**Minor Project Report on**

**Open Source Software**

**Submitted in partial fulfillment of Integrated**

**Mtech (IT)**

**International Institute of Professional Studies**

**Devi Ahilya Vishwa Vidhyalaya**

**Indore M.P.**

**Integrated Mtech Vth semester.**

**Batch 2K19**

**Submitted to Submitted By**

**Dr. Shaligram Prajapat Divyansh Patel**

**IT-2K19-15**

**BONAFIDE CERTIFICATE**

This is to certify that the project report entitled "Signal" submitted to Devi Ahilya University in partial fulfillment of the requirement for the award of the Master of Technology (Integrated MTech), is an authentic and original work carried out by Mr. Divyansh Patel (IT-2K19-15) under my guidance.

The matter embodied in this project is genuine work done by the student and has not been submitted whether to this University or to any other UNIVERSITY/ Institute for the fulfillment of the requirements of any course of study.

Internal Examiner: External Examiner:

**RECOMMENDATION**

The Project wok entitled "Hostel Management S" submitted by Divyansh Patel is satisfactory account of the bona fide work under my supervision and is recommended towards the end of their V th semester of Mtech Integrated 2021.

**Guided by:**

**Dr. Shaligram Prajapat**

**ACKNOWLEDGMENT**

We acknowledge our sincere thanks to those who have contributed significantly to this project. It is a pleasure to extend deep gratitude to our internal guide Mr. Shaligram Prajapat, IIPS, for his valuable guidance and support and to continuously prompt us for the progress of the project. We thank him for his valuable suggestions towards our project, which helped us in making this project more efficient and user friendly.

We thank and acknowledge each and every ones efforts that helped us in some or the other way for small and significant things.

**ABSTRACT**

The Signal messenger aims at defining a completely Integrated Application that is designed to cater to every administrative requirement like a proper functioning Instant messaging application . Signal Messenger is a smartphone messaging application that can be used on Android, Symbian, BlackBerry, iPhone and Windows phones. The application enables users to exchange instant messages with their contacts. This project explores how Signal can be used to enhance learning in a University course.

**TABLE OF CONTENTS**

1. **INTRODUCTION**

1.1 Problem definition

1.2 Proposed system

1.3 Aim

1.4

1. **FEASIBLITY STUDY**

2.1 Feasibility Analysis

2.2 Technical Feasibility

2.3 Operational Feasibility

2.4 Economical Feasibility

**3. ANALYSIS**

3.1 Requirement Analysis

3.1.1 Functional requirements

3.1.2 External Interface requirements

3.1.3 Non Functional requirements

3.1.4 Software quality attributes

3.2 Time and Cost analysis

**4. DESIGN**

4.1 Logical design

4.1.1 Entity definition

4.1.2 Attribute definition

4.1.3 Relationships

4.1.4 E-R diagram

4.1.5 Data Flow Diagram

4.1.6 Use Case Diagram

4.2 Physical design

4.2.1 User interface design

4.3 Architecture

4.3.1 Architecture of signal

4.3.2 low level diagram of signal

4.3.3 High level diagram of signal

**1.INTRODUCTION**

**1.1Problem Definition**

* Before whatapp and signal, text based user communication is based on emails, messaging services provided by one’s telecommunications provider or through the messaging application built inside a social media platform such as Facebook.
* Each of these poses a unique issue that prevents user from getting the cheap, reliable, easy to use messaging services. Emails tend to be too verbose and don’t associate user’s phone contacts with their email Ids.
* Thus, to send an email message a separate database of email ids, need to be maintained.
* It also restricts the use of inline media such as emoticons that can be attached to the message.
* Text messaging services provided by telecommunications providers are not cost effective and involve paying a monthly, quarterly or annual subscription fee.
* Social media based messaging applications might provide media rich, free online messaging services, but often require the user to install the heavy encompassing social media application.
* The problem of having a separate database of contacts, not linked to one’s phone’s contacts, also arises in this case.

Signal app aims at tackling the above issues and providing a seamlessly integrated, web based, mobile messaging application, that identifies the contacts on a user’s mobile device and enables free text messaging services. For better interaction and user experience, Signal intends to provide support for different media types such as audio, video and emoticons with a better message encryption than WhatsApp.

**1.2Proposed System Signal**

Signal is a cross-platform centralized encrypted instant messaging service developed by the non-profit Signal Technology Foundation and Signal Messenger LLC. Users can send one-to-one and group messages, which can include files, voice notes, images and videos .It can also be used to make one-to-one and group voice and video calls and the Android version can optionally function as an SMS app. The Signal Protocol also supports end-to-end encrypted group chats. The group chat protocol is a combination of a pairwise double ratchet and multicast encryption.

All communications between Signal users are automatically end-to-end encrypted (the encryption keys are generated and stored on the phones, not on the servers). To verify that a correspondent is really the person that they claim to be, Signal users can compare key fingerprints (or scan QR codes) out-of-band. The app employs a trust-on-first-use mechanism in order to notify the user if a correspondent's key changes.

The non-profit Signal Foundation was launched in February 2018 with initial funding of $50 million from Brian Acton. As of January 2021, Signal had more than 105 million total downloads, and the app had approximately 40 million monthly active users.Signal has been installed on more than 50 million Android devices.

**1.3 Aim**

The aim of signal is to build a reliable, secure, and private communication experience that is broadly accessible and simple to use.

**1.4 Objectives**

* To provide real time communication
* To provide Better Encryption to the messages
* No need to make separate database of users with the help of phone numbers the user can chat
* To provide Convenient group chat
* To increase speed of communication
* To provide more security
* To provide unlimited messaging at very low data cost
* To boost the productivity

**2.1 FEASIBILTY ANALYSIS**

The main objective of the feasibility study is to treat the technical operational and economic

feasibility of developing the application. Feasibility is the determination of whether or not project is worth doing. The process followed in making this determination is called feasibility study. All systems are feasible, given unlimited resources and infinite time. The feasibility study to be conducted for this project involves:

• Technical Feasibility

• Operational Feasibility

• Economic Feasibility

**2.1.1 TECHINAL FEASIBILTY**

It is the measure of the specific technical solution and the availability of the technical resources and expertise. It is one of the first studies that must be conducted after tool has been identified. A technical study of feasibility is an assessment of the logistical aspects of business operation. This is considered with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably but should include the facility to produce outputs in a given time, response time under certain conditions and the ability to process a certain amount of transaction at a certain speed.

**From user end**

* To install the application ,user must have a Google Play Store , App Store or any downloadable resource
* the signal application will require a stable internet access as it is a messaging application
* the application is emphasized with a good user interface to expect consistency and stability in user messaging environment.

The signal application is developed using two pieces of software from open whisper system

* RedPhone: Encrypted voice calling app
* TextSecure :Encrypted texting platform

which are open source software for the continued development of textsecure and RedPhone. The idea is to implement Encryption to data to make better standard of messaging.

**2.1.2 OPERATIONAL FEASIBILTY**

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented, whether there will be resistance from the users which will affect the possible application benefits. It is the ability to utilize, support and perform the necessary tasks of a system or program. It includes everyone who creates, operates or uses the system or program. It is the measure of how well a proposed system solves the problem and takes advantages of the opportunities identified during the scope definition and problem analysis phases. This system helps in many ways. It shows the number of users using Signal and gives the data information of their sharing data. Which is organized in Pie-chart and Bar chart

* + 1. **ECONOMIC FEASIBILITY**

Economic feasibility is the measure of the cost effectiveness of an information system solution. Without a doubt, this measure is most often and important one of the three. Information systems are often viewed as capital investments for the business, and, as such should be subjected to the same type of investment analysis as other capital investments.

Economic analysis is used for evaluating the effectiveness of the proposed system. In economic feasibility, the most important is cost-benefit analysis.

**From user side:**

The intended application Signal developed is available for free, thus the project is economically feasible for the user. The only requirement for a user is a functional smart phone with Android Operating System.

**From Company side:**

At the company side the Project cannot be regarded as economically feasible because the signal app possess different costs of the resources needed for the development of system like Framework, Network, database,API ,documentation , infrastructure , storage , Startup cost, recruitment of personnel. This project is not economical as it mainly depends on the sharing of data between two phones.

**3.1Requirements Analysis**

**3.1.1 Functional Requirements**

Functional requirements are properties that must exist in the final system. For the Signal , we need to download the application from the play store. The application is free to download. To use the application the user needs to register and login to the application after installing by providing login information. Once, he or she logins into the application, they can use all the features.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | **Requirements** | Input | Output |
| 1. | The user shall be able to register in the system | Username  Phone Number  SMS Verification Code  Pin | New account |
| 2. | The user shall be able to login through a username  and a password | Username  Pin | Login/Error |
| 3. | The user shall be able to see their profile | - | - |
| 4. | The user shall be able to modify their personal  information |  |  |
| 5. | The user shall be able to search for a contact | Search query | Results |
| 6. | The user shall be able to add new contacts | Add request | Accept /  Block |
|  | The user shall be able to send messages |  |  |
| 7. | The user shall be able to delete a contact |  | Delete  contact |
| 8. | The user shall be able to see other contacts’ profiles |  |  |
| 9. | The user shall be able to begin a new conversation  with a selected contact |  | Messages |
| 10. | The user must be able to take picture or videos from Camera or choose from storage |  |  |
| 11. | The user must be able to make new group |  |  |
| 12. | The user must be able to users various symbols or emojis during conversation |  |  |
| 13. | The user shall be able to get and see push  notifications whenever an event happen(messages) |  |  |

**General Requirement**

**Functional requirement 1:**

**Description**: If the application on any device is not connected to the Internet, the system should display initial display.

