

1. Problem Statement & objectives

Existing chat applications face issues like **security vulnerabilities, poor user experience, limited cross-platform support, unreliable message delivery, and lack of advanced features**. Users need a **secure, efficient, and feature-rich** messaging solution for seamless communication.

Project Objectives

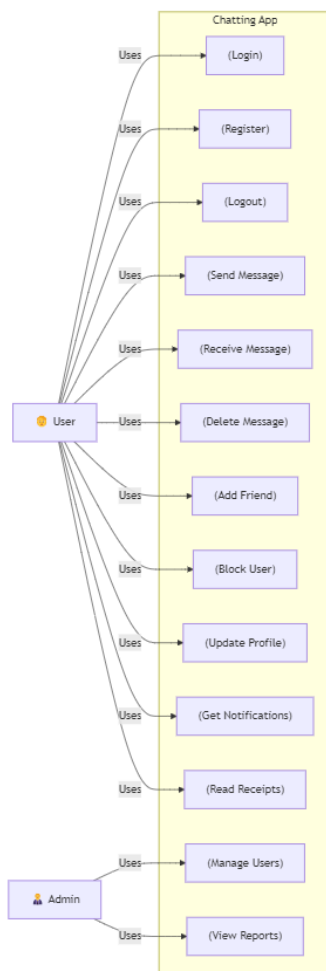
1. **Enhance Security & Privacy** – Implement end-to-end encryption and two-factor authentication.
2. **Improve User Experience** – Offer a clean UI, real-time messaging, and customization options.
3. **Expand Communication Features** – Enable text, voice, video calls, and media sharing.
4. **Integrate Smart Features** – AI chatbots, message scheduling, and reminders.

Use case Diagram:

Actors: User & Administrator

Interactions: user has ability to login, register, send messages, delete messages, add friends, block user, updates profile ... etc.

Admin has ability to manage user and extract data to be analyzed.



Functional requirements:

- >User to user chatting
- >Group chatting up to 256 users
- >Image w emoji sharing
- >File Sharing
- >Message deletion/editing

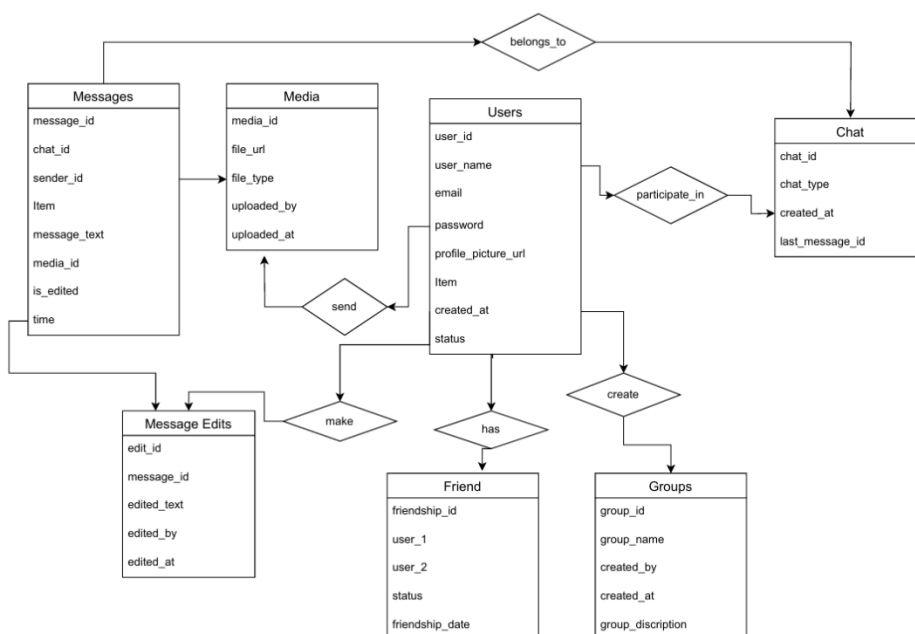
Non-functional requirement:

- >End-to-end encryption
- >Response-time: 2sec
- >Usability: Clean Web UI
- >Reliability: Server up time 3-hours, Message saving

