Software Requirements Specification
Athena

First Draft: February 8, 2010
Final Revision: February 16, 2010
Date of Submission: February 18, 2010

Gregory LeBlanc
Norman Maclennan
Stephen Failla

1 Introduction 1
1.1 Purpose of this Document 1
1.2 Scope of the Development Project 1
1.2.1 Project Name 1
1.2.2 Functionality Overview 1
1.2.3 Benefits 1
1.3 Definitions, Acronyms and Abbreviations 1
1.4 References 1
1.5 Document Overview 2
2 General Description 2
2.1 User Characteristics 2
2.2 Product Perspective 2
2.3 Overview of Functional Requirements 2
2.3.1. Data Communication 2
2.3.2 Communication Interface 3
2.3.2.1 Offline Interface 3
2.3.2.2 Online Interface 3
2.3.3. Interface Menus 3
2.3.3.1 File Drop-down Menu 3
2.3.3.2 Edit Drop-down Menu 4
2.3.3.3 View Drop-down Menu 4
2.3.3.4 Encryption Drop-down Menu 4
2.3.3.5 Help Drop-down Menu 4
2.4 Overview of Data Requirements 4
2.5 General Constraints, Assumptions, Dependencies, Guidelines 4
2.5.1 Constraints 4
2.5.1.1 Hardware 4
2.5.1.2 Software 5
2.5.2 Assumptions 5

2.5.3 Dependencies 5
2.5.4 Guidelines 5
2.6 User View of Product Use 5
Figure 2.6.1 5
Figure 2.6.2 6
Figure 2.6.3 7
3. Specific Requirements 7
3.1 External Interface Requirements 7
3.1.1 User Interfaces 7
3.1.2 Hardware Interfaces 8
3.1.3 Software Interfaces 8
3.1.4 Communication Interfaces 8
3.2 Detailed Description of Functional Requirements 8
3.2.1 Template for Describing Functional Requirements 8
3.2.2 Communication Interface 8
3.2.2.1 Offline Interface 8
3.2.2.2 Online Interface 9
3.2.3 Interface Menus 11
3.2.3.1 File Drop-down Menu 11
3.2.3.2 Edit Drop-down Menu 12
3.2.3.3 View Drop-down Menu 13
3.2.3.4 Encryption Drop-down Menu 13
3.2.3.5 Help Drop-down Menu 13
3.3 Performance Requirements 13
3.4 Quality Attributes 14
3.4.1 Security 14
3.4.2 Reliability 14
3.4.3 Availability 14

3.4.4 Maintainability 14
3.4.5 Portability 14
4 Other Requirements 14
4.1 Packaging Requirements 14
5 Use Cases 14
Appendix A 16
Appendix B 18

1 Introduction

1.1 Purpose of this Document

This Software Requirements Specification defines and describes the overall function and performance of the software. This specification will identify the functional, performance, development, and user requirements for the software application.

1.2 Scope of the Development Project

This scope of the development project can be described in three sections:

1.2.1 Project Name

Athena

1.2.2 Functionality Overview

Athena will be a standalone Java-based communication application that will allow users to exchange encrypted instant messages through a secure central server. Athena will have the ability to process encrypted messages in real time. Athena will have a constant connection to a database, and users will be able to observe the status of other users through a contact list in the main window of the application.

Athena will also have the ability to send encrypted files between online users, and exchange encrypted emails through a dedicated mail server on the Athena domain. Athena will give users the option to create a user name, password, and encryption key pair for data security. This option will be available through the Athena web site as well as the program's interface.

The software interface will consist of an offline interface and an online interface. The software interface look and feel of Athena will be customizable to the user. Basic appearance and layout characteristics will have multiple options for active users.

1.2.3 Benefits

Athena users will have the protection of RSA encryption during all online messaging, email, and file transfer sessions. Athena's level of encryption will provide substantial data protection and confidentiality for all users. The security aspects of Athena will not complicate the user experience. The interface of Athena will remain simple and intuitive.

1.3 Definitions, Acronyms and Abbreviations

See Appendix A

1.4 References

(n.d.). Retrieved Feb 2010, from Wikipedia:
http://wikipedia.org
AthenaChat.org. (n.d.). Retrieved Jan 2010, from AthenaChat:
http://athenachat.org
Deligiannidis, L.
(2009, Aug). Network Security.
How To Safely Store a Password. (n.d.). Retrieved
Feb 2010, from Codahale: http://codahale.com/how-to-safely-store-a-password/

http://github.com . (n.d.). Retrieved Jan 2010, from GitHub: http://github.com/maclennann/Athena-Chat
Java SE5 API. (n.d.). Retrieved Jan 2010, from Java Sun: http://java.sun.com/j2se/1.5.0/docs/api/

1.5 Document Overview

Section 1 of this document identifies the purpose and scope of the software and the document and lists all definitions, acronyms, and references. Section 2 is an overview of the description of application characteristics, requirements, and basic rationale for client use and experience. Section 3 contains a more detailed analysis of the functional, data, and hardware requirements and constraints. Data flow and relational concept diagrams will also be included in Section 3.