**Functional requirement 2:**

**Description**: Send a message function sends a text message to another user of app.

**Inputs**: inputs are characters, including numbers and the receiver contact from contact list.(Name of the receiver in this case)

**Processing**: Reads the message from the buffer, stores it in the sender' s device DB, pass it to the Central DB and passes the message to receiver and stores it in the receiver s device DB. Sender then gets indication of receiving the message by the receiver and indication of reading the message by the receiver. (Sender gets mark on the message when receiver gets the message and double mark when receiver opens the message)

**Outputs**: Mark at the sender' s side -when receiver gets the message in his device.

Double mark at the sender's side when receiver opens the message.

The message itself - stored in Central DB, Stored in the sender's and the receiver's devices DB for history. Notification at sender's side if the message wasn't transferred to receiver.

**Functional requirement 3:**

**Description**: User Registration User is able to register for the application through a valid phone number. On installing the application, user must be prompted to register their phone number. If user skips this step, application will close. The user's phone number will be the unique identifier of his/her account on WhatsApp.

**Functional requirement 4:**

**Description**: Adding New Contacts The application is able detect all contacts from the user's phone book. If any of the contacts have user accounts with Signal those contacts must automatically be added to the user's contact list on WhatsApp. If any of the contacts have not yet registered on Signal, user is provided With an invite option that sends those contacts a regular text message asking them to join Signal along with a link to the Signal application on Google Playstore or respective store depending on the Operating system the device is running.

**Functional requirement 5:**

**Description**: Send Message User is able to send instant message to any contact on his/her Signal contact list. User will be notified when message is successfully delivered to the recipient by displaying a tick sign next to the message sent.

**Functional requirement 6:**

**Description**: Send Attachments User is able to send audio, video and images as attachments.

Audio formats that the application should support: mp3, wav.

Video formats that the application should support: avi, mp4, flv, gif.

Image formats that the application should support: jpeg, png.

**Functional requirement 7:**

**Description**: Broadcast Message User is able to create groups of contacts. User is able to broadcast messages to these groups.

**Limitations**: The recipient should have the sender in his contacts to receive the broadcast.

**Functional requirement 8**:

**Description**: Message Status User is be able to get information on whether the message sent has been read by the intended recipient. If recipient reads the message, 2 ticks will appear next to the message read.

**Functional requirement 9:**

**Description**: Auto Backup User is able to have all his messages backed up on Internal Storage without ever being prompted. User has the choice of setting the frequency by which the backup can be made.

**Functional requirement 10:**

**Description**: The Application should be able to connect to Internet automatically even though the application isn't opened by the user. The Application should stay connect to internet at every point of time if the network resources are available on the device.

**Functional requirement 11:**

**Description**: Signal should be able to distinguish between the network resources like Wi-Fi and Mobile Data in order to perform media respective download operation with respect to said so network resource.

**Functional requirement 12:**

**Description**: User is able to set media download preferences with respect to various network resources available. Media download preferences are with respect to Wi-Fi and Mobile data and different.

**Functional requirement 13:**

**Description**: Share photo function sends a picture (photo) to another user of Application.

**Inputs**: Input is the picture in JPEG, GIF etc. formats and the receiver contact from contact list.(Name of the receiver in this case)

**Sender sends existing photo**

Reads photo from the device DB, sends it to receiver, stores it in the receiver's device DB, Sender then gets indication of receiving the photo by the receiver and indication when the photo is viewed by the receiver. (Sender gets mark on the message when receiver gets the photo and double mark when receiver opens the message with the photo)

**Sender takes new picture and sends it**

When sender chooses to send new photo, the camera of sender' s device turns on and sender can take photo. After that, stores photo in sender' s device DB, sends it to the receiver and stores it in the receiver' s device. Sender then gets indication of receiving

the photo by the receiver and indication when the photo is viewed by the receiver.(Sender gets mark on the message when receiver gets the photo and double mark when receiver opens the message with the photo)

**Functional requirement 14:**

**Description**: Using User's contact list from the device.

**Processing** Imports contacts from user's device to contacts DB. Then checks which accounts have the Signal app, by using accounts DB, and then stores this information in the contacts DB (synchronization). All contacts that have Signal Messenger, appear in Signal Messenger contact list on user' s device with status. If new contact added to contacts list of the device, the synchronization executed and new contact added (if new contact has messenger).

**Outputs**: Signal messenger contact list.

**3.1.2 External interface requirements**

**i) User Interfaces**

The Signal user interface should be intuitive, such that 99.9% of all new Signal users are able to use Signal without any assistance

**ii) Hardware Interfaces**

The hardware should have following specifications:

* Ability to run the latest version of application
* Ability to read gallery
* Ability to exchange data over network
* Touch screen for convenience
* Keypad (in case touchpad not available)
* Continuous power Supply
* Ability to connect to signal network
* Ability to take input from user
* Ability to validate user

**iii) Software Interfaces**

The software interfaces are specific to open source, cross-platform, encrypted instant messaging client

**3.1.3 Nonfunctional Requirements**

1. **Performance Requirements**

* It must be able to perform in adverse conditions extremely sL

slow

* Internet connections and low battery on device.
* Uninterrupted connections High data transfer rate

1. **Security Requirements**

* Signal provides encryption so as to prevent unauthorized

access to a message midway.

* Signal messages are encrypted with the Signal Protocol (formerly known as the Text Secure Protocol). The protocol combines the Double Ratchet Algorithm, prekeys, and an Extended Triple Diffie–Hellman (X3DH) handshake and uses **Curve25519, AES-256, and HMAC-SHA256** as cryptographic primitives.

**3.1.4 Software Quality Attributes**

**i)  Availability**:

The Signal  internal Server has to be available 24 hours a day.

ii)  Security:

The Application should provide maximal security in order to make that much more transparent there are the following requirements

It must be impossible to plug into the internal server network.

**iii) Maintainability:**

Only maintainers are allowed to connect into internal servers

**iv) Space**

The mobile application(signal) shouldn’t take a lot of space to fit the Android criteria.

Signal app can be downloaded on Android from Google Play store and iPhone from Apple App store. On iOS Signal weighs 133.5 MB while on Android it weighs **96.7 MB**

**v) Usability**

The user interface should be easy to use and understand

**3.2 Time and cost analysis**

Efforts and Time to create a Signal Application

These are certain grounds on which secure application development takes place. Here we will explain a few important modules for the proposed messaging application, along with the time and resources it may take.

• **Registration Module** –

This is the basic module, where the user registers himself/herself and then provides further information like name, surname, sex, country, and phone number. Once the information is furnished, the application sends an authorization message to the end-user. The application can read and then submit the verification code automatically to register the end user. Past consumer registration, the application syncs all the contacts to display the people who are already available on the signal. messaging platform.

**Development Time**: This module is quite the basic one, and it may. take up to 50-55 hours.

**• Real-time messaging Module** - Signal provides One on one and group chats as well. Since we are willing to develop our own Signal application, it is obvious to incorporate all the messaging features, so that end users can avail one of one chat and multi-user chat. For messaging applications, we generally use extensible Messaging and Presence Protocol (XMMP).which ensures the message sending and delivery mechanism works flawlessly. We also must enable the other chat features like stickers, emojis, and GIF files.

• **Development Time**: This is an important module, which takes up to 210-220 hours, for both frontend and backend

development

• **Voice Calling** - Signal offers a free calling feature which Internet traffic to make calls, rather than using a cellular service plan on your SIM connection. Several messaging applications use PJSIP library to implement VoIP (Voice over IP) protocol for end-to-end calling facility. It is an open-source multimedia communication library, which is written in C language. PJSIP uses a high-level Application Programming Interface by combining the Multimedia framework with SIP (signalling protocol)and NAT traversal functionality.

• **Development Time**: It could take up to 200-220 hours for building both frontend and backend components.

• **Video Calling module**- Video calling feature is the bare minimum requirement for any messaging application these days.

With the availability of high-speed internet, people are now preferring to make video calls with individuals or groups. We

have seen that many applications which did not have video calling features are now looking at the possibility of

incorporating this feature to remain relevant among their users.

• **Development time:** To develop a video calling feature, it may take 180-220 hours to build a frontend and backend interface.

This time could be vary depending on the operating system like Android or iOS.

• **Multimedia sharing module** -For any messaging application, we must integrate the capability of multimedia sharing. The

application should enable its users to share document files, pictures, voice notes, and video files. The media transmission

is executed by uploading a multimedia file, that to be delivered to an HTTP server. Which then uses its Base64 encoded

thumbnail while sending the link to the content together.

• **Development Time:** To incorporate the multimedia sharing module, it may take up to 40-50 Hours. Though it depends on

Android and iOS and several other factors like usage of an API or building it from scratch.

• **Settings Module** - This is an important module for any application, as it provides the end-user to make some customized

changes in the application by modifying the setting. The settings module consists of a user account and profile setting,

Privacy Settings, and Help etc. From this module, users can make changes in their user profile, update the settings like

switch on or off the read receipts, manage blocked contacts, delete accounts, change status, change contact numbers,

customize the notification settings, and change the appearance of the application, etc.

• **Development time:** This is an important module and may take up to 110-120 hours for development

• **Contact Sharing feature** - This feature enables users to send and receive contacts from their phone memories directly with

their friends. This is a very basic feature that uses XMPP protocol for message delivery. We can use several available APIs

to enable this feature and may develop it from scratch as well.

• **Development time**: This feature may take 70-80 hours of effort to develop.

• **Location sharing capability** - sharing the geolocation with our contacts is a must-have feature that should be integrated

into the application. Through this feature, we can share our current location or the location of any specific place with our

friends. We can also share our live location with any person for a specific amount of time. We can easily use Google Maps

Android API to enable this feature for android, for the iOS environment, we can utilize the Map Kit framework.

• **Development Time**: It could take 25-30 hours to develop and integrate this feature into your application.

• **Notifications** - On any messaging application, the user is notified about a personal chat message or a group message

through mobile app notifications. We can enable the feature, where the user may receive a pop-up notification when the

phone is locked. This feature can be developed from scratch, but we would suggest using Apple Push Notifications or

Google Cloud Messaging to enable this feature for our application.

• **Development time**: It may take hardly 15-20 hours to integrate this feature into our app.

