Optimized Mail Client

Report – I
1 March, 2017

Vanshaj Bhatia
Sonam Mittal

Table of Contents

REVISION HISTORY .... **II**

DOCUMENT APPROVAL .... **II**

1. INTRODUCTION **1**

1.1 Purpose 1
1.2 Scope 1
1.3 Definitions, Acronyms, and Abbreviations **1**
1.4 References 2
1.5 Overview 2

2. GENERAL DESCRIPTION **2**

2.1 Product Perspective 2
2.2 Product Functions 2
2.3 User Characteristics 2
2.4 General Constraints 3
2.5 Assumptions and Dependencies 3

3. SPECIFIC REQUIREMENTS **2**

3.1 External Interface Requirements 3
3.1.1 User Interfaces *3*
3.1.2 Hardware Interfaces *3*
3.1.3 Software Interfaces *3*
3.1.4 Communications Interfaces *4*
3.2 Functional Requirements 4
3.2.1 <Functional Requirement or Feature #1> *4*
3.2.2 <Functional Requirement or Feature #2> *4*
3.3 Use Cases 5
3.3.1 Use Case #1 .... *5*
3.3.2 Use Case #2 .... *5*
3.4 Non-Functional Requirements 6
3.4.1 Performance *6*
3.4.2 Reliability *6*
3.4.3 Availability *6*
3.4.4 Security *6*
3.4.5 Maintainability *6*
3.4.6 Portability *6*
3.5 Inverse Requirements 6
3.6 Design Constraints 7
3.7 Logical Database Requirements 7
3.8 Other Requirements 7

4. ANALYSIS MODELS **7**

4.1 USE CASE Diagrams 8
4. 2CONTEXT Diagrams 8
4.3 DATA FLOW DIAGRAMS 8
4.4 STATE TRANSITION DIAGRAM 8
4.5 ENTITY RELATIONSHIP DIAGRAM 8

A. APPENDICES **9**

A.1 Appendix 1 9
A.2 Appendix 2 9

1. Introduction

Client is meant to be a role. For example, a web application which provides message management, composition, and reception functions may internally act as an email client; as a whole, it is commonly referred to as webmail. Likewise, email client may be referred to a piece of computer hardware or software world whose important role is to work as an email client.

1.1 Purpose

This Application is used for sending and receiving mail to and from the recipient in an optimized way.

1.2 Scope

1) This Application named Optimized mail client will be used to send and receive messages and attachments offline i.e. without the use of any browser.
2) Unlike the other mail servers this Application can be used for the other users who are not registered in the Optimized mail client.
3) Even if user lacks internet connection then also sending mails is possible.
4) Mail Server like yahoo, Gmail, Hotmail, outlook can be used for accessing the mail send by the sender.

1.3 Definitions, Acronyms, and Abbreviations

SMTP- **Simple Mail Transfer Protocol** is an Internet standard for electronic mail (email) transmission. First **defined** by RFC 821 in 1982, it was last updated in 2008 with the Extended **SMTP** additions by RFC 5321—which is the protocol in widespread use today. **SMTP** by default uses TCP port 25.

POP3 (**Post Office Protocol 3**) is the most recent version of a standard protocol for receiving e-mail. **POP3** is a client/server protocol in which e-mail is received and held for you by your Internet server.

IMAP4- In computing, the Internet Message Access Protocol (**IMAP**) is an Internet standard protocol used by e-mail clients to retrieve e-mail messages from a mail server over a TCP/IP connection. **IMAP** is **defined** by RFC 3501.

MIME-**Multipurpose Internet Mail Extensions** (**MIME**) is an Internet standard

- Text in character sets other than ASCII
- Non-text attachments: audio, video, images, application programs etc.
- Message bodies with multiple parts
- Header information in non-ASCII character sets

1.4 References

[1] Mail Client. (2016, August 20). Retrieved from https://en.wikipedia.org/wiki/Email_client

[2] Java Mail API Tag Library. (2016 August 20). Retrieved from https://javamail.java.net/nonav/docs/api/

[3] Java Mail. (2016, August 20). Retrieved from Mail Client. (2016, August 20). Retrieved from https://en.wikipedia.org/wiki/Email_client.

1.5 Overview

While popular protocols for retrieving mail include POP3 and IMAP4, sending mail is usually done using the SMTP protocol. Another important standard supported by most email clients is MIME, which is used to send file,

email attachments. Attachments are files that are not part of the email proper, but are sent with the email. The email clients will perform formatting according to RFC 5322 for headers and body, and MIME for non-textual content and attachments. Headers include the destination fields, To, Cc, and Bcc, and the originator fields from which is the message's author(s), Sender in case there are more authors, and Reply-To in case responses should be addressed to a different mailbox.

