|  |
| --- |
| University of Waterloo  Faculty of Engineering  Department of Electrical and Computer Engineering |
| Real-Time Instant Messaging System |
| Project Specifications and Risk Assessment |

|  |
| --- |
| Group #2015.010  Professor Werner Dietl (Consultant)  Qi Liu (20358515)  Asif Arman (20349964)  SangHoon Lee (20357600)  Danny Yan (20387735)  June 1, 2014 |

Table of Contents

[1. High-Level Project Description 2](#_Toc389435720)

[1.1 Motivation 2](#_Toc389435721)

[1.2 Problem Statement 2](#_Toc389435722)

[1.3 Block Diagram 3](#_Toc389435723)

[2. Project Specifications 4](#_Toc389435724)

[2.1 Functional Specifications 4](#_Toc389435725)

[2.2 Non-Functional Specifications 5](#_Toc389435726)

[3. Risk Assessment 6](#_Toc389435727)

[References 9](#_Toc389435728)

# High-Level Project Description

## 1.1 Motivation

Instant messaging systems of today are instant in name only.

Messages are sent when the send button is pressed by the sender, and no feedback to the receiver is presented during the waiting period except a wholly inadequate “user is typing” notification. This disruptive downtime between message and reply in modern instant messaging systems is becoming more apparent day by day.

In a 2007 study conducted by scientists at the Dortmund Institute for German Language and Literature, it was discovered that the average instant messaging system user discards 20% of their composed messages [1]. Evidently, messages in instant messaging systems of today are at constant risk of becoming obsolete before the send button is pressed.

Much of this awkward inefficiency could be avoided if there exists an instant messaging system that is true to its name: a system that provides instant, real-time communication between participants, without being bound by the archaic timing constraint dictated by the send button.

More recently, leaks provided by Edward Snowden on NSA’s overreach in its information collection practices has sparked a global discussion regarding government surveillance, and a surge in public demand for truly private and secure communication systems.

Some believe that the widely popular instant messaging network, Skype, is the solution to this problem as it makes use of a peer-to-peer architecture. However, this is no longer true of the network since 2012, when Microsoft replaced all of the decentralized supernodes in the Skype network (peers that had enough resources to act as relays for traffic between other peers) with servers under their control [2]. A quick look at the current Skype privacy policy verifies that Microsoft indeed reserves the right to collect “Content of instant messaging communications, Voice messages, and video messages” [3].

As of today, the demand for truly private and secure instant messaging systems has yet to be met.

## Problem Statement

We aim to design an instant messaging system that mimics the free-flowing experience of natural, in-person conversations, and at the same time protects the privacy of its users as an utmost priority.

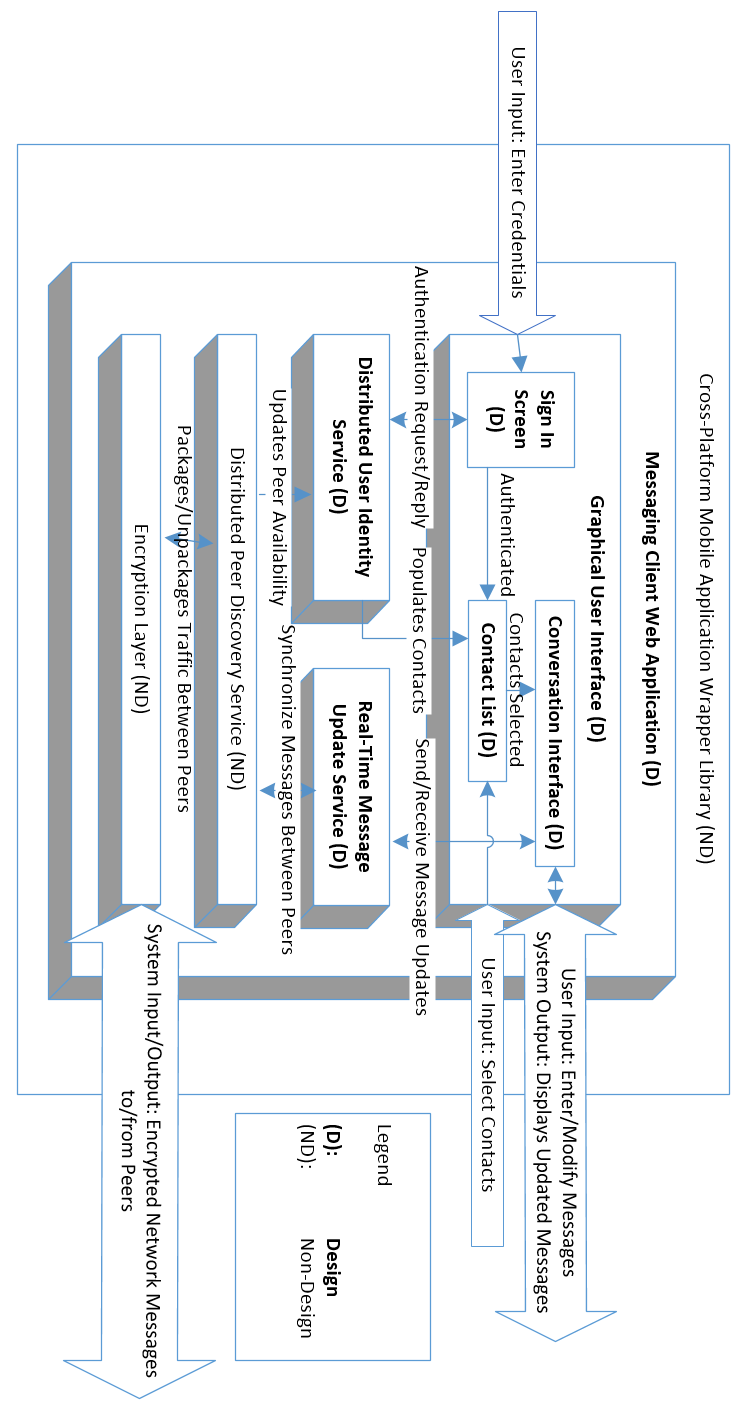
Our system will meet the former objective by updating messages on the receiver side as they are being written in real time, removing the disruptive delay between message and reply necessitated by the send button.