• **Contact Sync module** - This module is used to sync the contacts in the phone to the cloud our applications database.

• **Development time:** It may take hardly 15-20 hours to integrate this feature into our app.

• **Call History -** This module displays the Call history (Incoming, outgoing, rejected call) for a user. It provides the end-user

with an account of different kinds of calls they have made and received.

• **Development tim*e***: It may take hardly 18-25 hours to integrate this feature into our app.

• **Connect Web/Desktop/QR Scanner -** This is a new feature, which is used to scan QR codes to visit any specific website or

make the payment as well.

• **Development time**: It may take hardly 15-20 hours to integrate this feature into our app.

• **Chat Backup -** It is an important module that helps us take the backup of our chats and other data, so that could be

accessed at any time later.

• **Development time:** It may take hardly 25-35 hours to integrate this feature into our app.

• **Data Storage** - This module helps us to keep a track of the Data storage usage of our application. We can make the

change in the storage setting to ensure the application does not take unnecessary space on the end user's mobile phone.

• **Development time:** It may take hardly 5-10 hours to integrate this feature into our app.

• **About Application** - This is again an important section, where end-user can browse important information and FAQs about

the application. It helps end-users with several queries about the application and offers support material as well.

• **Development time**: It may take hardly 20-30 hours to integrate this feature into our app.

• **End-to-end encryption** - This module helps a user to make changes in encryption settings.

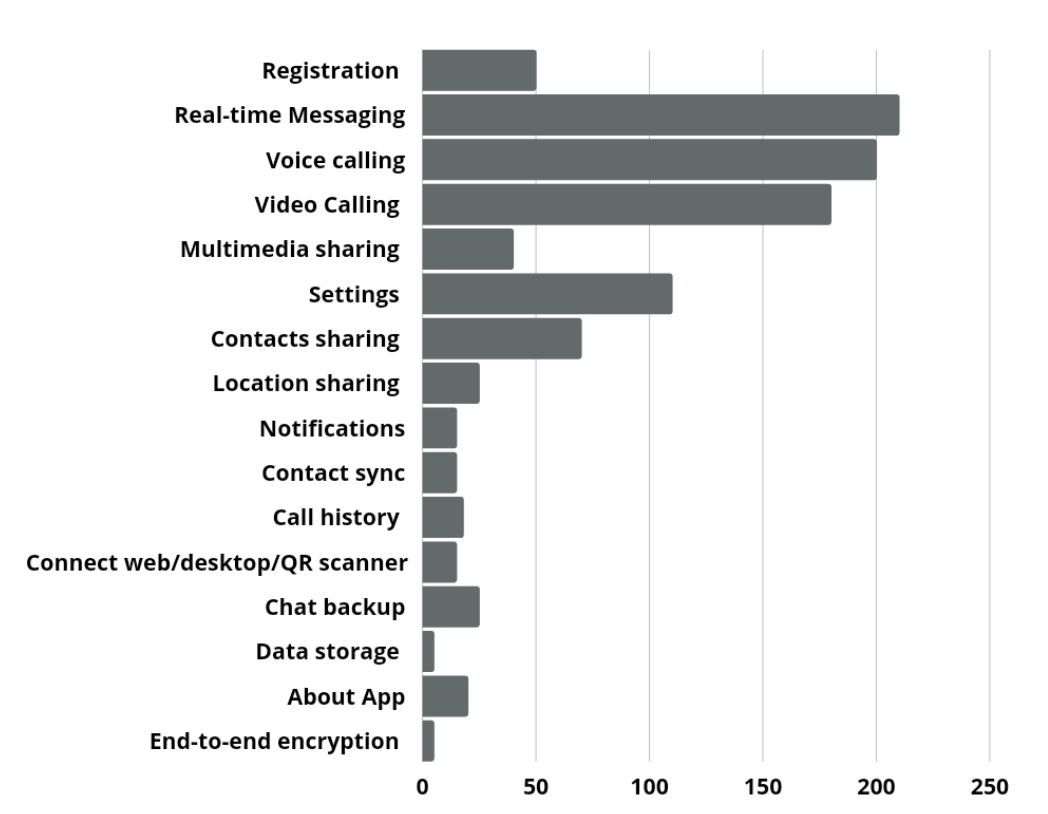
• **Development time**: It may take hardly 5-10 hours to integrate this feature into our app.

If 50 USD/hour is given.

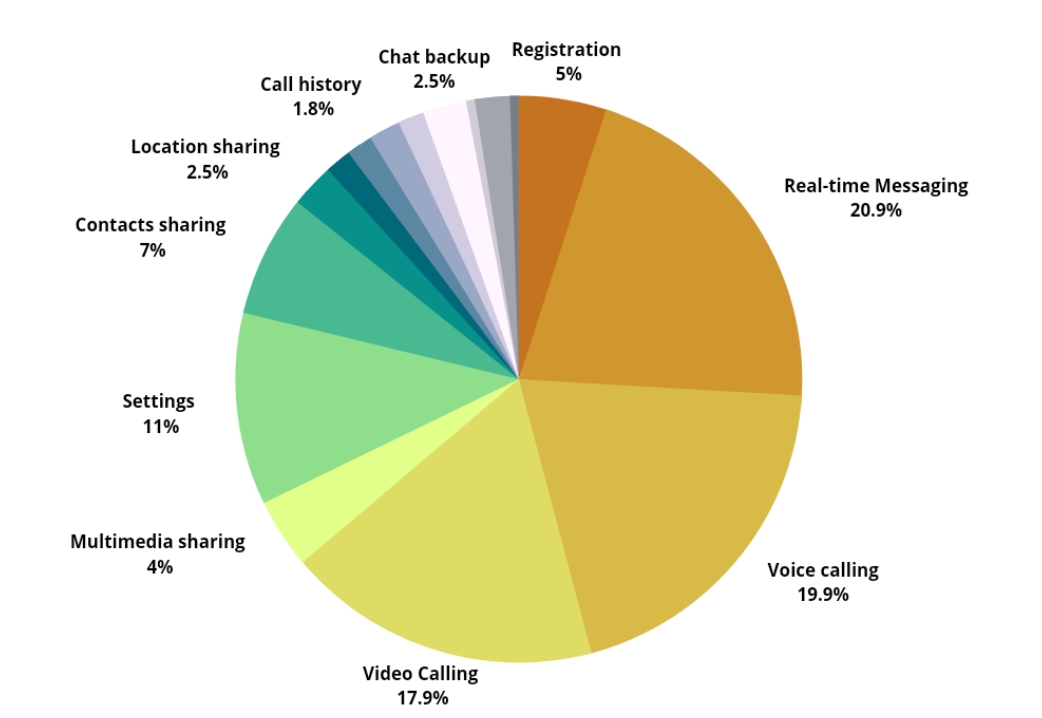
|  |  |  |
| --- | --- | --- |
| **FEATURE** | **DEVELOPMENT TIME(in hours)** | **DEVELOPMENT COST(in USD)** |
| Registration Module | 50-55 | 2500 |
| Real-time messaging Module | 210-220 | 10500 |
| Voice Calling | 200-220 | 10000 |
| Video Calling module | 180-220 | 9000 |
| Multimedia sharing module | 40-45 | 2000 |
| Settings Module | 110-120 | 5500 |
| Contact Sharing feature | 70-80 | 3500 |
| Location sharing capability | 25-30 | 1250 |
| Notifications | 15-20 | 750 |
| Contact Sync module | 15-20 | 750 |
| Call History | 18-25 | 900 |
| Connect Web/Desktop/QR Scanner | 15-20 | 2500 |
| Chat Backup | 25-35 | 1250 |
| Data Storage | 5-10 | 250 |
| About Application | 20-30 | 1000 |
| End-to-end encryption | 5-10 | 250 |

|  |  |  |
| --- | --- | --- |
| PLATFORM | APP DEVELOPMENT TIME | APP DEVELOPMENT COST |
| IOS | 800-900+ hours | USD 40,000-USD 45,000 |
| ANDROID | 900-1100+ hours | USD 45,000-USD 55,000 |

Row Chart for signal app with respect to time:-



Pie chart of signal app with respect to money:-



**4.Design**

This project is based on the functional design approach, which helps in understanding the design of the project in a simpler way by explaining its flow, use cases, and implementation more like a modular approach. For example, there are different modules in this project which have separate functionality and, other sub functionalities/modules. All the modules are designed, implemented and integrated together to make a flawless working application Signal.

**4.1 logical design**

**4.1.1 Entity definition**

Entities are the principal data object about which information is to be collected. Entities are either concrete or abstract, such as person, places, things, or events, which have relevance to the database.

The following is the list of entities that have been identified during analysis:

* Entity :User

Description :There are large number of users using Signal app

Identifier: user\_id

* Entity :Registration

Description : helps to put the account into signal database to use the Signal app

Identifier: Registration ID

* Entity: Profile

Description : Each registered user will have its profile to chat to others

Identifier: profile\_id

* Entity: Friend

Description : It is an another user

Identifier: F\_id

* Entity : Group

Description :collection of users

Identifier : group\_id

* Entity: Conversation

Description : among different users to whom you are having conversation

Identifier: conversation ID

* Entity: Messages

Description : A message can be a text, Images,videos,audio,document,etc

Identifier:Message\_id

**4.1.2 Attribute Definitions**

The attributes that are identified as part of the entities are listed along with their description.