2. General Description

Client is meant to be a role. This application which provides message management, composition, and reception functions may internally act as an email client; as a whole, it is commonly referred to as Optimized mail client. Likewise, email

Client may be referred to a piece of computer hardware or software whose primary or most visible role is to work as an email client.

2.1 Product Perspective

The Application will be a Windows-based, self-contained and independent software product.

2.2 Product Functions

The System will allow access only to authorized users with specific role. Depending upon the user's role he/she will be able to access only specific module of the system.

- LOGIN- A Login facility for enabling only authorized access to the system.
- SENDING MAIL- The user can send their composed mail along with the attachment using this module.
- RECEIVING MAIL- The receiver client can view the pop-up notification using receiving module.
- Downloading attachments.
- Facility of latest news updates.

2.3 User Characteristics

- User have elementary computer knowledge and the knowledge of use of Application.
- The user should know the details of its corresponding email account.

2.4 General Constraints

The sender email-id and password is required. The mail will be sent to particular mail server provided in the drop down menu.

2.5 Assumptions and Dependencies

These are some following assumptions:

- The system is having required configuration as well as Windows operating system.
- Full working of Optimized mail client Application is dependents on the availability of Internet connection.

3. Specific Requirements

None.

3.1 External Interface Requirements

3.1.1 User Interfaces

Main/Central window:
This is the first screen that will be displayed and allows the user to enter his email-id and password. User also needs to select the host server whether its **smtp.gmail.com** for sending mail using Gmail or other.

Help window:
This is the screen/window available under the menu option "help" and it provides a general and brief introduction the proposed system.

About window:
This is the screen/window available under the menu option "help" providing ownership and build information about the software system.

3.1.2 Hardware Interfaces

- An IBM compatible PC is required to support and run the given software
- An x86, x86_64, arm, ultrasparc based processor. Java virtual machine is compiled and tested for these CPU architectures.
- At least 512 megabytes of RAM is required to run both the software and the java virtual machine.

3.1.3 Software Interfaces

- An IBM compatible PC running Windows/Linux/*BSD/*nix operating system.
- Java Runtime Environment to run and test the software.
- Java Development Kit to build and develop the software.
- Modelio for generating and viewing reports.

Optimized Mail Client

3.1.4 Communications Interfaces

None

3.2 Functional Requirements

- Ability to send mail.
- Ability to receive mail
- A standard SMTP client
- A standard IMAP4 client.

3.3 Use Cases

3.3.1 Login

3.3.1.1 Brief Description

This use case describes how a user can login into his email account using the software system.

3.3.1.2 Actors

The following actor(s) interact and participate in this use case:

- user
- SMTP Server
- POP3/IMAP Server

3.3.1.3 Flow of Events

- Basic flow: This use case starts when the actor wishes to send or read mail. The "login into account" dialog is presented to the user whereby he can choose the desired email server. The actor selects the SMTP/IMAP server with a mouse click and logins into email server using login credentials.
- Alternative Flows: In case the user is unable to login into the account using the software system, an error is displayed accordingly.

3.3.2 Compose Mail

3.3.2.1 Brief Description

This use case describes how a user can compose the mail using various editing options. User can send attachment as well.

3.3.2.2 Actors

The following actor(s) interact and participate in this use case:

- user

3.3.2.3 Flow of Events

- Basic flow: This use case starts when the actor has already logged in. As the user presses the compose mail option a window showing text area including editing options for text appears where user can compose mail.
- Alternative Flows: In case the screen does not successfully appears in the software system, an error is displayed accordingly.

3.3.3 Send Mail

3.3.3.1 Brief Description

This use case describes how a user can Send Mail which he has composed.

3.3.3.2 Actors

The following actor(s) interact and participate in this use case:

- user
- SMTP server.

3.3.3.3 Flow of Events

- Basic flow: This use case starts when the actor wishes to send the mail he has composed. The "Send" button is presented to the user on compose mail screen.
- Alternative Flows: In case the mail could not be sent, an error is displayed accordingly.

3.3.4 Receive Mail

3.3.4.1 Brief Description

This use case describes how a user can read his emails which he has received.

3.3.4.2 Actors

The following actor(s) interact and participate in this use case:

- user
- POP3/IMAP server.

3.3.4.3 Flow of Events

- Basic flow: This use case starts when the actor wishes to read the emails. The "Received mail/Inbox" menu provides the user with the desired facility.
- Alternative Flows: In case the received mail cannot be retrieved, an error is displayed accordingly.

3.3.5 Download Attachments

3.3.1.1 Brief Description

This use case describes how an attachment file can be downloaded from mail server.

3.3.1.2 Actors

The following actor(s) interact and participate in this use case:

- User
- POP3/IMAP server.

