Software Requirements Specification

1. Introduction

1.1 Purpose

This document aims to build a messaging application.
The system will allow users to create separate chats and channels for any count of participants and send real-time messages between users, in chats or channels, with the possibility to send files, images, videos, and create polls.

1.2 Document conventions

API	Application Programming Interface
DB	Database
ER	Entity Relationship
UML	Unified Modelling Language

1.3 Intended audience and reading suggestions

The application will be useful for all Windows users to communicate with each other.
The intended audience is 12 years old English-speaking audience but isn’t limited to it.
In perspective, localization can be introduced to the project to make the audience broader.

1.4 Project Scope

The project is a desktop 3-tier application, with PostgreSQL for storing group messages and user data.
The purpose of this system is scalability, convenience, speed, and reliability.
The desktop interface should be intuitively understandable and comfortable to end users.

2. Overall Description

2.1 Product Perspective

The messaging app offers such features:

- **Messaging.**
  The system allows communication with other users directly or within groups or even with a broad audience with chats or channels in real time.
- **Chats and channels**. The key difference between chat and channel is the structure. The chat is just a group messaging space, while channels are blogs with discussion threads.
- **The flexibility of chat/channel rights**.
  The system allows configuring permissions sets for chat/channel participants, such as unauthorized content view, creating polls, sending multimedia and files, filter settings, mutes, and blacklists.