1. **Entity Name: User**

|  |  |
| --- | --- |
| Attributes | Description |
| User\_id | Id of the user |
| User\_name | Name of the user |
| Phone\_no | Phone number of the user |
| DOB | Date of birth of the user |
| Age | Age of the user |

1. **Entity Name: Registration**

|  |  |
| --- | --- |
| Attributes | Description |
| Registration\_id | Registration id of the user |
| Phone\_number | Phone number for registering of user |
| Pin | Pin set by the user for the security of the app |
| name | Name Used while registration |
| Verification\_code | Code which came through sms for User verification |

1. **Entity Name: profile**

|  |  |
| --- | --- |
| Attributes | Description |
| profile\_name | Name of user profile |
| about | Description of user in about section |

1. **Entity Name: Friend**

|  |  |
| --- | --- |
| Attributes | Description |
| F\_id | Unique Id of the friend |
| F\_name | Name of the friend |
| Phone\_no | Phone number of the friend |

1. **Entity Name: group**

|  |  |
| --- | --- |
| Attributes | Description |
| group\_id | Unique id of the group |
| group\_name | Name of the group |
| group\_description | Description for the group |

1. **Entity Name: Conversation**

|  |  |
| --- | --- |
| Attributes | Description |
| Conversation\_id | Unique conversation id for specific friend Or group whom you are having conversation |

1. **Entity Name: Messages**

|  |  |
| --- | --- |
| Attributes | Description |
| Message\_id | Unique Id of the message |
| Message\_date | Date of the message |
| Message\_type | UC |
| Message\_content | Content of the message |

