| Design DocumentFor  Chatty Messenger App  **All Rights Reserved**  This document is a property of Trianz. No part of this document may be copied or reproduced in any form or by any means, electronic, mechanical or otherwise without the prior written permission of Trianz. | |
| --- | --- |
| **Owner: Nandini /Vinay/ Aman** | **RID:** QMS/5101/0293 |
| **Email: uccnandinisinha2gmail.com** | **Version: 0.1** |
| **Date: 07-Sept-23** | **Total Pages: 6** |
| **Distribution List:** All members of group | |
|  | |

**Table of Contents**

1. Project Overview 3

2. Scope 3

3. References 3

4. Definitions/Acronyms 3

5. Environment 3

6. System Architecture 4

7. Design Strategy 4

8. High Level Design 4

9. Detailed Design 4

9.1 Detailed Design of <Component/Module 1> 4

9.2 Detailed Design of <Component/Module n> 5

10. Design Quality Objectives 6

11. Traceability Matrix 6

12. Revision History 6

# Project Overview

# 1. Introduction

CHATY is a social-networking tool that leverages technology advancement thereby allowing its users to communicate and share media. It offers a wonderful one-stop shopping experience for keeping in touch with people you know. It can be used for messaging, placing voice messages, making voice and video calls, share updates and photos, share locations, enhance local socializing in pidgin English, play games and make monetary financial transactions.**2.Objective**

CHATTY enormous social power will be let loosed as it will translate to become one of the main ways people will have to communicate in Nigeria and sub-Sahara Africa. Even when doing business, people will prefer CHATTY to email. With inbuilt plug ins that allow individuals to load pictures and videos and their contacts can make comments about them.

# 3 Scope

One of the primary uses of CHATTY is messaging. Just like other social apps, you have a list of conversations that you’re engaged in. This feature is pivotal as you can add people in a variety of ways aside from the conventional way of details collection. When fully operational, you will be amazed how individuals will have to scan their phones during details collection.

**4.Definitions/Abbreviations/Acronyms**

| Role | Responsibility |
| --- | --- |
| SQA | Software Quality Assurance |
| SOW | Statement Of Work |
| PM | Project Manager |
| PL | Project Leader [who assumes the role of senior design engineer] |
| CCB | Configuration Control Board |
| SR | Service Request |
| PT | Porting Team |

1. **5. Reference**s[[1]](#footnote-0)

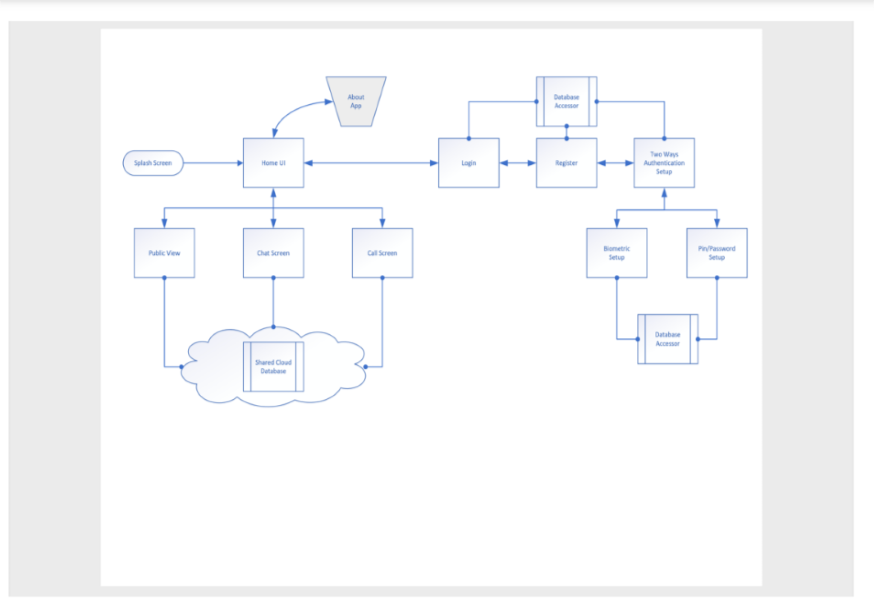
| **S.No.** | **Title** | **Reference Number** | **Owner/ Author/Supplied By** | **Location/Folder where available** |
| --- | --- | --- | --- | --- |
| 1. | Design Process | 0199/020 | HMT | OneDrive |
| 2. | Design Template | 1009/34 | HMT | One drive |

# 6. Environment

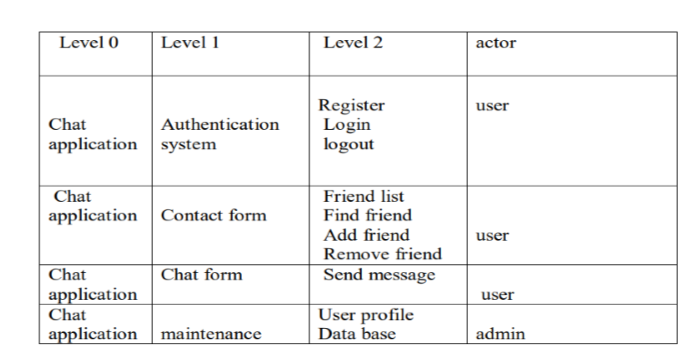
1. Support one-to-one chats
2. Support offline sending message
3. Support to send messages to other users even the user is offline
4. support group chats
5. Chatty group with up to **30 participants**
6. Support video chat
7. Support group video chat
8. Allow voice messages
9. Support image, video, and file-sharing capabilities
10. Support encrypted message
11. Video cannot exceed 16 MB or 90 seconds to 3 minutes
12. Indicate read/receipt of messages
13. Last seen time of users (depends on a few scenarios)
14. Sent + Delivered + Read receipts ticks
15. File size can be shared is limited to 100MB
16. Support to play YouTube, audio, and video on the display screen
17. Supports video formats — MP4, 3GP, MKV, AVI, and MOV
18. Notification will be shown once your file, video, audio, and image exceed the size limitation
19. Support to share the file, video, audio, and image to other applications which includes
20. Support reply to a particular message in a group chat
21. Allow us to forward any messages
22. Allow us to copy the message
23. Allow us to save the message
24. Allow us to archive the message
25. Allow us to reply privately to the particular message
26. Allow us to send and view videos and images one time and delete them once opened

