# Deliverable #1 Template : Software Requirement Specification (SRS)

SE 3A04: Software Design II – Large System Design

**Tutorial Number:** T03
**Group Number:** G06
**Group Members:**

- Virochaan Ravichandran Gowri
- Alex Yoon
- Noah Goldschmied
- Krish Dogra
- Leo Vugert

## 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 <u>underline</u>
- 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

- Provide an overview of the document/SRS.

## 1.1 Purpose

This Software Requirement Specification has been created to specify the requirements needed to develop a secure communication app (VanklComm) for our organization. This SRS will ensure to cover functional requirements specifying how the app will perform the secure communication, including viewpoints from stakeholders and common business events and use cases and non-functional requirements outlining specifications of the system.

## 1.2 Scope

- Identify the software product(s) to be produced, and name each (e.g., Host DBMS, Report Generator, etc.)
- Explain what the software product(s) will do (and, if necessary, also state what they will not do).
- Describe the application of the software being specified, including relevant benefits, objectives, and goals.

## 1.3 Definitions, Acronyms, and Abbreviations

- Provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS.
- This should be in alphabetical order.

## 1.4 References

- Provide a complete list of all documents referenced elsewhere in the SRS.
- Identify each document by title, report number (if applicable), date, and publishing organization.
- Specify the sources from which the references can be obtained.
- Order this list in some sensible manner (alphabetical by author, or something else that makes more sense).

## 1.5 Overview

- Describe what the remainder of the document/SRS contains.
  (e.g. "Section 2 discusses...Section 3...")

