Deliverable #1: Software Requirement Specification (SRS)

SE 3A04: Software Design II – Large System Design

Febuary 16, 2023

Tutorial Number: T02
Group Number: G6
Group Members:

- Emily Perica
- Harman Bassi
- Kyen So
- Kelly Deng
- Swesan Pathmanathan

IMPORTANT NOTES

- Be sure to include all sections of the template in your document regardless whether you have something to write for each or not
  - If you do not have anything to write in a section, indicate this by the *N/A*, *void*, *none*, etc.
- Uniquely number each of your requirements for easy identification and cross-referencing
- Highlight terms that are defined in Section 1.3 (**Definitions, Acronyms, and Abbreviations**) with **bold**, *italic* or underline
- For Deliverable 1, please highlight, in some fashion, all (you may have more than one) creative and innovative features. Your creative and innovative features will generally be described in Section 2.2 (**Product Functions**), but it will depend on the type of creative or innovative features you are including.

1 Introduction

This SRS describes the software requirements for our messaging application, MacMessenger. The application is meant to provide a secure means of communication within a company. This document will outline the purpose, project scope, requirements, and use cases of the product.

1.1 Purpose

This SRS is meant to provide an overview of the requirements imposed upon the MacMessenger system, providing information on functional and non-functional requirements, as well as use cases and potential employee characteristics. The intended audience of this SRS is all stakeholders involved in the MacMessenger product. It will provide a high-level overview of the requirements needed to begin designing the product, using a non-technical language that may be understood by stakeholders who may have a broad range in their understanding of technical terms.

1.2 Scope

MacMessenger is a secure messaging application that allows employee to send messages on company issued Android devices through sessions of key authentications and store chat logs on the server. It also allows employee to create group chats, attaching files and customizing account.

Employees are required to sign-in or create an account to access the application. Main services of the application include “Sending messages”, “Creating group chat”, “Attaching files” and “Customizing account”. In “Sending messages”, employee will fetch a key from KDC and authenticates with the server to start a session and send messages. System will encrypt/decrypt the message using symmetric key cryptography and store the chat log onto the server. In “Creating group chat”, employee(s) can add contacts to group chat creation and others will receive notification for being added. In “Attaching files”, an employee can add a file by dragging it into chat box and click “Send” to display it in chat interface. In “Customizing account”, an employee can modify their account information such as avatar, nick names etc.

The objective of this application is to avoid corporate espionage within the organization by establishing secure communication via authentication keys on company android devices, so to reduce the risks of information leakage and cybersecurity attack. As the software will be used by employees who may have minimal technical background, the software will be intuitive to navigate and accessible. The software will also have fast response time since it is for business use.

The goal of this software is to promote safe communication means between employees, as companies heavily rely on online communications nowadays.

1.3 Definitions, Acronyms, and Abbreviations

DoS: Denial of Service attack. A type of cyberattack in which the attacker attempts to stall services by flooding the server with requests.

Employee: A worker employed by the company utilizing MacMessenger. Used interchangeably with ‘user’ throughout the SRS.

KDC: Key Distribution Center. Used to generate all keys used in a secure communication session.

MacMessenger: Secure messaging application.

Symmetric-key cryptosystem: An encryption scheme in which both involved parties have access to a private, pre-determined key.

1.4 References

References

[1] Gillis, A. (2022, June).What is a security audit?TechTarget. https://www.techtarget.com/searchcio/definition/security-audit

[2] Make Google Messages more accessible - Google Messages. (n.d.). Support.google.com. https://support.google.com/messages/answer/6105764?hl=en

[3] Non-functional Requirements: Examples, Types, How to Approach. (2022, July 26). AltexSoft. https://www.altexsoft.com/blog/non-functional-requirements/

[4] Understanding SOLID Principles and Clean Architecture. (n.d.). Www.yourteaminindia.com. Retrieved February 18, 2024, from https://www.yourteaminindia.com/blog/understanding-solid-principles-and-clean-architecture: :text=Architecture

[5] NIST. (2022, July 21).NIST Cybersecurity Privacy Program[Review ofNIST Cybersecurity Privacy Program].https://www.nist.gov/system/files/documents/2022/07/21/Extended

[6] frank@v51.com. (2019).Your privacy rights - IPC. IPC. https://www.ipc.on.ca/privacy-individuals/your-privacy-rights/

[7] Wang, Y., Huang, Y., Li, J., Zhang, J. (2021). The effect of mobile applications' initial loading pages on users' mental state and behavior.Displays,68, 102007. https://doi.org/10.1016/j.displa.2021.102007