# 7. System Architecture

**Flow Chart**

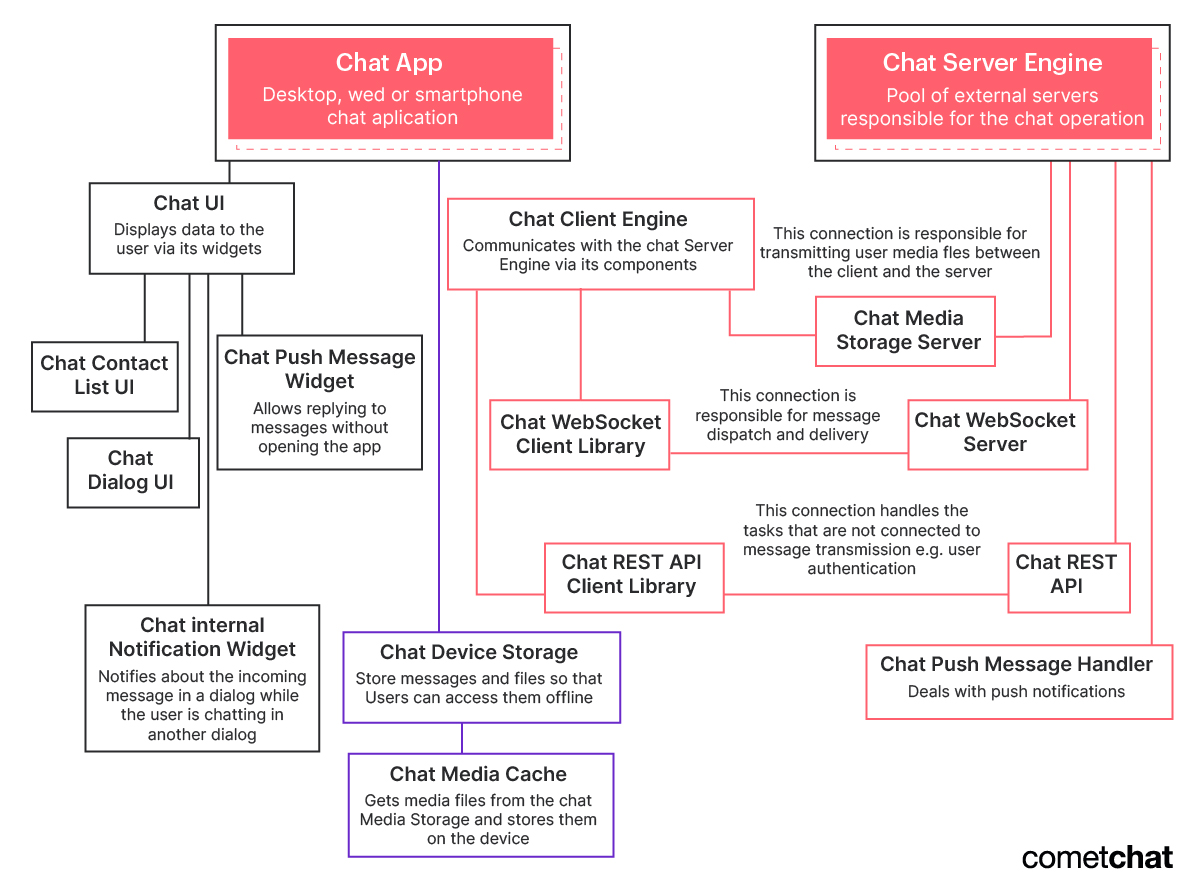


**Workflow**



**8. Design Strategy**

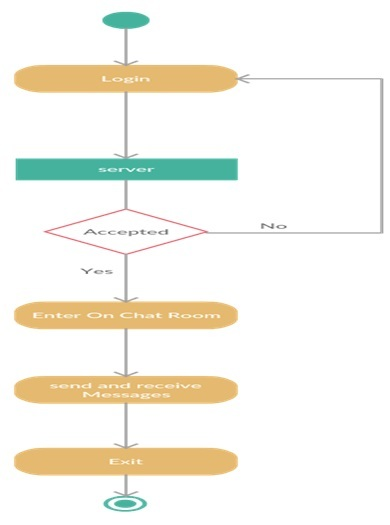
**Architectural Design**



1.When a user just clicked a Send button the message is sent from a client device to web servers (clock icon).

1. When the message came to a server it sends a notification back to the client. At that moment the icon changed to a single check.
2. The web server looks for a connection that is established between it and a recipient and sends a message. The recipient client device sends back a notification to a server and after that, the server can send a notification to the sender that the message was delivered, and the icon changed to double-check.
3. Eventually, the recipient opens an application and reads the message. His device sends one more notification to the server and the server communicates to the sender that the message was read. The icon changed to blue colored double-check.

**Activity Diagram**



# 9. High Level Design

The messenger app has to handle more than 100 billion messages daily or roughly a million per second. The messenger app follows the distributed design to take such a load and the important part of it is Load Balancer (LB). Load Balancer applies a particular strategy to equally distribute effort between different servers.

Messenger aims to quickly deliver messages between users and servers. Web sockets are used to ensure it. A web socket is a protocol providing duplex communication over a TCP connection.

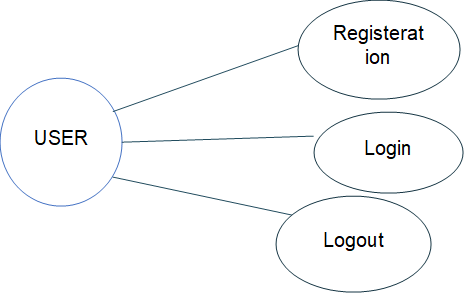
Keeping a web socket connection is cheaper than an HTTP connection. The client and server subscribe to each other and wait for new messages. That’s why all arrows on the scheme are bi-directional.

Different users might be connected to different web servers, so we need to have Web Socket Manager. This service is responsible for providing information about connections and routing messages to a proper web server.

