**SOFTWARE**

**REQUIREMENTS SPECIFICATION**

**For**

**CHAT APPLICATION WITH MESSAGE STORAGE**

**Prepared by:-**

**KAVIN S**

**RISHI**

**DINESH KARTHICK**

## 1. Introduction

### 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the requirements for the development of a Chat Application with Message Storage. This document serves as a blueprint for the development team, ensuring a clear understanding of the functionalities, features, and constraints of the proposed software.

### 1.2 Document Conventions

Bold Text: Represents section headings.

Italic Text: Emphasizes important points.

Monospace Text: Indicates code snippets or technical terms.

### 1.3 Scope of Development Project

The Chat Application with Message Storage is intended to facilitate real-time communication between users. It will support text-based messaging, multimedia sharing, and persistent message storage, allowing users to access their conversation history across devices.

### 1.4 Definitions, Acronyms and Abbreviations

JAVA -> platform independence

SQL-> Structured query Language

ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment

SRS-> Software Requirement Specification

### 1.5 References

VIDEO : <https://www.youtube.com/watch?v=ii7tE8SdN1E>

VIDEO : <https://www.youtube.com/watch?v=_1nqY-DKP9A>

WEBLINK : <https://www.javaguides.net/2019/07/javafx-jdbc-mysql-tutorial.html>

## 2. Overall Descriptions

### 2.1 Product Perspective

Use Case Diagram of Library Management System

Student

Staff

search\_book

verify\_member

check\_limit

check\_availability

calculate\_fine

User

View\_logs

issue\_book

>>

<<

include

return\_book

renew\_book

<<

include

>>

<<

include

>>

<<

extend

>>

*searches*

*requests*

*request\_renew*

*give\_book*

add\_book

Librarian

*monitors\_request*

*take\_book*

*monitors\_renew*

update\_record

*adds\_new\_book*

*perform\_transaction\_updation*

1

\*

1

1..\*

*performs*

1

0..\*

1

0..\*

1

\*

1

\*

1

\*

1

1..\*

1

1..\*

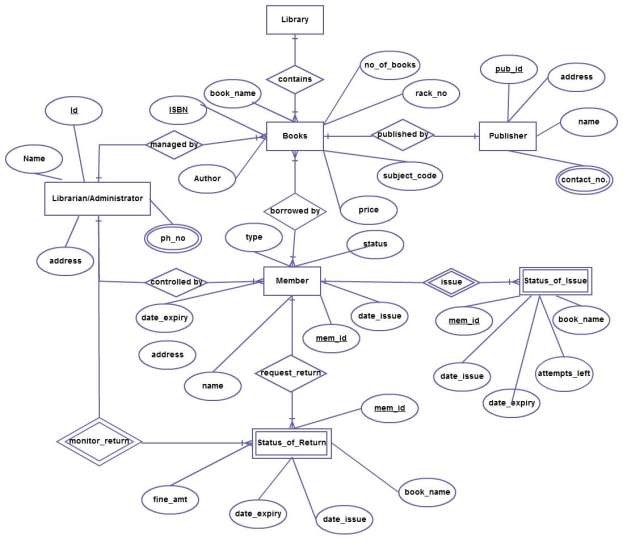
1

1..\*

The chat application will be a standalone system with integration capabilities for user authentication and potentially third-party APIs for additional features.

### 2.2 Product Function

Entity Relationship Diagram of Library Management System



The Online Library System provides online real time information about the books available in the Library and the user information. The main purpose of this project is to reduce the manual work. This software is capable of managing Book Issues, Returns, Calculating/Managing Fine, Generating various Reports for Record-Keeping according to end user requirements. The Librarian will act as the administrator to control members and manage books. The member’s status of issue/return is maintained in the library database. The member’s details can be fetched by the librarian from the database as and when required. The valid members are also allowed to view their account information.

### 2.3 User Classes and Characteristics

Regular Users:

Characteristics:

* Have the ability to send and receive text messages.
* Can create and manage personal profiles.
* Access a contact list to connect with friends and colleagues.
* Customize notification settings.
* View and manage chat history.
* Access basic chat functionalities.

Guest Users:

Characteristics:

* Limited access to the application's features.
* Can join public chat rooms as guests.
* View and participate in discussions within the chat rooms. And No access to personalized settings or chat history

### 2.4 Operating Environment

The product will be operating in windows environment. The Chat Application With Message Storage is a webApplication and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer,Google Chrome,and Mozilla Firefox.Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.