[8] Android OS. (n.d.). Endoflife.date. https://endoflife.date/android

[9] Core app quality. (n.d.). Android Developers. https://developer.android.com/docs/quality-guidelines/core-app-quality

1.5 Overview

Section 2 provides a general description of the product, delving into the perspectives, functions, users, constraints, and assumptions that must be considered when designing the system. Section 3 presents a Use Case Diagram to provide the reader with a visual understanding of the desired flow of the system. Section 4 describes the Functional Requirements of the system, and section 5 describes the Non-Functional Requirements of the system. Lastly, Section A contains the Division of Labour of this SRS.

2 Overall Product Description

2.1 Product Perspective

Compared to applications like WhatsApp and Singal the main property of the MacMessenger will be to provide proper security between the employees of the company. The end-to-end encryption ensures that the messages in the application are not easily attainable like Signal. MacMessenger focuses on providing a key that would be used to ensure encryption like Signal. Signal makes sure that the key is also not stored in external servers to ensure discrepancy. In terms of looks, the application would resemble Facebook Messenger as it provides the user with a very intuitive design making it easy to understand how to use the messenger app. MacMessenger also allows the user to add or remove people from chat groups. This would resemble all the previous related apps mentioned.

The application itself is self-contained as all the main applications are within the company. The product communicates with the businesses database to ensure the employees have an account for specific authorization. Overall, it depends on existing interfaces that exist within the company to complete authorization and beyond that everything else is done within the MacMessenger ecosystem. This is to ensure secrecy and protect valuable data.

Interfaces within the Overall System:

The User Interface is an important aspect when it comes to what the employee sees and interacts with. The employee is presented with the front-end and is allowed to interact with it, given the certain perameters of the application. This would be the system that takes in the user inputs to provide wanted results.

The Messaging System is where the encryption and overall distribution of messages occurs. When the employee sends their message it enters the system, encrypted it and then sends it to the other user. The system is also responsible for communicating with the other systems to ensure the authorization of the user and the proper distribution of keys.

Authentication Server is where the system would connect to the business's database to ensure that the employee is entering their correct information and if they are allowed access to the application. This entity would communicate with the Messaging System to authenticate the user.

The KDC Server is responsible for providing and storing the keys that would be sent to the employee. This is to ensure it helps encrypt the messages sent by the user. This will regularly be updating itself to ensure that users will always be provided with new keys to ensure security.

The Chat Log Database will store all the chat logs from the application. After each message is sent it is stored within the chat log database so that an authorized person has access to it when need be. This is to ensure that every chat can be accessed even after chat deletion to ensure a proper record of every company discussion within the application.

2.2 Product Functions

tbd.

2.3 User Characteristics

This system is meant to be user friendly and simple to navigate; it will be designed with the following stakeholders in mind:
User group 1: Interns (Primary stakeholder)

- Part-way through a bachelor's degree
- 19-23 years old
- Android users
- Have internet access.
- Little to no experience in their field

User group 2: Full time employees (Primary stakeholder)

- Minimum education: bachelor's degree
- 23+ years old
- Android users
- Have internet access.
- Experience ranging from junior level to upper management.

User group 3: IT Employees/Technical Maintainers (Tertiary stakeholder)

- Have at least a bachelor's degree
- Tech-savvy
- 23+ years old
- Have internet access.
- Have a thorough understanding of the communication system and its technical maintenance.

2.4 Constraints

The project's development is bound by several key constraints that affect its timeline, design, and functionality. The necessity to complete the project by a specified due date, alongside the imperative to balance security measures with a employee-friendly design, notably influences the development approach. Compatibility requirements with Android devices, considerations for mobile resource limitations, and the need to ensure the application operates reliably across varied network conditions further restrict development choices. Additionally, adherence to data protection laws and integration with existing organizational infrastructure pose significant considerations.

Furthermore, the application's design must prioritize scalability and ease of maintenance to accommodate future growth, all while adhering to a defined budget. Secure data management, including chat history and compliance with data retention policies, alongside comprehensive testing, and quality assurance processes, are crucial for safeguarding employee information and ensuring the app's reliability. These constraints collectively guide the development process, ensuring the final product meets organizational needs, regulatory requirements, and employee expectations, even within the constraints of cost, performance, and security.