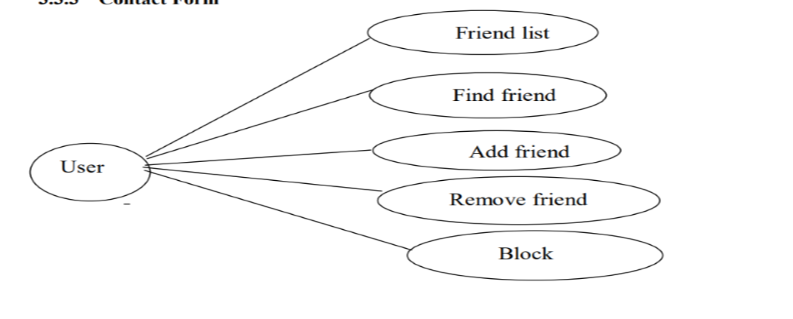
If the recipient is offline the message should be stored. The database should be optimized for frequent write and delete operations. Delete operation performed after the client is online and all messages are delivered to him. NoSQL database will be a choice for this case, e.g. some messengers use HBase and Cassandra to solve the task.

Use Case Diagrams

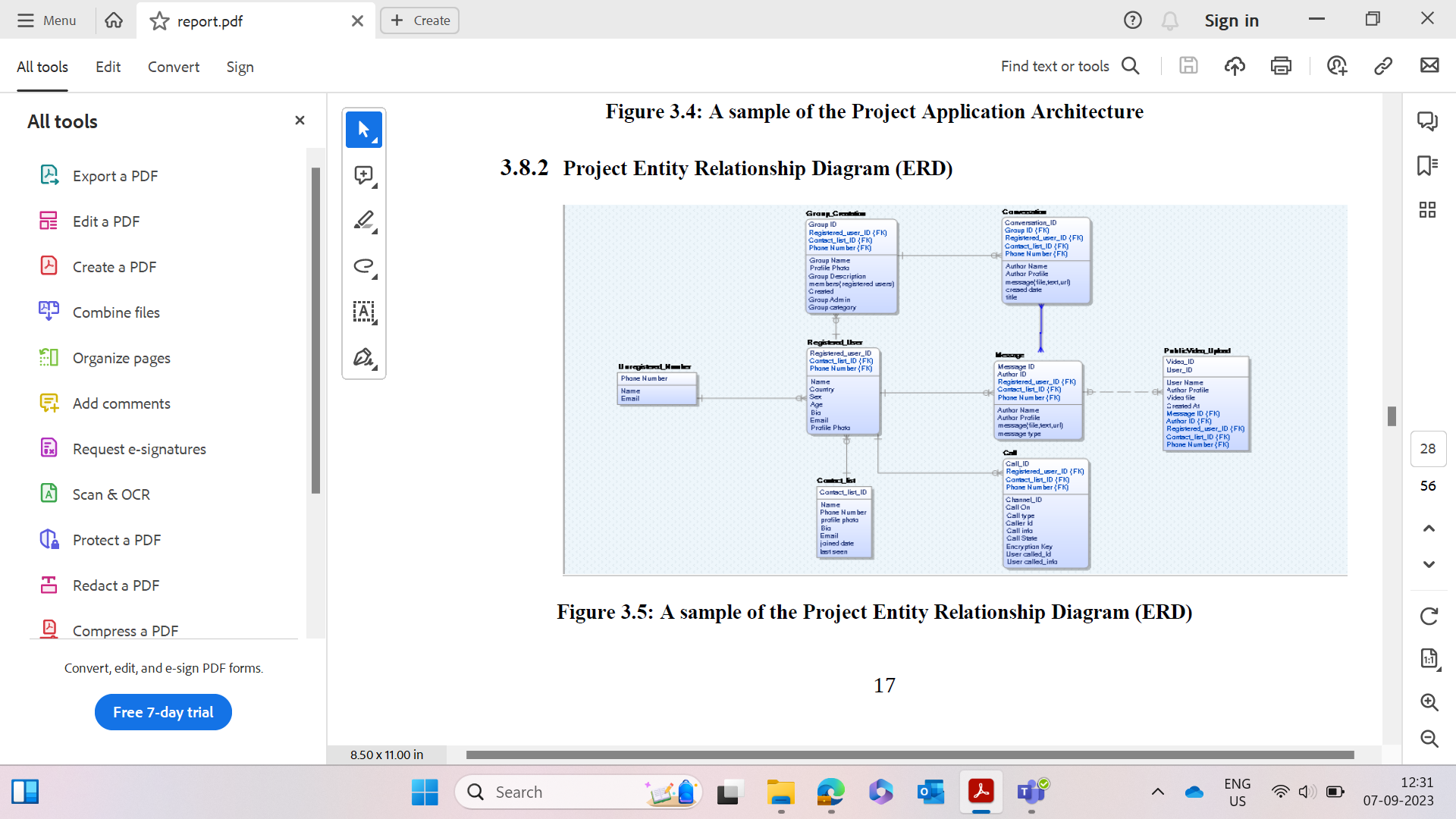
1. Main diagram



# b) Contact form Use case diagram

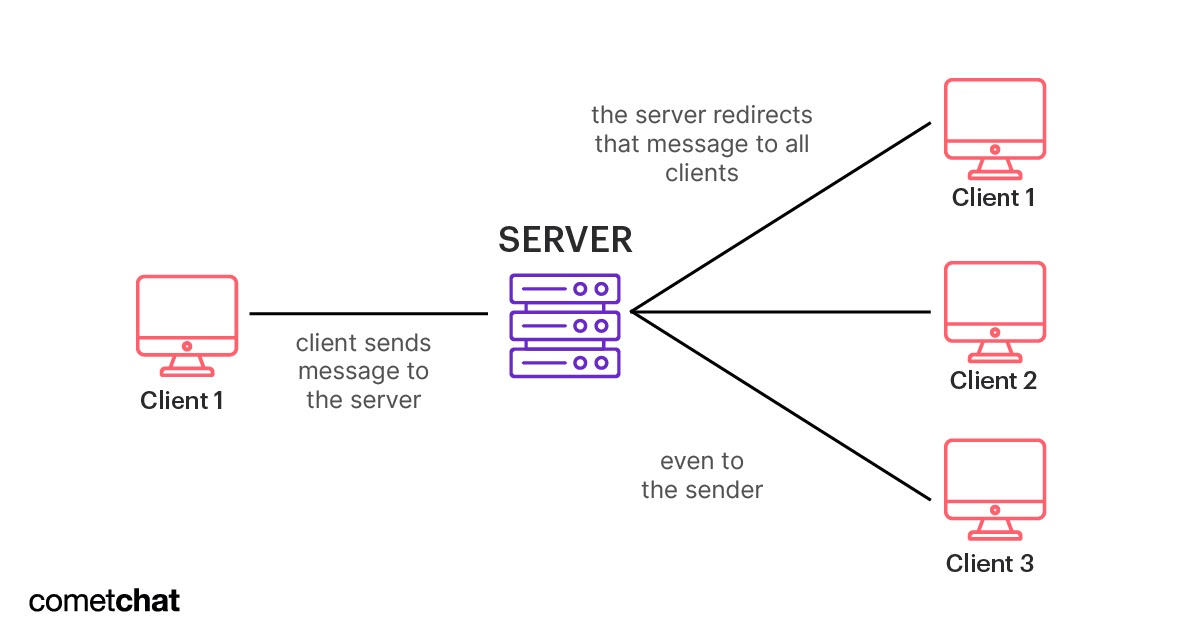


**C) ER Diagram**



# 10. Detailed Design

**10.1Detailed Design of <Component/Module 1>[[2]](#footnote-1)**



### **Chat Client**

The chat clientis what the user experiences. A desktop, web or smartphone chat application, the chat client is responsible for interacting with the operating system (i.e. your computer, browser or smartphone).

Interactions include sending push notifications, displaying data to the user and storing messages and files. When you type a message and hit send, the chat client transmits that message to the other major component: the chat server.

### **Chat Server**

The chat server is just that, a server (or usually many many servers) that hosts all the software, frameworks and databases necessary for the chat app to operate. This server, or pool of servers, is responsible for securely receiving a message, identifying the correct recipient, queuing for the message and then forwarding the message to the recipient's chat client.

The chat server’s resources can include a REST API, a WebSocket server, an AWS instance for media storage, etc

**Chat Rest API**

A Chat REST API is most often used to facilitate the functionality of the chat app outside of messaging. For example, authentication, profile settings and notification settings can all be managed through a REST API. All these functionalities are built into CometChat’s SDKs and Chat APIs. [This article](https://www.cometchat.com/tutorials/what-is-chat-api) talks about Chat APIs, their features and how they work in more deta

