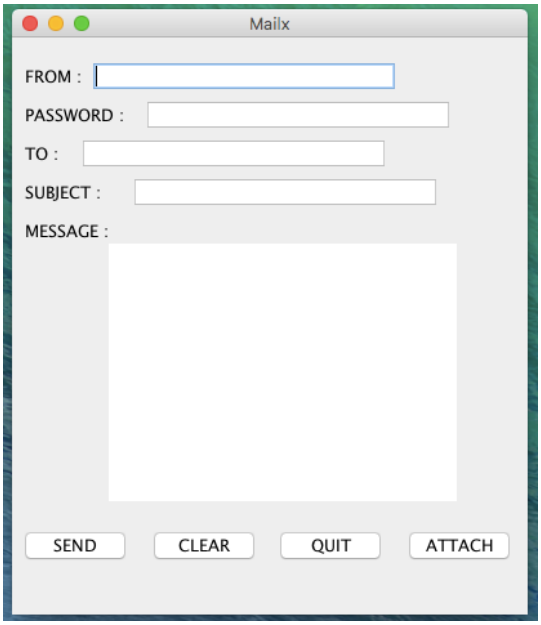


Figure 1. showing Interface

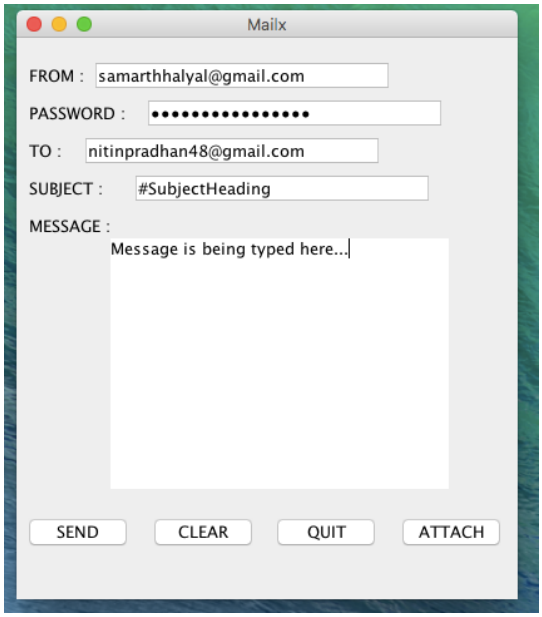


Figure 2. showing Data-Entering

there exist many button respectively for sending(send), clearing the previously typed text(clear), quitting the interface(quit), and attaching files(attach).
the following picture shows the attachment of files.

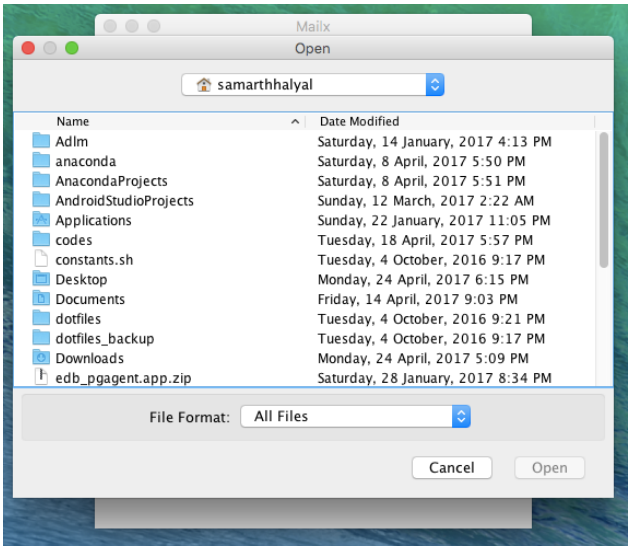


Figure 3. showing the file attachment

3.1.2. Hardware Interfaces

<<There are no visible hardware Interfaces.>>

3.1.3. Software Interfaces

The Software Interface is as shown above in user interface.
The data that will be transferred will be as in form of normal Text or file format (any kind). the extensions that will be sent depends on the Mail Server.

3.1.4. Communications Interfaces

The Protocols we are using here are SMTP, FTP.
SMTP :: stands for simple mail transfer protocol, this help to send mail from senders machine to his/her E-Mail Servers, which is forwarded to respective E-Mail server and then to the receiver's machine. something called POP and IMAP are involved he certain steps which out of scope of this Document.
FTP :: stands for file transfer protocol, helps in sending file for one machine to another.