Software architecture: Client-server architecture

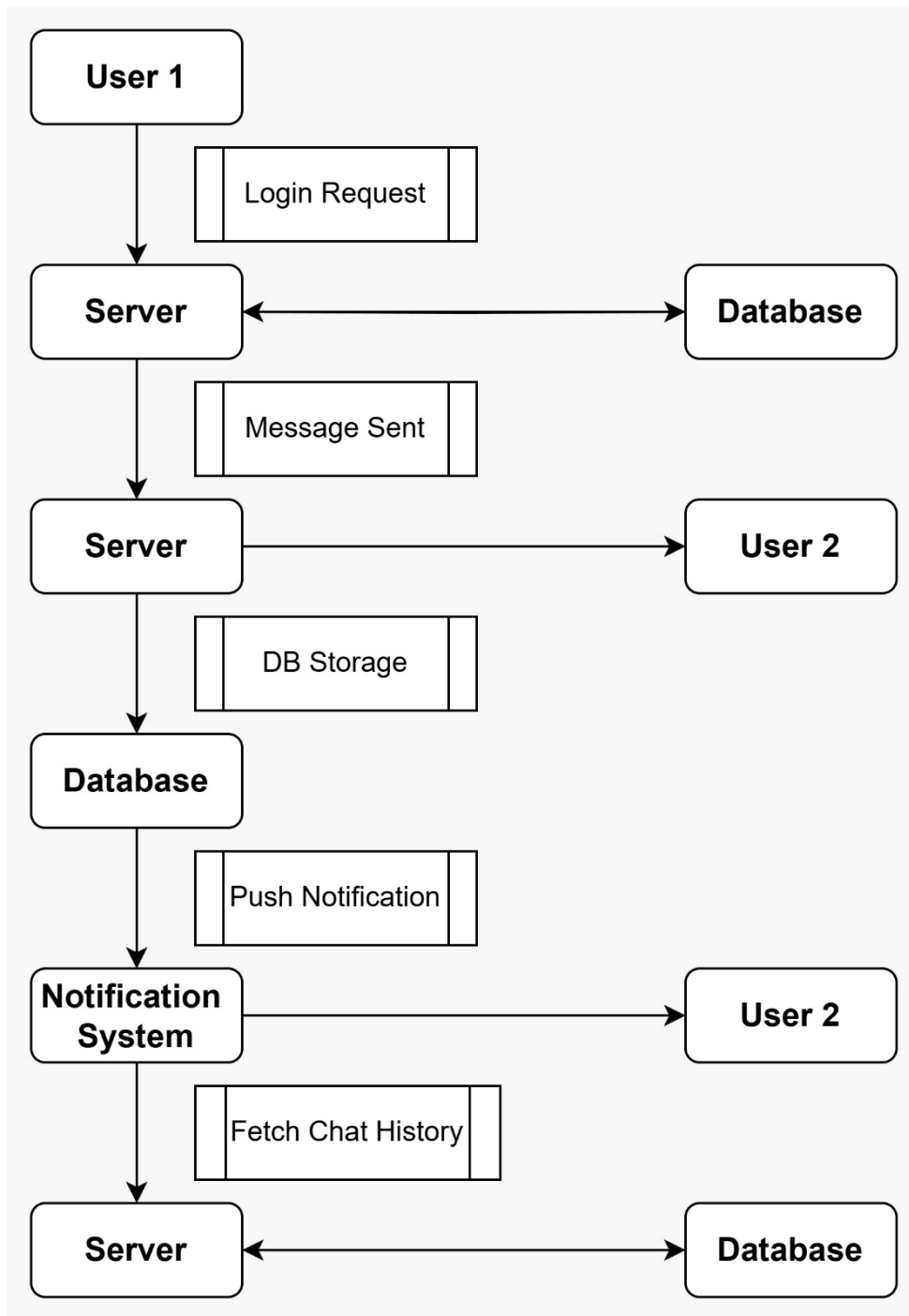
- >Clients (Users): Send and receive messages via web site.
- >Server: Acts as a central hub, handling message routing, authentication, back up data storage.
- >Database: Stores messages, user details, and chat history.
- >WebSocket: Enables real-time, bi-directional communication between client and server.