2.5 Assumptions and Dependencies

- List any assumptions you made in interpreting what the software being developed is aiming to achieve
- The software will only be provided on company-issued Android devices. Not supported on other devices in the market.
- Devices will have stable connection to the Internet during communication.
- It is assumed that the server for storing chat history logs will be provided, and will always have sufficient storage space for chat logs.
- It is assumed that there is a database for storing existing employee names/passwords
- It is assumed that there is no restriction on what types of authentication protocols and cryptosystem that will be implemented in the software (? More like constraints?)
- The software will only be used within Canada.

2.6 Apportioning of Requirements

Certain requirements may be earmarked for implementation in future versions of the system to ensure that the initial release is manageable, focuses on core functionalities, and can meet the deadline and budget constraints. Advanced features such as enhanced encryption algorithms for even higher security, deeper integration with other organizational systems, additional employee customization options, expanded compatibility with a broader range of Android versions, or more sophisticated chat functionalities (like file attachment or editing messages, group management features) might be deferred.

3 Use Case Diagram

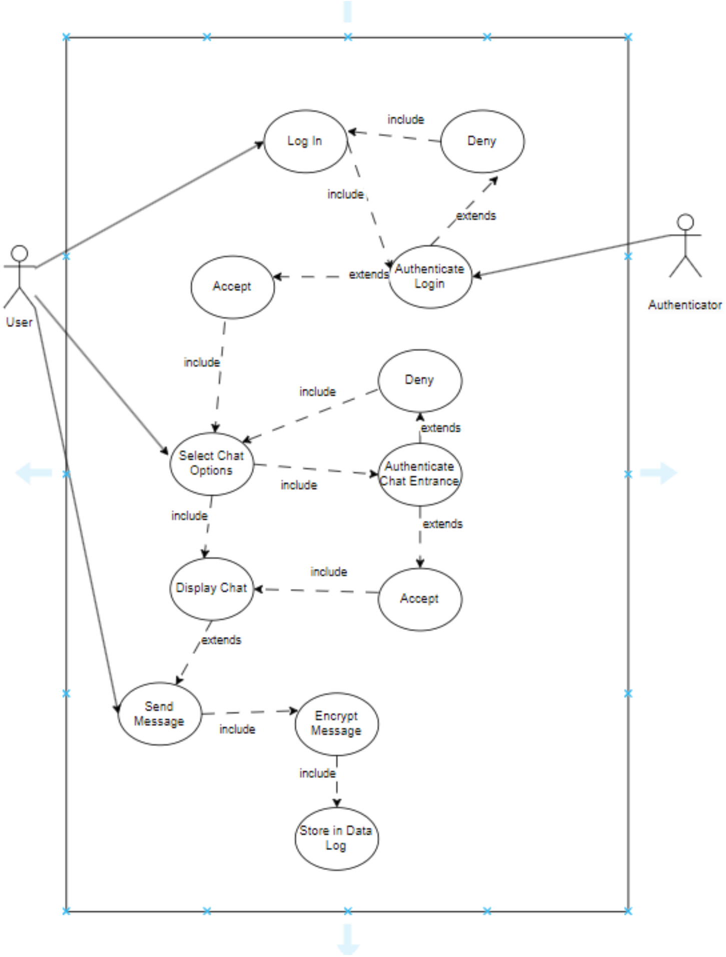

Figure 2: Use Case diagram for Sending a Message

Sending Message is the business event that the use case diagram represents. The use case diagram is to highlight how the employee would go about sending a secure message through the application. So, the user will login in, then select the chat room and send the message. There are also some secondary scenarios covered like if the login is denied. This is to cover the main possibilities within the sending business events.

4 Highlights of Functional Requirements

The business events we will consider:

BE1. Sending message
BE2. Group chat creation (Creative Feature)
BE3. Password recovery (Creative Feature)
BE4. Customize account (Creative Feature)
BE5. Logging in
BE6. Account register
BE7. Attaching file (Creative Feature)
BE8. Accessing chatlog
BE9. Updating key

The viewpoints we will consider:

VP1. Employee

VP2. Administrator
VP3. Customer support
VP4. Maintenance
VP5. Cybersecurity
VP6. Management

Interpretation: Specify any liberties you took in interpreting business events, if necessary.

BE1. Sending Message
Pre-condition: The employee must already have an account.

VP1. Employee
Main Success Scenario:

1. Employee opens the MacMessenger app on their phone.
2. System requires the employee to login and displays the login fields.
3. Employee logs into the app.
4. System authenticates the employee.
5. System shows the account, and chat options.
6. Employee chooses chat option.
7. System authenticates employee.
8. System displays chat box.
9. Employee enters message into chat box and clicks send.
10. System encrypts message.
11. System sends message to the recipient.
12. Recipient receives encrypted message.
13. System decrypts message with key.
14. System stores message in the chatlog.