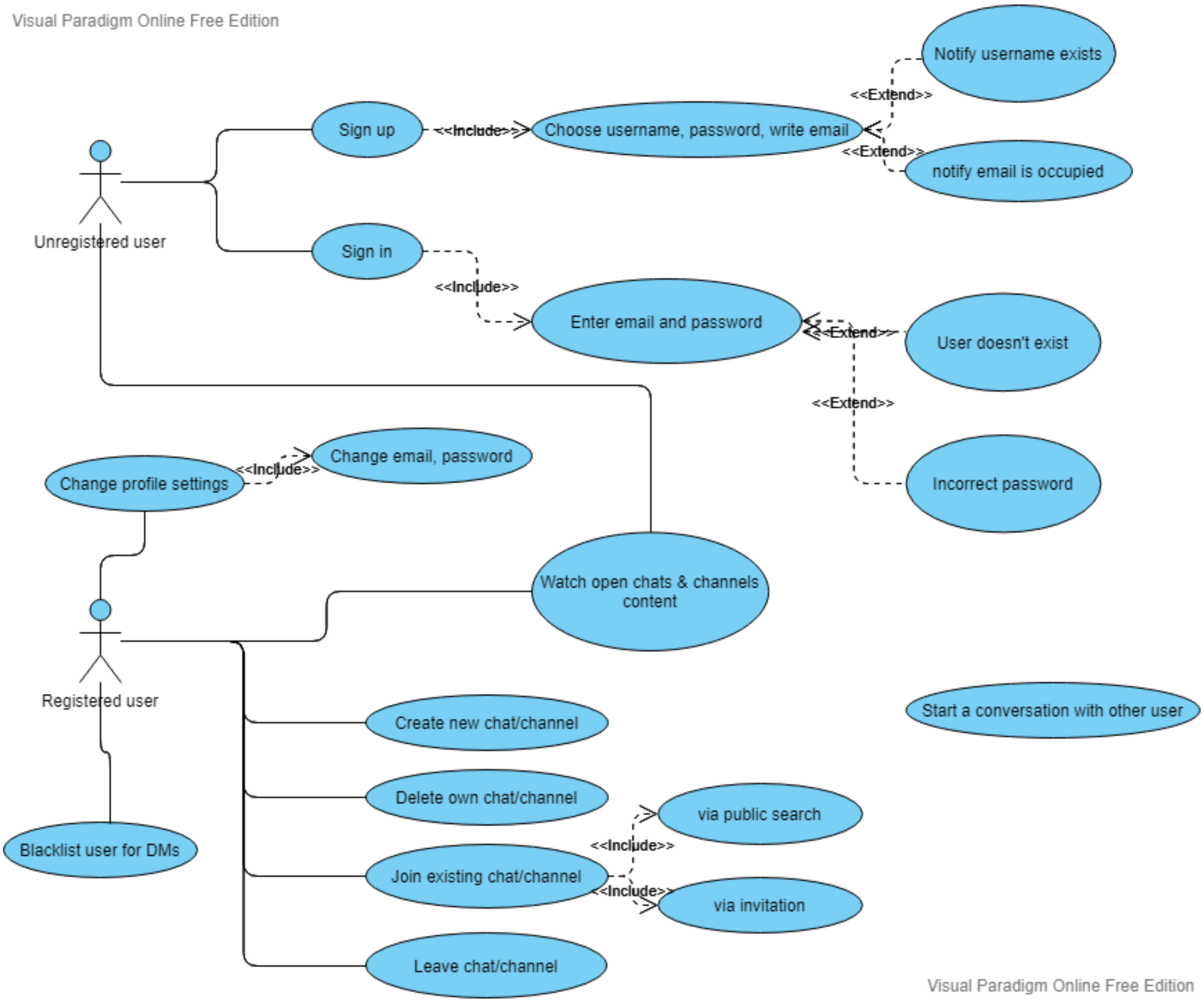
2.2 Product features

- Unregistered users can create a new account. During this process, they have to choose a username and password and provide an email and might provide phone number.
    - When users choose a username that already exists, they should be notified about that and they have to choose another one.
    - When users choose a weak password, they should be warned about that.
    - When users complete the signup, the confirmation email comes to their inbox.
- Registered users can log in. To do so, they should enter their email and password
    - In case of entering the wrong email or password, they should be notified about it and will be asked to enter another email/password.
    - In case of users forget the password, they can reset it via email or phone.
    - In case of entering a valid email and password, they log in.
- Registered users can edit their profiles.
- Registered users can write direct messages to other users unless some side of the conversation blacklisted another.
- Registered users can join other chats and channels unless blacklisted
    - Users can search public chats in the search bar and join them
    - Users can join public chats via links
    - Users can be invited by other users
    - Users can invite others when it’s allowed
- All registered users can join private chats and channels
    - Users can be invited by admins or users with permission to invite
    - Users can request join via link if possible
- Both registered and unregistered users can watch the content of public chats without having joined chat unless it is banned by chat permissions
- All registered users can write messages in public or private chats unless it is banned by chat permissions

