
# SWAMI VIVEKANANDA INSTITUTE OF MODERN SCIENCE

# SOFTWARE REQUIREMENT SPECIFICATION

# CHAT APPLICATION

## SUBMITTED BY:

**Sudip Utthasini [Roll No: 26401222009]**

**Samiran Bhattacharya [Roll No: 26401222024]**

**Sawan Kumar Murmu [Roll No: 26401222029]**

**Sayan Adhikary [Roll No: 26401222033]**

**Sudhangsu Shekhar Bairagi [Roll No: 26401222092]**

**Internal Project Guide:**

**Prof. Jyotirmoyee Roy Chowdhury**

**Swami Vivekananda Institute Of Modern Science**

# INDEX

# ABSTRACT

This project is about developing a web-Application capable of Users can create accounts, log in, and join chat rooms to communicate with others in real-time. The chat rooms leverage PHP database, ensuring quick and efficient updates as messages are sent and received.

The user interface is designed using HTML and CSS, with a responsive layout adaptable to different screen sizes. Key features include message deletion, user blocking, and notification alerts for new messages.

# INTRODUCTION

**Project Title: Chat Application**

**Statement of Project:**

- The Online Chat Application is a web-based platform that enables real-time communication between users. It allows individuals to exchange messages, share files, and participate in group chats.
- **The web application mainly associates with:**
  - User registration and authentication
  - One-on-one chat
  - Group chat
  - Message history
  - Notifications
- The application caters to a wide audience, including students, professionals, and social groups.

# OBJECTIVES OF THIS PROJECT

The primary objective of our online chat application project is to develop a robust and user-friendly platform that facilitates seamless communication and collaboration among users in real-time. The application should allow users to exchange messages, share files, and participate in group chats.

**The Goals of Our System:**

- The project aims to optimize database usage, ensuring efficient storage and retrieval of chat history. Proper indexing, data modeling, and query optimization are essential to achieve this objective.
- The chat application should provide real-time updates, ensuring that messages are delivered promptly without noticeable lag times.
- The chat interface should be responsive and adaptable to different screen sizes (e.g., desktop, tablet, mobile). Users should have a consistent experience across devices.
- Implement robust user authentication mechanisms to secure user accounts. Only authorized users should have access to chat features. Additionally, consider encryption for message privacy.

# SOFTWARE AND HARDWARE SPECIFICATION

- **Hardware Requirement Specification**

| Client Machine | | Server Machine | |
|---|---|---|---|
| SSD | 500GB | SDD | 500GB |
| Processor | Intel i3 / AMD Ryzen 3 or Higher | Processor | Intel i3 / AMD Ryzen 3 or Higher |
| Memory | 4GB or more | Memory | 4GB or more |

- **Software Requirement Specification**

| Client Machine | | Server Machine | |
|---|---|---|---|
| OS | Windows/Linux/Mac | OS | Windows 11 |
| BROWSER | Any browser | WEBSERVER | Apache 2.4.58 |
| | | FRONTEND | HTML5,CSS3, JavaScript |
| | | BACKEND | PHP 8.2.12 |
| | | DATABASE | MySQL 8.2.12 |

# PROJECT MODULE

- **User Registration Module:** This module allows new users to create an account within the chat application. Registration form for entering user details (username, password, email, etc.).Validation of user inputs to ensure data integrity and security. Storage of user data into the User Database. Error handling and feedback for invalid registration attempts.
- **User Authentication Module:** This module manages user login and authentication processes. Login form for users to enter their credentials. Verification of credentials against stored user data in the User Database. Management of user sessions upon successful login. Feedback to users for successful or failed login attempts.
- **Real-Time Chat Module:** This module enables users to communicate with each other in real time. Sending and receiving text messages between users. Storage of chat messages in the Chat Database. Retrieval of chat history and display of messages in real time. Notification system for incoming messages.
- **User Management Module:** This module provides functionalities for managing user profiles and searching for other users.
- **Session Management and Security Module**: This module handles user session management and security features to ensure safe interactions. Management of user sessions to maintain active login states. Secure logout functionality to end user sessions and prevent unauthorized access. Encryption of sensitive data such as passwords. Prevention of security vulnerabilities like SQL injection and cross-site scripting (XSS).
- **Database Management Module:** This module manages the interaction between the application and the databases.
- **User Interface Module:** This module is responsible for the visual elements and user interactions within the application.