Secondary Scenario:

4i. System fails to authenticate the employee in login.
4i.1 System prompts for re-entry of login details.
4i.2 Sending Message failed.
7i. System fails to authenticate the employee for the chat session.
7i.1 System logs the employee out.
7i.2 Sending Message failed.
9i. employee enters an invalid format or restricted content in the message.
9i.1 System displays an error message and requests correction.
9i.2 Sending Message failed.
10i. System fails to encrypt the message.
10i.1 System notifies the employee of the encryption error.
10i.2 Sending Message failed.
11i. System fails to send the message to the recipient.
11i.1 System notifies the employee of the sending error.
11i.2 Sending Message failed.
13i. System fails to decrypt the message for the recipient.
13i.1 Recipient receives a notification of decryption error.
13i.2 Message is stored encrypted, pending resolution.

VP2. Administrator
N/A

VP3. Customer support

4i. System displays an error message to try again or contact customer support for assistance.
  - 4i.1 System provides a direct link or contact details for the MacMessenger customer support team.

9i. Employee enters an invalid format or restricted content in the message.
  - 9i.1 System displays an error message and requests correction.
  - 9i.2 Sending Message failed.

10i. System fails to encrypt the message.
  - 10i.1 System notifies the employee of the encryption error.
  - 10i.2 Sending Message failed.

11i. System fails to send the message to the recipient.
  - 11i.1 System notifies the employee of the sending error.
  - 11i.2 Sending Message failed.

i. Maintenance
N/A

ii. Cybersecurity
N/A

iii. Management
N/A

Global Scenario:

Precondition: The employee must already have an account.

Main Success Scenario:

1. Employee opens the MacMessenger app on their Android device.
2. System requires the employee to login and

Insert Scenario Here

VP4. Business Event Name #2

VP1. Viewpoint Name #1
Insert Scenario Here

VP2. Viewpoint Name #2
Insert Scenario Here

Global Scenario:
Insert Scenario Here

BE7. Attaching file
Pre-condition: Employee has an account and has already logged in.

VP1. Employee
Main Success Scenario:

1. KDC generates a new communication key for employee.
2. Employee fetches a key from the KDC.
3. Employee enters the communication key to start the chat.
4. System receives the communication key and authenticates it and displays the chat interface.
5. Employee adds a file into the chat box and clicks ‘Send’.
6. System receives the sending request from employee and posts the file into chat interface.

7. System displays that the file was sent successfully.
8. System stores the chat log and the file info onto the server.
9. Employee quits the chat interface.

Secondary Scenario:

4i. Employee unable to authenticate with the generated key

4i.1 Employee unable to authenticate with the generated key.

4i.2 Sending file failed.

5i. Employee added a file exceeding the maximum size limit.

5i.1 Employee added a file exceeding the maximum size limit.

5i.2 Attaching file failed.

7i. Employee saw the file sent was failed

7i.1 Employee saw the file was failed to send over

7i.2 Sending file failed

VP2. Administrator

N/A

VP3. Customer support

4i. System prompts the employee to re-try or connect with Customer Support

7i. The system gives the employee two choices, either re-send or suggests connecting with Customer Support.

8i. System receives the error while storing the chat log, and auto-creates a bug ticket for Customer Support. (Out of scope).

VP4. Maintenance

N/A

VP5. Cybersecurity

N/A

VP6. Management

N/A

Global Scenario:

Precondition: Employee has an account and has already logged in.

Main Success Scenario:

1. KDC generates a new communication key for employee.
2. Employee fetches a key from the KDC.
3. Employee enters the communication key to start the chat.
4. System receives the communication key and authenticates it and displays the chat interface.
5. Employee adds a file into the chat box and clicks ‘Send’.
6. System receives the sending request from employee and posts the file into chat interface.
7. System displays that the file was sent successfully.
8. System stores the chat log and the file info onto the server.
9. Employee quits the chat interface.

5 Non-Functional Requirements

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF-A1. *The application must fill to fit the screen of all Android devices.*
Rationale: The system will be used on a variety of different Android devices.

LF-A2. *The application must use a minimalistic design style.*
Rationale: A minimalistic design will allow a new user to more easily grasp the main functions of the application.

