



Kreogist Mail Development Documentation

Software Requirement Specification

January 19, 2016



All information provided here is subject to change without notice. Contact Kreogist Dev Team to obtain the latest Kreogist product specifications and roadmaps.

Kreogist Mail may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

No computer system can be absolutely secure. Kreogist does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

The products described may contain design defects or errors which may cause the product to deviate from published specifications. Current characterized errata are available on request.

All the other documents mentioned in this document could be found at the official site of Kreogist Dev Team. Contact Kreogist Dev Team if there's any trouble.

Intel, Intel Core and the Intel logo are trademarks of Intel Corporation in the U. S. and/or other countries.

Linux is a trademark of Linus Torvalds in the U. S., other countries, or both.

Microsoft and Microsoft Windows are trademarks of Microsoft Corporation in the U. S. and/or other countries.

Macintosh is a trademark of Apple Inc. in the U. S., other countries, or both.

UNIX is a registered trademark of The Open Group in the U. S. and other countries.

*Other names and brands may be claimed as the property of others.

First Edition (Jan 2016)

This edition applies to Version 0.1 of Kreogist Mail.

Copyright © 2016, Kreogist Dev Team. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License". Note to government users restricted rights.

Kreogist Dev Team

**Kreogist Mail
Development Documentation
Software Requirement Specification**

January 19, 2016

Contents

Revision History	IV
Preface	V
1 Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Definitions, Acronyms, and Abbreviations	2
1.4 References	2
1.5 Overview	2
2 Overall description	3
2.1 Product Perspective	3
2.1.1 System Interfaces	3
2.1.2 User Interfaces	3
2.1.3 Hardware Interfaces	3
2.1.4 Software Interfaces	4
2.1.5 Communications Interfaces	5
2.1.6 Memory	5
2.1.7 Operations	5
2.2 Product Functions	5
2.3 User Characteristics	5
2.4 Constraints	6
2.5 Assumptions and Dependencies	6
3 Specific Requirements	7
3.1 External Interfaces	7
3.2 Functions	7
3.3 Performance	8
3.4 Logical Database	8
3.5 Design Constraints	8
3.6 Software System Attributes	8
3.6.1 Reliability	9
3.6.2 Availability	9
3.6.3 Security	9
3.6.4 Maintainability	10
3.6.5 Portability	10
Appendices	11
A Definitions	11
B Acronyms and Abbreviations	12

Revision History

Revision	Version	Description	Date
KMKOT01	-001	Initial release	Jan. 11th, 2016
KMKOT01	-002	Add preface section	Jan. 13th, 2016
KMKOT01	-003	Add appendix B	Jan. 14th, 2016

Preface

This document is an update to the specifications contained in the "Affected Documents" table below. This document is a part of product (project) Kreogist Mail.

This document may also contain information that was not previously published.

Affected Documents

Document Title	Document Number
Kreogist Mail Software Design Specification	KMKOT03

Related Documents

Document Title	Document Number
Kreogist Mail Software Project Management Plan	KMKOT02
Kreogist Mail Software Quality Assurance Plan	KMKOT04
Kreogist Mail Software Verification and Validation Plan	KMKOT05

1 Introduction

The introduction section of this document gives a general description and provides an overview of the entire software requirements specification (SRS). It provides the scope of the software, and the most accurate definitions of the abbreviations which will be used in the following sections.

1.1 Purpose

The purpose of this SRS is to give an exhaustive description of the requirements of Kreogist Mail (Mail) software. It will make a description of the objective and integrated declaration for system development. It will interpret the constraints, external interfaces and interactions with people, the system's hardware, other hardware, and other software as well. This document is written for schedule arrangement and development/testing organizing.

This is a reference document for project managers, designers, developers, testers and end users.

1.2 Scope

Mail is an E-mail management application for all mainstream desktop platform. It's a single user application. User could manage their mails in one or more E-mail accounts within a single window.

When user launches Mail for the first time, a wizard will be popup for asking user to login their Kreogist Account. Account will store their application settings, E-mail account information, etc. to server. When the next time user login their account on a new machine, their settings will be download automatically.

If user login their account on Mail for the first time (or sign up and login their account), wizard should ask user to add one E-mail account to application. User could add it later. But application won't work until there's one valid E-mail account.

