So

IT-314 : Software Engineering

Software requirements specifications

Group : 12

CHAT APPLICATION

Introduction

In today’s digital era, communication has become an essential aspect of our daily lives. Messaging applications are increasingly becoming the primary medium for individuals and organizations to interact with one another. To cater to the diverse communication needs of users, our project focuses on developing a chat application that offers both individual and group chat functionality with a rich set of features designed to enhance user experience, security, and privacy. The application aims to provide a seamless, user-friendly platform that allows real-time conversations, media sharing, profile customization, and more.

This chat application not only focuses on providing fundamental chat features, but also emphasizes account security, user management, and content moderation to ensure a safe and enjoyable environment for all users.

Problem Statement

The problem addressed by this project is the increasing demand for a secure, user-friendly, and feature-rich messaging platform that facilitates real-time communication. Existing chat applications often lack flexibility in terms of user control over privacy, notifications, and profile management. Moreover, these platforms sometimes fall short in ensuring effective moderation of inappropriate user behavior and in preventing spam. Therefore, our chat application aims to solve the following key problems:

1. **Account Management**: Simplified user creation, authentication, password recovery, and account deletion.
2. **Privacy Concerns**: Providing users with complete control over their profile visibility and communication preferences.
3. **Security**: Ensuring secure communication through features like two-factor authentication, password resets, and spam detection.
4. **Real-Time Interaction**: Facilitating real-time private and group chats with features for media sharing, status updates, and conversation history.

REQUIREMENTS

FUNCTIONAL REQUIREMENTS:

1. User Registration and Authentication

- Users can register with name, email, username, and password.
- A confirmation email is sent for verification.
- Users can log in using their credentials and enable two-factor authentication (2FA).
- Users can reset forgotten passwords.

2. User Profile Management

- Users can create, edit, and delete their profiles.
- Users can upload profile pictures and control profile visibility (public or private).

3. Chat Functionality

- Users can send and receive text, images, videos, and files in private or group chats.
- Real-time messaging with read receipts.

4. Group Chat

- It enables us to stay connected, share updates, and collaborate in real-time within your chosen group circles!

5. SmartText Enhance

- SmartText Enhance is an AI-driven messaging assistant that refines user input by suggesting more polished, concise, or expressive alternatives.
- It ensures your messages are professional, impactful making communication more effective and engaging.

NON-FUNCTIONAL REQUIREMENTS:

1. Performance

- Real-time message delivery with minimal latency.
- Scalable to support large user bases (thousands of concurrent users).
- High availability (99.9% uptime).

2. Security

- Strong encryption for all sensitive data.
- Secure login and Authentication.

3. Usability

- Intuitive and responsive user interface across devices (mobile).

4. Reliability

- Backup and disaster recovery to prevent data loss.
- Fault tolerance for high reliability.

5. Maintainability

- Modular, well-documented codebase.
- Logging and monitoring for system health and error tracking.
- Version control for easy collaboration.

Actors

1. **End User**: A person who registers, logs in, and interacts with the chat application by sending messages, joining groups, updating their profile, etc.
2. **Admin**: A person who manages user accounts, moderates reports, suspends or deletes users.
3. **External Systems**: Email service provider (for sending confirmation emails, password resets).
4. **System**: The chat application itself, which manages user accounts, chat messages, notifications, security protocols, and more.

Use Cases

1. Create User Account

- **Actor**: User
- **Description**: The user creates a new account by entering necessary details, such as name, email, username, and password.

Preconditions:

- The user is not already registered.
- The system is online and accessible.

Postconditions:

- The user's account is successfully created and stored in the database.
- The user is logged in and redirected to the main application interface.

Main Flow:

1. The user navigates to the "Create Account" page.
2. The user fills in the required fields (name, email, username, password).
3. The system validates the user input (e.g., ensuring the email format is correct, username is unique).
4. The system sends a confirmation email to the user.
5. The user verifies the email by clicking on the confirmation link.
6. The system confirms the email and creates the user account.
7. The user is successfully logged into the application.

Alternate Flow:

1. If the email format is incorrect, the system displays an error message asking the user to provide a valid email.
2. If the email is not verified within a specified time frame, the system sends a reminder to the user to verify their email.

2. Authenticate User (Login)

- **Actor**: User
- **Description**: The user logs into the chat application using their credentials (username/email and password).