2 General Description

2.1 User Characteristics

Aside from basic consumer experience with a computer, no specific knowledge, skill, or training will be required on the part of the user prior to using this product. The user will not be expected to learn or memorize any commands or controls prior to installing and using this application. Commands, controls, and user preferences will be available through menus, tips, and help screens within the application. Users will be guided through the initial setup process via menus and screen prompts. Users will be ensured an encrypted connection before online communication can begin.

2.2 Product Perspective

The software described in this specification will be a standalone application that requires the Java Runtime Environment (Java Standard Edition 5 or greater) to run. The application will identify any existing Java environment during installation, and install the required environment if it doesn't exist on the target system. The software will use the Java Swing API to generate and present the GUI. The application will communicate with a dedicated server and database that is hosted by the Athena domain. Internet connectivity will be required by the user to create a successful connection with the application's host server. The application will provide support and functionality for all common hardware setups.

If the software is obtained for open-source development purposes, a connection to the host server will not be necessary. The source code for the server application code will be able to run in a Java development environment separately and simultaneously with the client application code.

2.3 Overview of Functional Requirements

The functionality of the software can be categorized into three major sections. These overview sections correspond to the detailed requirements in Section 3.2.

2.3.1. Data Communication

The software will be able to send and receive fully encrypted data in multiple formats. The software will permit the exchange of plaintext messages, emails, and files through an encrypted connection. The application will allow users to send and receive emails from a temporary session email address or a permanent address from an

external domain. The software will permit encrypted file transfer of local files on the user's computer or network.

2.3.2 Communication Interface

The software will provide an intuitive user interface for efficient data communication and menu navigation. The interface will consist of two essential sub-interfaces:

2.3.2.1 Offline Interface

The offline interface will provide the user with the opportunity to log in to the Athena database and create a secure connection to the server to initiate an online session. The software will provide a text location to enter a user name and password, and a selection of buttons that will allow the user to either proceed with login or exit the application. The user will be notified if any login information is invalid.

The offline interface will also provide an option to generate a new user account in the Athena database if necessary or desired. If a user name or password has been forgotten, an option will be available to send new login information to a specified email address pending verification.

2.3.2.2 Online Interface

The online interface will provide the user with a selection of options and settings during an online session. This interface will display a "contact list," a customizable list that contains other known Athena users that the current session holder has added as contacts. This list will provide buttons for adding and removing contacts from the list and creating groups within the list.

The interface will also display text areas for displaying plaintext messages that are sent and received through the established connection. The interface window will allow for customizable font style and color through a variety of buttons.

The application will provide an interface method of initiating a file transfer via a visible button.

The user will be able to begin or end an encrypted communication session with one or more online users. The online interface will organize the individual communication windows in a tabbed format. The interface window will also allow the user to scroll through the text communication area in the vertical direction. The software will also have the option to play a sound upon sending or receiving data in a session. The online interface will feature a menu bar with additional options and settings (See 2.3.3 for menu functionality overview).

2.3.3. Interface Menus

The online interface will provide a system of menus for extensive settings and options. The interface menu will consist of five drop-down menus:

2.3.3.1 File Drop-down Menu

The File menu will allow the online user to disconnect from a current session, display a status message, send a one-time-address email, send a permanent-address email, or exit the application.

2.3.3.2 Edit Drop-down Menu

The Edit menu will allow the online user to edit application preferences or change current password.

2.3.3.3 View Drop-down Menu

The View menu will allow the online user to view the temporary session inbox or adjust the visibility of the contact list.

2.3.3.4 Encryption Drop-down Menu

The Encryption menu will allow the online user to generate a new encryption key pair or export the current key pair.

2.3.3.5 Help Drop-down Menu

The Help menu will allow the online user to select an About tab that displays information and links about the software application.

2.4 Overview of Data Requirements

The application will prompt the user to create a new account upon successful installation. The user will be able to input a user name and password that will be stored in an online database. The user will not have access or viewing permissions to the database, but the user name and password that is created will be added to the database and the user will be notified upon successful account creation.

The application will display all text data exchanged in online sessions. Text data between users will be displayed in a plaintext communication area on a session tab. The user will be able to view sent and received data from other users. If the user is exchanging data with multiple contacts, text data will be displayed in exclusive tabbed windows corresponding to each contact. All visible data will be in decrypted format. User will be able to view sent and received emails in decrypted format. No data will be displayed to the user in encrypted format.

No communication data will be stored in the database or system after user terminates the secure connection. Sent and received text data will be displayed on the screen during an online session, and emails will be stored in a temporary online mailbox. After user logs off, all text and email data will be erased and unrecoverable.