Ideally, we will meet the latter objective by implementing our system as a truly decentralized peer-to-peer architecture with no centrally controlled servers of any kind. Messages in this system will thus travel directly from the sender to the recipient, through a completely encrypted channel, giving no opportunity for any third-party to access message contents.

Alternatively, if a pure peer-to-peer system proves to be impractical for any reason, we will attempt to design a hybrid system that makes use of central servers for only peer discovery and authentication.

## Block Diagram

Figure 1: Block diagram of project components and inputs/outputs



# Project Specifications

## 2.1 Functional Specifications

Table 1: List of functional specifications

|  |  |  |
| --- | --- | --- |
| Functional Specification # | Essential /  Non-Essential | Description |
| 1 | Essential | Messages should be displayed in real time, the receiver should see the message that the user is typing as it is being typed. The average update time should be 2 seconds (or lower) assuming both sender and receiver are under an ideal and stable network. |
| 2 | Essential | Every message sent over the network will be encrypted, messages should use at least 128 bit encryption scheme |
| 3 | Essential | Users are not limited to one conversation, they will be able to participate in multiple conversations at a time |
| 4 | Essential | Users will have access to a contact a list that remains synced across all devices |
| 5 | Essential | Each member of the users contact list will display an availability indicator showing they are available to communicate |
| 6 | Essential | Users must properly authenticate themselves in a sign-on screen to obtain access to their contact list as well as send and receive messages |
| 7 | Non-Essential | The frequency of real time message updates should be adjustable to user. Both sender and receiver should be able to modify how often the message is being update within 0.1 second level of granularity. Users should be able to turn off the real time component completely so that the application behaves like traditional messaging systems. |
| 8 | Non-Essential | Communication between users is done using a private P2P channel using Distributed Hash Tables (DHT) for network discovery and connectivity. Client server architecture will be used as a fall back in case suitable DHT libraries are not found |
| 9 | Non-Essential | Message queuing will be supported, if the receiver of the message is offline (unavailable), it will be queued and the receiver will get the message when he/she is online |
| 10 | Non-Essential | Group chat should be supported, at least 4 users will be able to participate in the same conversation |
| 11 | Non-Essential | Group conversation messages should be accurately synced, every user will have the same view of the messages in terms of ordering |
| 12 | Non-Essential | Conversations history will be saved, when a user reconnects to their account (on a different device) they will be able to view the previous conversations between the other users and groups |
| 13 | Non-Essential | Real time widgets will be supported that allow users to perform more than just sending messages; such as posting a voting poll in a group conversation |