**WebSocket Server**

A WebSocket server and client library are useful components for a chat app. Typical communication between a client and server is done using HTTP and requires that the client makes a request for data from the server. The server itself can’t push data to the client without the client first making a request.

In a chat app, this quickly leads to inefficiencies since the client would have to poll the server every second for new messages. A WebSocket is a persistent connection between client and server that provides a bidirectional communication pathway.

That means, the server can send data to the client without first getting a request. Real-time chat is the perfect use case for WebSockets. [This article](https://www.cometchat.com/tutorials/what-is-websockets) talks about websockets in more detail

### **Media Storage**

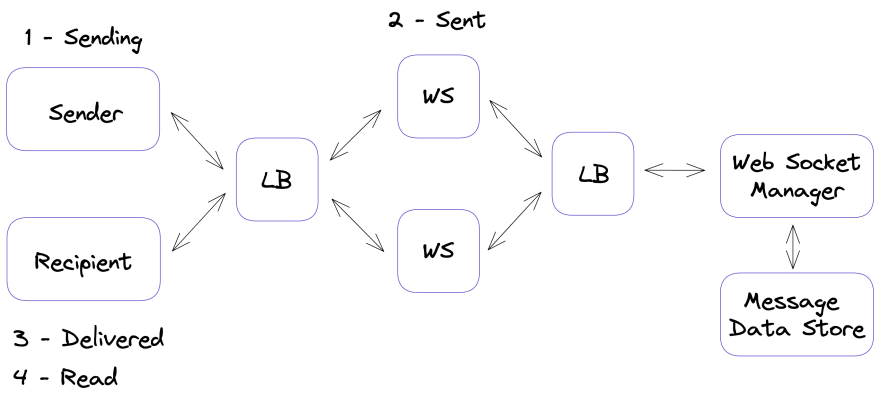
Of course, almost every mobile or web app requires some kind of data storage. Chat apps require both data storage and media storage. Profiles, messages and media files need to be accessible at all times.

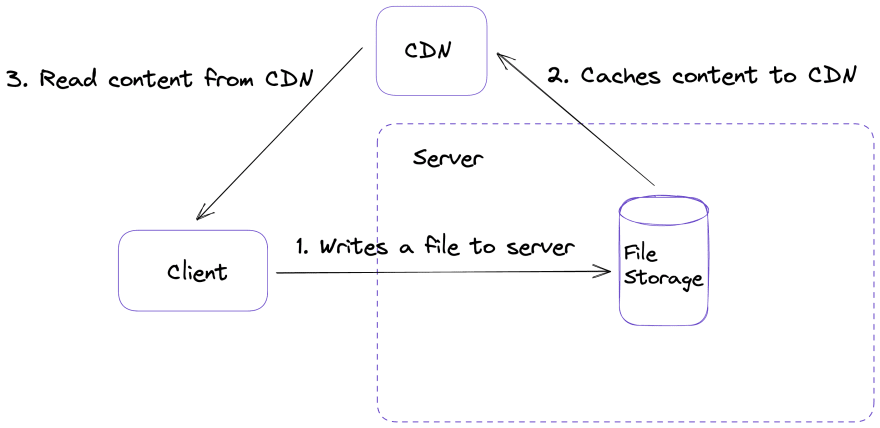
Your choice of database, whether it’s a relational database or a NoSQL database (like MongoDB), relies heavily on your use cases, what’s important to your users, and what your development team is familiar with.

You can even use a mix of both, using a reliable and robust relational database for generic data like profile settings and using a NoSQL database for the messages. Since key-value stores allow for easier horizontal scaling and low latency to access data, NoSQL databases like Cassandra are perfect for storing messages.