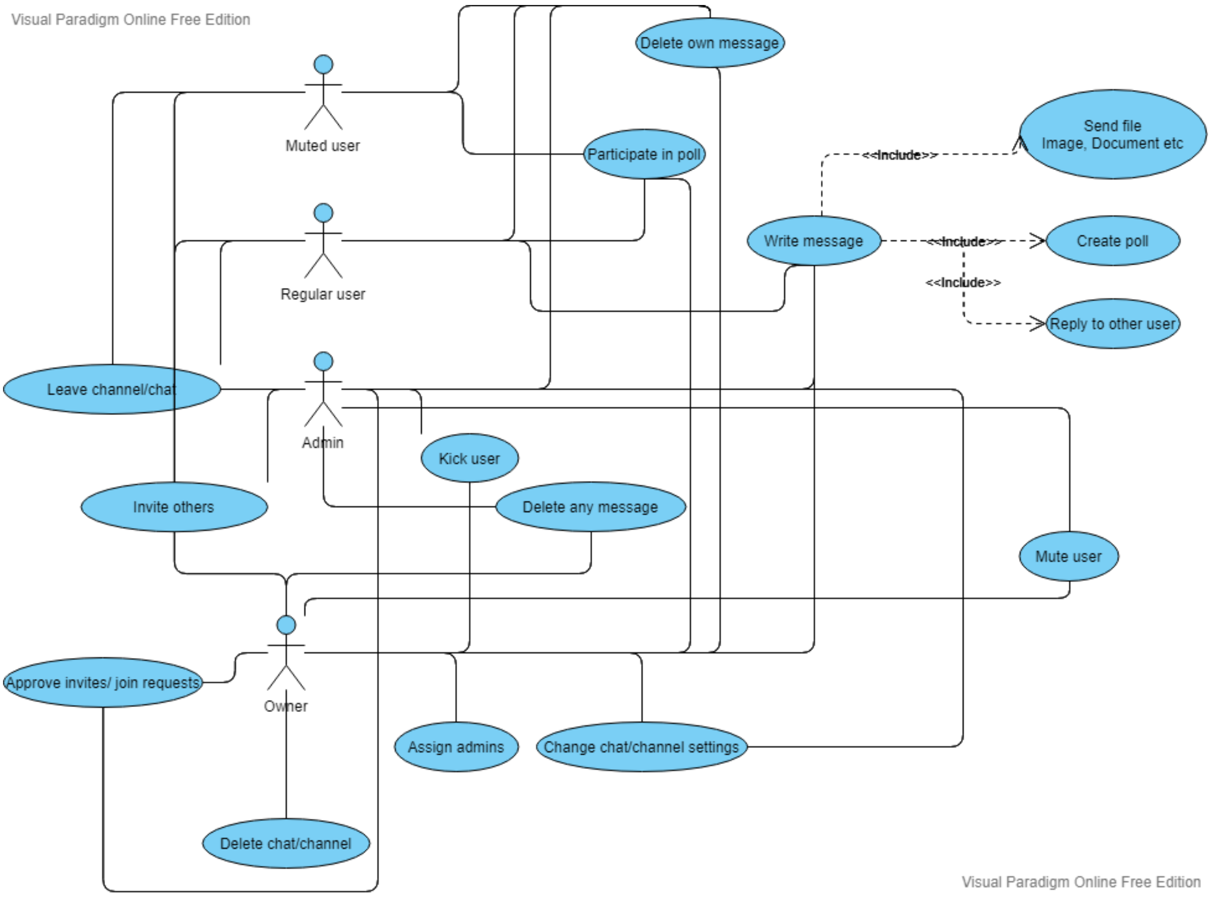
    - They can send messages in real-time or schedule it on some time (unless muted or blacklisted)
        - It can be the answer to another message
    - Chat participants can create polls if they have permission to do so
    - User can pin the message if allowed by chat permissions
    - User can send files and multimedia if allowed by chat permissions
- All registered users can keep their messages in draft
    - They might keep any count of drafted messages per chat
- All registered users can delete their own messages from chats/channels
    - Deleted messages should not be visible to other users, but might be visible in chat history if enabled
- All registered users can leave chats or channels unless they are their owner
    - In case of rejoining the chat, users do not restore special permissions given before leave, but restore special limitations such as message ban
    - Owners cannot leave their own chats / channels, but they can delete them
- Regular users can create chats and channels
    - The chat creator becames owner and has all permissions to this chat unless they are transferred to other user
    - The chat creator can assign admins for the chat/channel
    - The chat creator can configure global chat/channel settings:
        - If chat / channel is private or public
        - If public chat / channel content can be seen without joining it
        - Whether chat participants can invite other users
        - Define admins special rights and limitations
            - Mute or blacklist other users (and reverse these actions)
            - Invite other people, approve requests to join
            - Manage limits for regular users: writing and pinning messages, sending files and multimedia, creating polls, cooldown for messages
        - Define regular user special rights and limitations: writing and pinning messages, sending files and multimedia, creating polls, cooldown for messages
        - Saving history. That allows recently joined users to acknowledge the older discussion in chat/channel.
    - The chat/channel creator can mute or blacklist any user in the chat or remove
    - The chat/channel creator cannot have fewer rights than admins
- Chat/channel admins can do these things depending on their rights:
    - Mute or blacklist other users, and reverse these actions (but they cannot do it with themselves)
    - Invite other people, approve requests to join
    - Manage limits for regular users: writing and pinning messages, sending files and multimedia, creating polls, cooldown for messages
    - Assign other admins with no more rights than they have
- Chat/channel admins cannot have fewer rights than regular users and more rights than chat/channel owners
- Registered users can delete their chats
    - All the content of deleted chats/channels is deleted and can’t be restored
    - Other chat participants automatically leave the chat/channels

- Registered users can blacklist other users to prevent from receiving direct messages or undesired chat invitations
- Blacklisted users cannot:
    - Watch the chat content when it’s not allowed to unregistered users
    - Rejoin the chat / channel
    - Be invited to the chat/channel by other participants
- Muted chat/channel participants:
    - Cannot:
        - write messages
        - send files and multimedia
        - create polls
        - pin / unpin messages
    - But they can:
        - watch all the content of chat/channel
        - vote in polls
- Any chat/channels participant can vote in polls
    - The vote can be cancelled by the user if allowed by poll, or user is blacklisted from chat/channel
    - If chat/channel is public, the user’s vote is not revoked until blacklisted
    - Polls can be open or anonymous