Preconditions:

- The user has already registered and has a valid account.
- The system is online and functioning.

Postconditions:

- The user is successfully logged into the application and can access its features.

Main Flow:

1. The user enters their username/email and password on the login page.
2. The system checks if the entered credentials are valid.
3. If valid, the system authenticates the user and grants access to the chat application.

Alternate Flow:

1. If the entered credentials are incorrect, the system displays an error message, prompting the user to check their credentials.
2. If the user forgets their password, the system provides a "Forgot Password" link, guiding the user through the process to reset their password.

3. Send a Message (Private or Group Chat)

- **Actor**: User
- **Description**: A user sends a message in a private chat or group chat.

Preconditions:

- The user is authenticated and logged into the system.
- A chat (private or group) must exist.

Postconditions:

- The message is successfully sent and displayed in the chat window.
- The message is stored in the database for future retrieval.

Main Flow:

1. The user opens a chat (either private or group).
2. The user types a message in the text box.
3. The user clicks "Send."
4. The message appears instantly in the chat window for both the sender and the recipient(s).
5. The system stores the message in the database for future retrieval (in case of searching or history).

Alternate Flow:

1. If there is a network failure or the message cannot be sent, the system shows an error message and allows the user to try again.
2. If the message fails to deliver due to server issues, the system retries sending the message in the background.

4. Reset User Password

- **Actor**: User
- **Description**: The user resets their password if they forget it.

Preconditions:

- The user has a registered account with the system.
- The user has access to their registered email address.

Postconditions:

- The user's password is updated in the database.
- The user can log in using the new password.

Main Flow:

1. The user clicks on the "Forgot Password" link on the login page.
2. The user enters their registered email address.
3. The system sends a password reset link to the user's email.
4. The user clicks the link and is redirected to a page where they can enter a new password.
5. The user submits the new password.
6. The system validates the new password (ensuring it meets security requirements) and updates the user's account.
7. The user is notified that their password has been successfully reset.

Alternate Flow:

1. If the email entered is not registered, the system displays an error message indicating that the email does not exist in the system.
2. If the user fails to enter the new password correctly (e.g., passwords do not match), the system displays an error message and prompts the user to try again.
3. If the reset link expires (After 1 hour & will get access only one time), the system informs the user and asks them to request a new reset link.

5. View Message Read Receipts

- **Actor**: User
- **Description**: The user can see whether their sent messages have been read by the recipient.

Preconditions:

- The user has sent a message in a chat (private or group).

Postconditions:

- The user sees the current status of the message (e.g., "Read" or "Delivered").

Main Flow:

1. The user sends a message in a private chat.
2. The system displays a "Read" status when the recipient opens and reads the message.
3. The user sees a "Read" or "Delivered" status next to the message.

Alternate Flow:

1. If the recipient has disabled read receipts, the user is shown a "Delivered" status but not the "Read" status.
2. If the message fails to be delivered due to network issues, the system may show a "Failed to Deliver" status.

6. Create Group Chat

Actor: User
Description: The user creates a new group chat with multiple participants.

Preconditions:

- The user is authenticated and logged into the system.

Postconditions:

- A new group chat is successfully created and stored in the database.
- The group chat details are shared with all participants.

Main Flow:

1. The user clicks on "Create Group Chat."

2. The user enters a group name and selects participants.
3. The system validates the inputs (e.g., ensures a group name is provided).
4. The system creates the group in the database and notifies all selected participants.

Alternate Flow:

- If the group name is missing, the system displays an error message and asks the user to provide a valid name.
- If no participants are selected, the system prompts the user to add at least one participant.

7. Update Profile

Actor: User
Description: The user updates their profile information (e.g., name, email, or profile picture).

Preconditions:

- The user is authenticated and logged into the system.

Postconditions:

- The updated profile information is successfully stored in the database.
- The user sees the updated profile details in the application.

Main Flow:

1. The user navigates to the "Profile" section.
2. The user updates the desired fields (e.g., name, email, profile picture).
3. The system validates the updated inputs (e.g., checks if the new email format is valid).
4. The system saves the updated information in the database.
5. The system confirms the update and displays the updated profile to the user.

Alternate Flow:

- If the new inputs are invalid (e.g., email format is incorrect), the system displays an error message and prompts the user to correct it.
- If the update fails due to server issues, the system displays an error message and allows the user to retry.