**4.1.3 Relationships**

**Binary relationship**

**One to One**

User < ---- > profile

User < ---- >Registration

**One to many:-**

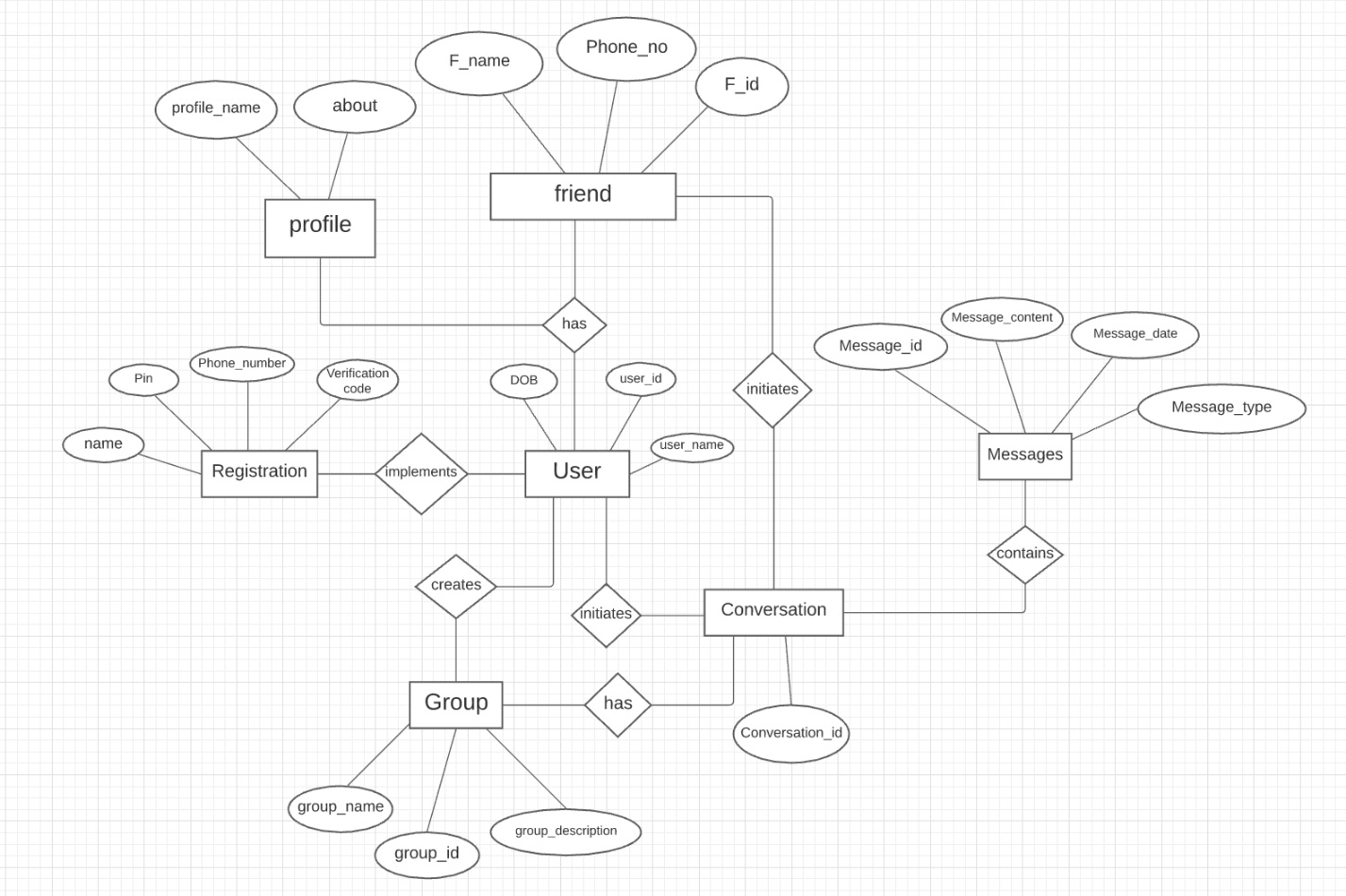
User< ---- > Friend

User< ---- > group

Conversation < ---- > Messages

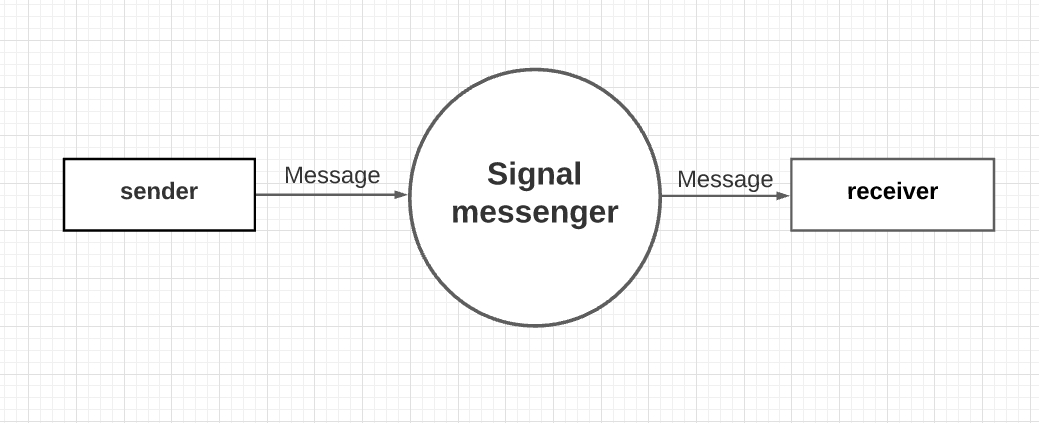
Group < ---- > Conversations

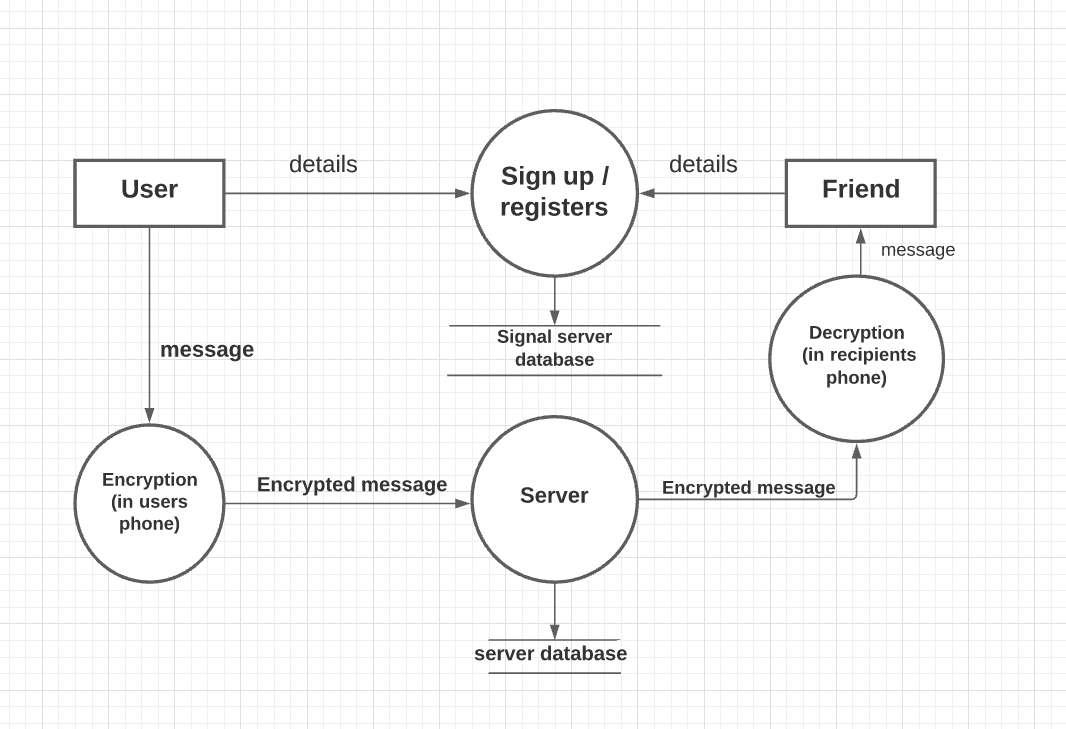
**4.1.4**  **The E-R diagram**



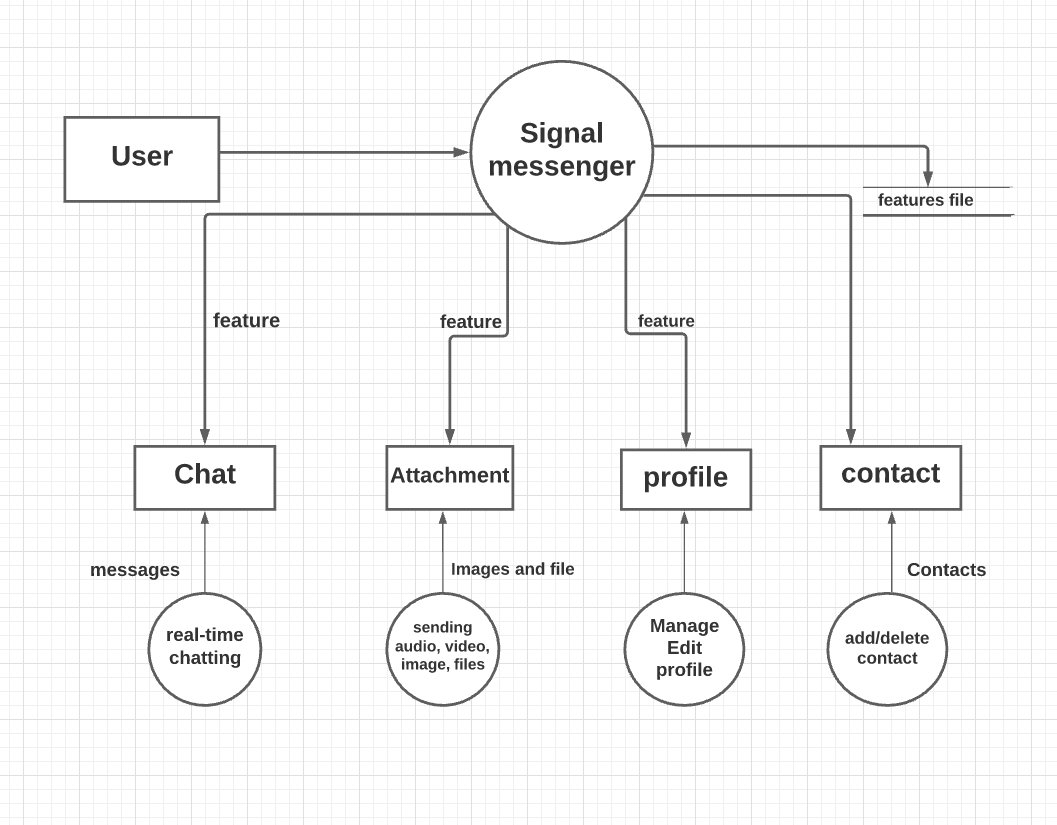
**4.1.5 Data flow diagram**

**Level 1 DFD : Message sending**



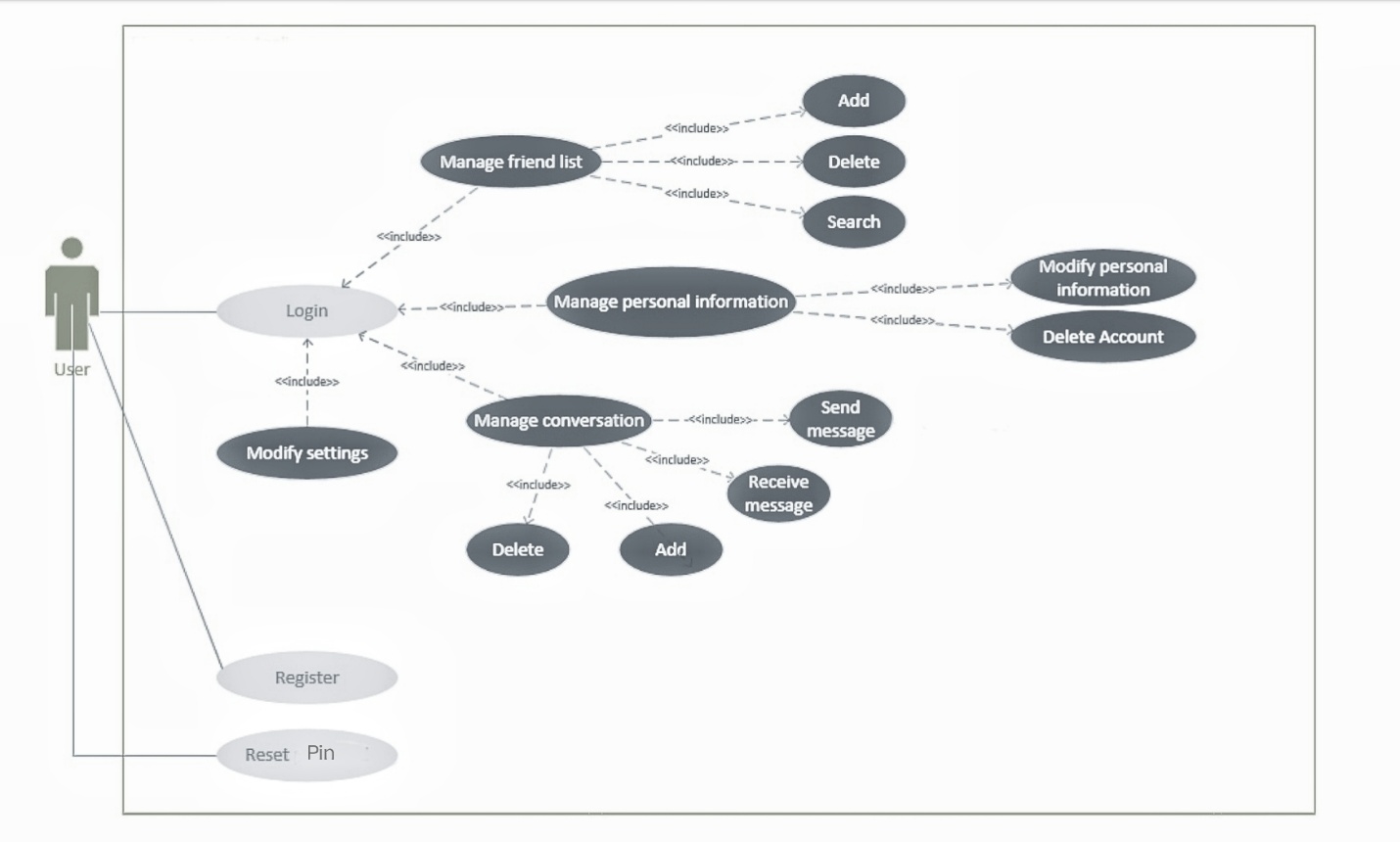
**Level 2 DFD : Encrypted Message**

**Level 3 DFD : features**



**4.1.6 Use case Diagram**

The Use Case Diagram represents the interaction between the actors and the system. In our project, we have only considered one actor, which is the end user.