3.3.1.3 Flow of Events

- Basic flow: This use case starts when the actor has already logged into the email account and has read the received mail. The mail if contains attached files, they can be downloaded easily.
- Alternative Flows: In case the file is not successfully loaded in the software system, an error is displayed accordingly.

3.4 Non-Functional Requirements

Following are the non-functional requirements of the given software system.

3.4.1 Performance

The software should be able to send, receive mail, read news to an extent permitted by the operating system without affecting the overall performance of the system.

3.4.2 Reliability

The software system must provide reliability regarding

- The downloaded files should be in consistent state in case of failures.
- The unethical access of email account is protected as it uses strong networking protocols.

3.4.3 Availability

The software is supposed to be responsive in all possible and imaginable states, in that it should not leave a user astray.

3.4.4 Security

No specific concerns.

3.4.5 Maintainability

The software is supposed to be easily maintainable and easy to change.

3.4.6 Portability

The software is portable in that it is expected to run on the following platforms

- Linux
- mac
- windows

3.5 Inverse Requirements

The software system in no case should,

- Login into account with invalid login credentials.
- Should download the files in same format as they were received.

3.6 Design Constraints

The design constrains as applicable to the given software system are,

- The software is to be built using windows as development platforms.
- All the development tools licensed under some open or free license.
- Open JDK is used as java implementation.
- Developed on an x86 or x86_64 compatible processors.
- The coding style used is as per Oracle's *Code Conventions for the Java Programming Language.*

3.7 Logical Database Requirements

None.

3.8 Other Requirements

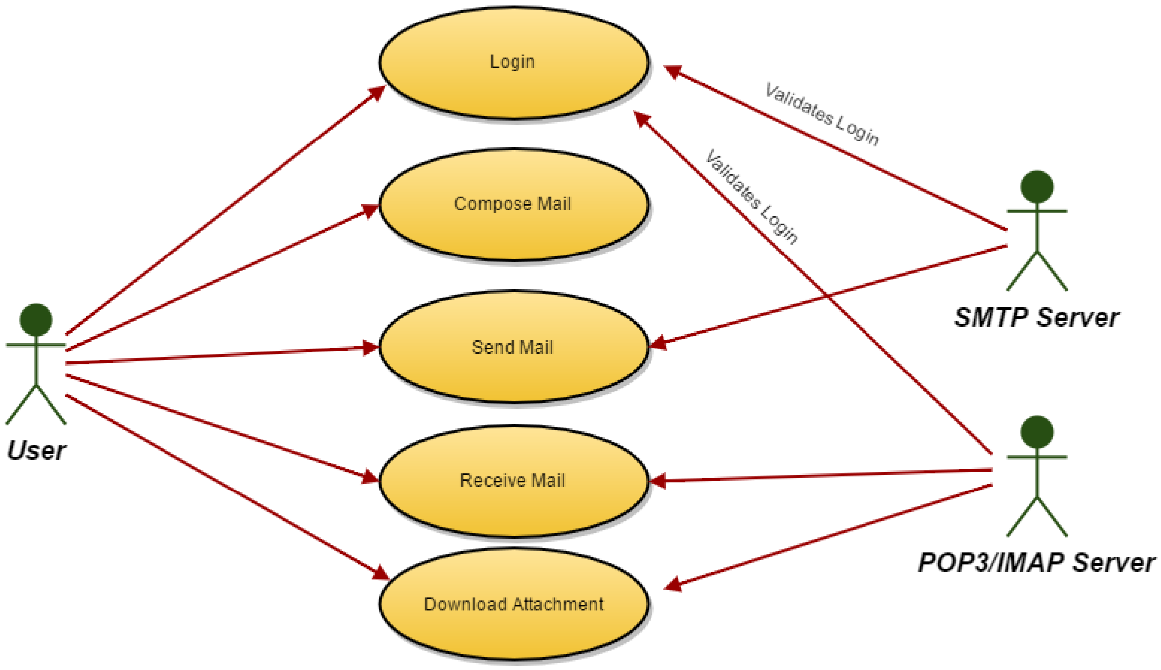
If possible following requirements can be implemented,

- Use swing for java for development of the proposed software system.
- Implementing an inbuilt library for network protocols.