2.3 User class and characteristics



(https://online.visual-paradigm.com/share.jsp?id=313839383633362d31)



(https://online.visual-paradigm.com/share.jsp?id=313839383633362d32)

2.4 Operating environment

- Database: PostgreSQL
- Server: ADO.NET + Entity Framework
- Client: WPF
- CI/CD Environment: Circle CI

2.5 Dependencies

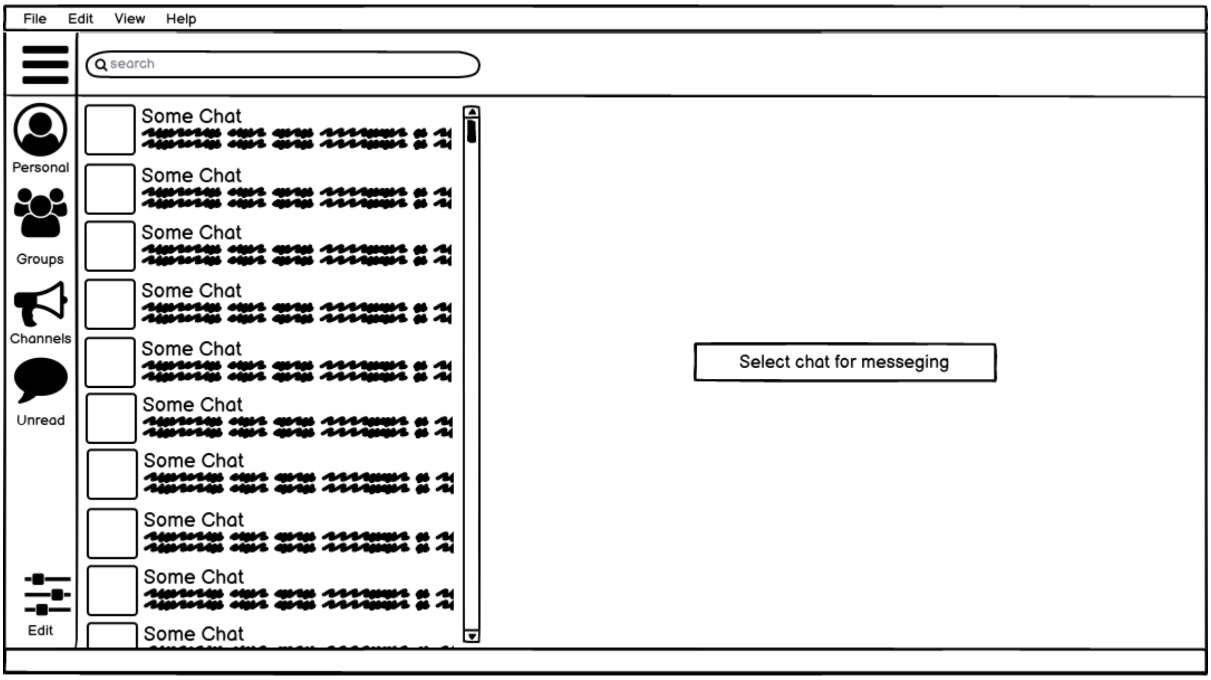
The service will be self-contained without significant dependencies on third-party vendors. The system will require a web-hosting environment to be always online and accessible for the end users.

