Telegram Clone App - Software Architecture (MVC)

This document describes the software architecture of a Telegram Clone app built using Flutter and Firebase,

following the MVC (Model-View-Controller) pattern.

MODEL

- UserModel: stores user ID, email, and metadata.
- MessageModel: for public chats.
- PrivateMessageModel: includes message content, sender, receiver, timestamp, status (sent/seen/typing).

Data is stored in Firebase Firestore:

- users (collection)
- public_chats (collection)
- private_chats (collection)
  - messages (subcollection)
  - metadata fields: lastMessage, lastMessageTime, participants, seen, typing

----

VIEW

Flutter Screens (UI Layer):

- LoginScreen

- RegisterScreen

- HomeScreen

- PublicChatScreen

- PrivateChatListScreen

- PrivateChatScreen

- UserListScreen (for starting new private chats)

CONTROLLER

Handles logic, state, and business flow:

- AuthController (login, register, logout)

- PublicChatController (handles public messaging)

- PrivateChatController (handles all 1-to-1 messaging logic, emoji, typing, seen status)

INTERACTION FLOW

View -> calls methods in -> Controller

Controller -> updates/fetches data from -> Model (via Firebase)

Model <-> persists data in -> Firebase (Firestore, Auth)