# Functional Components of The PROJECT

## 1. User Authentication and Management :

- **Purpose:** Allow users to register, log in, and manage their accounts.
- **Registration:** Users need to sign up with their details (e.g., username, email, password).
- **Login/Logout:** Users should be able to log in and out securely.
- **Session Management:** Maintain user sessions to keep track of logged-in users.
- **Password Management**: Securely store and manage user passwords (e.g., hashing passwords before storing).

## 2. Chat Interface :

- **Purpose:** Provide a user interface for sending and receiving messages.
- **Message Display:** Show chat messages in a real-time chat window.
- **Input Field:** Allow users to type and send messages.
- **User List:** Display a list of active users if desired.

## 3. Error Handling and Logging :

- **Purpose:** Manage and log errors to ensure the application runs smoothly and issues are addressed promptly.
- **Error Handling:** Display user-friendly error messages and handle exceptions gracefully.
- **Logging:** Record errors and significant events for troubleshooting and monitoring.

## 4. Message Handling :

- **Purpose:** Manage the sending and retrieval of chat messages.
- **Send Messages:** Save new messages to the database.
- **Message History:** Optionally, implement pagination or infinite scrolling for message history.

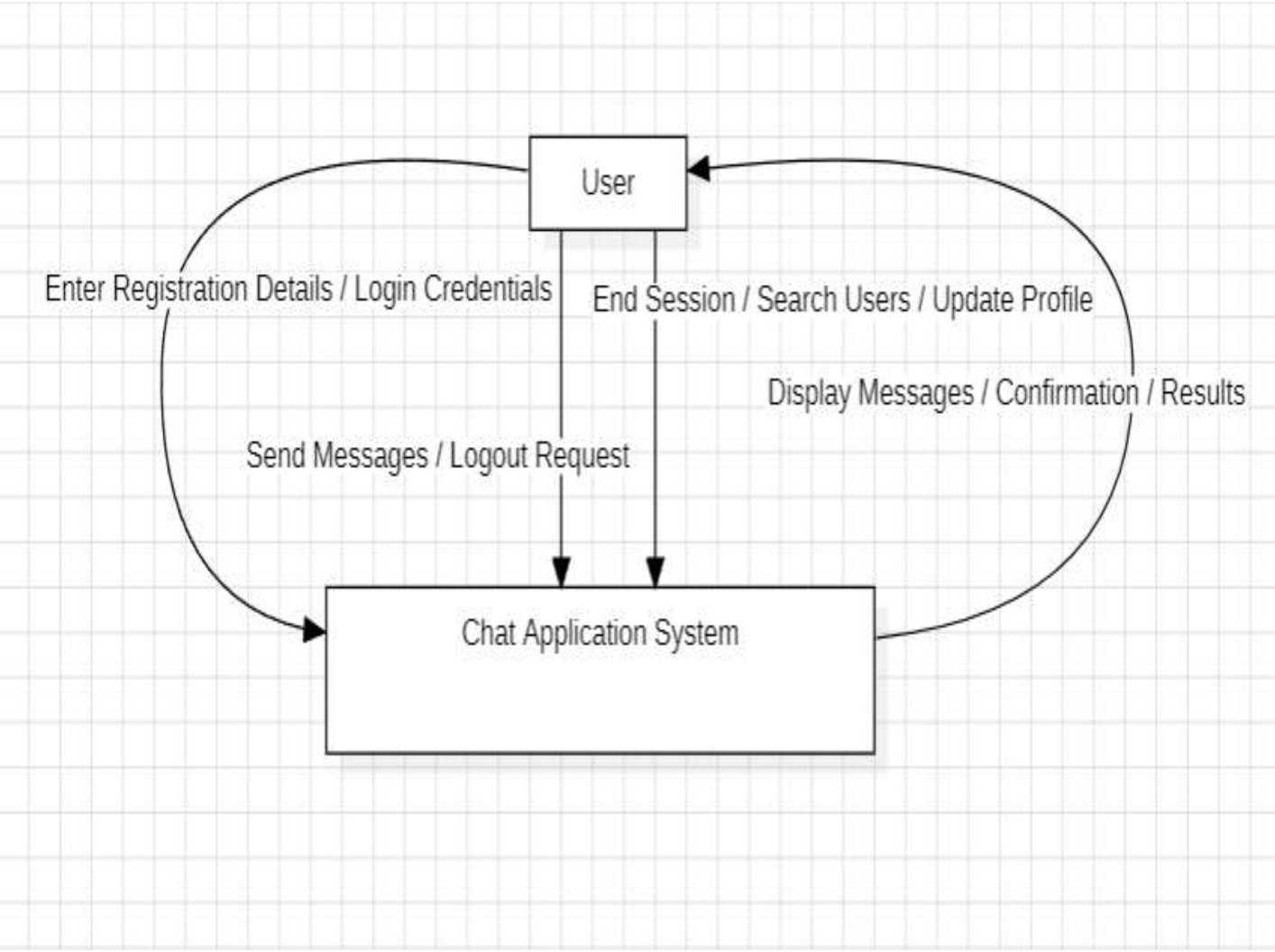
5. **Real-Time Communication :**
   - **Purpose:** Ensure that messages are updated in real-time without requiring page refreshes.
6. **Database Management :**
   - **Purpose:** Store user information and chat messages securely and efficiently.
   - **Error Handling:** Display user-friendly error messages and handle exceptions gracefully.
   - **Logging:** Record errors and significant events for troubleshooting and monitoring.