## Non-Functional Specifications

Table 2: List of non-functional specifications

|  |  |  |
| --- | --- | --- |
| Non-functional specification | Essential /  Non-Essential | Description |
| Efficiency | Essential | Efficiency is one of the most important non-functional specifications that our system requires. The system needs to be as efficient as possible in both local and network resource usage. In terms of network usage, on average it should not surpass 1MB per 10000 characters sent/received. In terms of local resource usage, it should never use more than 50MB of memory as a mobile or web application. |
| Portability | Non-Essential | As our system may need to run on multiple platforms, portability becomes important aspect of our system. It should be able to run on android, iOS mobile platforms in addition to the original web platform, retaining all of our functional and non-functional specifications. This can be done using open source libraries that compiles into different platforms once we have completed the core specifications. |
| Dependability | Essential | Our system needs to be very dependable. Since it is an instant messaging system it should be very reliable and robust. It should perform within our design limits without failure over time and should be able to respond adequately to unanticipated runtime conditions. Therefore, all messages should be delivered in the correct order and without any errors under ideal network conditions. Also, all failures in delivery due to poor networking conditions needs to be notified to the user. |
| Usability | Essential | Our system should be highly usable, with an intuitive and responsive interface. Since the main objective of our project is it to be real time. Our system needs to have very low input latency under all supported platforms. There should not be more than 0.5 second of lag between user input and interface response (not counting network latency) as long as the system has sufficient local resources for normal operation. |

# Risk Assessment

Table 3: List of possible risks, their probabilities and potential impact

|  |  |  |  |
| --- | --- | --- | --- |
|  | Nature of Situation | Potential impact | Probability of the Situation |
| Quality and maturity of open-source libraries needed for certain project components | Many of the project’s depended open-source libraries are in early development. Thus, these libraries have a higher probability of containing bugs or incomplete features.  In addition, the compatibility of the libraries or technologies may have potential issues or conflicts that might be caused by one another. | This situation can cause a devastating impact to our project success. If any of the depended open source libraries creates conflict during the development of the project, it can greatly delay our project completion time and decrease the chances of the project successfulness. | The probability of this situation is considered to be high. Due to the usage of the new open source libraries, where some libraries can be fairly new and still in under development, there is no ensure or guarantee in which the project can be constructed successfully by all the depended open source libraries.  Finding all possible alternative to each libraries can greatly reduce or avoid this situation. Having a list of alternative libraries can substitute dysfunctional or incompatible libraries which we can avoid failure or halting of our project.  Nonetheless, it is still possible for the worst case to occur. Therefore, to mediate the damage caused by the situation, the project may have to remove some features or fallback to its alternative features. |
| Require more time to complete the project. | The nature of this situation are time management and possibly insufficient skill or knowledge possessed by the group members. Time management is a great influence in determining the completion of project before the deadline because it depends on the time invested in completing each component of the project. The components of the project involves many new technologies in which time estimation for completing each component can be difficult to estimate.  Furthermore, group members are likely require to learn and develop in new languages and environment which require time, and not all group members can learn at the same speed. Thus, there will definitely be unexpected delays in the project. | Failure to complete the project under the deadline can result to academic penalties. The produced delays by poor time management can also increase or create more potential delays in the project progress. | The probability of this situation is medium because the group members have a flexible schedule throughout the duration of project to reduce the chances of this situation occurring. |
| Team member leaving or disbandment of the team. | The nature of this situation is group dynamic problems. In every project involving two or more people, it is inevitable that there will be conflicts between team members when working together. | In such a case where the situation of a team member were to leave the group, it will greatly affect the project and the remaining team members. Addition time required to complete the remaining work of the leaving member, and the spirit of the team will be heavily affected. In the worst case, the entire team can disband and abandon the project. | The probability of this situation were to occur is low. The team members have worked on previous projects involving cooperation and teamwork. |

# References

|  |  |
| --- | --- |
| [1] | T. Simonite, "New Scientist Blogs," 07 December 2007. [Online]. Available: http://www.newscientist.com/blog/technology/2007/12/instant-message-irrelevance.html. [Accessed 01 June 2014]. |
| [2] | D. Goodin, "Skype replaces P2P supernodes with Linux boxes hosted by Microsoft (updated)," 01 May 2012. [Online]. Available: http://arstechnica.com/business/2012/05/skype-replaces-p2p-supernodes-with-linux-boxes-hosted-by-microsoft/. [Accessed 01 June 2014]. |
| [3] | Microsoft, "Skype Privacy Policy," 2014. [Online]. Available: http://www.skype.com/en/legal/privacy/. [Accessed 01 June 2014]. |