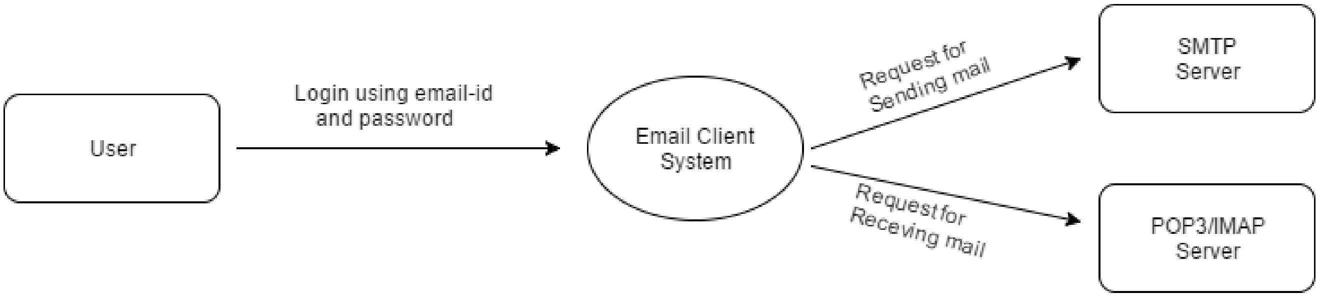
4. Analysis Models

None.

4.1 Use Case Diagram

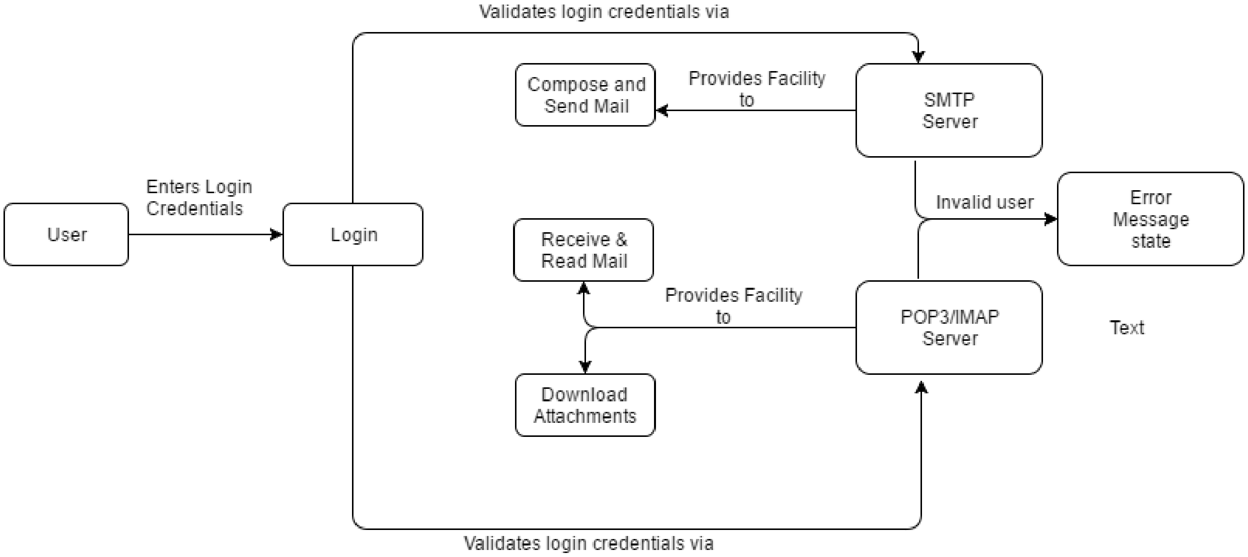


4.2 Context Diagram



Context Diagram-Email Client

4.3 Data Flow Diagrams (DFD)



Level-1 Data Flow Diagram

4.4 State-Transition Diagrams (STD)

None.

4.5 Entity Relationship Diagram (ERD)

None.

5. Change Management Process

The scope of the software does not indicate *future* changes in SRS document, in case a change is required, it should be accomplished by the following members of the development team,

- Vanshaj Bhatia
- Sonam Mittal

A. Appendices

A.1 Appendix 1

- **Open JDK :** Open JDK (Open Java Development Kit) is a free and open source implementation of the Java Platform, Standard Edition (Java SE). It is the result of an effort Sun Microsystems began in 2006.

- **Modelio:** Modelio is an Open Source UML tool developed by Modeliosoft, based in Paris, France. It supports the UML2 and BPMN standards.

.

A.2 Appendix 2

SMTP- **Simple Mail Transfer Protocol** (**SMTP**) is an Internet standard for electronic mail (email) transmission. First **defined** by RFC 821 in 1982, it was last updated in 2008 with the Extended **SMTP** additions by RFC 5321—which is the protocol in widespread use today. **SMTP** by default uses TCP port 25.

POP3 (**Post Office Protocol 3**) is the most recent version of a standard protocol for receiving e-mail. **POP3** is a client/server protocol in which e-mail is received and held for you by your Internet server.

IMAP4-In computing, the Internet Message Access Protocol (**IMAP**) is an Internet standard protocol used by e-mail clients to retrieve e-mail messages from a mail server over a TCP/IP connection. **IMAP** is **defined** by RFC 3501.