# NON FUNCTIONAL REQUIREMENTS

1. Secure access of confidential data (user's details) should be provided.
2. 24*7 availability of the application.
3. Application should be accessible over internet.
4. Better component design to get better performance at peak time.
5. Flexible service based architecture will be highly desirable for future extension

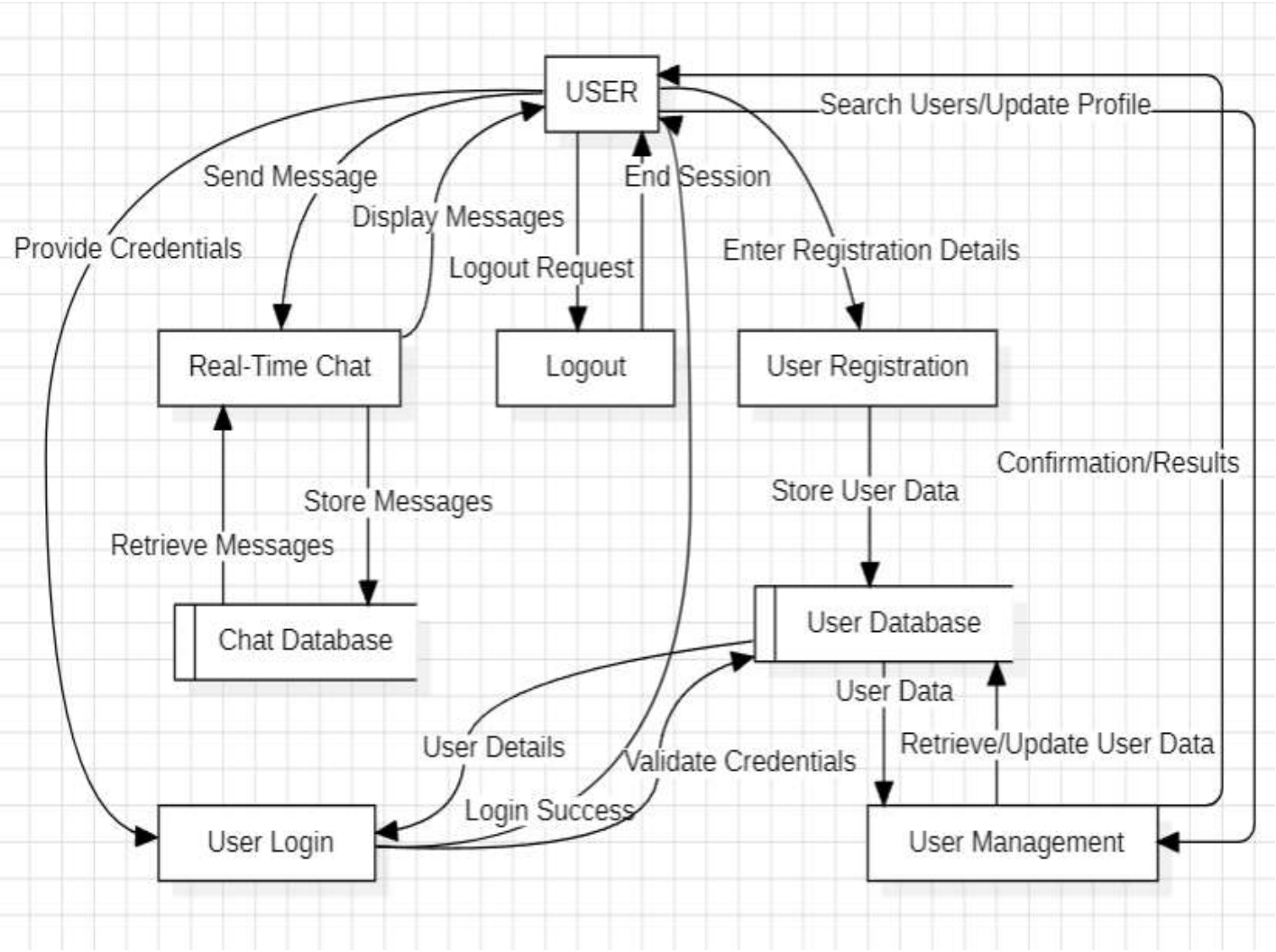
# DATA FLOW DIAGRAM

## 0 Level DFD



# 1 Level DED

.


USER
Search Users/Update Profile
Send Message
End Session
Display Messages
Provide Credentials
Logout Request
Enter Registration Details
Real-Time Chat
Logout
User Registration
Confirmation/Results
Store User Data
Store Messages
Retrieve Messages
Chat Database
User Database
User Data
User Details
Validate Credentials
Retrieve/Update User Data
Login Success
User Login
User Management

# TESTING AND VALIDATION

## Testing:

- Software testing is a critical element of software quality assurance and represent the ultimate of specification, design, coding.
- The purpose of testing in our project is to check proper connection of webpage with the database , Whether the credential are properly stored in data base or not, to check whether the connection are on real time, whether sender and receiver can see each others messages in real time or not.

## Validation:

- Chat Application cannot be access if necessary credential are not filled in the login page
- Only administrator can perform sophisticated works of backend like deleting data form database, adding constrain on which information are to be required to access this application and retrieve data of users
- For security purpose the Email of user is required in case he/she forgets his/her password and wants to retrieve that.

# SYSTEM SECURITY MEASURES

- Security prompting the user for a user id and password in our application is a potential security threat. So credential information is transferred from the browser to server are encrypted.
- Cookies are an easy and useful way to keep user-specification, information available. However, because cookies are send to the browser's computer, they are vulnerable to spooling or other malicious use.
- Do not store any critical information in cookies. For example, do not store a user's password in a cookie, even temporarily
- Avoid permanent cookies if possible. Consider encrypting information in cookies. Set expiration date on cookies to the shortest practical time if possible.