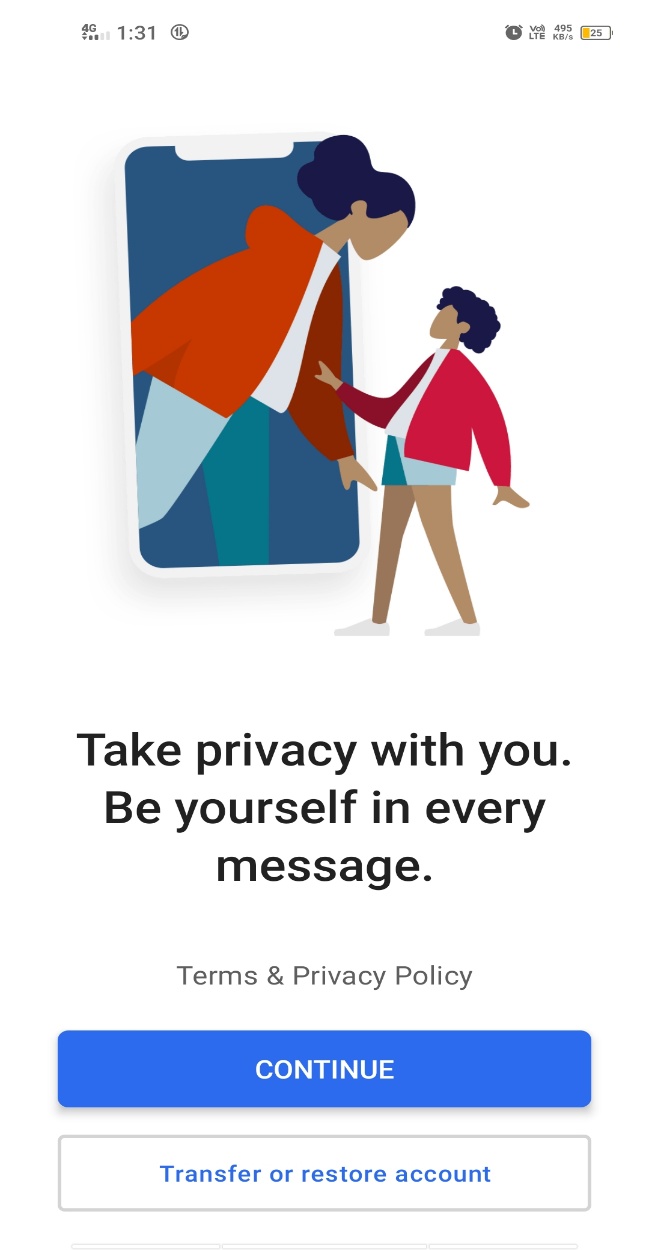


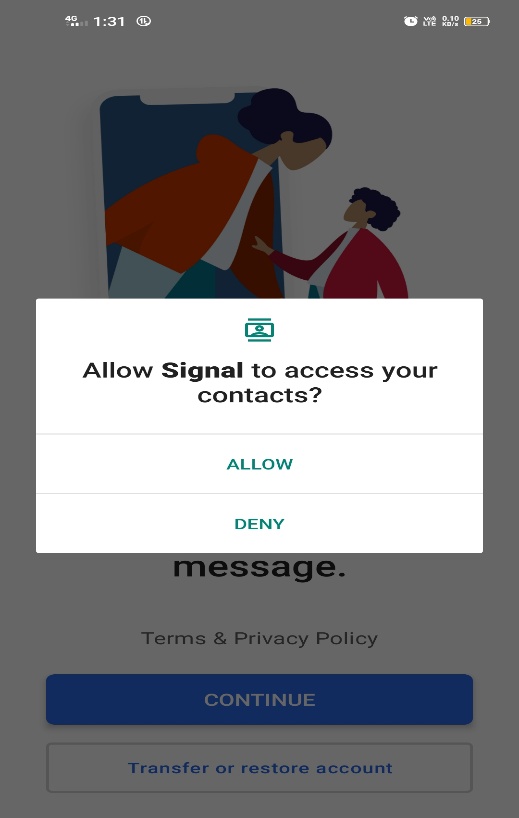
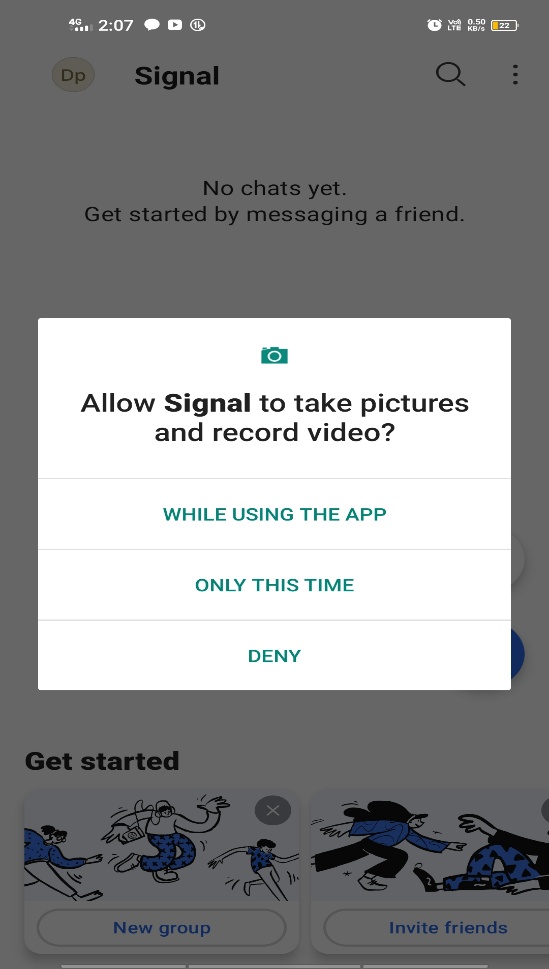
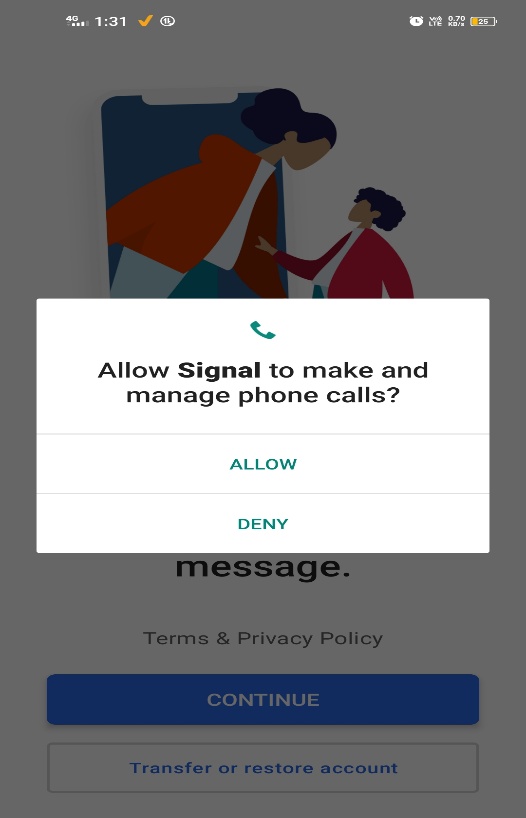
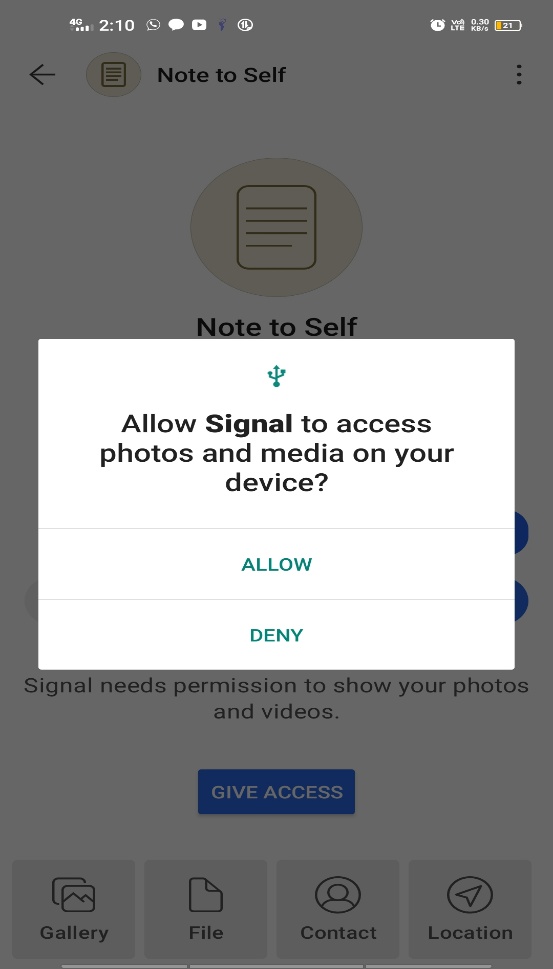
**4.2 physical design**

**4.2.1 User interface design**

# **First Impressions**

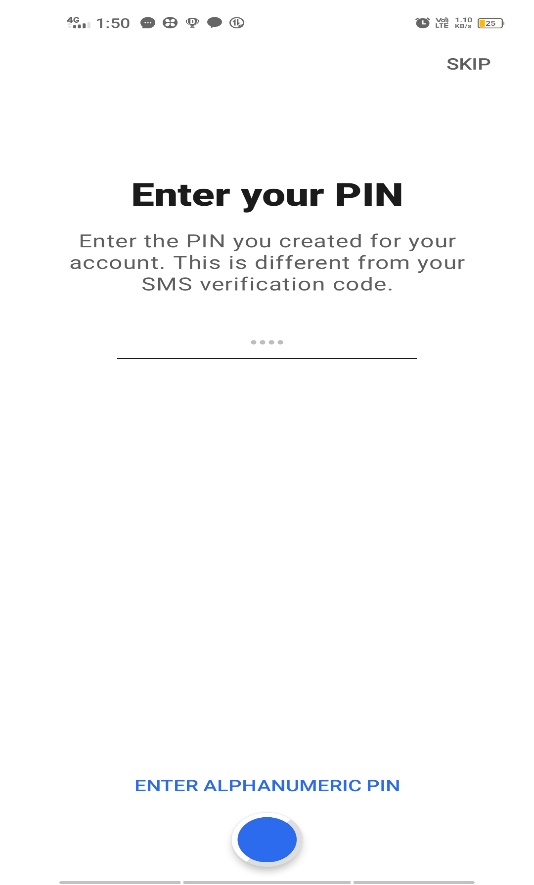
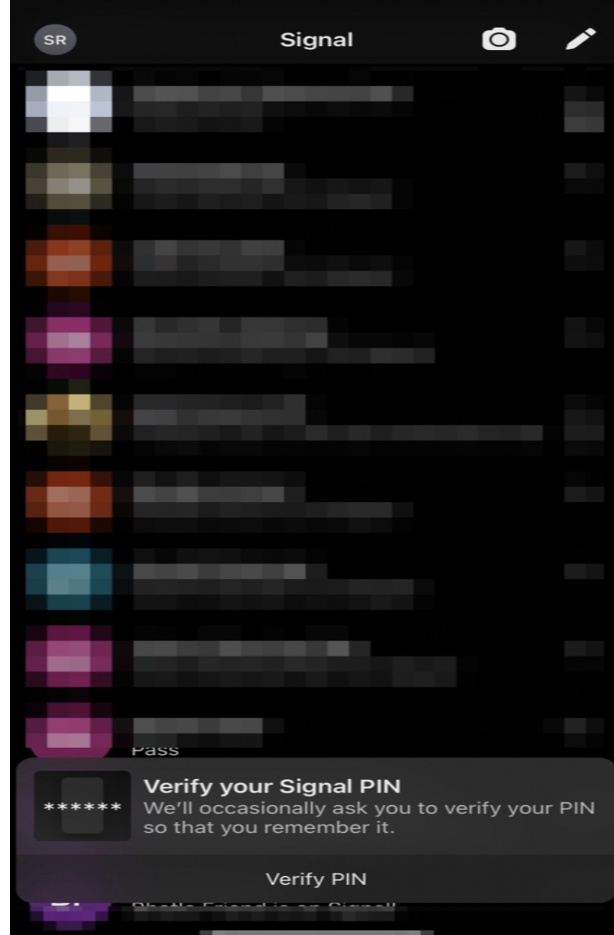
On the first impression, Signal seems quite similar to Whatsapp with most of the functions being the same — one-to-one messaging, groups, stickers, photos, file transfers, voice calls, and video calls.



