Text

Description automatically generated

**Activity based**

**Project Report on**

**Software Engineering**

**Project Module - III**

**Submitted to Vishwakarma University, Pune**

**Under the Initiative of**

**Contemporary Curriculum, Pedagogy, and Practice (C2P2)**

**By**

**Shravan Meshram**

**SRN No : 202101425**

**Roll No : 31**

**Div : E**

**Third Year Engineering**

**Department of Computer Engineering**

**Faculty of Science and Technology**

**Academic Year**

**2023-2024**

**Software Engineering: Project Module II**

**Project Name : UML Diagram for Telegram**

**Draw following UML diagram**

* **Communication Diagram**

A communication diagram for Telegram can provide a visual representation of how various components or actors interact within the Telegram messaging system. Here's a brief introduction to how you might structure such a diagram:

Title: Communication Diagram for Telegram

Description: The communication diagram illustrates the interactions between different components and actors within the Telegram messaging system. It outlines the flow of messages and interactions that occur when users send and receive messages through the Telegram application.

Components/Actors:

1. User: Represents individuals using the Telegram application to send and receive messages.
2. Telegram Client: The client-side application running on users' devices (e.g., smartphones, computers).
3. Telegram Server: The central server responsible for routing, storing, and managing messages.
4. Recipient Device: Represents the device(s) receiving messages sent via Telegram.

Key Interactions:

1. User ↔ Telegram Client:

* The user interacts with the Telegram client to compose and send messages.
* The Telegram client displays messages received from other users.

1. Telegram Client ↔ Telegram Server:

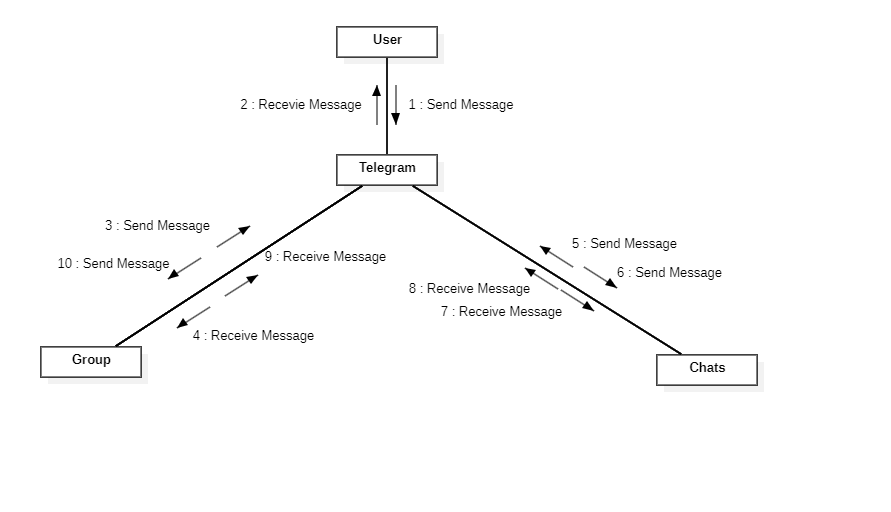
* The Telegram client communicates with the Telegram server to send outgoing messages.
* The Telegram server receives incoming messages from clients and processes them for delivery.

1. Telegram Server ↔ Recipient Device:

* The Telegram server delivers incoming messages to the recipient device(s).
* The recipient device displays received messages to the user.

Message Flow:

1. User composes a message in the Telegram client.
2. Telegram client sends the message to the Telegram server for delivery.
3. Telegram server receives and processes the message, determining the recipient(s).
4. Telegram server delivers the message to the recipient device(s).
5. Recipient device receives and displays the message to the user.