When exporting an encryption key pair, a file with key pair information will be saved to a specified location on the user's local storage device. A user folder containing configuration files will also be generated upon installation. This folder will include a saved text file of the user's contact list from the most recent session. Other configuration files will be edited through the online interface menus.

2.5 General Constraints, Assumptions, Dependencies, Guidelines

2.5.1 Constraints

The constraints for the software application will be assessed in two parts:

2.5.1.1 Hardware

The application does not have any known hardware constraints or dependencies as of current release.

2.5.1.2 Software

The user must have an active internet connection in order for the application to initiate an online session with the server. The application can be opened in offline mode, but no online features will be available, and any attempts to log into the server will be refused until an internet connection is established.

The application may not function properly if certain corporate-level firewalls attempt to block attempted server connections. This is an unlikely event, but if a connection cannot be established, confirm that a firewall is not restricting use.

The application may not be able to export an encryption key pair to a location on the user computer if certain file or folder permissions have been established by a company administration or otherwise. If a permission error occurs while attempting to export a key pair, confirm that system permissions are not restricting use.

2.5.2 Assumptions

It is assumed that the encryption and hashing methods and algorithms used for creating secure connections within the application meet current and updated encryption standards and guidelines.

It is assumed that the requirements described in this document have different levels of priority. It is assumed that certain low-priority requirements may be postponed to ensure timely and successful completion of high-priority requirements. Requirements should only be removed or postponed pending agreement with customers and prospective clients. While there is currently no documentation of requirement priority, it is assumed that a requirement will not be removed unless it can be clearly justified as having no dependencies or connections to high-priority requirements.

2.5.3 Dependencies

The application requires the Java Runtime Environment to be installed on the computer of the user trying to run the application. Java Standard Edition 5 or greater is recommended for optimal performance. During installation, the application will check for an active version of Java, and install a working version automatically if one is not found upon installation.

This application is not dependent on any other hardware or software after the Java Runtime Environment has been installed.

2.5.4 Guidelines

This application does not have any specific guidelines for use.

2.6 User View of Product Use

Figure 2.6.1 shows the prototype layout of Athena. Designed with a singular window approach, the contact list appears in the same window as the messages.

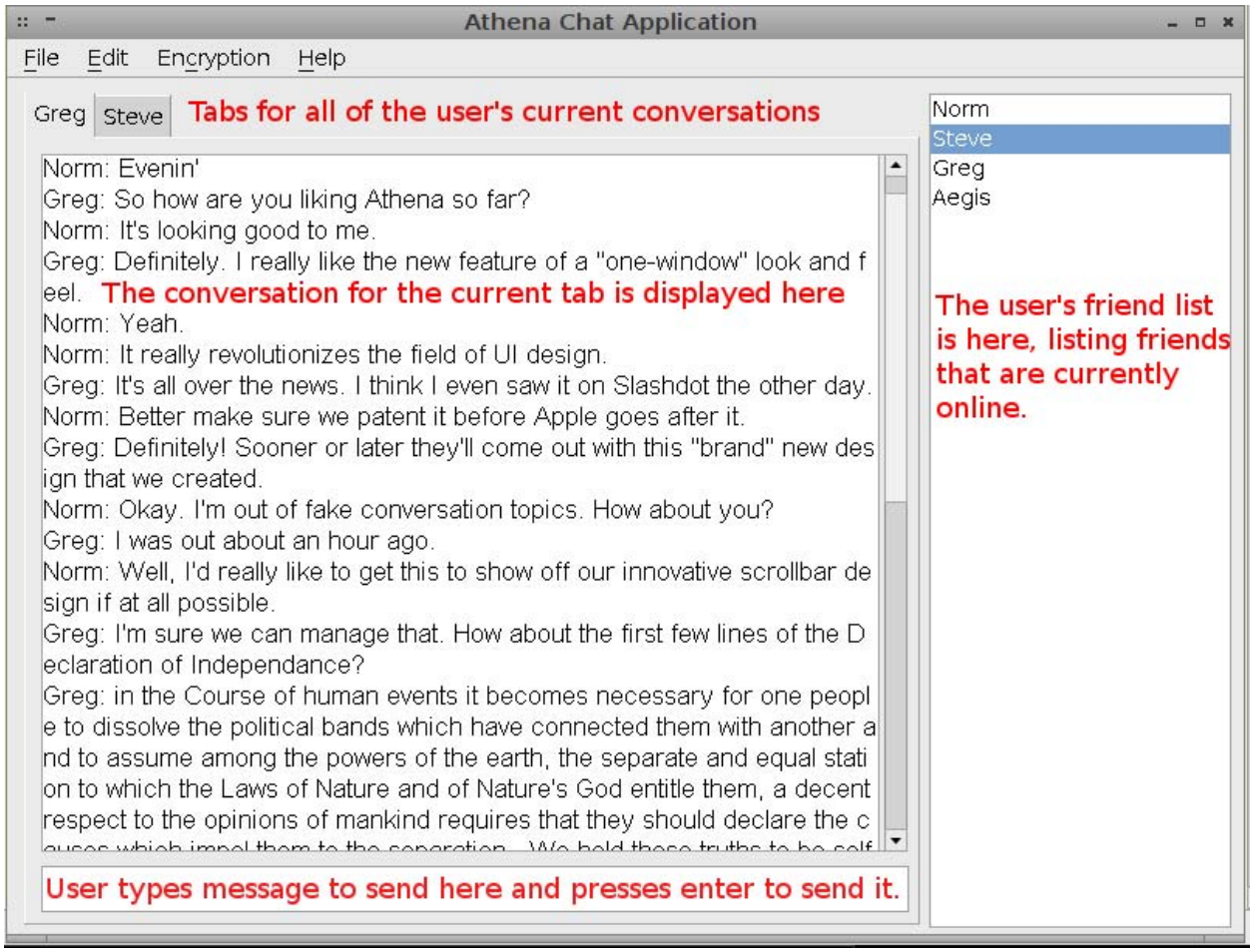
Figure 2.6.2 shows a marked prototype layout for easier understanding.