Mail will show the main user interface. It should display the following widgets: folder selector, mail list and mail viewer. Folder selector will provide a list which all the E-mail account contains. Mail list should display all the mail in the folder which the folder selector selected. The mail viewer will display the selected mail in the mail list. And provides the button to manage the mail, such as: reply, move to folder, delete and so on.

Mail should also provide a simple configure dialog for user to manage all the settings of the application. User could log out and switch to another Kreogist account from the configuration panel. When user log out, their E-mail information will be removed as well, including all the mails.

When user writing a new mail, Mail should save the draft automatically every 3 minutes. User could change the interval in configuration panel. User could also select the mail box which will be used to send the editing mail.

Mail should provides the port for other application to launch it and asking for sending mail. It should also support the **mailto** protocol for Internet explorers to send the mail.

1.3 Definitions, Acronyms, and Abbreviations

Please visit Appendix A and B.

1.4 References

- [1]. IEEE Software Engineering Standards Committee. "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.
- [2]. Kreogist Dev Team. "Kreogist Mail Software Design Descriptions", January, 2016.
- [3]. Free Software Foundation, "GNU Free Documentation License", See <http://www.gnu.org/licenses/fdl.html> (last checked January 13th, 2016), November 3, 2008.

1.5 Overview

The document surplus two chapters and appendices.

In the second chapter, the Overall Description chapter, it will give an overview of the functionality of the product, describe the general factors that affect the product and its requirements and the application interaction with the client devices. This chapter is used to establish a context of the technique requirements specification, clarify the constraints and assumptions about this product.

In the third chapter, the Specific Requirements chapter, it will contain all of the software requirements and describe the product in detailed and technique terms. This part is written for developers and maintainers.

The Appendices at the end of this document provides some supporting and background information that can help readers of the SRS.

2 Overall description

This section will give an overview of the whole Mail software, including both the client side and the server side. Application will be explained in its context to show how it interacts with user, operating systems, peripheral device and introduce the basic functionality. Constraints and assumptions for system will be presented at the end of this section.

2.1 Product Perspective

Mail software will be divided into two parts: client application and content saving server. Client application could be set up on most desktop personal computer. Mail should support all the mainstreams operating systems. It will be used to gather the user data, save user configuration to the content server and get information from the mail server according to the information provided by user.

Client application will have only desktop version. Mobile version will be developed as another single application.

Content saving server part will use Kreogist User System which hosted on Bomb.cn.

2.1.1 System Interfaces

In order to launch this application on the client side, following are the minimum system requirements:

- Graphics: OpenGL 4.0+ on UNIX based OS. DirectX 9.0 or DirectX 11 for ANGLE on Windows.
- Internet Connection: A hardwired or wireless broadband Internet access.
- Bandwidth: 512kbps(64KB/s)
- Operating System: Please visit section 2.1.3. i.e., *Hardware Interfaces*.

2.1.2 User Interfaces

Please visit document *Kreogist Mail Software Design Descriptions*^[2].

2.1.3 Hardware Interfaces

Minimum hardware requirements for client:

Linux

- CPU: 2.0GHz Intel® Core™ 2 Duo CPU
- Memory: 1GB DDR II 800
- Disk Space: 1GB of available storage
- OS: Linux kernel 3.13 with a mainstream desktop environment installed Qt 5.5, e.g. KDE Plasma 5, Gnome 3.

OS X

- CPU: 2.0GHz Intel® Core™ Series Duo CPU
- Memory: 1GB DDR II 800
- Disk Space: 1GB of available storage
- OS: OS X Mountain Lion

Your Mac should be no older than the following models:

- MacBook Air (2012)
- MacBook Pro (2012)
- iMac (2012)
- Mac mini (2012)
- Mac Pro (Late 2013)

Windows

- CPU: 2.5GHz Quad Intel® Core™ i5 CPU
- Memory: 2GB DDR II 800
- Disk Space: 2GB of available storage
- OS: Windows XP Services Pack 3, Windows Vista Service Pack 1, Windows 7 or later.

2.1.4 Software Interfaces

1. Qt 5.5, the following module must be installed:
 - Core
 - Gui
 - Network
 - Widgets
 - All the official plugins for current platform

2.1.5 Communications Interfaces

Nowadays, mainstream E-mail solutions are divided into two groups: Internet Mail Access Protocol (IMAP) and Post Office Protocol - Version 3 (POP3). Mail should support both of them for receiving E-mail. All the mainstream E-mail solutions are supporting Simple Mail Transfer Protocol (SMTP) for sending E-mail. Mail should also support it.