* **Activity Diagram**

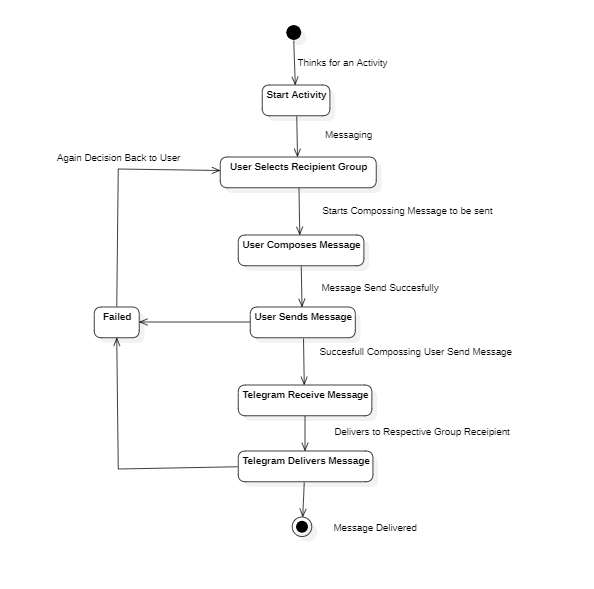
The activity diagram depicts the various activities and actions involved in sending and receiving messages through the Telegram application. It outlines the steps users take to compose, send, receive, and manage messages within the Telegram interface.

Activities/Actions:

1. Start: Represents the initiation of the messaging process.
2. Compose Message: Users compose a new message by entering text, attaching media, or selecting recipients.
3. Send Message: Users send the composed message to selected recipients.
4. Receive Message: Users receive incoming messages from other users.
5. Read Message: Users read and view the contents of received messages.
6. Reply to Message: Users compose and send replies to received messages.
7. Forward Message: Users forward received messages to other contacts or groups.
8. Delete Message: Users delete unwanted or outdated messages from their chat history.
9. Manage Contacts: Users manage their contact list, adding, removing, or blocking contacts as needed.
10. Logout: Users log out of the Telegram application, ending the messaging session.

Flow:

1. Start: The messaging process begins when the user launches the Telegram application.
2. Compose Message:
3. Users initiate the composition of a new message.
4. They enter text, attach media, select recipients, and customize message settings.
5. Send Message: Users send the composed message to selected recipients.
6. Receive Message: Users receive incoming messages from other users.
7. Read Message: Users view the contents of received messages.
8. They can read text messages, view attached media, or listen to voice messages.
9. Reply to Message: Users compose and send replies to received messages.
10. Forward Message: Users forward received messages to other contacts or groups.
11. Delete Message: Users delete unwanted or outdated messages from their chat history.
12. Manage Contacts: Users manage their contact list, adding, removing, or blocking contacts as needed.
13. Logout: Users log out of the Telegram application, ending the messaging session.



* **Timeline/Timing Diagram**

The timeline diagram presents a chronological sequence of key events and developments in the history of the Telegram messaging platform. It illustrates the evolution of features, technological advancements, user milestones, and other noteworthy occurrences.

Timeline Events:

1. Founding of Telegram: Pavel Durov and his brother Nikolai Durov found Telegram Messenger LLP.
2. Initial Release: Telegram is launched as a messaging app for iOS and Android platforms.
3. End-to-End Encryption: Telegram introduces end-to-end encryption for Secret Chats.
4. Voice Calling: Telegram introduces voice calling functionality.
5. Telegram Bots: Telegram introduces support for bots on the platform, enabling automated interactions and services.
6. Video Calling: Telegram adds video calling functionality to the app.
7. Group Video Calls: Telegram introduces group video calls, allowing multiple users to participate in video conversations.
8. Channels: Telegram introduces Channels, allowing users to broadcast messages to large audiences.
9. Voice Chats: Telegram introduces Voice Chats in groups and channels, enabling live voice conversations.
10. Payments: Telegram introduces payment features, allowing users to make purchases within the app.
11. End-to-End Encrypted Cloud Storage: Telegram announces the launch of End-to-End Encrypted Cloud Storage for all file types.
12. Milestone User Base: Telegram surpasses a milestone number of active users (e.g., 500 million).

Timeline Representation:

* The timeline diagram presents a linear timeline with labeled intervals representing years or specific time periods.
* Each timeline event is depicted as a labeled marker on the timeline, indicating the occurrence of significant milestones or developments.
* Events may be accompanied by additional details, such as brief descriptions or icons representing the nature of the event (e.g., messaging icon for feature updates, globe icon for user milestones).