# FUTURE SCOPE AND ENHENCEMENT

1. **Enhanced Security and Privacy:**
   - **End-to-End Encryption:** Ensuring all messages are encrypted, protecting user data from unauthorized access.
   - **Decentralized Networks:** Exploring block-chain or decentralized technologies for more secure and private communication.
   - **Advanced Authentication**: Implementing multi-factor authentication (MFA) and biometric verification to enhance user security.
2. **Rich Media and Interactivity :**
   - **File Sharing**: Improved support for various file types, including high-resolution images, videos, and documents.
   - **Video and Voice Integration**: Seamless integration of voice and video calling features with high quality streaming and low latency.
3. **Cross-Platform Compatibility :**
   - **Unified Experience**: Ensuring smooth functionality across different devices and operating systems, including desktops, tablets, and mobile devices.
   - **Progressive Web Apps (PWAs):** Leveraging PWAs for a more app-like experience within web browsers, including offline capabilities and push notifications.
4. **Integration with Other Services:**
   - **API and Web-hooks:** Providing robust APIs and web-hooks for integration with other applications and services, such as CRM systems, project management tools, and social media platforms.
   - **Third-Party Integrations*: Supporting popular third-party apps and services, including calendar scheduling, file storage, and productivity tools.***
5. **Real-Time Collaboration Features:**
   - **Shared Workspaces:** Implementing collaborative features like shared documents, whiteboards, and project management tools.
   - **Real-Time Updates**: Providing real-time synchronization of content and interactions across users and devices.

# LIMITATIONS

## 1. Real-Time Communication :

- Polling vs. WebSockets.
- Latency Issues.

## 2. Security :

- SQL Injection: While prepared statements mitigate this risk, it's crucial to always validate and sanitize user inputs to prevent SQL injection attacks.
- Password Management: Storing passwords securely requires using proper hashing algorithms and handling password reset mechanisms securely.
- Session Management: Securing user sessions to prevent session hijacking and fixation is essential.

## 3. User Authentication and Management :

- Session Security: Implementing secure session management can be challenging, particularly with regard to preventing session fixation and ensuring secure cookie handling.
- User Identity Verification: Without robust authentication methods, user identity verification and account management can be weak, leading to potential impersonation or abuse.

## 4. User Experience :

- Scalability of Features: Adding advanced features like message search, file sharing, or video chat can be complex and may require additional infrastructure or technology.
- User Interface: Ensuring a smooth, responsive, and intuitive user interface can be challenging, especially when designing for various devices and screen sizes.

# CONCLUSION

Building a web-based chat application with PHP and MySQL offers a practical approach for creating real-time communication solutions, leveraging the strengths of these technologies. Here's a summary of the key points and considerations for such a project:

## 1. Core Features ->

- User Authentication: Allows users to register, log in, and manage their accounts, ensuring personalized and secure interactions.
- Chat Interface: Provides an intuitive and user-friendly interface for sending and receiving messages.
- Message Handling: Efficiently manages message storage, retrieval, and display, supporting both real-time and historical message access.
- Real-Time Communication: Implements methods (e.g., AJAX polling or WebSockets) to provide immediate message updates.
- Administrative Tools: Offers functionalities for managing users, moderating content, and overseeing application health.

## 2. Benefits ->

- Familiar Technologies: PHP and MySQL are widely used and well-documented, making it easier to find resources and support.
- Cost-Effective: Both technologies are open-source and free to use, reducing initial development costs.
- Scalability Potential: With proper optimization and architecture, PHP and MySQL can scale to handle increasing numbers of users and messages.

## 3. Limitations and Challenges ->

- Scalability Issues: As the user base grows, the performance of PHP and MySQL can be challenged, requiring optimization and potentially additional infrastructure.

- Real-Time Performance: PHP's synchronous nature and MySQL's lack of native real-time support can complicate the implementation of seamless, instant messaging**.**
- Security Considerations: Ensuring robust security practices is critical to protect user data and prevent vulnerabilities such as SQL injection and XSS attacks.
- User Experience: Designing a responsive and intuitive chat interface across various devices and browsers requires careful planning and testing.

Developing a web-based chat application using PHP and MySQL is a valuable project that provides a solid foundation in web development and real-time communication. While there are limitations to be aware of, careful planning and implementation of best practices can result in a functional and scalable chat application. By addressing challenges proactively and incorporating user feedback, you can build a robust communication platform that meets user needs and adapts to future demands.

This project serves as an excellent learning opportunity and a practical example of integrating web technologies to solve real-world problems, paving the way for more complex and feature-rich applications in the future.