Figure 2.6.3 shows a high-level encrypted message transfer operation pattern.

Figure 2.6.1

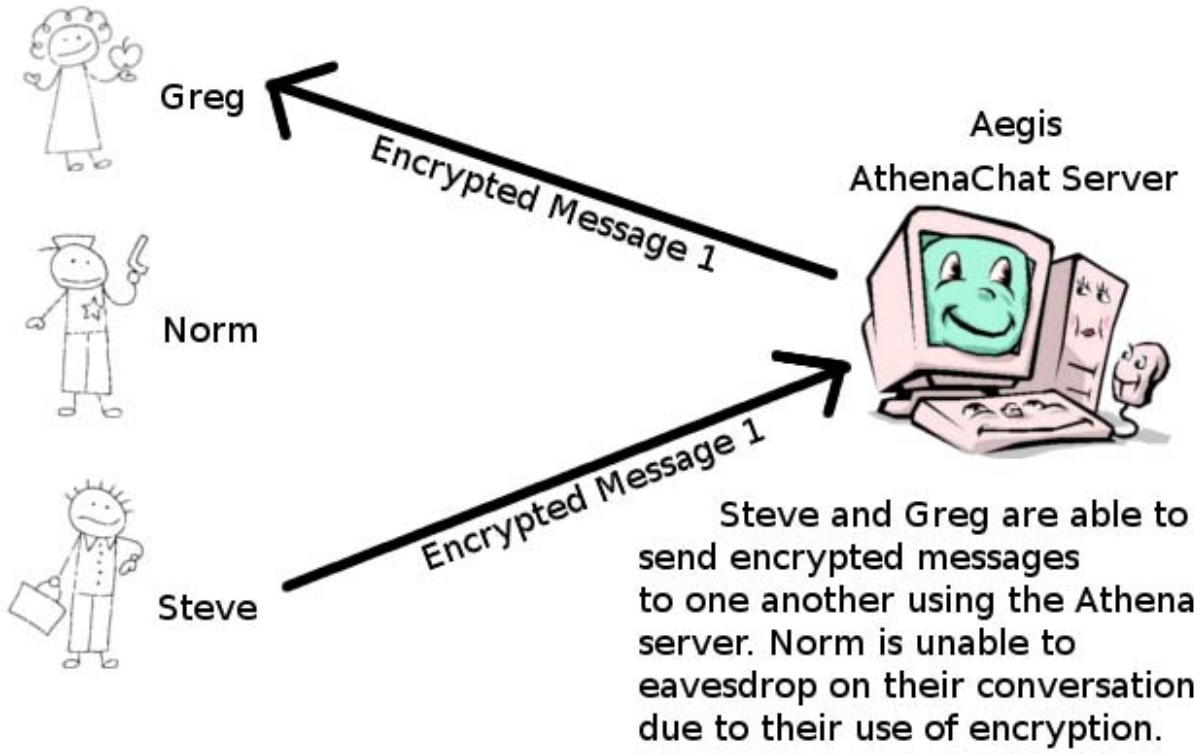
See Appendix B

Figure 2.6.2



Above is a sample conversation between two test users. The contact list and the other components in the window are also included.

Figure 2.6.3



Above is a diagram depicting sample message transaction logic.

3. Specific Requirements

3.1 External Interface Requirements

The external interface requirements can be described on four levels:

3.1.1 User Interfaces

All interaction between the user and the software will be supported by the software GUI. The interface of the application will provide either an offline or online interface to the user depending on connection status. When creating an account, the offline interface will provide visual notification to the user if an account already exists. When logging in to the server, if the user name and password combination entered is incorrect, the user will be visually notified. When using message tabs in the online interface, the user will be provided with the ability to scroll through text communication data that can not fit in the message tab window space. The online interface will provide the user with a separate window to compose and manipulate received and sent emails.