3. External Interface Requirements

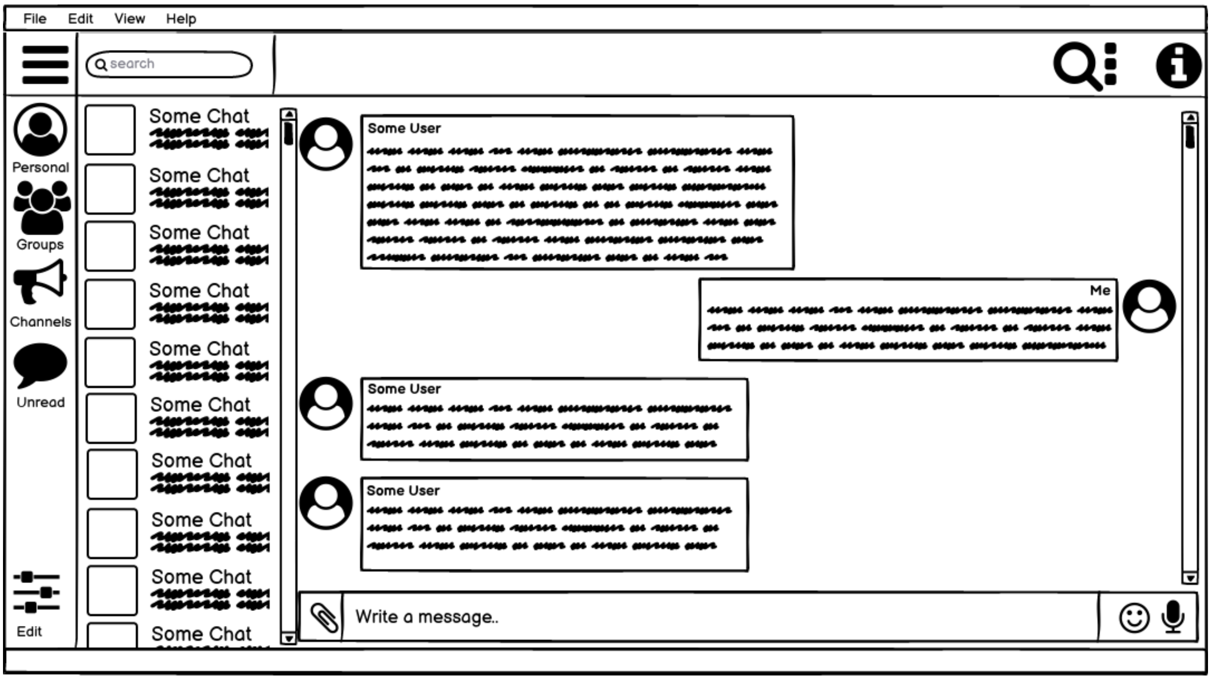
3.1 User Interfaces

Here are some mockups for user interfaces

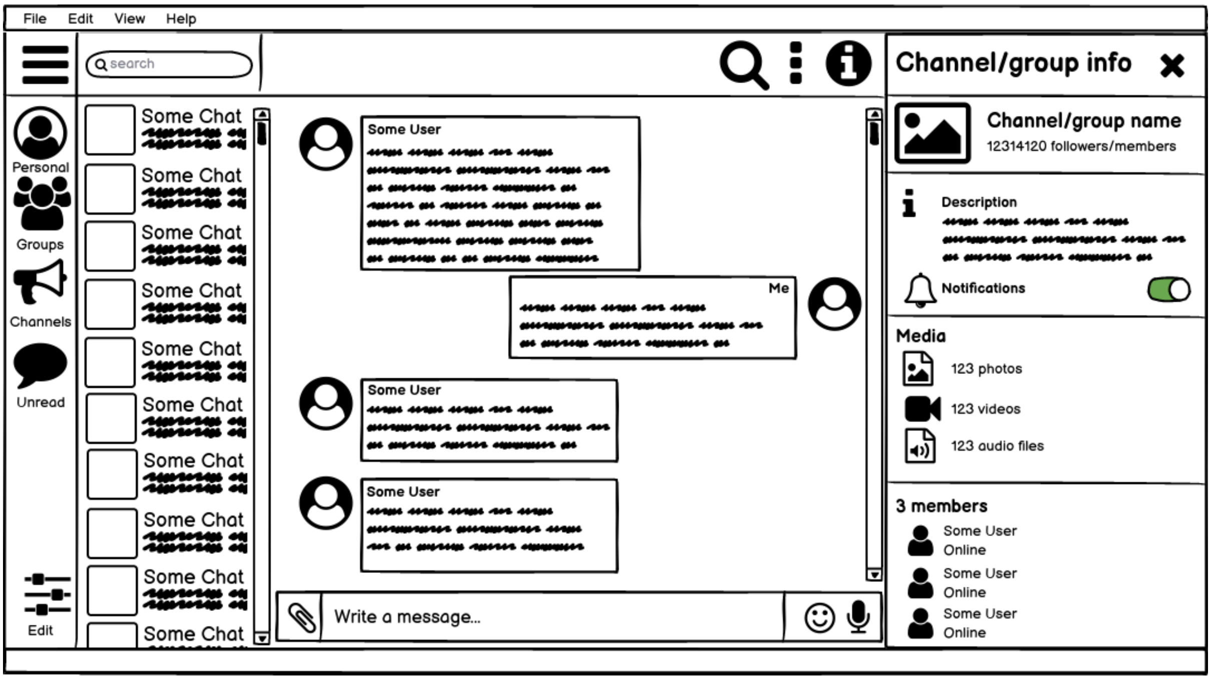
1) Main menu



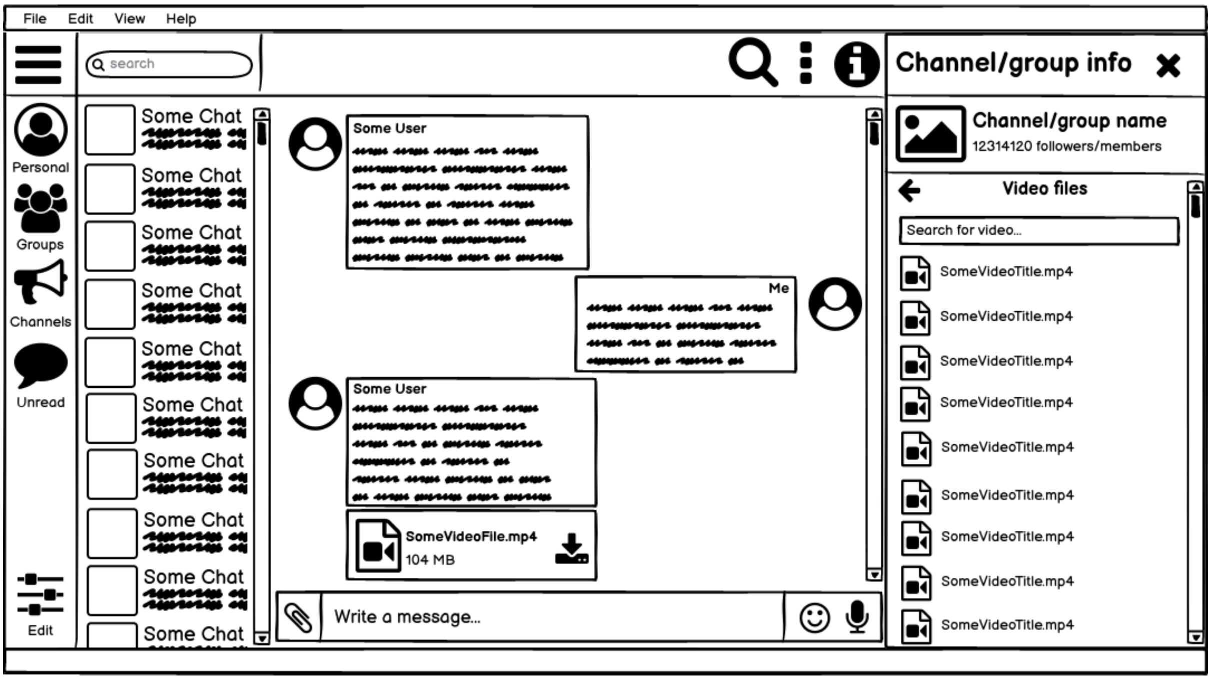
2) Chat layout



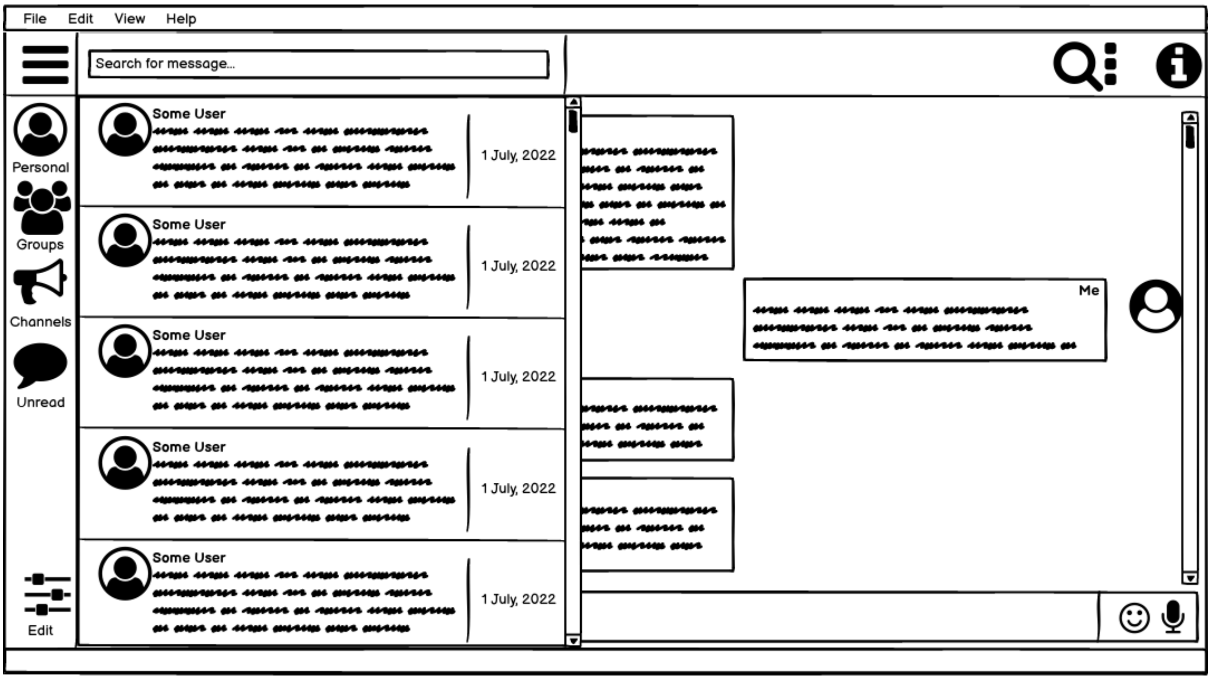
3) Channel/group info



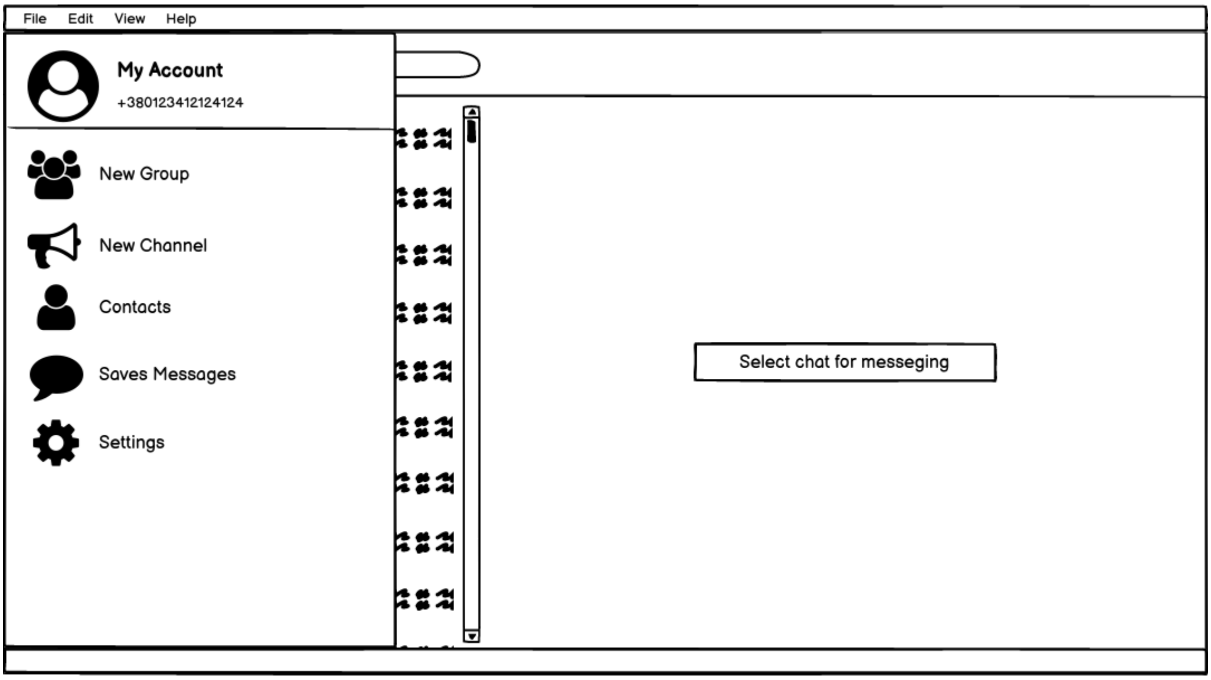
4) Channel/group media files list



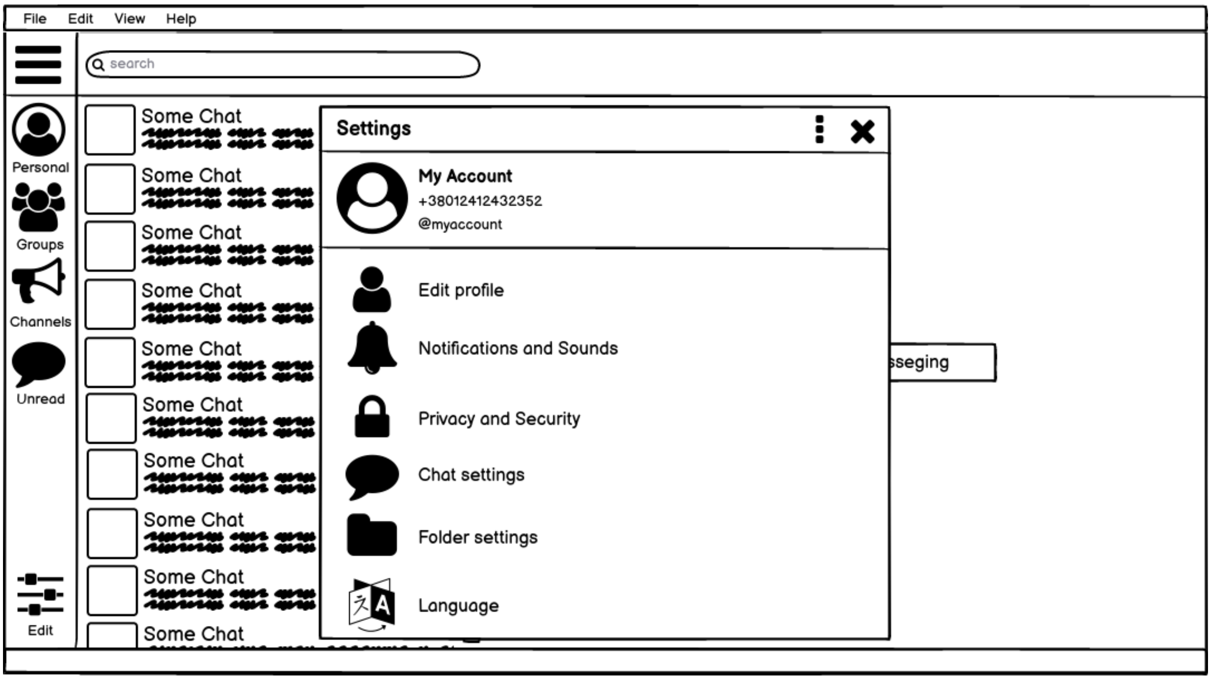
5) Channel/group message filtering



6) Burger menu



7) Settings



3.2 Hardware interfaces

Computer with Windows

3.3 Software Interfaces

Software used	Description
Operating system	Windows
Backend framework	ADO.NET + EF, the most suitable technologies for this task
Desktop framework	WPF, not the most convenient, but we are asked to use it
Database management system	PostgreSQL, has support for JsonB fields

3.4 Communication Interfaces

For the current stage of development, only Windows desktop application will be available for this project

4. Non-functional Requirements

4.1 Performance Requirements

The program should be able to quickly change chats without waiting, as well as download other users' data. The speed of sending and receiving a message may depend on the user's Internet speed, but in general, it should not be more than one second. Images and other media files should be compressed for better performance.

4.2 Safety Requirement

No personal information of the user, i.e. his password, login, and mobile number, can be known to other users through the program. However, the program cannot ensure the user's safety if he transfers this information to another person. The application database cannot be accessed by regular users.

4.3 Security Requirement

When logging in, you always need a password, which cannot be simple. Messages are encrypted when sent, so they cannot be "intercepted" by someone other than the addressee. The encryption method must be reliable so that fraudsters cannot decrypt the data. If desired, the user can add two-factor authentication to his account.

4.4 Software Quality Attributes

The software should be flexible to easily implement further features without total app redesign.