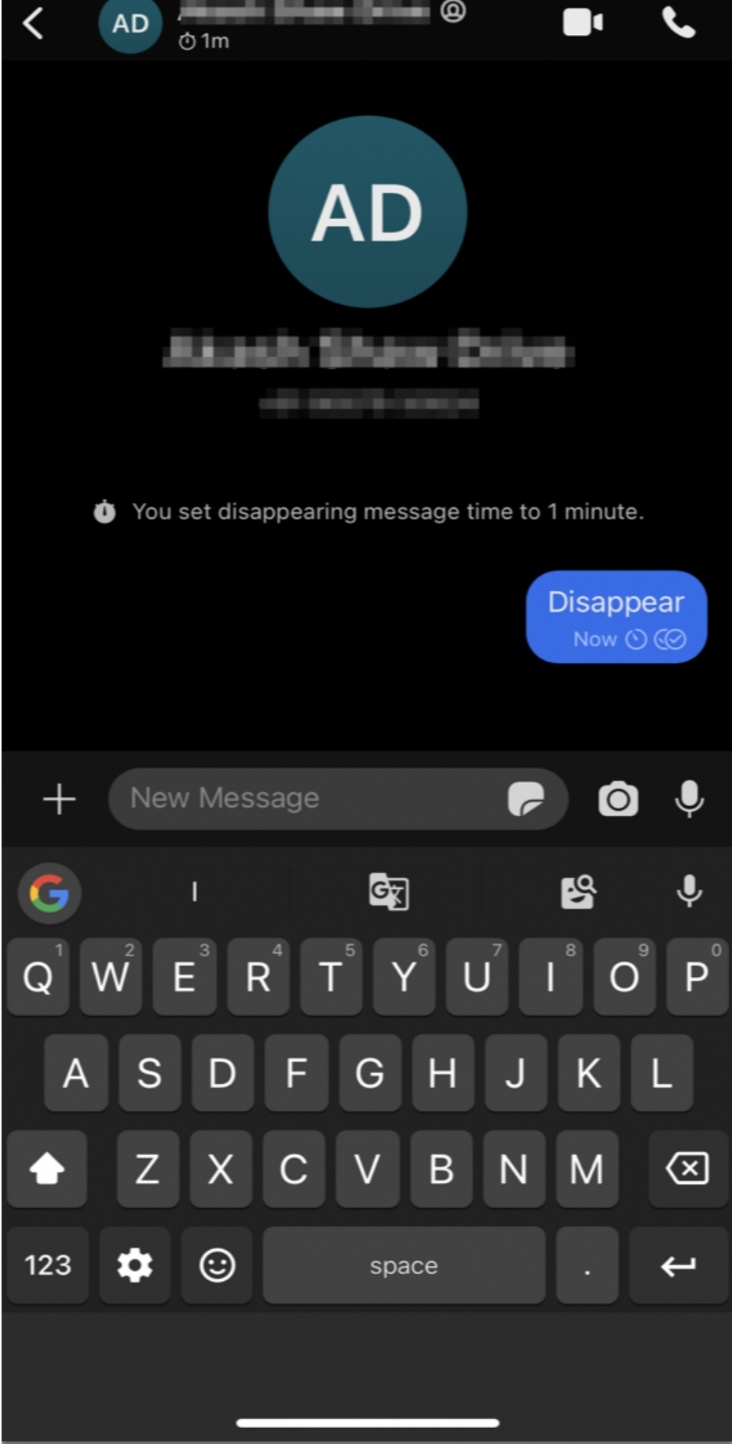
**Permissions**

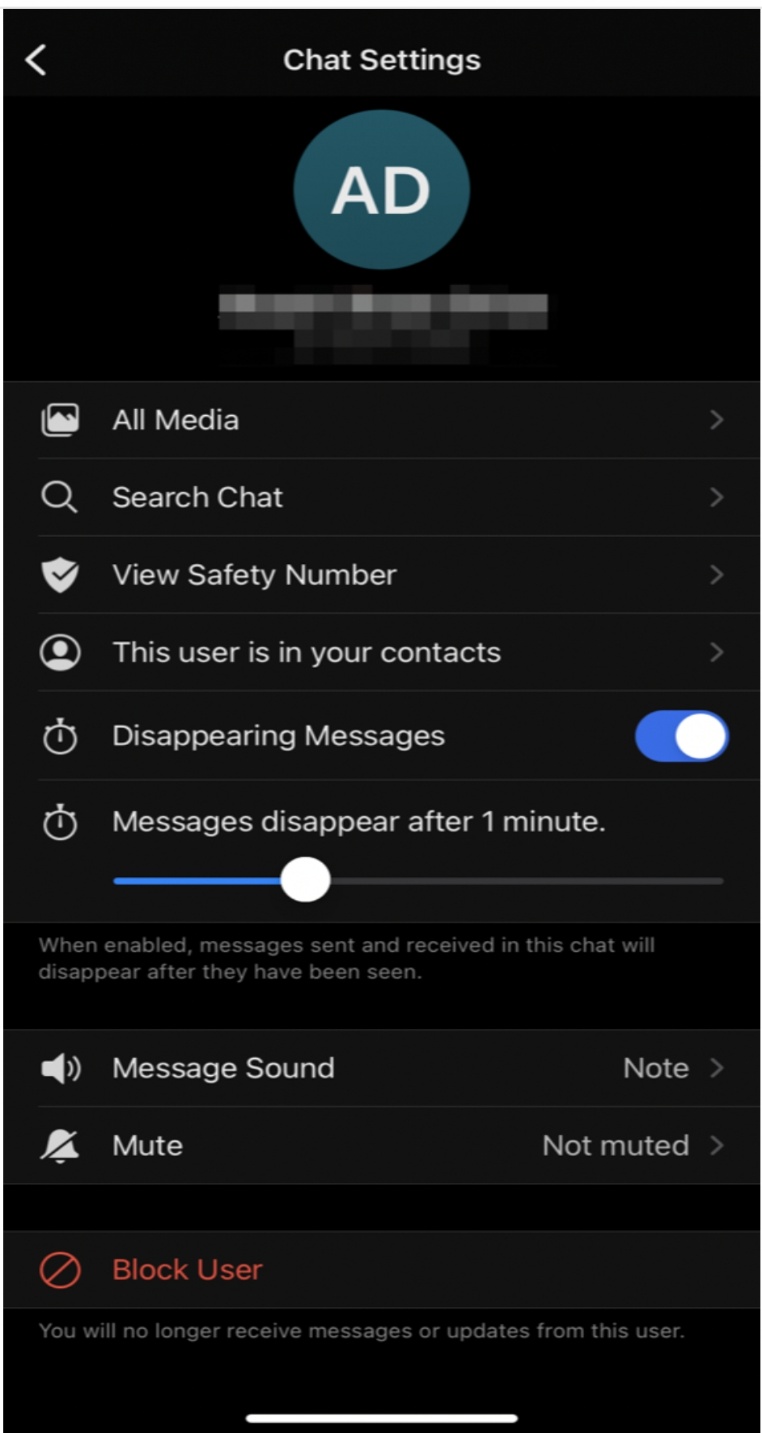
# **The PIN**

One of the key features which embraces Signal’s focus around data privacy and security is the PIN. The PIN is meant to support features like non-phone number based unique identifiers. If you ever lose your device or change it, you can recover your profile, settings and contacts simply by using this PIN. It also acts as a natural registration lock for your Signal account, preventing any other person from being able to enter and misuse your Signal account. In the future, the PIN feature will also enable new features the Signal community has been requesting for such as addressing without using phone numbers and chatting with contacts not saved in the address book.

After you set your PIN for the first time, Signal will remind you to verify your PIN after predefined time internals: 12 Hours, 1 Day, 3 Days, 7 Days and lastly after 14 Days.