When a user attempts to disconnect from an online session, visual notification will be displayed confirming this choice. The online interface will also provide a separate settings window when a user chooses to edit application preferences during a session. If at any time a connection is refused or interrupted, the user will be visually notified of the connection error and a selection of possible causes will be displayed. The GUI will also provide the user with individual status windows while exporting or generating encryption key pairs.

Both the offline and online interfaces will have the ability to display a help window with important product information.

3.1.2 Hardware Interfaces

There are no external hardware interface requirements for the software.

3.1.3 Software Interfaces

The software will have the ability to use the operating system's file management system to locate and select files when initiating a file transfer during a session. The external Athena database interface will not be accessible to the user. Therefore, no requirements for the database will be defined.

3.1.4 Communication Interfaces

The software will have the ability to export encryption key pairs in a common format that is compatible with other encryption software applications using the same method of encryption.

3.2 Detailed Description of Functional Requirements

The tables provided in this section describe the detailed requirements of the software.

3.2.1 Template for Describing Functional Requirements

The format below will be the template for the proceeding section regarding the detailed functional requirements of the software:

3.2.x Primary Component
3.2.x.x Specific Component Interface

Component Name	Purpose	Inputs	Process

3.2.2 Communication Interface

The tables in this section describe the detailed requirements according to interface level.

3.2.2.1 Offline Interface

Component Name	Purpose	Inputs	Processing	Outputs
Create Account	Connect to the Athena database	User name, password, email, age	Verification of the user name, password, age, and password requirements.	Information will be entered into the database. One record per user. Any error messages will be displayed in a pop-up style window.
User Name Entry	User name is entered here as login procedure	User name	User name is sent to the database to verify that it is registered. If not, the user will be prompted to register.	If user name found and password is correct, a secure connection is established.

Component Name	Purpose	Inputs	Processing	Outputs
Password Entry	Password is entered here as login procedure	Password	Password is sent (hashed) to the database to verify that it matches. If not, the user will be prompted to try again. After 5 attempts the user is locked out for a predetermined amount of time.	When matched, the user gets access to Athena.
Forgot Password	If user forgets password - new password can be created	User name, email	Email is sent to user and user will be redirected to AthenaChat.org where they will create a new password	The new password is created and added to the database.
Connect	This initiates the connection with the server	The user name and password from the above fields, respectively	The information is sent to the server which allows the client to login.	User is logged in to Athena.
Exit	Exit Athena	None	All application windows are terminated.	Athena is closed and connections are terminated.

3.2.2.2 Online Interface

Component Name	***Purpose***	***Inputs***	***Processing***	***Outputs***
Send Message	Send text data to other users	The message the user wants to send	Verification will be placed upon the message before it is sent. The message will be encrypted using the user's private key before it is sent.	Message is sent to the recipient.
Receive Message	Receive text data from another user	The message is displayed in the user's window	Verification will be placed upon the received message before it is displayed. The incoming message will be decrypted using the sender's public key.	The decrypted message will be displayed in the user's window.
Contact List	View the user's available "contacts"	List is generated from the contact list file on the user's computer	Logic is applied to verify that the users on the "Contact List" are online.	The list will be displayed in a section of the "Online Interface."

Component Name	***Purpose***	***Inputs***	***Processing***	***Outputs***
View Contact Status	View a contact's current online status	The contact list will have the status listed	Validation will be in place to make sure all contact statuses are accurate and real time.	Contact statuses will be displayed below their user name in the contact list.
Add Contact	Adding a user to his/her contact list. This enables encrypted communication	The user name the user wishes to add	Validation will check to see if the user is an actual user of Athena.	The user is added to the contact list.
Remove Contact	Remove contact from the user's contact list	User name of contact to remove	An entry will be removed from the XML file containing user's contact list.	Removed contact will no longer appear in user's contact list.
Add Group	Allow user to organize contact list in a desirable format	Group name, contact(s) to add to group	The user's contact list XML will be updated, placing the specified contacts into the newly created group.	The user's contact list will update to reflect the new group and organization of contacts.
Remove Group	Allow user to organize contact list in a desirable format	Group to remove	The user's contact list XML file will be updated, removing the group and placing the members that were in the group in to a generic group.	The user's contact list will update to reflect the changes.
New Message Tab	Allow user to create conversation with exclusive contact	Contact to initiate connection with	A new tab is created inside the user's window, associated with the selected contact.	The user sees a new tab appear in the window, titled with the requested contact. Data communication can begin.
Close Message Tab	Allows the user to close an unwanted conversation	The user clicks on a small button in the tab title	The tab is removed from the window, and the object is destroyed.	The user sees the tab disappear from the window.
Send File	Allows the user to send an encrypted file to a specified recipient	The user chooses the recipient and the file to send	The file is broken into chunks, a hash of the file is created, and it is encrypted before being sent to the recipient, where it is decrypted and reassembled.	The user sees a message indicating that the file was successfully transferred. Recipient can view file in decrypted format.