While every chat app has a chat client and server, the size and functionality of those components will vary depending on the app’s intended use. The components can scale from [an MVP chat app](https://www.cometchat.com/blog/how-to-create-mvp-for-chat-app) with basic capabilities to a full-blown, mature chat app like WhatsApp that has over two billion active users.

### 10.2 Design Alternatives[[3]](#footnote-2)

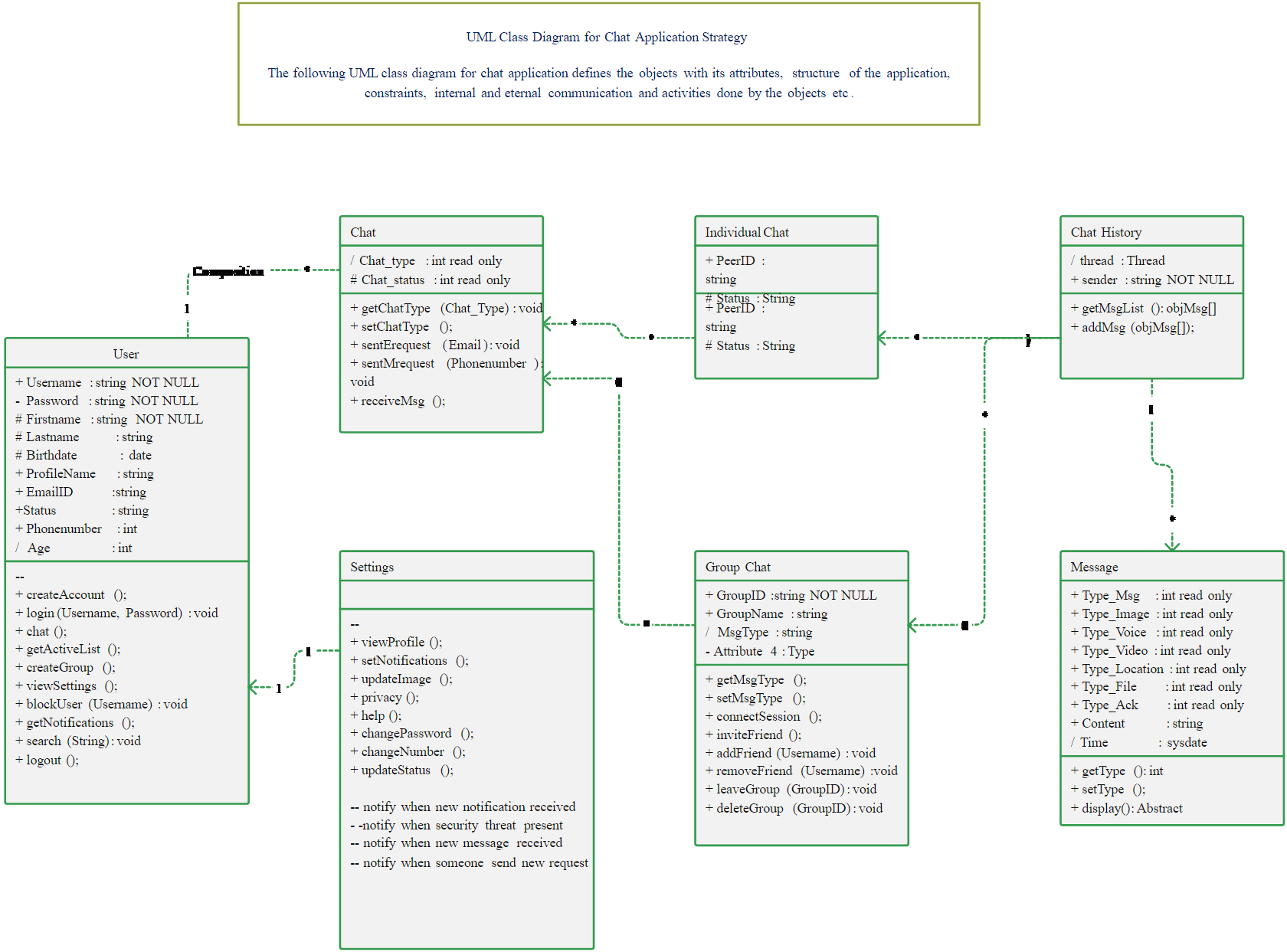




#### 10.3 Design Alternative 1

##### Design

Class diagram



**Hardware Interface**

Android phone 256 MB minimum RAM required Internet or LAN connections Processor with speed of 500MHz

**Tools and Technology**

Quality Planning -Software QFD

Product Innovation- Brainstorming, Mind-Map, TRIZ/ARIZ, Innovative algorithms

Software Analysis –Brainstorming, Mind-Map Design, patterns UML tools and technique