2. Database Design & Data Modeling



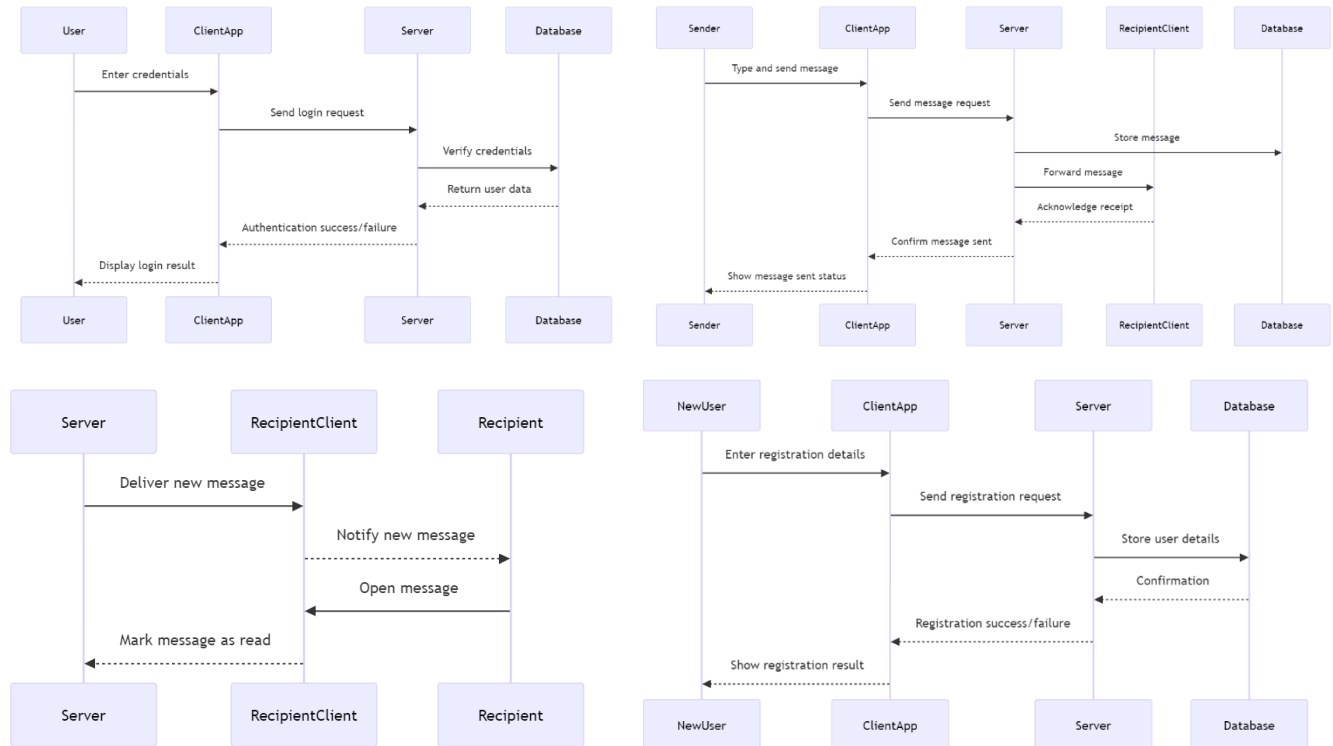
3. Data Flow & System Behavior

Data flow diagram



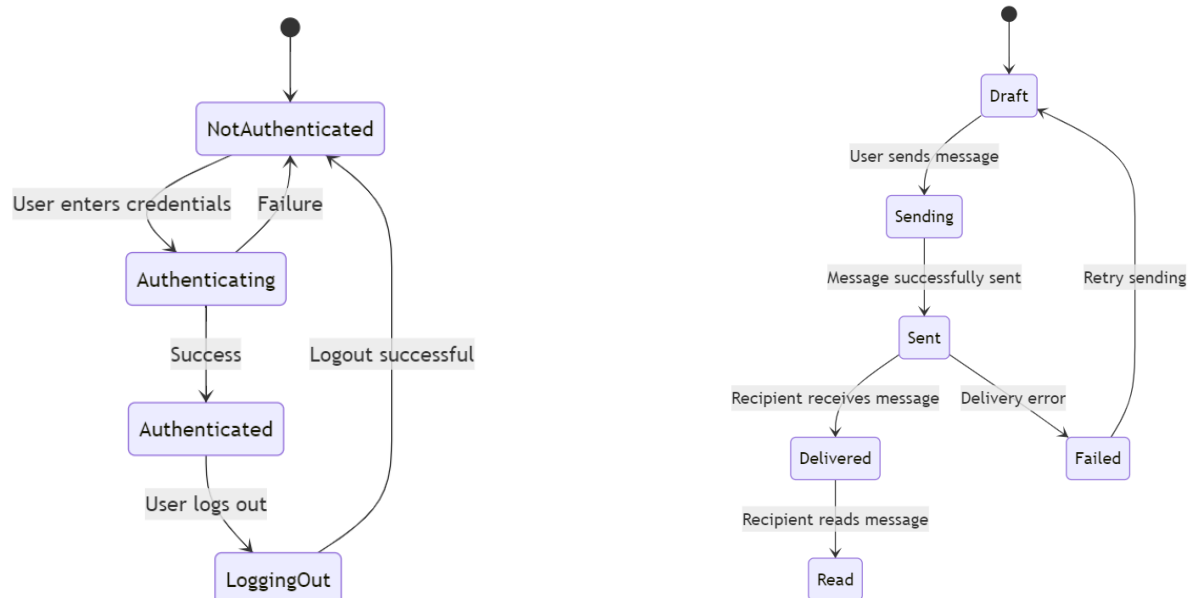