Component Name	***Purpose***	***Inputs***	***Processing***	***Outputs***
Text Font Style	Allows the user to change the font style displayed in current message tab	Select a font style to use from a drop-down list	The font style property of the text box in the current tab will be updated to reflect the user's choice.	The font style in the current message tab will change to reflect the user's choice.
Text Font Color	Allows the user to change the font color displayed in current message tab	Select a font color to user from a pop-up window	The font color of the current text box will be updated to reflect the user's choice.	The font color in the current message tab will change to reflect the user's choice.
Text Font Size	Allows the user to change the size of the font displayed in current message tab	Select a font size to use from a drop-down menu	The font size property of the current text box will be updated to reflect the user's choice.	The font size in the current message tab will change to reflect the user's choice.
Text View Area	Display message tab conversations	Sent and received text data in corresponding message tab	Messages are added to the text view area as they are sent by the current user or received from other users.	The messages for the current conversation are displayed in real time.
Text Scroll Vertical	Allows the user to scroll through a long conversation	Buttons to control direction of scroll bar	The current conversation is placed into a scrolling pane that the user can scroll within.	The user can examine different parts of the message based on the position of the scroll bar.

3.2.3 Interface Menus

The tables in this section describe the detailed requirements according to menu.

3.2.3.1 File Drop-down Menu

Component Name	***Purpose***	***Inputs***	***Processing***	***Outputs***
Disconnect	Terminate current connection with server	None	After confirmation, client socket connection with server is closed and session data is erased.	User is returned to the offline interface.

Component Name	***Purpose***	***Inputs***	***Processing***	***Outputs***
Status Message	Notify users on contact list of current tasks or locations	Custom text entered by the user to be displayed as a status	The text will be verified for length and appropriately formatted content.	The validated text status is displayed under the user's name in the contact list of all online contacts.
Send One-time-address Email	Send an email to a temporary session address of an online contact	Desired data contents to be sent	Email contents will be verified for length and format and data will be encrypted before leaving client.	Encrypted email will be decrypted and displayed in recipient inbox in readable format.
Send Permanent-address Email	Send an email to a permanent domain address of an online contact	Desired data contents to be sent	Email contents will be verified for length and format and data will be encrypted before leaving client.	Encrypted email will be decrypted and displayed in recipient inbox in readable format.
Exit	Close application completely	None	Upon confirmation, client will be disconnected from server and any message tabs will be erased.	All application windows and interfaces will close, all connections will be terminated.

3.2.3.2 Edit Drop-down Menu

Component Name	**Purpose**	**Inputs**	**Processing**	**Outputs**
Settings and Preferences	Edit application behavior preferences and visual/audio settings	Selection of buttons to adjust corresponding settings and preferences	The corresponding setting that is adjusted by the user will be processed by the application.	All changes made will take effect upon confirmation and be visible to the user.
Change Password	Reset or change current account password for security or personal purposes	Old password, new password, new password confirmation	Old password will be verified for validity, new password and confirmation will be verified for appropriate length and format.	If any data is incorrect, user will receive a message to retry password change. Otherwise, user will receive successful confirmation message.

3.2.3.3 View Drop-down Menu

Component Name	Purpose	Inputs	Processing	Outputs
Session Inbox	View temporary session email mailbox	None	User's session inbox will be accessed, and any encrypted files will be decrypted for viewing.	Any emails that were sent or received during active session will be displayed.
View Contact list	Allow user to hide or show contact list	Check-box option	Status of box will be confirmed by application and view setting will be applied.	If box is checked, user's contact list will be visible in online window. Otherwise, contact list will be hidden.

3.2.3.4 Encryption Drop-down Menu

Component Name	Purpose	Inputs	Processing	Outputs
Generate New Encryption Key Pair	Create new encryption key if current key is compromised or unreliable	Key password	Key password is used to generate a new private and public key pair based on encryption algorithm.	Confirmation message is displayed notifying user of successful key pair generation.
Export Current Key Pair	Export current encryption key for use with other encryption applications	None	Encryption key file will be written to a file in the user application folder.	Confirmation message will be displayed notifying user of location of exported file.

3.2.3.5 Help Drop-down Menu

Component Name	Purpose	Inputs	Processing	Outputs
About Athena	Access useful information from product web site	Option to click on URL links to view web site	Application will identify default browser and send product link to the browser.	Browser will load link in new window and display product help page on the screen.

3.3 Performance Requirements

Specific performance requirements have not yet been established for the software.

3.4 Quality Attributes

The quality attributes of the software are described according to category.