LF-A3. *The application must not utilize overly bright or saturated colours.*
Rationale: Bright and oversaturated colours may be difficult to read, or make it difficult to look at the application for extended periods of time.

5.1.2 Style Requirements

LF-S1. *The application must provide strong contrast between lettering and background.*
Rationale: Strong contrast will increase readability.

LF-S2. *The application must utilize the same colour palette across the user interface.*
Rationale: A standardized colour palette will provide a sense of unity across the different function of the application.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH-EOU1. *Buttons and similar interactive components throughout the application must be clearly labelled, unless their usage is intuitive.*
Rationale: The main purpose of the application is for communication; thus it should not require any more knowledge than having an in-person conversation would. Many interactive components are labelled similarly throughout the variety of technologies common in the modern world, and so these can be assumed to be within the users existing knowledge base.

5.2.2 Personalization and Internationalization Requirements

UH-PI1. *International keyboards must be supported to allow messages with all global alphabets, accents, and characters.*
Rationale: A global company using this application may have employees wishing to communicate in many different languages.

5.2.3 Learning Requirements

UH-L1. *The employee must be able to understand how to use the application within the first 10 minutes of signing in.*
Rationale: Some employees may have minimal technical knowledge and prefer a small learning curve to make usage of the app simple.

5.2.4 Understandability and Politeness Requirements

N/A

5.2.5 Accessibility Requirements

UH-A1. *Text-to-speech must be provided throughout the app.*
Rationale: Accessibility for users with limited/no vision.

UH-A2. *The app must integrate with the native OS's accessibility services.*
Rationale: Decrease learning curve of the application by providing services the user is already familiar with.

UH-A3. *The user must be able to increase or decrease the size of text displayed [2].*
Rationale: Users may have different levels of visual impairments.

UH-A4. *The application must provide support for speech-to-text [2].*
Rationale: Accessibility for users with visual or fine-motor impairments, or similar.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR-SL1. *Messages must be updated in a chat within 10ms, provided each user has a strong internet connection.*
Rationale: The expectation of users of most messaging apps is that messages are received as soon as they are sent, so the message sending process must appear to be instantaneous (studies show that 100ms is that maximum value perceived by humans to be instantaneous [7]).

PR-SL2. *Application start-up time must be less than 5 seconds.*
Rationale: 10 seconds is the upper limit for how long a user will keep their attention on a loading page [7].

5.3.2 Safety-Critical Requirements

PR-SC1. *The application must include a 'Report User' function to notify the company of inappropriate behaviour.*
Rationale: Any user who feels uncomfortable or threatened by another user should make the inappropriate behaviour known to their employee, to be dealt with accordingly.

5.3.3 Precision or Accuracy Requirements

PR-PA1. *Time stamps on sent/received messages must be accurate to the minute.*
Rationale: Showing accuracy of time to the minute is a standard across all messaging apps.

5.3.4 Reliability and Availability Requirements

PR-RA1. *The application must be available to users 24/7.*
Rationale: All messaging between employees should occur within a secure environment; continuous service of the application will eliminate any need for the user to use some other, less secure messaging application.

PR-RA2. *The application must perform with a success rate of 95% [3].*
Rationale: Users will be less likely to trust the application if they often experience failures and errors.

PR-RA3. *The application must be able to recover from errors without experiencing data loss or service failures [3].*
Rationale: In the case that an error occurs the system should be able to continue from its last valid state in order to maintain the trust of its users and prevent any catastrophic data loss.

5.3.5 Robustness or Fault-Tolerance Requirements

PR-RFT1. *The app must display accurate error messages in the case of server-side or client-side errors.*
Rationale: The user should know why a failure in the application occurred so they can troubleshoot client-side errors or report server-side errors.

5.3.6 Capacity Requirements

PR-C1. *The application must perform the same given the minimum number of employees and 5 times the total number of employees at the company.*
Rationale: The performance of the application should not change based on the total number of users, and so the system must be prepared to handle the event where every employee is simultaneously using the application. Ensuring performance up to 5 times the expected traffic will also provide protection against DoS attacks.

5.3.7 Scalability or Extensibility Requirements

PR-SE1. *The application must be built on proper design principles, SOLID and follow the proper design patterns to ensure scalability of the app.*
Rationale: The SOLID principle is designed to ensure that the application can be scalable [4].

5.3.8 Longevity Requirements

N/A

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE-EPE1. *The application must be readable in an office environment.*
Rationale: Main usage is expected to occur during business hours in the office.

