SOFTWARE REQUIREMENTS SPECIFICATION

For

Chat Application with Message Storage

Done by:

NITIN.P

VARSHA.AR

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the software requirements for the development of a Chat Application with Message Storage. This application aims to facilitate real-time communication between users and store messages for future reference.

1.2 Scope

The Chat Application will allow users to register, log in, and exchange messages in real-time. Additionally, the application will include a message storage feature to retain chat history for users.

1.3 Definitions, Acronyms, and Abbreviations

SRS: Software Requirements Specification

UI: User Interface

API: Application Programming Interface

2. Overall Description

2.1 Product Perspective

The Chat Application will be a standalone system that allows users to communicate with each other. It will be built as a client-server application, with clients connecting to a centralized server to exchange messages. The server will be responsible for storing and retrieving messages.

2.2 User Classes and Characteristics

Registered Users: Individuals who have created an account and can log in to use the chat application.

Guest Users: Users who can access the application without registration but have limited functionality.

2.3 Operating Environment

The application will be developed to run on multiple platforms, including Windows, macOS, and Linux. It will also be accessible through popular web browsers.

2.4 Design and Implementation Constraints

The application will be developed using [programming language/technology].

Compatibility with major web browsers and operating systems must be ensured.

3. System Features

3.1 User Registration and Authentication

Users can register by providing necessary information.

Authentication mechanisms (e.g., email verification, password hashing) will be implemented.

3.2 User Profile Management

Users can create and update their profiles.

Profile information may include a display name, profile picture, etc.

3.3 Real-Time Messaging

Users can send and receive real-time messages.

Group and private chat functionality will be supported.

3.4 Message Storage

Messages will be stored on the server for future retrieval.

Users can view and search their chat history.

3.5 Notifications

Users will receive notifications for new messages and other relevant events.

4. External Interface Requirements

4.1 User Interfaces

The application will have an intuitive and user-friendly interface with the following components:

Login and registration screens

Chat interface

Profile management

4.2 Hardware Interfaces

The application will rely on standard hardware components such as computers, smartphones, and tablets.

4.3 Software Interfaces

The application will interact with a database for message storage.

Integration with third-party authentication services may be considered.

5. Non-functional Requirements

5.1 Performance Requirements

The system should support a minimum of [X] concurrent users.

Messages should be delivered within [Y] seconds under normal operating conditions.

5.2 Security Requirements

User data should be encrypted during transmission.

Authentication mechanisms should be robust to prevent unauthorized access.

5.3 Reliability and Availability

The system should have an uptime of at least 99%.

Regular backups of message data should be performed.

