1. Stakeholder Analysis: Identifying Key Stakeholders and Their Needs

Stakeholders:

- **End Users:**
  - **Needs:** A seamless and intuitive chat experience with real-time communication, support for multiple message types (text, voice, video), user-friendly interface, and security for private messages.
- **Product Owner/Project Manager:**
  - **Needs:** Clear project timelines, budget management, progress updates, and a finished product that meets end-user requirements.
- **Developers (Front-end & Back-end):**
  - **Needs:** Clear technical requirements, access to tools and resources, and user feedback to refine the system.
- **QA Testers:**
  - **Needs:** Access to test environments, detailed use cases, and clear success criteria to ensure the system is bug-free.
- **Security Specialists:**
  - **Needs:** Clear guidelines on data protection, secure user authentication, and encryption of communication channels.
- **Admin/Support Team:**
  - **Needs:** User management tools, support for moderating content, and reporting mechanisms for activity or issues within the chat platform.
- **Third-Party Integrators (if any):**
  - **Needs:** Clear API documentation for integrations (e.g., payment gateways, social media logins).

2. User Stories & Use Cases: Scenarios Illustrating How Users Interact with the System

User Stories:

1. **As a user, I want to be able to register an account** so that I can use the chat platform with personalized features.
2. **As a user, I want to send text messages** so that I can communicate with my friends or group members.

3. **As a user, I want to send voice messages** so that I can communicate without typing.
4. **As a user, I want to create a group chat** so that I can communicate with multiple people at the same time.
5. **As a user, I want to be notified of new messages** so that I don't miss any important communication.
6. **As a user, I want to delete my message** in case I sent something by mistake.
7. **As a user, I want to be able to log out** so that I can safely exit my account.

Use Cases:

1. **Use Case 1: User Registration**
    a. **Actor:** New User
    b. **Description:** The user accesses the registration page, fills in required details (name, email, password), and submits. An email confirmation is sent for verification.
    c. **Pre-condition:** The user is not logged in.
    d. **Post-condition:** The user is registered and can log in.
2. **Use Case 2: Sending a Message**
    a. **Actor:** Registered User
    b. **Description:** The user selects a chat (private or group), types a message, and presses send. The message is delivered to the recipient(s) in real-time.
    c. **Pre-condition:** The user is logged in.
    d. **Post-condition:** The recipient(s) receive the message.
3. **Use Case 3: Group Chat Creation**
    a. **Actor:** Registered User
    b. **Description:** The user creates a group chat by inviting multiple users, sets a group name, and starts the conversation.
    c. **Pre-condition:** The user is logged in and has at least two contacts.
    d. **Post-condition:** A new group chat is created, and users can begin chatting.
4. **Use Case 4: Logging Out**
    a. **Actor:** Registered User
    b. **Description:** The user clicks the logout button to safely exit the chat platform.
    c. **Pre-condition:** The user is logged in.
    d. **Post-condition:** The user is logged out and redirected to the login page.

3. Functional Requirements: List of Features and Functionalities

1. **User Authentication:**
    a. Sign up, log in, and password reset functionality.
    b. Email verification during registration.
2. **Real-time Messaging:**
    a. Text messages between users.
    b. Voice and video messaging support.
    c. Group chat functionality.
    d. Message status (sent, delivered, read).
    e. Typing indicators for real-time communication.
3. **User Profile Management:**
    a. Ability to update profile information (name, profile picture).
    b. Option to set a status (online, offline, custom status).
4. **Notifications:**
    a. Push notifications for new messages (both text and media).
    b. Notification when a user joins or leaves a group.
5. **Message History:**
    a. Store chat history for all users (text, voice, video).
    b. Searchable message history by keywords.
6. **Security and Privacy:**
    a. Encrypted communication between users.
    b. Two-factor authentication for secure logins.
    c. Private messages and group privacy controls.
7. **Moderation:**
    a. Admin panel to manage users and content (mute, ban, report users).
    b. Report functionality for inappropriate content.
8. **File Sharing:**
    a. Ability to share images, files, and documents within chats.
    b. Preview of images or files in the chat window.
9. **Cross-platform Support:**
    a. The system should work across multiple devices, including desktops, tablets, and smartphones.

4. Non-functional Requirements: Performance, Security, Usability, and Reliability Criteria

1. **Performance:**
    a. **Response Time:** The time taken to send and receive a message should be less than 1 second.
    b. **Scalability:** The system should support up to 10,000 concurrent users.
    c. **Throughput:** The system should handle up to 100 messages per second without performance degradation.
2. **Security:**
    a. **Data Encryption:** All messages and user data should be encrypted using SSL/TLS for transmission and AES for stored data.
    b. **Authentication:** Use of multi-factor authentication for user login.
    c. **Session Management:** Secure session management and automatic session expiration after a period of inactivity.
3. **Usability:**
    a. **User Interface:** The interface should be intuitive and easy to navigate, with minimal steps to start chatting.
    b. **Accessibility:** The platform should be accessible to people with disabilities (e.g., support for screen readers).
    c. **Responsiveness:** The website should be responsive and work well on different screen sizes (desktop, tablet, mobile).
4. **Reliability:**
    a. **Availability:** The platform should have 99.9% uptime.
    b. **Backup and Recovery:** Regular backups should be taken for user data and chat history. In case of failure, data should be recoverable.
    c. **Error Handling:** Proper error messages and recovery procedures should be in place for failed operations.