3.4.1 Security

The encryption keys should not be susceptible to compromise. With the use of the international standards of encryption the keys and hashes should not be able to be spoofed or susceptible to a MITM (Man in the Middle) attack. All data will be encrypted during an online session, and no encrypted data will be stored after a session is terminated. Total user confidentiality is the fundamental priority of the Athena security profile.

3.4.2 Reliability

Downtime will be reduced to a minimal level. There will be weekly server down time for database maintenance and system patching. The product website will have constant updates and news feeds for users to obtain the latest information. The application will have a system of recoverable error handling procedures for common software errors.

3.4.3 Availability

Athena will be accessible from anywhere in the world. All software will be open-source and available for user development and experimentation.

3.4.4 Maintainability

Athena will have daily build updates as well as timely patches and critical bug fixes. The source code files will have comments concerning date of last change/revision to allow easier future modification and open source modulation.

3.4.5 Portability

Athena will be able to run on the following operating systems:

- Windows 2000 or later
- Linux
- Mac OSX

4 Other Requirements

Additional requirements for the software are described according to category.

4.1 Packaging Requirements

The software will be packaged as a single installation file. All source code and user documentation will be provided in the package file. When the package file is installed and extracted, a set of files will be created in an application folder that will include the core Athena files (Athena.exe), configuration files (Athena.conf) and user based files (contactlist.xml). Un-installation will be normally available through the user operating system.

5 Use Cases

Figure 5.1 High Level Data Flow Model for a simple message transfer

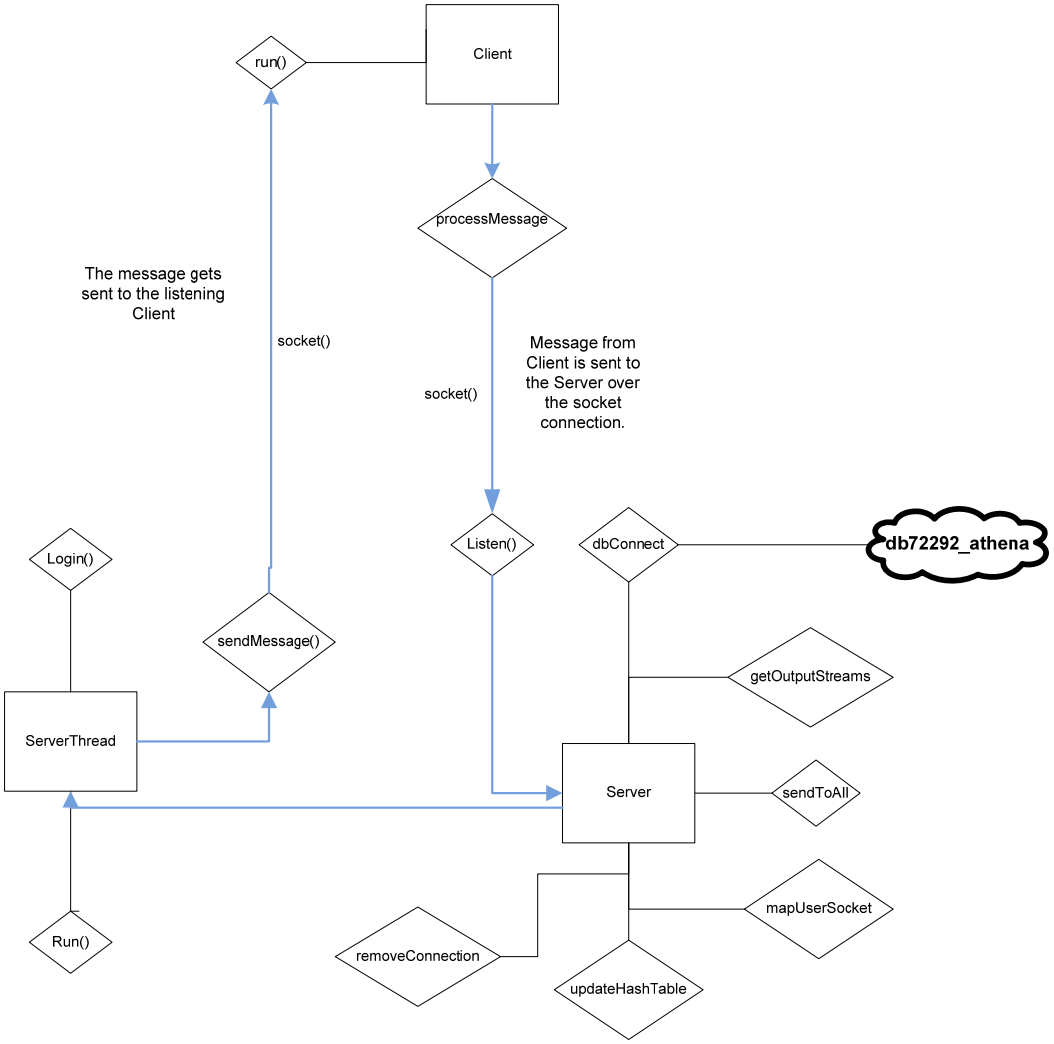
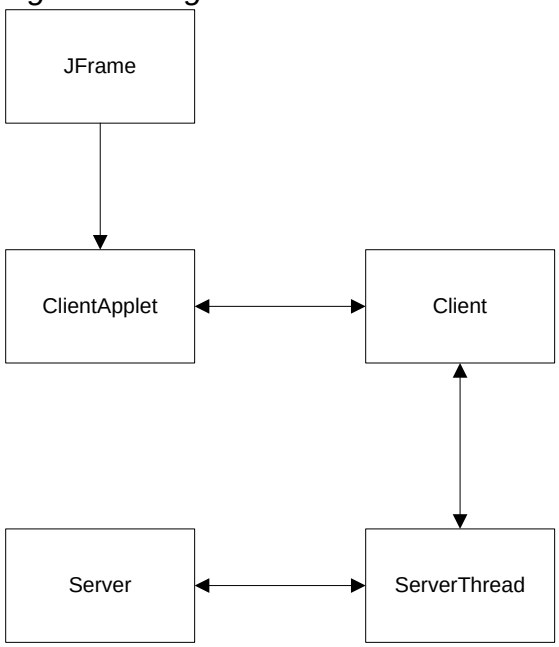