3.2. Functional Requirements

1. security restrictions for sending email (can't send as someone else)
2. security against reading outgoing mail (leaving the system, or within the system)
3. ability to receive email 24/7 (a server that's always on)
4. interpret email message content (attachments, headers, etc)
5. provide an interface for users to view sent email
6. provide an interface for users to view already read email

3.3. Non-Functional Requirements

1. Internet availability
2. Space availability on disk for running MailX

et cetera

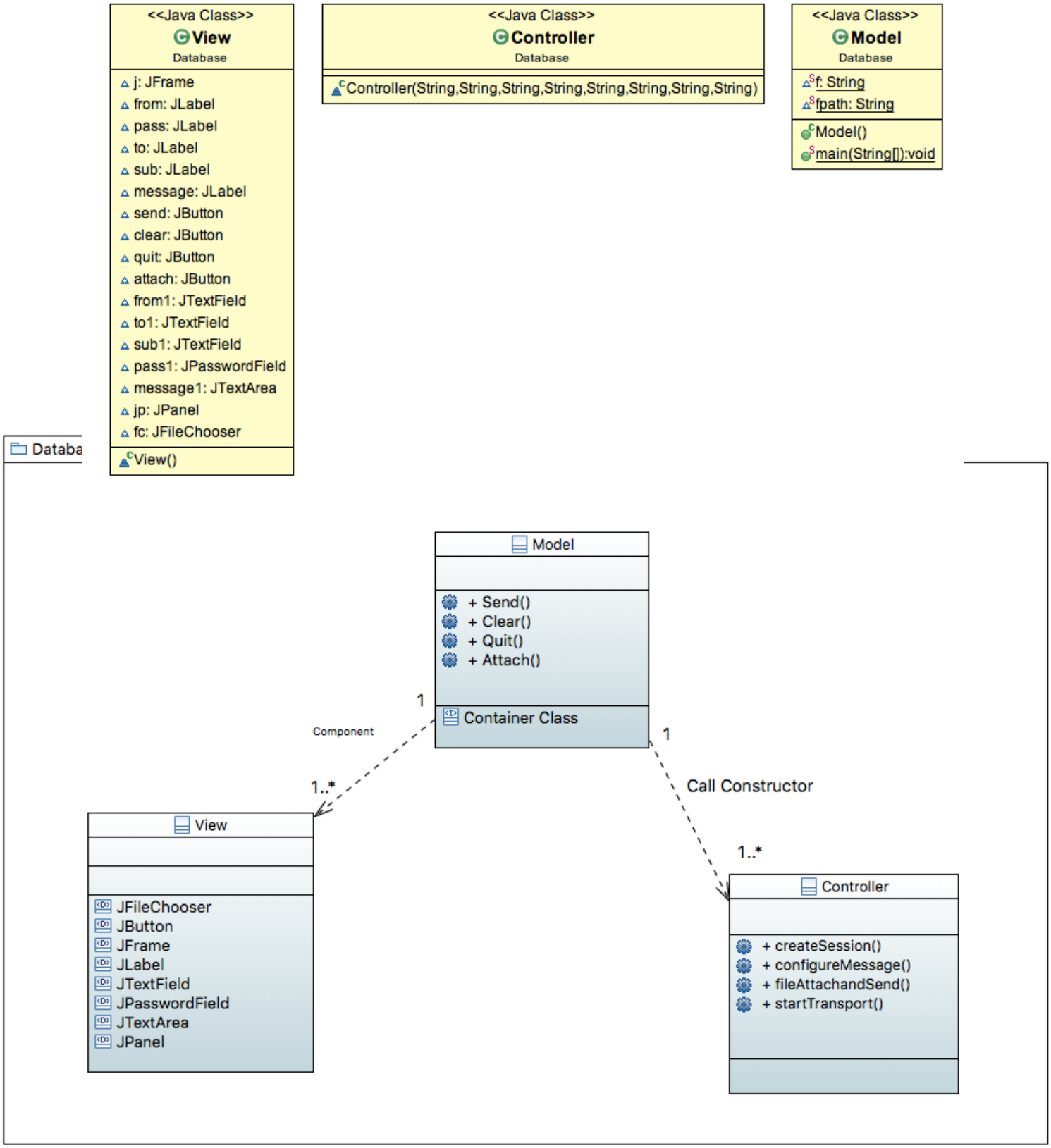
3.4. Performance Requirements

PE-1: The System shall give 24X7 service to all user without timeout as is a standalone application

PE-4: The system shall display confirmation messages to users within 10 seconds after the user submits information to the system.

3.5. Design Constraints

This work well with java.mail API version 1.4.4 (may work with newer , may not work with lower), JAVA 1.8 or higher is required to be present on clients machine.