Sequence diagram

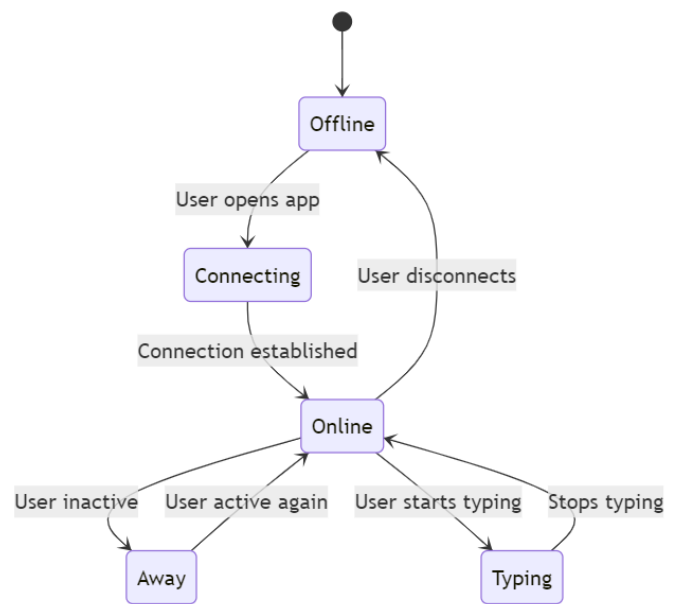
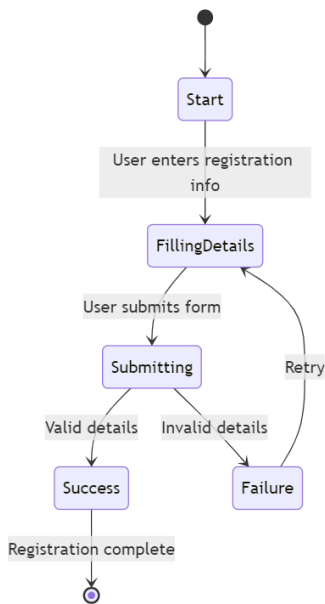
Login – message sending – message alert - registration