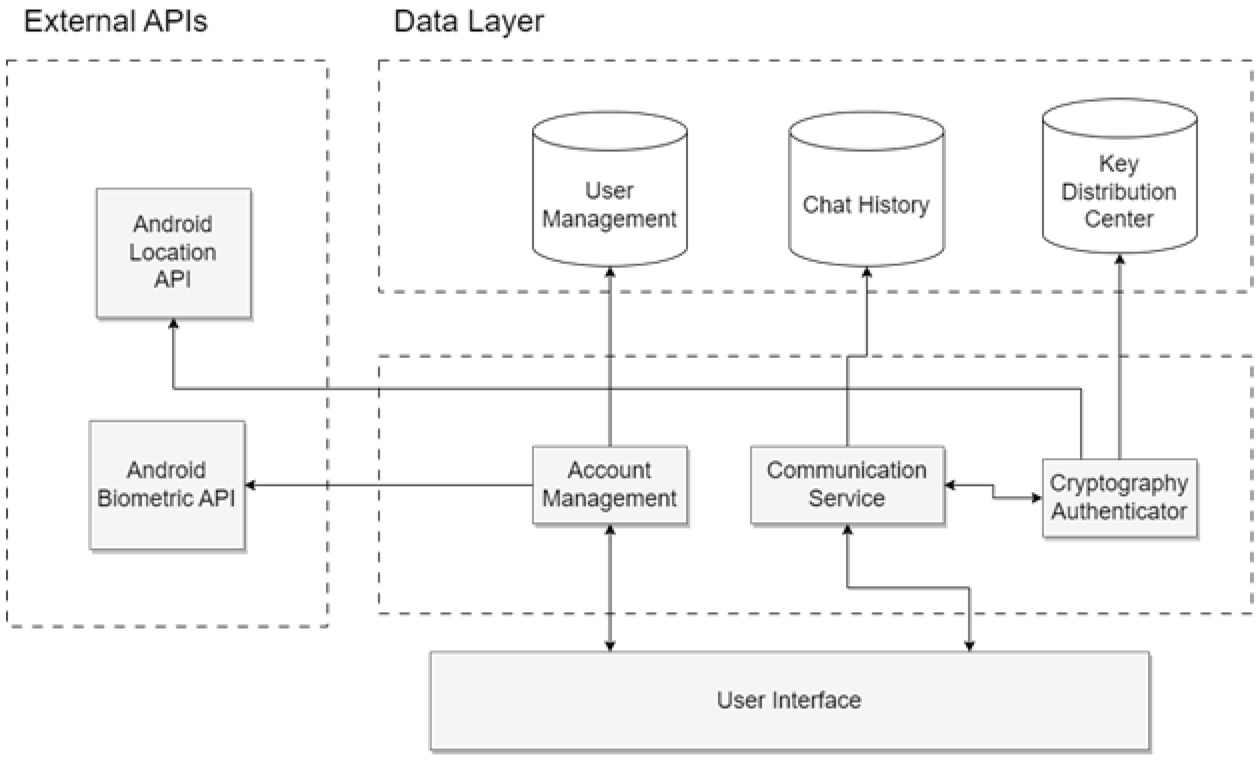
# 2 Overall Product Description

- This section should describe the general factors that affect the product and its requirements.
- It does not state specific requirements.
- It provides a *background* for those requirements and makes them easier to understand.

## 2.1 Product Perspective

VanklComm is a secure chat Application developed for Android and to be used solely on company devices. Other products similar to this app include other secure communication apps such as Signal and WhatsApp which allow for secure communication through End-to-End encryption of messages. VanklComm will have a focus on only allowing communication between registered agents (employees) within the company, allowing for a secure communication channel. The app will allow users to communicate securely through text and with a file sharing feature to securely transfer files between users. The user can store contacts within a contact book for fast access to other authorized users and past chat logs. The app will utilize geolocation software to ensure the app is only used in authorized locations and will utilize biometric identification for enhanced security through 2-Factor authentication.

The app will interface with Android APIs for biometric identification and geolocation. The biometric identification API will be used to allow entry into the application itself. The geolocation API will ensure usage of the app solely within the authorized locations allowed by the organization and will prompt users to reenter zones where they are allowed to use the app.



## 2.2 Product Functions

- Provide a *summary* of the major functions that the software will perform.
  - **Example**: An SRS for an accounting program may use this part to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires.
- Functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time
- Present the functions in a list format - each item should be one function, with a brief description of it
- Textual or graphical methods can be used to show the different functions and their relationships

  - Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables

### 2.3 User Characteristics

- Describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise
- Since there will be many users, you may wish to divide into different user types or personas

### 2.4 Constraints

- Provide a general description of any constraints that will limit the developer's options

### 2.5 Assumptions and Dependencies

- List any assumptions you made in interpreting what the software being developed is aiming to achieve
- List any other assumptions you made that, if it fails to hold, could require you to change the requirements
  - **Example**: An assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

### 2.6 Apportioning of Requirements

- Identify requirements that may be delayed until future versions of the system

## 3 Use Case Diagram

- Provide the use case diagram for the system being developed.
- You do not need to provide the textual description of any of the use cases here (these will be specified under "Highlights of Functional Requirements").

## 4 Highlights of Functional Requirements

- Specify all use cases (or other scenarios triggered by other events), organized by Business Event.
- For each Business Event, show the scenario from every Viewpoint. You should have the same set of Viewpoints across all Business Events. If a Viewpoint doesn't participate, write N/A so we know you considered it still. You can choose how to present this - keep in mind it should be easy to follow.
- At the end, combine them all into a Global Scenario.
- Your focus should be on what the system needs to do, not how to do it. Specify it in enough detail that it clearly specifies what needs to be accomplished, but not so detailed that you start programming or making design decisions.
- Keep the length of each use case (Global Scenario) manageable. If it's getting too long, split into sub-cases.
- You are *not* specifying a complete and consistent set of functional requirements here. (i.e. you are providing them in the form of use cases/global scenarios, not a refined list). For the purpose of this project, you do not need to reduce them to a list; the global scenarios format is all you need.

- Red text below is just to highlight where you need to insert a scenario - don't actually write it all in red.

**Main Business Events:** List out all the main business events you are presenting. If you sub-divided into smaller ones, you don't need to include the smaller ones in this list.