Use QNetworkAccessManager to realize a high-level API for IMAP/POP3 and SMTP, all the other module shouldn't call the QNetworkAccessManager or any low level API classes directly.

2.1.6 Memory

Please refer section 2.1.3 i.e., *Hardware Interfaces*.

2.1.7 Operations

Please visit document *Kreogist Mail Software Design Descriptions*^[2].

2.2 Product Functions

Table 4: Operation List of Mail

S.No.	Particulars
1	Kreogist Account registration
2	Kreogist Account login
3	E-mail account management (Added, Modified, Removed)
4	Mail receiving (IMAP/POP3, Automatic/Manual)
5	Mail display
6	Mail editing
7	Mail sending (SMTP)
8	E-mail account and password automatically backup

2.3 User Characteristics

The mainly user of Mail software should be:

1. User may have a low education level. Our user may only know the basic computer operation. e.g. click, double click, type and knows the basic technique words.
2. User must be familiar with its E-mail account, which means user should know the E-mail account, password, and all the basic operation of E-mail system.

3. They shouldn't ever use Mail application before, so they couldn't have any experience on the Mail software.
4. They may have none of technical expertise on any parts of E-mail system.

2.4 Constraints

Since, the proposed Mail application to be developed using open-source environment/projects/technology, therefore following are limitations pertaining to these selected technology:

- More developer-oriented;
- Some unsolved bugs in the framework;
- Lack of professional official support;
- Documentation is not completed.

2.5 Assumptions and Dependencies

Mail should always be used on terminal devices which have enough performance. If the device doesn't have enough hardware resources for Mail, e.g. system might allocated resources for other applications, Mail may not work as intended or even at all.

Mobile computer should be well charged. If device runs out of battery, the latest data process on the device which still not have a chance to update will totally lost.

Users should be familiar with their device. They should know the basic operation to their device. e.g. clicking with their track pad, typing with the fixed or touch keyboard and turn on the special wireless switches.

Mail should be deployed on terminal device with all the Qt runtime library on it.

3 Specific Requirements

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

3.1 External Interfaces

Kreogist Mail doesn't need any special external interfaces.

3.2 Functions

1. Kreogist Account registration

We suppose all the users register a Kreogist Account. Kreogist Account could automatically backup your E-mail account settings, encrypt it and upload to Kreogist server.

2. Kreogist Account login

When user login the Kreogist Account, Mail will download the E-mail account and application configuration from server, and set it up automatically for user. They won't need to do all the configuration again.

3. E-mail account management

User could create a new mail, modified a draft mail and remove all kinds of mail from their mail folder.

4. Mail receiving

Mail could use IMAP or POP3 protocol to download mail from your E-mail server. It will update your mail box every 5 minutes. User could also update it manually by clicking a button. The waiting time will be reset if mail list updated manually.

5. Mail display

Mail will display the selected E-mail file. It should support rich-text format and HTML format E-mail.

6. Mail editing

User could edit any mail and resend it. User couldn't change the original mail content.

7. Mail sending

User could send E-mail via SMTP protocol.

8. E-mail account and password automatically backup

When our user login with its Kreogist Account, Mail will automatically encrypt the E-mail account and password and upload it to Kreogist Account server. User could block this backup in the configuration panel.

3.3 Performance

The requirements in this section provide a detailed specification of the player interaction with the software and measurements placed on the system performance.

Mail should complete the following listed items:

1. 99% of input operation should be processed and responded on user interface within 1 second.
2. 90% of Internet operation should be responded on user interface within 5 seconds.
3. 80% of online operation should be completed under minimal requirement within 30 seconds.

3.4 Logical Database

This section specifies the logical requirements for any information that is to be placed into a database.

All the E-mail information will be stored at local hard drive. We will use JSON format object to save in a UTF-8 format file. Read and write the file using `QJsonDocument` class.

E-mail account and password information will be stored as JSON format. The raw data will be encrypt with user object via AES-256 or better.

3.5 Design Constraints

Framework Mail shall be a stand-alone application running on mainstream desktop operating system. It should be developed using open source frameworks as mentioned in section 2.1.4. i.e., *Software Interfaces*.