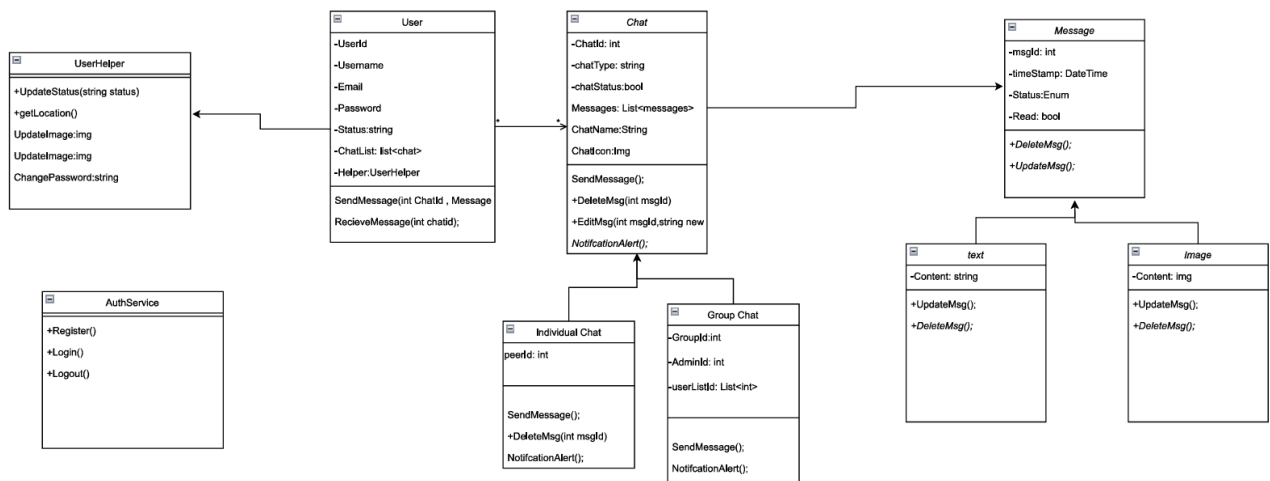
State diagram

Login – message sending – message alert - registration





Class diagram



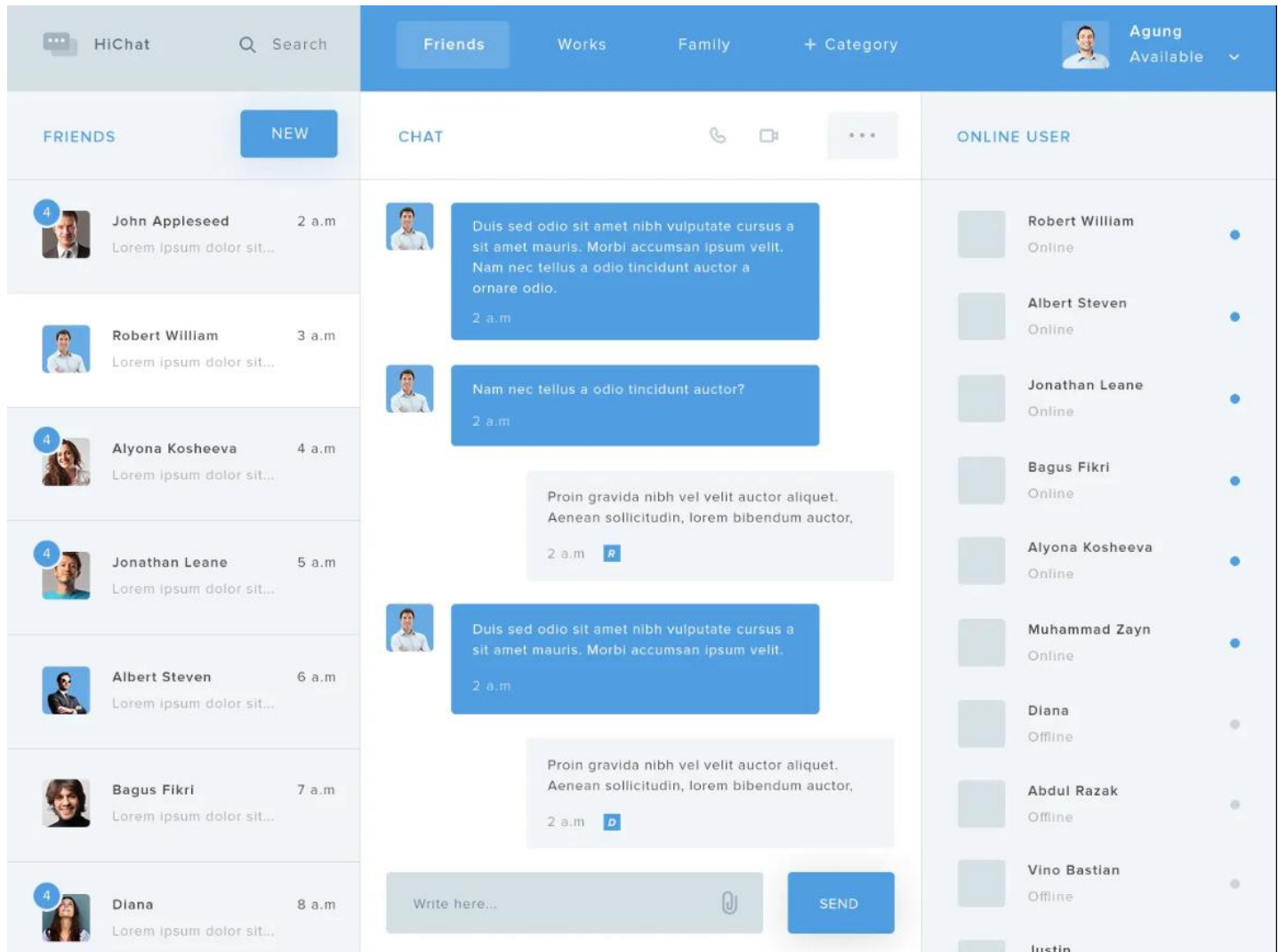
4. UI/UX Design & Prototyping

1. UI Guidelines

- Consistent Design
- Typography & Layout: Clear, readable fonts with proper spacing and padding.
- Icons & Buttons: Intuitive icons and well-sized buttons for easy navigation.
- Profile Avatars: Display clear user images for quick identification.

2. UX Guidelines

- Easy Navigation: Organized chat categories, recent chats, and online users.
- Message Readability: Different colors for sent/received messages, clear timestamps, and status indicators.
- Real-time Feedback: Typing indicators, message delivery/read receipts.



5. System Deployment & Integration

- Technology Stack:
 - >Backend: .Net
 - >Frontend: html/CSS - JavaScript
 - >Database: SQL Server
- Deployment Diagram: single server - single responsibility
- Component Diagram: Later