6. Other Requirements

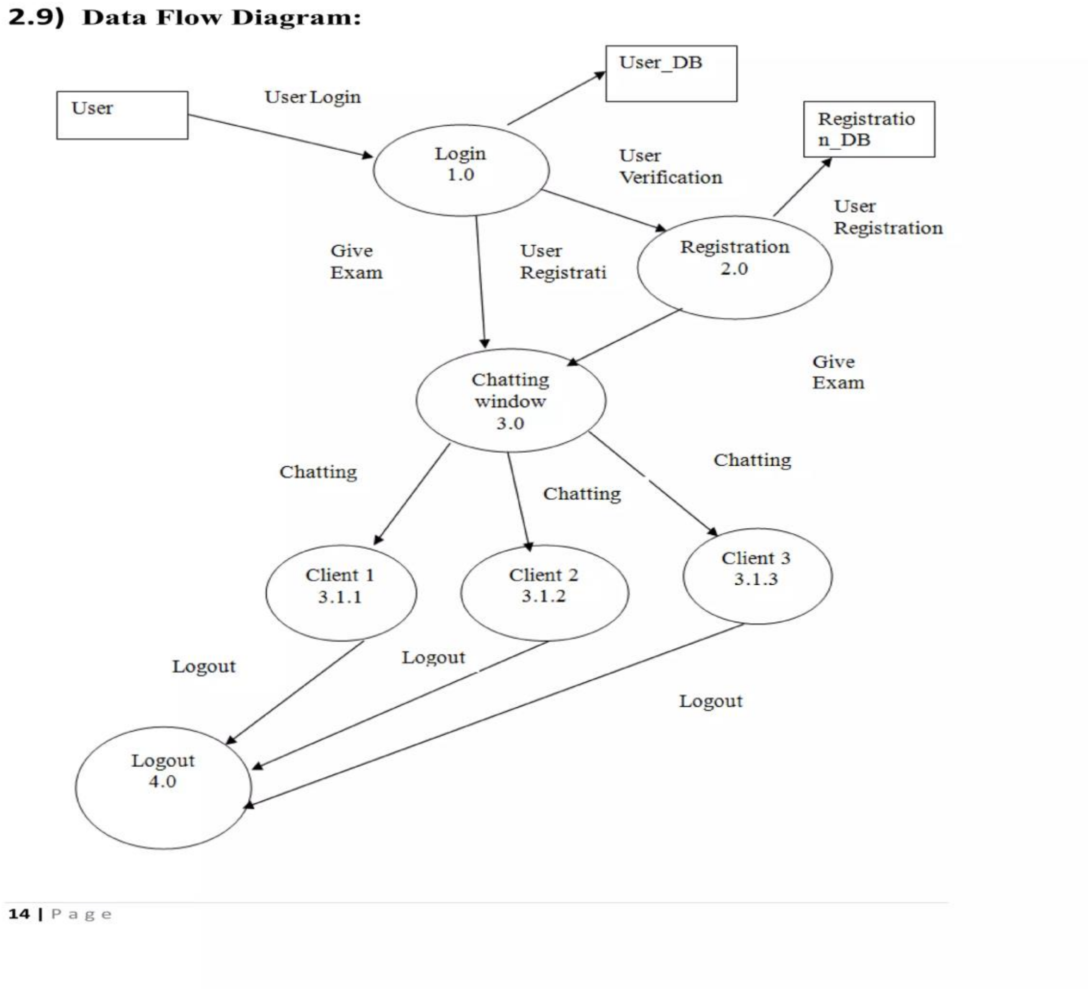
6.1 Legal and Regulatory Requirements

The application should comply with data protection laws and regulations.

6.2 Documentation Requirements

Comprehensive user documentation and developer documentation should be provided.

7.Data flow Diagram



2.9) Data Flow Diagram:

14 | Page

8. System Architecture

8.1 High-Level Architecture

The application will follow a three-tier architecture, consisting of:

Presentation Tier: User interfaces for registration, login, profile management, and chat.

Application Tier: Business logic for user authentication, message handling, and storage.

Data Tier: Database for storing user profiles and message history.

8.2 Scalability

The system should be designed to scale horizontally to handle an increasing number of users. This can be achieved through load balancing and distributed database architecture.

9. Usability Requirements

9.1 User Training

The application should be intuitive enough for users to navigate without extensive training. A brief onboarding tutorial may be provided.

9.2 Accessibility

The application should comply with accessibility standards (e.g., WCAG) to ensure usability for users with disabilities.

10. User Stories and Use Cases

10.1 User Registration

As a new user, I want to register with the application by providing my email and creating a secure password.

10.2 Real-Time Messaging

As a user, I want to send messages in real-time to individual contacts or groups.

As a user, I want to receive notifications for new messages.

10.3 Message Storage

As a user, I want to view my chat history and search for specific messages.

11. Testing Requirements

11.1 Unit Testing

Individual components of the application, such as registration, authentication, and message handling, should undergo thorough unit testing.

11.2 Integration Testing

Test the integration of different modules to ensure seamless communication between components.

11.3 User Acceptance Testing (UAT)

Conduct UAT with actual users to validate that the application meets their needs and expectations.

12. Maintenance and Support

12.1 Bug Fixes and Updates

A system for identifying and fixing bugs should be established, and updates should be released periodically to improve functionality and security.

12.2 User Support

Provide a help desk or support system for users to report issues and seek assistance.

13. Risk Management

13.1 Security Risks

Regular security audits and vulnerability assessments should be conducted to mitigate potential security risks.

13.2 Data Loss Prevention

Implement measures to prevent data loss, including regular backups and redundancy in data storage.

14. Future Enhancements

14.1 Multimedia Support

Consider adding support for multimedia messages, such as images and files.

14.2 Integration with External Services

Explore integration possibilities with external services, such as cloud storage for file sharing.

15. Conclusion

This Software Requirements Specification outlines the necessary features, interfaces, and performance requirements for the development of the Chat Application with Message Storage. It serves as a reference for developers, testers, and other stakeholders involved in the project. Regular updates to this document may be necessary to reflect changes in project requirements or scope.

16. Reference

- https://www.slideshare.net/atulrockx/srs-of-3
- https://www.academia.edu/40438435/Chatting_application_SRS