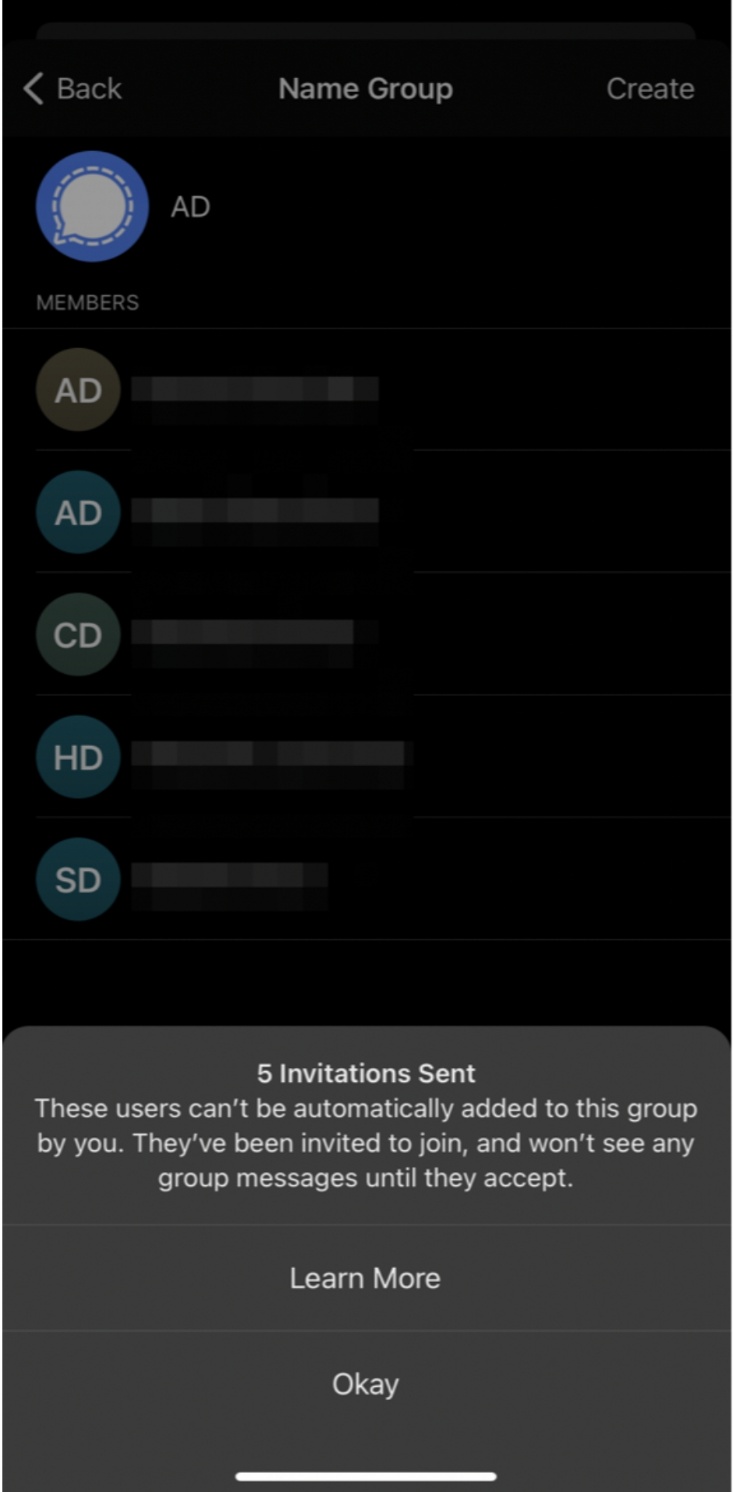
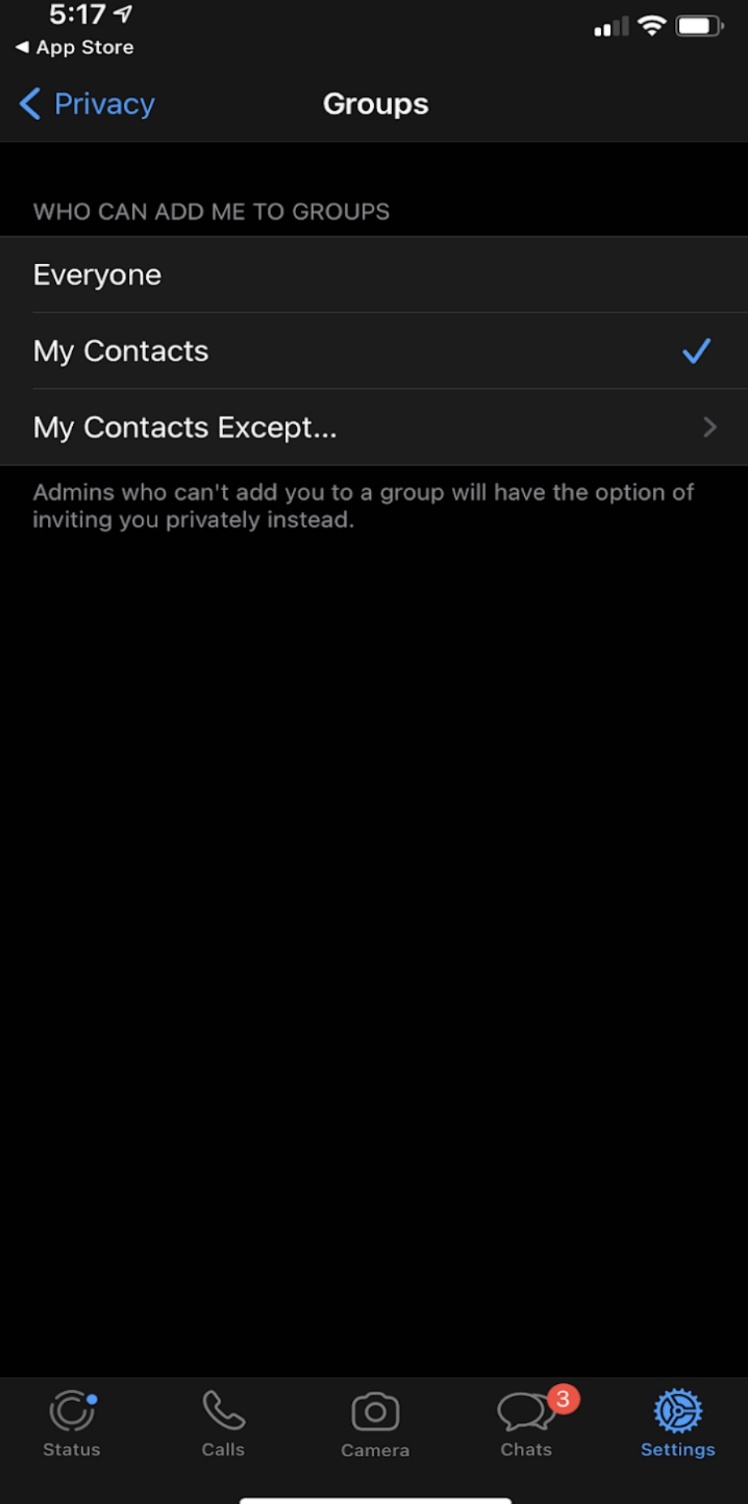
# **Disappearing Messages**

Signal solves the fear of messages being seen by other who takes your phone by giving you the flexibility to decide whether you want your messages to disappear or not at an individual contact level, letting you decide the contacts for whom you want to keep the disappearing messages feature ON.



**Groups are designed to be less intrusive**

Respecting your privacy seems to be at the heart of every feature on Signal. Most WhatsApp users hate it when people add them to random Whatsapp groups without their seeking permission.

 . On Signal, if you’re adding members to a group , you will have to seek their consent by waiting for them to accept your invitation.

**4.3 Architecture**

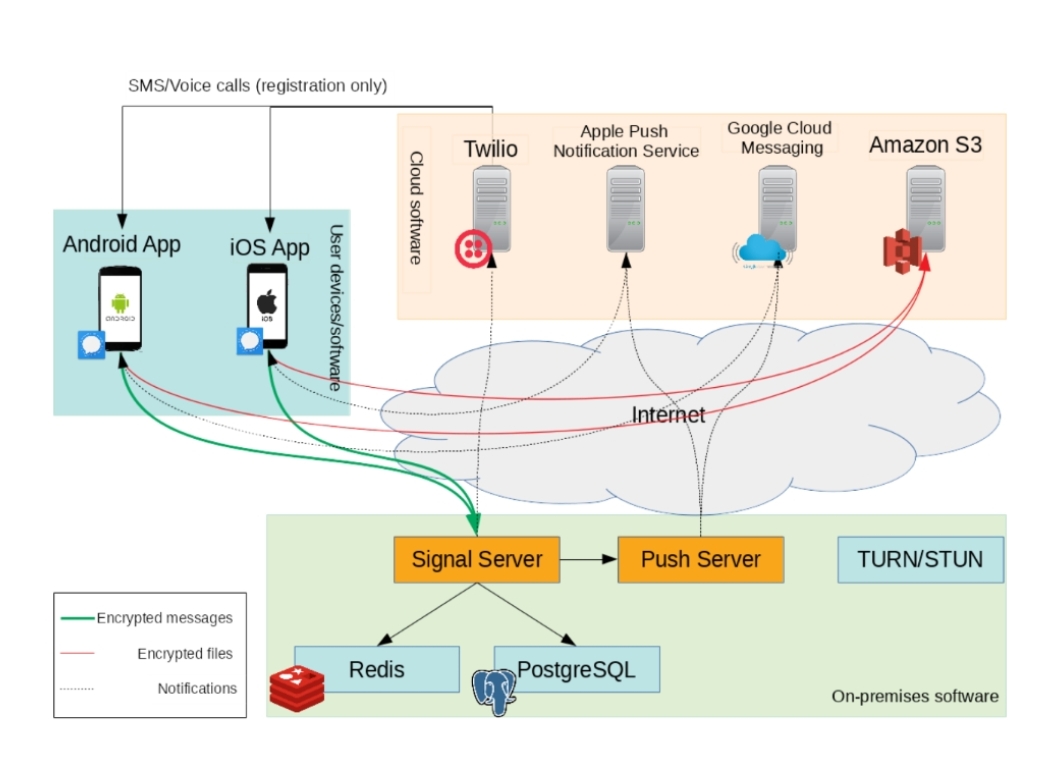
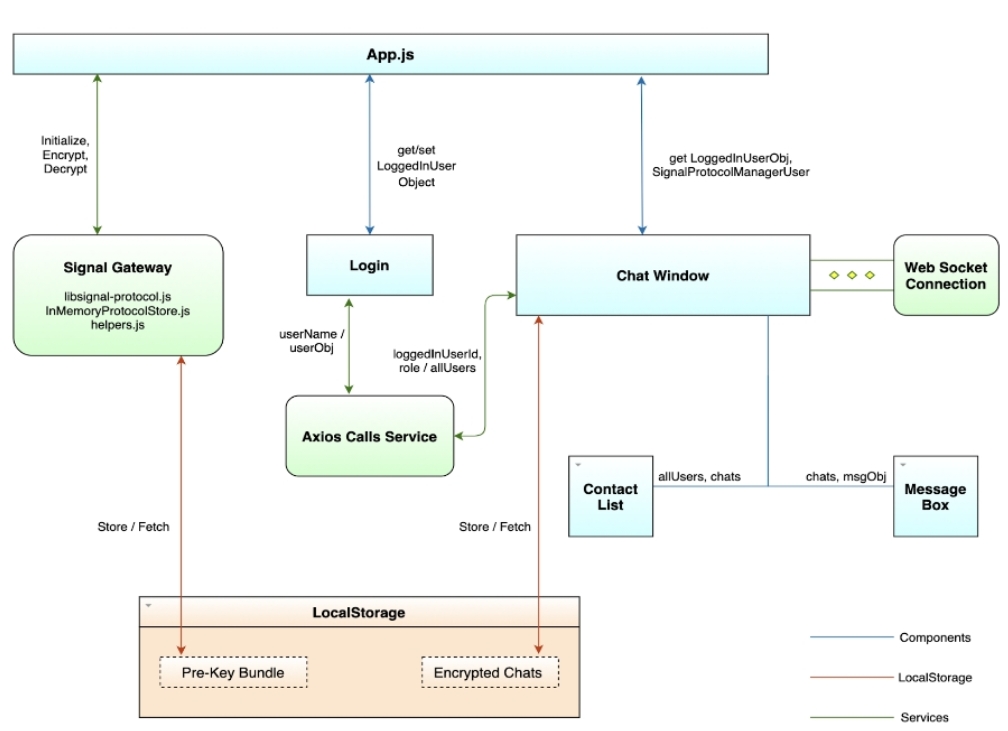
**4.3.1 Architecture of signal messenger :**

Fig. Architecture of Signal messenger

**4.3.2 Low level diagram of server-client(web browser)**



**4.3.3 High level diagram client-server**