Database modelling tools- Mongo dB Compass, Mongoose, Mongo dB Driver

Software Development Methodology- Agile, scrum

Programming Language -Typescript, JavaScript, Dart, Json, yaml, swift, java

##### B) Design Advantage[[4]](#footnote-3)

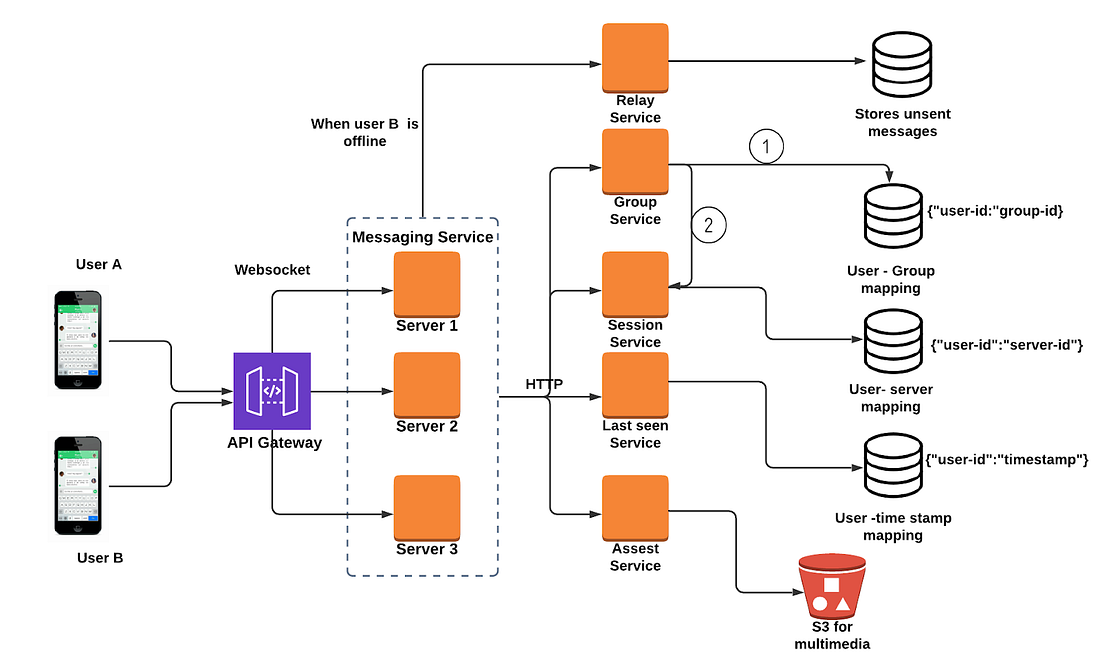
* Messenger requires an internet connection, so there's no additional fee for using it. In fact, you can chat with anyone from anywhere as long as they have a smartphone and are connected to the internet.
* Messenger is much faster than text messaging, which can be delayed when too many users are in the same area using it at once. Messenger uses data instead of texts, so you don't have to wait for your messages to go through.
* Messenger is easy to download and sign in, plus it requires no additional information other than your phone number. All you have to do is open the app on your smartphone, create a profile if you don't already have one or find another user who has added their contact details into Messenger already, then send them a message. You can also send a photo, video, or voice message quickly and easily.
* This is a great money saver if you and your friends and family members don't already have an app like Skype installed on all of your devices. All you need to do is open the Messenger app, click "video call" or "voice call," then choose who you want to talk to. Money saving benefits of Facebook messenger are worth noticing when you are living abroad and have friends and family members to call often.

##### C)Design Limitation[[5]](#footnote-4)

* You can sign up for an account using your email address or phone number, but the only way you're able to use Messenger is if you already have an active page on their website. If this isn't something that's ok with you, then there are other free messaging apps available in app stores worldwide.
* Messenger is only available for smartphones, so if you don't have one or your smartphone isn't compatible with it, you won't use this app. If you don't have a smartphone or an internet connection, other messaging apps work on both devices and desktop computers.
* Messenger is only available in English, so if you don't read or speak it, then this app isn't for you. Other messaging apps support multiple languages and dialects, though.
* If you have it on your phone, be aware that it will take much of your storage. Messenger runs in the background to get you an instant notification about active conversations. The continuous activity is a sure and short way to drain your device’s battery. Provided it requires internet connectivity to operate; battery discharge rate may stagger in areas where connectivity is poor.

#### 11. Design Alternative

##### Design



##### **Design Advantage**

**MESSAGING**

One of the primary use of CHATTY is messaging. Just like other social apps, you have a list of conversations that you’re engaged in. This feature is pivotal as you can add people in a variety of ways aside the conventional way of details collection. When fully operational, you will be amazed how individuals will have to scan their phones during details collection.

This is made possible as each CHATTY user will have a unique barcode known as a QR code. One person can scan the other user’s QR code to add them to CHATTY. Users of CHATTY can also use a phone number to add a person to their contact list and even search for people nearby.

CHATTY enormous social power will be let loosed as it will translate to become one of the main ways people will have to communicate in Nigeria and sub-Sahara Africa. Even when doing business, people will prefer CHATTY to email. With inbuilt plug ins that allows individuals to load pictures and videos and their contacts can make comments about them,

then the barrier of environmental distance will be broken and knowledge will increase as people will be able to explore the beautiful landscape of Africa without really travelling to those regions physically.

**FILE TRANSFER**

With the sophistication design of the CHATTY app, individuals will be able to share files without size constrains ranging from images, videos, to large documents files like zip, dmg, and so on. As long as you have a smart phone with android or iOS running on it, it can be linked to your CHATTY app to achieve this.

The uniqueness of this app is that is works seamlessly with your use case for it and saves you the stress of uploading and downloading the file as the mobile device become a host you can download files from with the simple aid of your smart phone at download cost, thereby helping the world become a closer with ease.

**SOCIAL BARRIER BREAKER**

With the introduction of the Pidgin English feature, CHATTY is intended to ensure we all socialize within our limits and create more friends irrespective of our ranks and file. Just like the BBC pidgin that is bringing the news room closer to the lower strata, CHATTY hopes to rekindle the spirit of humanity in her natural state devoid of privileges some individuals have over others. In conceptualizing this app, careful thought and study has been given to a methodology of really defining the peaceful co-existence of human and, CHATTY will really serve as a means of promoting and forestalling unity as it helps in defining the place of equality of the human race devoid of sentiment, religion and ethnicity.

##### C**) Design Limitation**

The following challenges were observed after development of this system

• Only registered users can use the system

• Internet must be available to use the application

• There must be minimum of two users per time for interactive chatting

#### 12. Design Selected

##### **Criteria for Selection of Design Alternative[[6]](#footnote-5)**

* Communication is a means for people to exchange messages. It has started since the beginning of human creation. Distant communication began as early as 1800 century with the introduction of televisions, telegraphs and then telephony.
* Interestingly enough, telephone communication stands out as the fastest growing technology, from fixed lines to mobile wireless, from voice call to data transfer. The emergence of computer network and telecommunication technologies bears the same objective that is to allow people to communicate. All this while, much efforts has been drawn towards consolidating the device into one and therefore indiscriminate the service.
* Chatting is a method of using technology to bring people and ideas together despite of the geographical barriers. The technology has been available for years, but the acceptance was quite recent. This project is an example of a real time chat app, it is made up of the user application which runs on the user mobile and the server application, which runs on any PC on the network. To start chatting the user should get connected to a server where he can do group and private chatting.