Figure 5.1 High Level Class Hierarchies



Appendix A

Name	***Formal Definition***
Advanced Encryption Standard (AES)	A method of encryption using symmetric keys. The same key is used to encrypt and decrypt the target message. This method is faster than RSA, but an agreement on a symmetric key must first be established.
Application	Computer software designed to help the user to perform a single or multiple related specific tasks. Such programs are also called software applications, applications or apps.
Application Programming Interface (API):	This is a public "interface" that allows software to interact with other software and users.
Central Processing Unit (CPU	The "brain" of the computer that executes the program's instructions, and performs all of the calculations for the encryption algorithms.
Client-Server Architecture	A common program architecture in which a user or users run a small "client" program which interacts with a main "server" to exchange information. The Internet is an example of client-server architecture.
Compiler	A software application used to convert source code into machine-language executable programs to be run by a computer.
Cross-platform	The ability for a program to run on different operating systems or computer types. For example, a cross-platform program is able to run on both Windows and Macintosh computers.
Diffie-Hellman	A method of generating one-time-use encryption keys (e.g. AES) secretively. Through the use of large prime numbers, users are able to agree upon and separately generate the same key without ever communicating over an unsecure connection.
Encryption	A method of hiding or obscuring information, generally using a mathematical algorithm. This information can only be recovered using a secret password or key.
File Transfer	A way of moving or copying a file from one location to another.

Name	***Formal Definition***
Hashing	A "one-way" algorithm. This is similar to encryption, but slightly distinctive in nature. A hashing algorithm will generate a unique string of characters of any data passed to it. These algorithms are designed to never generate the same string from different data, making it perfect for securely storing and verifying passwords.
Java	An object-oriented programming language that allows programmers to create applications on any operating system. Java uses a "Virtual Machine" to run its programs.
Java Virtual Machine (JVM)	Uses the java instructions generated by a Java compiler and translates them into machine code for execution. This makes java programs inherently cross-platform, because any java program will run on any computer capable of running the JVM.
Local-area Network (LAN):	A group of computers in a small physical area that are connected one another. LANs are smaller in scope than the Internet; the Internet is made up of a large group of interconnected LANs.
Machine Language	Instructions a program executes at a very low level that allows the application to run.
Man-in-the-middle Attack (MITM):	A common vulnerability in communications infrastructures in which an attacker can capture messages from one endpoint, and, after capturing the information, transparently relay the information to the intended recipient.
Operating System (OS):	An interface between hardware and user that is responsible for the management and coordination of activities and the sharing of the resources of a computer. The OS acts as a host for computing applications that run on the machine.
Plaintext	A message or text that is unencrypted and in human-readable form.
Random-access Memory (RAM):	Temporary cache memory commonly referred to as virtual memory. When computer programs are run or files are opened, they are loaded into the RAM for later access when it is needed.

Name	***Formal Definition***
Relational Database	A collection of tables used to organize specific information. A company may use a relational database to keep information on their employees and customers.
Rivest Shamir Adleman (RSA):	A method of encryption using a key pair comprised of a public and a private key. A message encrypted with the public key can only be decrypted with the private key, and vice-versa. This ensures confidentiality and represents a digital signature. The acronym references the names of the method's original designers.
Specific Hashing Algorithm 256 (SHA-256)	Part of the "SHA-2" family of hash algorithms. It creates a hash of a message that is 256 bits long. A bit is the basic unit of information in computing (e.g. A one or a zero).

Appendix B

Figure 2.6.1