**Viewpoints:** List out all the viewpoints you will be considering.

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

**BE1.** Business Event Name #1

  **VP1.** Viewpoint Name #1
  Insert Scenario Here

  **VP2.** Viewpoint Name #2
  Insert Scenario Here

  **Global Scenario:**
  Insert Scenario Here

**BE2.** Business Event Name #2

  **VP1.** Viewpoint Name #1
  Insert Scenario Here

  **VP2.** Viewpoint Name #2
  Insert Scenario Here

  **Global Scenario:**
  Insert Scenario Here

## 5 Non-Functional Requirements

- For each non-functional requirement, provide a justification/rationale for it.
  **Example:**
  SC1. *The device should not explode in a customer's pocket.*
  **Rationale** Other companies have had issues with the batteries they used in their phones randomly exploding [insert citation]. This causes a safety issue, as the phone is often carried in a person's hand or pocket.
- If you need to make a guess because you couldn't really talk to stakeholders, you can say "We imagined stakeholders would want...because..."
- Each requirement should have a unique label/number for it.
- In the list below, if a particular section doesn't apply, just write N/A so we know you considered it.

### 5.1 Look and Feel Requirements

#### 5.1.1 Appearance Requirements

LF-A1.

#### 5.1.2 Style Requirements

LF-S1.

## 5.2 Usability and Humanity Requirements

### 5.2.1 Ease of Use Requirements

UH-EOU1. Users should be allowed to report errors and areas of improvement through the organization's in-app feedback tool.
**Rationale:** The system will provide a way of reporting bugs and errors and provide general feedback or areas of improvement which the organization can then act upon. It will be done through the organization to provide maximum transparency for all stakeholders involved and keep them in the loop about future developments.

UH-EOU2. On first download, users should be given a short tutorial on the correct usage of the app.
**Rationale:** The tutorial will allow every employee to have a base understanding of the usage of the app and will protect the organization to make sure users properly authenticate before communicating. The tutorial will also indicate certain accessibility features for users that require it to ensure effective usage of the app.

UH-EOU3. The system should provide users with a help section including FAQs and recapping the Tutorial.
**Rationale:** While using the app, if a user should face difficulty or forget the location or usage of certain features, they can use the help section to see if their question or problem can be resolved. It will also recap the tutorial including basic usage and accessibility features of the app.

### 5.2.2 Personalization and Internationalization Requirements

UH-PI1. The system should allow users to change the language to any of the provided languages based on their preferences.
**Rationale:** Users must be able to utilize the app effectively, so it is crucial they understand the different headings and labels within the app. This is also represented within the tutorial as it will be translated into the provided languages. We will only provide languages that the organization uses in day-to-day work as there is an expectation that the employees will understand these languages.

UH-PI2. The user should be allowed to change the font based on one's own preference.
**Rationale:** Allows the user to customize the font which will apply to all facets of the app for an overall better experience. Some people like different fonts for readability and giving them this option can only raise the overall user experience of using the app.

UH-PI3. The user should be allowed to change the background of the chat and home screens.
**Rationale:** Provides users with greater customization to make the app more personal to them. Increasing the options available to the user can improve the user experience and make the app more personal to them.

### 5.2.3 Learning Requirements

UH-L1. Users should be able to set up and use the app within 10 minutes of downloading.
**Rationale:** The process for registering a user's account and then using the app after watching the tutorial should be able to be completed within 10 minutes of the first usage of the app. Keeping the whole set up process short increases user retention using the app and a short, yet comprehensive tutorial ensures that information on usage is more easily understood.

### 5.2.4 Understandability and Politeness Requirements

UH-UP1. Symbols for messaging and adding users should be universally recognized and the standard for all mobile apps.
**Rationale:** To properly utilize the app the user must be able to understand all iconograph. By matching our icons with the ones that are used in other similar apps we can ensure that the

average user will be able to seamlessly make the transition to using our app without confusion and contribute to a lower learning curve for the app.

#### 5.2.5 Accessibility Requirements

UH-A1. The user should be allowed to change font size for better visibility.
**Rationale:** This feature should be implemented to ensure ease of use for persons who are more visually impaired. This font size should be global for the app to ensure that users with visual difficulties can always understand what is on the screen.