##### **Philosophy[[7]](#footnote-6)**

* As the first decade of the new millennium closed, an enterprise chat app renaissance slowly began. Though email had been widely used for the previous twenty years, companies soon began looking for a better way to communicate quickly; email-based workflows are slower and do not allow for many business functions that are now critical to work, like screen-sharing or video calls.
* Some one-to-one chat applications existed, like GChat and Outlook Messenger, but group messaging applications had yet to take off. In January 2010, three graduates of Rensselaer Polytechnic Institute—Chris Rivers, Garret Heaton, and Pete Curley—launched Hipchat, a web-based chat and instant messaging service. Shortly thereafter, Atlassian acquired it in March 2012. Its premium version addressed several enterprise concerns by adding screens haring, history retention controls, and the ability to run within corporate firewalls.
* In August 2013, the enterprise chat space exploded. Slack, which stands for “Searchable Log of All Conversation and Knowledge,” grew out of an internal tool used during the development of Glitch, a defunct online game.
* Though not the first (or even the most revolutionary) office chat app, Slack’s growth—it now has four million users—has dwarfed most other programs. A mere 1.25 years after launching, it reached a valuation of $1 billion. By April 2015, Slack was worth almost three times that. Slack–with its channels, DMs, and private groups–is reminiscent of IRC, which might help explain its popularity.
* A host of similar apps have since incorporated these features. 2015 saw the launch of Cisco Spark, which has since evolved into an entire ecosystem of SDKs, APIs, and a developer site. Microsoft designed MS Teams to compete directly with Slack; it launched in early 2017 as an integrated component of Office 365. As businesses prioritize digital transformation, enterprise chat apps continue to enjoy overwhelming popularity. What’s to come? Chatbots, machine learning, and increased integration are all popular with both enterprise and commercial users, who discover new ways to chat every day.

##### **Design Assumptions[[8]](#footnote-7)**

1 Processes

2 Registrations for new Members

3 Login Features

4 Adding Friend

5 Chat Forms

6 Settings

1 **Processes**

Installing the chatty on an android mobile device is as simple as installing the apk file.

2. **Registrations for new members**

To register a new user, Chatty app require the using to provide their email and phone number. It checks if the phone number or email already has a user in the database, if it doesn’t the user is then taken to a screen where they fill in their details which include full name, date of birth and a short description of themselves. The user is then asked to confirm their email and phone number through the email message or SMS sent to them which contains a verification code link to their account. Finally, the user goes through it local biometric lock and adds their profile picture if need be, then done. The user will now have a full active account.

**3. Login Features**

Chatty uses a two-factor login system which includes biometric authentication and email or phone number verification. First the phone number is verified to belong to a user and then a verification code is sent to both the users’ email and phone number if the phone number was found to belong to a user. Next the user is asking to continue with the login process with the device local biometric authentication, which could be either face id or fingerprint authentication.

**4. Adding Friends**

Adding friends through the chatty app can be done by clicking floating button at the bottom right corner of the app which after giving the app permission brings a list of all your contact list saved on your mobile phone.

**5. Chat Form**

The Chat Form hold and displays the current conversation going on between two users of the app, this conversation is not saved locally or on a live server, so a new session is created for every new instance in time. The chat form has two main items which are the message areas which on-going conversations can be seen and the chat form which carry an input field to send new messages area.

**6. Account Settings**

Using the Account settings is very simple and clear, as there are not too many functions on it. The Account settings allows you to logout, see your contact list count, view your details and open the app settings and app info.

##### **Design Risks/Constraints**

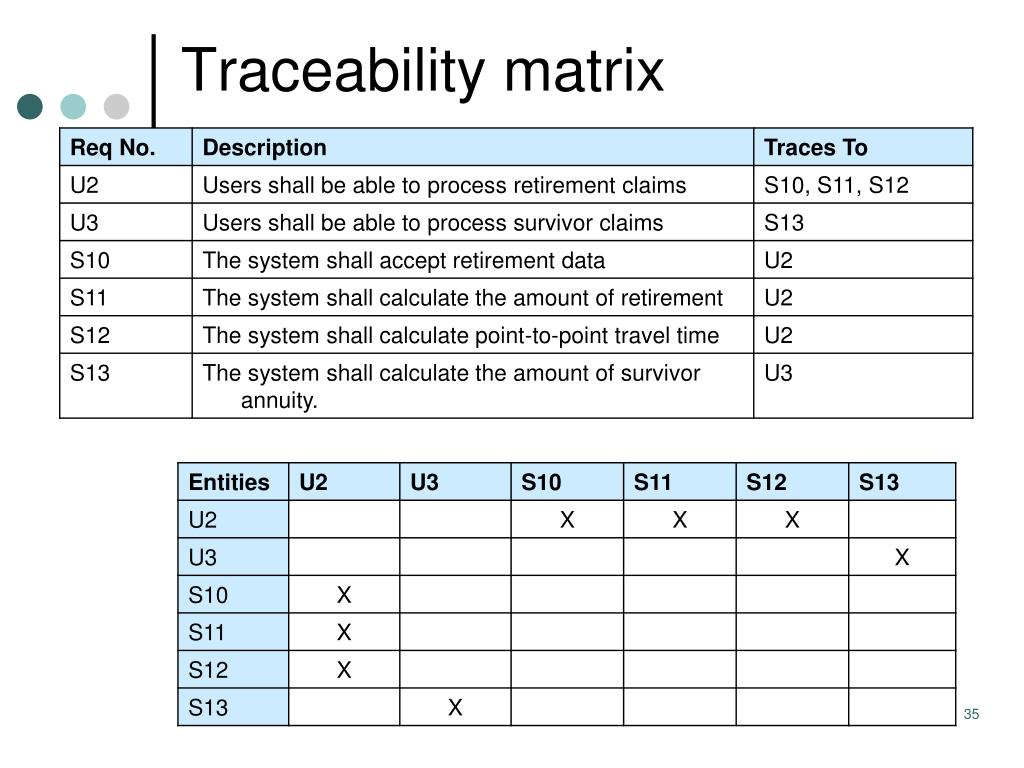
* Phishing is when you click on a link that redirects you to a malicious site that will steal or expose your confidential information. Scammers and hackers know that people communicating via IM or messaging apps are more likely to click on random links. They may not expect that a link will be fraudulent, unlike when they open an email.
* Bots are programs that pretend to be real-life users and send you spam, messages, advertisements, or attempt to link you to sales or phishing sites (outside the app). Other bots might suggest you play games or download additional apps.
* Getting your news solely through messaging may lead you to sites that publish false information. Some of these websites could be run by other countries or created just to spread fake news.
* Catfish would love to date you and whisk you off your feet! Romance scammers are con artists who pretend to be in love to get to the victim’s wallet.

