



COLLEGE CODE : 9504

COLLEGE NAME : DR.G.U.POPE COLLEGE OF ENGINEERING

DEPARTMENT : COMPUTER SCIENCE AND ENGINEERING

STUDENT NM ID : 7B485AC930EF52ABEEB4BB342F4C9245

ROLL NO : 46

DATE : 15.09.2025

Completed the Phase-01

PROJECT NAME : IBM-FE- CHAT APPLICATION UI

SUBMITTED BY,

**THANUSHA R
7708646205**

PROBLEM UNDERSTANDING & REQUIREMENT

PROBLEM STATEMENT

In an increasingly digital and remote world, the demand for real-time, intuitive, and reliable communication is more critical than ever. Many existing chat applications overwhelm users with bloated features, suffer from inconsistent performance, or lack a clean and accessible interface. These issues contribute to fragmented conversations, user frustration, and reduced productivity across teams and communities.

Our solution is a modern chat application designed with simplicity and speed at its core. It emphasizes a clean, responsive, and user-centric interface that fosters seamless interaction between individuals and groups. By eliminating unnecessary distractions and minimizing the learning curve, our platform empowers users to focus on meaningful communication. With a visually appealing design and optimized performance, it bridges the gap between functionality and user experience—making connection effortless, efficient, and enjoyable.

USERS & STAKEHOLDERS

ROLE	DESCRIPTION	PRIMARY INTEREST
End-user	The primary individual using the app to communicate with others.	An intuitive, reliable, and visually appealing interface for messaging.
Group Admin	A user who creates and manages group chats.	Controls for adding/removing members and setting group preferences.
UI /UX designer	Creates the visual and interactive design of the application.	Usability, accessibility, and a consistent, modern design system.
Frontend Developer	Implements the application's user interface.	Clear requirements, component definitions, and well-defined API endpoints.
Backend Developer	Develops the server, database, and app logic.	A UI design that aligns with achievable API contracts and database queries.
Product Owner	Defines the product vision and manages backlog.	Delivering a Minimum Viable Product (MVP) that validates core user needs.

USER STORIES

ID	USER STORY	PRIORITY
US-01	As a user, I want to see a list of my recent conversations so that I can quickly continue a chat.	High
US-02	As a user, I want to search for other users by their username so that I can start a new conversation.	High
US-03	As a user, I want to send and receive text messages in real-time so that the conversation feels fluid.	High
US-04	As a user, I want to see message timestamps and read receipts so I know the status of my communication.	Medium
US-05	As a user, I want to create a group chat, name it, and add participants to communicate with multiple people.	High
US-06	As a Group Admin, I want to add or remove participants from a group I manage.	Medium
US-07	As a user, I want to receive visual notifications for new messages so I am alerted to new activity.	Medium
US-08	As a user, I want a responsive interface so I can use the app on my desktop or mobile phone	High

MVP Features

1. User Authentication UI: Login and Registration screens with a mock "Forgot Password" flow.
2. Conversations List View: A sidebar displaying all chats with participant names, last message preview, timestamp, and unread count.
- 3.1-on-1 Chat Interface: A central message panel with a history of messages, distinct message bubbles, and a message input area.
4. Group Chat Creation & Interface: Ability to create a group via a modal and view group conversations.
5. Real-Time UI Simulation: The frontend is structured to handle real-time data, simulated for MVP with static data or timers.
6. Responsive Layout: A adaptive design that provides an optimal experience on desktop and mobile devices.

Wireframes & API Endpoints

Wireframe Overview

- ✧ Login/Signup Screen: Simple form with credentials fields.
- ✧ Main Application View: Two-panel layout (for desktop).
- ✧ Panel A (Sidebar): List of conversation items with a "New Chat" button.
- ✧ Panel B (Main Area): Dynamic content area. Shows either a welcome placeholder or the selected conversation view (header, message history, input area).
- ✧ Create Group Modal: Pop-over with group name field and a searchable list of contacts.

API Endpoint List

The UI will be designed to consume the following RESTful API endpoints. Data will be mocked for initial development.

Method	Endpoint	Description
POST	/auth/login	Authenticates user and returns token.
GET	/conversations	Retrieves the list of conversationsfor the authenticated user.
GET	/conversations/:id/messages	Retrieves the message history for a specific conversation.
POST	/conversations/:id/messages	Sends a new message to a conversation.
POST	/conversations/group	Creates a new group conversation.
GET	/users?search=term	Searches for users by username.

Acceptance Criteria

Feature: User Login

- AC1: User can input credentials and submit the form.
- AC2: Upon successful mock login, the user is redirected to the conversation list view.
- AC3: Invalid credentials trigger a user-friendly error message.

Feature: Conversations List

- AC4: The list is populated with data from the /conversations endpoint upon page load.
- AC5: Each list item displays the conversation name, last message preview, timestamp, and an unread count badge
(if count > 0).

Feature: Message Interface

- AC6: Selecting a conversation loads its message history from /conversations/:id/messages.
- AC7: Sent and received messages have distinct visual styling (e.g., aligned right/left, different colors).
- AC8: Sending a message triggers a POST request and the message is optimistically displayed in the UI.

Feature: Group Creation

- AC9: Clicking "New Group" opens a modal with a form.
- AC10: Submitting the form with a name and members triggers a POST to /conversations/group.

Feature: Responsive Design

- AC11: On viewports larger than 1024px, the application uses a side-by-side two-panel layout.
- AC12: On viewports smaller than 768px, the conversations list and chat view are separate, full-screen pages.