Appendix A: Glossary

SRS :: Software Requirements Specification
JRE :: Java Runtime environment

JDK :: Java Development Kit
API :: Application Programming Interface
SMTP :: Simple Mail Transfer Protocol
FTP :: File Transfer Protocol
POP :: Post Office Protocol (now POP3)
IMAP :: Internet Message Access Protocol

Appendix B: Analysis Model

Figure 4. Class Diagram for mailing process

Figure 5. The MailX Class
Figure 6. Use-Case Diagram

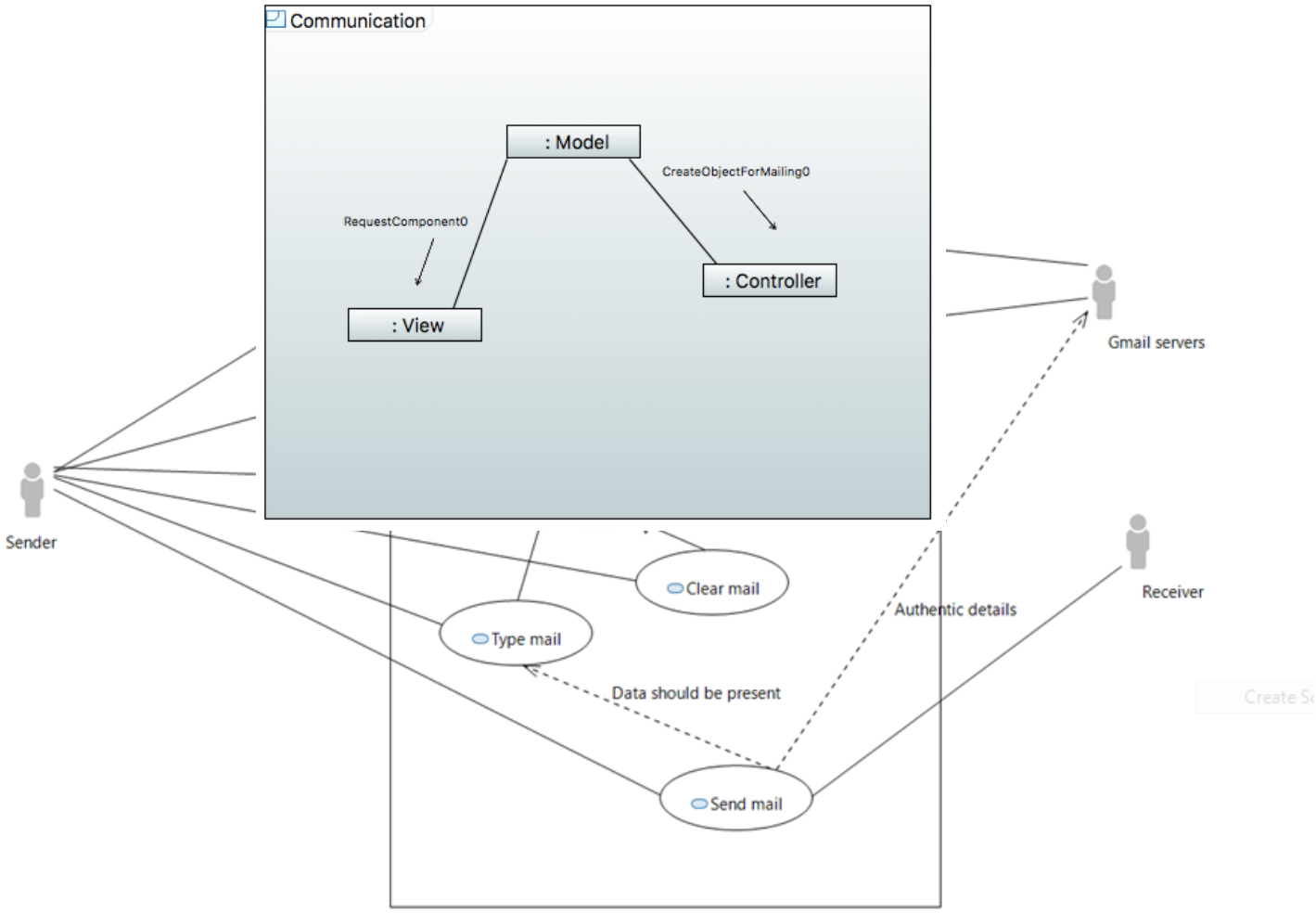


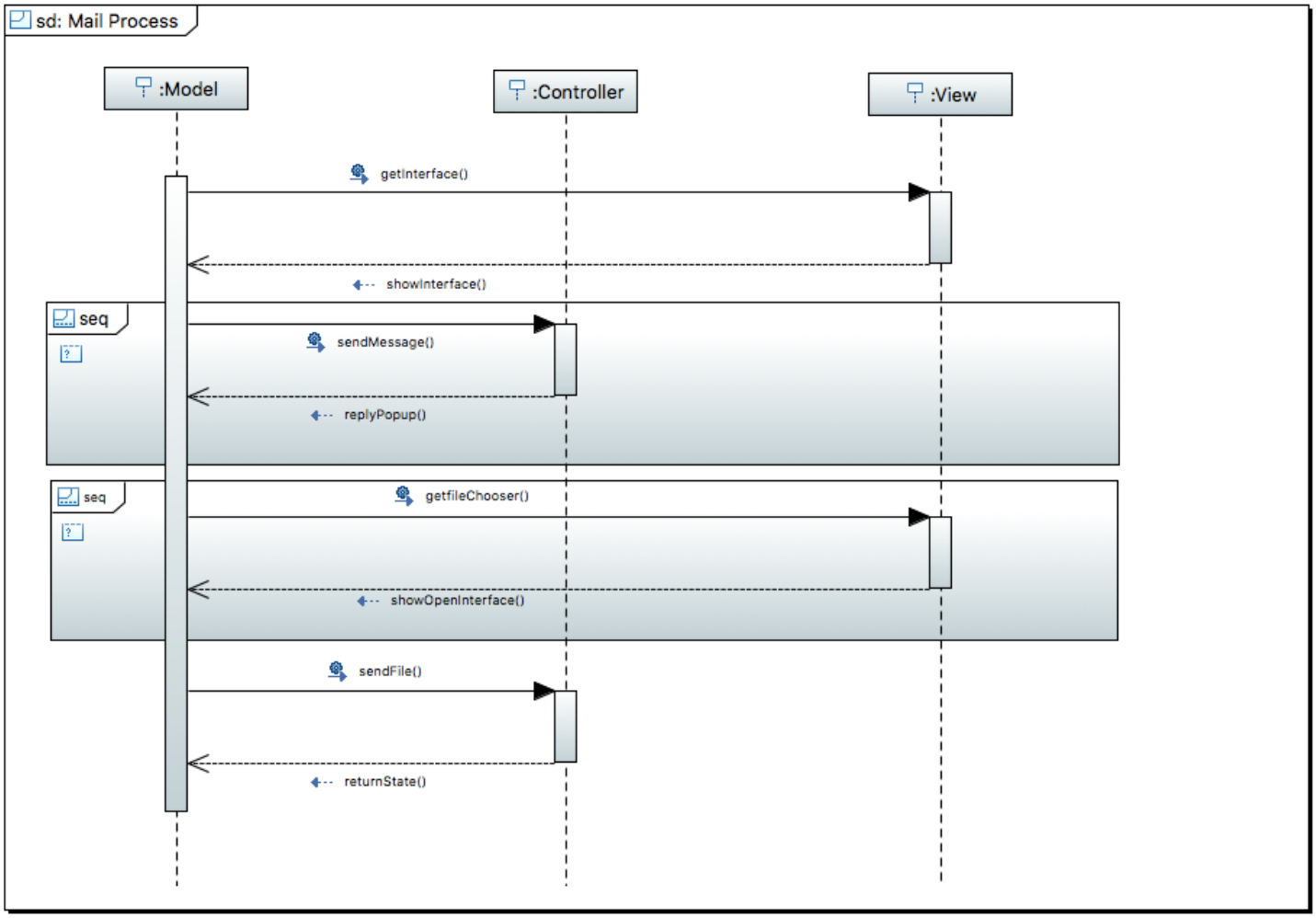
Figure 7. Communication Diagram
for sending Mails

Communication diagram (called collaboration diagram in UML 1.x) is a kind of UML interaction diagram which shows interactions between objects and/or parts (represented as lifelines) using sequenced messages in a free-form arrangement.

There is also no specific short form name for communication diagrams. Short form name sd (which is used for interaction diagrams in general) could be used. This sd is bit confusing as it looks like abbreviation of sequence diagram

Communication diagrams could be shown within a rectangular frame with the name in a compartment in the upper left corner.

Figure 8. Sequence Diagram for Mail Process


sd: Mail Process
:Model
:Controller
:View
getInterface()
showInterface()
seq
sendMessage()
replyPopup()
seq
getfileChooser()
showOpenInterface()
sendFile()
returnState()