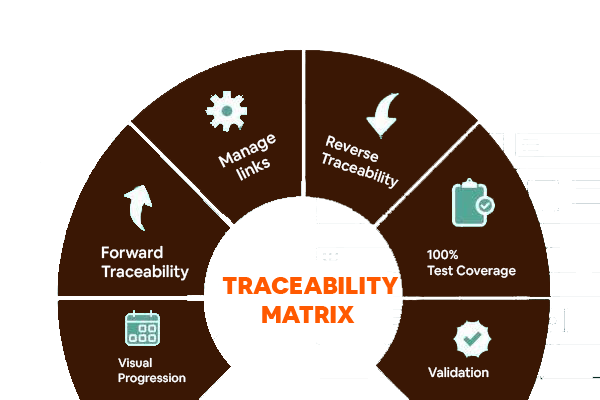
### Interface Requirements[[9]](#footnote-8)

# Design Quality Objectives[[10]](#footnote-9)

| **Attribute** | **Design Expectation** | **How addressed by selected design alternative( s)** | **Remarks** |
| --- | --- | --- | --- |
| Reusability[[11]](#footnote-10) | needed | High performance | Can be used |
| Maintainability[[12]](#footnote-11) | Required | needed | Done quaterly |
| Portability[[13]](#footnote-12) | MAC | Windows | Always |
| Extensibility[[14]](#footnote-13) | Client | Clent to client | Always |
| Other optimizations [[15]](#footnote-14) | Server to client | Client | Not needed always |
| Usability[[16]](#footnote-15) | Client | Client to client | always |
| Reliability[[17]](#footnote-16) | Server | Client | Always |
| Security[[18]](#footnote-17) | Server | Client to client | Always |
| Performance[[19]](#footnote-18) | Server | Server and client | Maintained always |

# 3. Traceability Matrix[[20]](#footnote-19)





# Revision History

| **Ver. No.** | **Author** | **Date** | **Reason for Change** | **Change Description** |
| --- | --- | --- | --- | --- |
| 1 | Aron P | 2020 | Websockets | Webrtc vs websockets.” |
| 2 | Bairam, W. | 2020 | golang | Making the switch from making the switch from node.js to golang. |

1. List of documents and other sources of information to be used in the project. This includes client-supplied documents, books, internal documents, etc. Designs can also be referenced here if being adopted from an existing project. Processes need not be referred [↑](#footnote-ref-0)
2. Your design should be represented in such a manner that the low level/detailed design forms the basis for programming. If required, this section can have further sub sections to represent your design. These sections and sub sections may contain description of each low level entity and describe special requirement that must be met by the entity that are not included in the HLD or SRS. It should identify all the sub entities/modules composing of detailed flow of the same. It may also contain description of relationship of this entity/module with other entities/modules. The dependencies should be identified clearly and are often graphically depicted by structure chart, data flow diagrams or transaction diagrams. The description should also include how an entity interacts with other entity. [↑](#footnote-ref-1)
3. Document the various design alternatives for the selected components that have been considered. A minimum of 2 design alternatives should be considered. [↑](#footnote-ref-2)
4. List the advantages of choosing this design alternative wrt meeting the objectives of the requirements – functional and non-functional [↑](#footnote-ref-3)
5. List the drawbacks/ limitations of the proposed design alternative. [↑](#footnote-ref-4)
6. The basis of selection of the design alternative should be elaborated here. The choice would normally depend on how well the alternative meets the functional and non-functional requirements of the client. [↑](#footnote-ref-5)
7. The basis of the design. On what grounds the designer has presumed this approach. What are the aspects that are considered for the design. This is a mandatory section and needs to be addressed. [↑](#footnote-ref-6)
8. Give a brief description of the assumptions while arriving at the design for the requirements. It is MANDATORY to address this section [↑](#footnote-ref-7)
9. Provide details and criteria for interface of different components/modules [↑](#footnote-ref-8)
10. The Quality Attributes that the design needs to satisfy based on the Quality requirements and Non-functional requirements. The attributes should be addressed to start with and subsequently when a design alternative is selected, the table should be completed to state how the selected design meets the requirements. [↑](#footnote-ref-9)
11. Describe the Reusability requirement as stated by the client. For the selected design alternative state as to how the design will address the requirement of reusability. If the system will reuse some already developed design entity/ components, explain them in details.

    Also identify what components of the system to be developed can be reused in future and state its area of application. [↑](#footnote-ref-10)
12. State the Maintainability requirement given by the client. Describe as to how the selected design will address the maintainability of the system [↑](#footnote-ref-11)
13. If portability is a requirement, explain what the requirement is ( that is to which all environments portability is expected) and how this requirement is addressed in the chosen design [↑](#footnote-ref-12)
14. Extensibility requirement should be stated here. The design alternative chosen should have provision to take care of future enhancement of the system then describe details in this section. [↑](#footnote-ref-13)
15. If the system has some specific optimization requirements, such as memory, speed, flexibility, then state what these requirements are and how the selected design addresses those requirements. [↑](#footnote-ref-14)
16. If usability is requirement in SRS , then state the same and explain as to how the selected design addresses this requirement [↑](#footnote-ref-15)
17. If reliability is a requirement in SRS, then explain as to how the selected design addresses this requirement [↑](#footnote-ref-16)
18. If there are any security requirements stated in SRS, then state the requirement and explain as to how the selected design address this requirement [↑](#footnote-ref-17)
19. If there is a performance requirement stated in SRS , then explain as to how the selected design addresses this requirement [↑](#footnote-ref-18)
20. Provide a link to the Traceability Matrix. The Traceability Matrix would have been started at the SRS stage and will be developed further at the design stage. The traceability matrix should address how the selected design addresses the User requirements and ensure that all the design requirements are addressed. [↑](#footnote-ref-19)