5.4.2 Requirements for Interfacing with Adjacent Systems

N/A

5.4.3 Productization Requirements

N/A

5.4.4 Release Requirements

OE-R1. *The application must be compatible with Android 11.0 or above*
Rationale: Android 10.0 and lower no longer receive security patches, increasing the risk that a device running these versions may be compromised [8].

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS-M1. *The system must have release bug fix updates at least once every 2 months.*
Rationale: To maintain high standards of quality, the application must regularly undergo review and publish bug fixes as bugs are discovered.

5.5.2 Supportability Requirements

MS-S1. *Messages sent using the application must support text input and attachments of any file extension.*
Rationale: This is the team's creative function. It will allow for an increased level of communication using the application and reduce the need for employees to communicate via some less secure third party.

MS-S2. *The application must include a function to allow users to report bugs.*
Rationale: Users interacting with the application should have a way to alert the developers of the application when they find bugs and unexpected/unwanted behaviours of the system.

5.5.3 Adaptability Requirements

N/A

5.6 Security Requirements

5.6.1 Access Requirements

SR-AC1. *Employees must have full access to their chat history unless they have deleted it.*
Rationale: Each user should have final say in the usage of their personal data.

SR-AC2. *The application must have an internet connection to send or receive messages.*
Rationale: An internet connection is required for messages to be sent and received between employees.

SR-AC3. *Employees cannot access chat logs of any other employee.*
Rationale: Ensure security and privacy of each individual employee.

SR-AC4. *Employees must provide authentication to gain access to their account.*
Rationale: Ensure only authorized individuals can gain access to a given account.

5.6.2 Integrity Requirements

SR-INT1. *Nonrepudiation must be ensured over each secure messaging channel.*
Rationale: Employees must not be able to deny that they sent a given message.

SR-INT2. *Identity of sender and receiver must be verified over each secure messaging channel.*
Rationale: Ensure that information being received has come from a known and trusted source.

SR-INT3. *The physical server containing chat logs must be stored in a secure environment.*
Rationale: Reduce probability of attacks such as side-channel attacks, i.e., using hardware to capture transmission waves.

5.6.3 Privacy Requirements

SR-P1. *Stored chat logs must be encrypted using a symmetric-key cryptosystem.*
Rationale: Ensure that chat logs of employees are kept private from unauthorized parties and are secure against known cyberattacks.

SR-P2. *Message channels must be encrypted using a symmetric-key cryptosystem.*
Rationale: Ensure that chats of employees are kept private from unauthorized parties and are secure against known cyberattacks.

SR-P3. *The application must implement a Key Distribution Centre which generates all keys to be used in the secure communication session.*
Rationale: Use of a Key Distribution Centre will ensure only authorized employees can access and use the application.

5.6.4 Audit Requirements

SR-AU1. *All security measures utilized in the application must adhere to NIST standards.*
Rationale: The National Institute of Standards and Technology (NIST) has developed a comprehensive set of guidelines according to American statutes and laws [5]. NIST standards are upheld worldwide despite being an American institute.

SR-AU2. *The application must pass penetration testing [1].*
Rationale: A penetration test will simulate a cyberattack and ensure the security of the application.

5.6.5 Immunity Requirements

N/A

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

CP-C1. *The application shall not use any offensive imagery.*
Rationale: Employees should feel comfortable and safe when using the application to ensure satisfaction with the product.

5.7.2 Political Requirements

N/A

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR-COMP1. *The application will follow all regional laws relating to the collection, usage, and disclosure of the personal information of its users.*
Rationale: All public institutions in Ontario are required by law to protect the information of their users and follow strict rules in the case any personal information must be disclosed to an outside party [6].

5.8.2 Standards Requirements

LR-STD1. *The app should store any sensitive company chatlogs into an internal database.*
Rationale: Standard app quality of application with consideration to the privacy and security of sensitive information [9]

A Division of Labour

Emily Perica Signature: Emily Perica	1.1. Purpose 1.2. Scope 2.3. User Characteristics 5. Nonfunctional Requirements
Harman Bassi Signature:	1.2. Scope 2.1. Product Perspective 5. Nonfunctional Requirements
Kyen So Signature:	1.2. Scope 2.2. Product Functions 5. Nonfunctional Requirements
Kelly Deng Signature: Yunqi Deng	1.2. Scope 2.5. Assumptions and Dependencies 4. Functional Requirements
Swesan Pathmanathan Signature:	1.2. Scope 2.6. Apportioning Requirements 4. Functional Requirements