UH-A2. Implement a colorblind mode which changes the color palette to use more accessible colours.
**Rationale:** This feature should be implemented to ensure ease of use for persons who are colour blind and have difficulty differentiating between colours. The color-blind mode should utilize the Deuteranomaly friendly colour palette so the user can differentiate the colours displayed. [1]

UH-A3. A high contrast/dark color mode should be implemented for better visibility and for usage during nighttime.
**Rationale:** This feature should be implemented to ensure ease of use for people who are more visually impaired and require better distinctions between certain colours. Can be a tool for people who want a dark mode as well and provide better distinction between icons, symbols, and lettering when being used during nighttime or in low lighting. Including a dark mode will greatly increase user experience and provide further customization for the user. [2]

UH-A4. The system should implement an on-screen reader for the visually impaired.
**Rationale:** An on-screen reader is essential for visually impaired people as it allows them to utilize the app more effectively. The on-screen reader should allow for easier navigation through the app as well as provide text-to-speech conversion of messages.

UH-A5. The system should implement a dictation feature which converts voice to messages.
**Rationale:** Provides the user with more methods of performing communication and sending messages. Can help persons with limited dexterity in their fingers by providing an alternative method of communication.

## 5.3 Performance Requirements

### 5.3.1 Speed and Latency Requirements

PR-SL1. The KDC should have less than a 2-second response time when returning keys.
**Rationale:** The KDC should be able to return keys within 2 seconds of receiving the request from an agent. This time is more than enough to send the key as it does not consider the generation/rotation of keys. This will also ensure agents do not spend a long time waiting to enter a chat due to the KDC having large delays providing a seamless experience.

PR-SL2. The time taken to encrypt and decrypt using key should be less than 2s.
**Rationale:** After receiving the key, the encryption and decryption of data should not take longer than 2 seconds since the payload of the data is quite small ranging in the kilobytes of size. We can see that through experimental testing of the algorithms 2 seconds should be more enough to encrypt and decrypt payloads of this size and this represents the worst-case scenario. [3]

PR-SL3. The message should be received by the receiving communicating agent in less than 10s after being sent.
**Rationale:** We want to ensure that global standards are met for delivering and receiving messages, but it also needs to be taken into account that these messages are being encrypted and decrypted, which adds extra time. This is why we believe 7.5 seconds is a good realistic timespan in which it doesn't introduce that large of a delay between send and receive while maintaining a high level of security.

PR-SL4. Time taken to access chat logs from the database server should be less than 10 seconds.
**Rationale:** Should an administrator require access to a certain chat log this should be done in a prompt manner and querying data should be fast and efficient. This can aid in Incident Response should there be a security breach, or some other concern related to an employee's behavior aiding in a fast investigation. 10 seconds provides both a realistic timeframe for querying through large amounts of data while still being prompt in the response time. [4]

### 5.3.2 Safety-Critical Requirements

PR-SC1. N/A

### 5.3.3 Precision or Accuracy Requirements

PR-PA1. The app must be only used in accepted locations with an error of location of maximum of 20 meters.
**Rationale:** Most Android Devices utilize some sort of GPS software to ensure accurate location of the device. The average accuracy of GPS enabled devices is around 5m but this can worsen due to location and spotty cell/Wi-Fi service. [5] By expanding the range to 20 meters we account for potential inaccuracies while still maintaining security.

PR-PA2. Fingerprint identification should have 99% accuracy in identifying users.
**Rationale:** The biometric identification feature should have a 99% accuracy to ensure only authorized users are able to gain entry to the access. This high level of accuracy ensures security by not misidentifying users while still attempting to minimize the total time taken to enter the app by fast identification. It also enables us to use 2-Factor Authentication which is a much more secure method of identification. [6]

### 5.3.4 Reliability and Availability Requirements

PR-RA1. The app should have a scheduled downtime of a maximum of 1 hour every 6 months to ensure proper functionality.
**Rationale:** Provides enough time for required maintenance on the app as necessary while maintaining a high level of availability. This higher time of downtime is justified due to the clandestine nature of the app and security being of utmost priority. Since this is a company-wide app it is easy to inform individuals using the app of the scheduled downtime limiting the risk and inconvenience.