Memory Usage The amount of system memory occupied by Mail should be limited.
Must No more than 100MB without any E-mail loading.
Plan No more than 70MB without any E-mail loading.
Wish No more than 30MB without any E-mail loading.

Disk Space The need of hard drive space of Mail should be limited.
Must No more than 100MB without any E-mail or cached data.
Plan No more than 70MB without any E-mail or cached data.
Wish No more than 30MB without any E-mail or cached data.

3.6 Software System Attributes

This section describes the required reliability, availability, security and maintainability of the client and the server side application.

3.6.1 Reliability

E-mail Receiving Mail should display all the E-mails received from the server correctly. But we cannot guarantee the network connections condition. So we have to suppose that there's nothing wrong with the Internet connection and the E-mail server.

Must 50% of E-mail files should be received correctly.

Plan 90% of E-mail files should be received correctly.

Wish 100% of E-mail files should be received correctly.

E-mail Display Mail should display all the mails correctly.

Must 100% of the received E-mail files should be display correctly.

Plan 100% of the received E-mail files should be display correctly.

Wish 100% of the received E-mail files should be display correctly.

3.6.2 Availability

Mail should be available throughout all E-mail sending process.

When the Internet connections is valid, Mail should always be available for E-mail receiving and user data synchronism.

3.6.3 Security

Kreogist Account system will encrypt user's password via MD5 and SHA-3. We won't stored the raw password.

User's E-mail account and password will be encrypt separately. E-mail account will be encrypt with AES-256 bit. Password will be encrypt as the following steps:

1. First encrypt the raw password with Elliptic Curve Cryptography (ECC), called N1.
2. Then append character '|', SHA-3 encrypt result, another '|' and MD5 encrypt result after N1, called N2.
3. Encrypt N2 with AES-512 bit, called N3.

Then upload the N3 to Kreogist Account server. When we download the N3 from the server, we have to decrypt the data with the following steps:

1. Decrypt N3 with AES-512 bit. We could get N2.
2. Separate N2 with character '|', encrypt first part with SHA-3 and MD5. Check the result with the second and the third part. If any of them is not the same, decrypt failed.
3. Decrypt the first part (N1) with ECC.

3.6.4 Maintainability

The client side application should be easy to extend. The code should be written in a way that it favors implementation of new functions.

None of user data will be fixed in code. There should not be any fixed binary image, sound or video data inside the code. All the multimedia files should be placed in a single resource file.

3.6.5 Portability

Mail should be portable with Windows, OS X and mainstream GUI Linux for desktop version. All of these platforms should be capable with the minimal system requirements. To know more about the requirements, please refer section 2.1.3. i.e., *Hardware Interfaces*.

A Definitions

The following table defines all the basic concepts in this document about Mail. Definitions given below are specific to this document and may not be identical to definitions of these terms in common use. The purpose of this part is to help users in understanding the document and the requirements of the system.

B Acronyms and Abbreviations

The following table lists the acronyms and abbreviations used in this document.

Table 5: Acronyms & Abbreviations

Acronyms	Definition
AES	Advanced Encryption Standard
ANGLE	Almost Native Graphics Layer Engine
API	Application Programming Interface
CPU	Central Processing Unit
DDR	Double Data Rate SDRAM
ECC	Elliptic Curve Cryptography
EDGE	Enhanced Data rates for GSM Evolution
e.g.	for example
GB	Gigabytes (10,737,418,240 Bytes)
GNU	GNU is Not Unix
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over TLS
i.e.	that is
IEEE	Institute of Electrical and Electronics Engineers
IMAP	Internet Mail Access Protocol
JSON	JavaScript Object Notation
KB/s	Kilo bytes per second
kbps	Kilo bit per second
MB	Million Bytes (10,485,760 Bytes)
MB/s	Million bytes per second
Mbps	Million bit per second
MD5	MD5 message-digest algorithm
OS	Operating System
POP3	Post Office Protocol - Version 3
SDRAM	Synchronous Dynamic Random-Access Memory
SHA-3	Secure Hash Algorithm 3
SMTP	Simple Mail Transfer Protocol
SRS	Software Requirements Specifications
TSL	Transport Layer Security
W-CDMA	Wideband Code Division Multiple Access
UMTS	Universal Mobile Telecommunications System