PR-RA2. The Database server should maintain a backup of the data in cold data storage.
**Rationale:** Ensuring that we keep a backup of the data but stored in cold data storage to save resources as hopefully we will not be needed to access the backup. The accessing of the data in the backup may take a long time if necessary and is primarily used just to ensure data security.

PR-RA3. The app should strive for 99.999% availability during normal operation.
**Rationale:** When the app is being used during standard situations, we should aim for it to have 99.999% availability as it will ensure that all members of the company remain connected. 99.999% or 5 nines uptime is industry standard and the highest realistic uptime necessary for the app to maintain continuous availability. [7]

### 5.3.5 Robustness or Fault-Tolerance Requirements

PR-RFT1. The app should function as long as there is cell service in the area.
**Rationale:** The app should be able to work on mobile data and is not limited to only Wi-Fi. This ensures that communication will still work when Wi-Fi networks are down, ensuring employees can remain connected.

#### 5.3.6 Capacity Requirements

PR-C1. The KDC should be able to handle 20% of the userbase's requests at the same time.
**Rationale:** Ensures that the KDC meets capacity while also considering scalability in terms of requests. We could say a fixed number such as 20 but if the company has 10000 people with the app there is a high likelihood that more than 20 people request keys at the same time during peak usage of the app. By maintaining this requirement as a percentage, we can maintain both scalability and have good capacity throughput.

PR-C2. The Initial Database Server should have a minimum of 1TB storage before scaling.
**Rationale:** We want to have a large base server size to ensure that we do not need to start scaling unnecessarily expending valuable resources on maintaining the database. 1TB of storage can store multiple years worth of data without

#### 5.3.7 Scalability or Extensibility Requirements

PR-SE1. The app should maintain at least 5% of the database empty to ensure space for future chat logs.
**Rationale:** Allow for scalability to store more chat logs in the future. Ensure that we never reach a situation where there are chats coming in but not being stored due to the full capacity in the database server. 5% also provides a good buffer space if there is a sudden influx of data providing time for the database to scale to the new capacity.

#### 5.3.8 Longevity Requirements

PR-L1. The app in its first iteration must be able to last a minimum of 2 years with avenues for extension or improvement.
**Rationale:** Current iteration should be usable by employees and based on feedback and bug reporting we can provide future updates or overhauls. This process could take a long time so it is important the current iteration lasts at least 2 years for feedback to be collected and updates developed.

## 5.4 Operational and Environmental Requirements

### 5.4.1 Expected Physical Environment

OE-EPE1.

### 5.4.2 Requirements for Interfacing with Adjacent Systems

OE-IA1.

### 5.4.3 Productization Requirements

OE-P1.

### 5.4.4 Release Requirements

OE-R1.

## 5.5 Maintainability and Support Requirements

### 5.5.1 Maintenance Requirements

MS-M1.

### 5.5.2 Supportability Requirements

MS-S1.

#### 5.5.3 Adaptability Requirements

MS-A1.

### 5.6 Security Requirements

#### 5.6.1 Access Requirements

SR-AC1.

#### 5.6.2 Integrity Requirements

SR-INT1.

#### 5.6.3 Privacy Requirements

SR-P1.

#### 5.6.4 Audit Requirements

SR-AU1.

#### 5.6.5 Immunity Requirements

SR-IM1.

### 5.7 Cultural and Political Requirements

#### 5.7.1 Cultural Requirements

CP-C1.

#### 5.7.2 Political Requirements

CP-P1.

### 5.8 Legal Requirements

#### 5.8.1 Compliance Requirements

LR-COMP1.

#### 5.8.2 Standards Requirements

LR-STD1.

## A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.

**Virochaan Ravichandran Gowri:**

- 1.1 - Purpose
- 2.1 - Product Perspective
- 2.1 - Block Diagram showcasing product connections
- 5.2, 5.3 - Usability and Humanity + Performance Non Functional Requirements.
- Reviewed the whole document as a group