The hardware configuration include Hard Disk: 40 GB, Monitor: 15” Color monitor, Keyboard: 122 keys. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

### 2.5 Assumptions and Dependencies

a. Assumptions:

- Users have access to compatible devices with internet connectivity.

- Users are responsible for maintaining the confidentiality of their login credentials.

b. Dependencies:

- Integration with third-party authentication services.

- Database management system for message storage.

In conclusion, this SRS document serves as a foundational guide for the development team to create a feature-rich chat application with message storage capabilities. It is crucial for all stakeholders to review and provide feedback to ensure that the final product aligns with the envisioned requirements and objectives.

### 2.6 Requirement

Software Configuration:-

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database. Operating System: Windows

Language: Java Runtime Environment, Net beans 7.0.1 (front end)

Database: MS SQL Server (back end)

Hardware Configuration:-

Processor: Pentium(R)Dual-core CPU

Hard Disk: 40GB

RAM: 256 MB or more

### 2.7 Data Requirement

User Data:

Usernames: Unique identifiers for users.

Profile Information: Optional details like profile pictures, status, and bio.

Authentication Data: Securely store user credentials for login.

Message Data:

Text Messages: Content of the messages sent between users.

Multimedia Messages: Images, videos, audio files shared in the chat.

Message Timestamps: Record the time when each message is sent.

Contact Data:

Friend List: Information about a user's contacts or friends.

Contact Requests: If the application allows users to send or receive friend requests.

Group Chat Data:

Group Details: Information about group chats, including name and description.

Group Members: List of users participating in a group chat.

Admins/Moderators: Information about users with administrative privilege

## 3. External Interface Requirement

### 3.1 GUI

The software provides good graphical interface for the user to operate on the system, performing the required task such as create account , chatting.

Login Interface:-

In case the user is not yet registered, he can enter the details and register to create his account. Once his account is created he can ‘Login’ which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message appears.

Main Chat Interface:

Chat Window: A clean and organized display of ongoing conversations.

Message Input: An area for users to type and send messages.

Emoji and Sticker Support: Integration of emoticons, emojis, and stickers for expressive communication.

Multimedia Attachment: An option to attach and send images, videos, and other multimedia files.

Group Chat Features:

Create Group: An option to create new group chats.

Group Details: Accessible information about group members, settings, and activities.

Group Invitations: A mechanism for inviting users to join group chats.

## 4. System Features

Users can create, edit, and delete contacts.

Search and filter options for managing contacts efficiently.

Real-time Messaging:

Users can send and receive text messages in real-time.

Support for multimedia messages (images, videos, etc.).

Group Chats:

Users can create and participate in group chats.

Group administrators can manage participants and settings.

Notifications and Alerts:

Customizable notification settings for messages and events.

Push notifications for mobile applications.

In-app notifications for web applications.

## 5. Other Non-functional Requirements

### 5.1 Performance Requirement

Response Time:

The chat application should have an average response time for sending and receiving messages of less than 1 second under normal operating conditions.

The system should maintain an acceptable response time even during peak usage, with a maximum response time not exceeding 3 seconds.

Scalability:

The system must be designed to scale horizontally to accommodate at least a 20% increase in concurrent users within a one-year period.

The application should handle a minimum of 100,000 simultaneous connections.

Throughput:

The system should support a minimum of 1,000 messages per second during peak usage.

System Uptime:

The chat application should have a minimum uptime of 99.9%. Planned maintenance activities should not result in more than 1 hour of downtime per month.

Data Durability:

Message data must be durably stored, with a data loss tolerance of less than 0.01% per year.

### 5.2 Safety Requirement

End-to-End Encryption:

Requirement: Implement end-to-end encryption for messages, ensuring that only the intended recipients can decrypt and read the messages.

User Authentication and Authorization:

Requirement: Enforce strong user authentication mechanisms, such as multi-factor authentication, to prevent unauthorized access. Implement strict authorization controls to regulate user access to various features and data.

Secure Password Handling:

Requirement: Store user passwords securely using industry-standard encryption algorithms and practices. Implement password policies to ensure strong and unique passwords.

### 5.3 Security Requirement

Password Policies:

Requirement: Enforce strong password policies, including minimum length, complexity requirements, and password expiration.

### 5.4 Business Rules

Internal Team Communication:

Role: Enhancing Team Collaboration

Use Case: Facilitating real-time communication among team members, departments, and project groups to improve collaboration and information sharing within the organization.

Customer Support and Service:

Role: Providing Customer Assistance

Use Case: Offering a platform for customers to interact with support agents, ask questions, and receive assistance. This can include live chat, automated bots, and multimedia support for issue resolution.

### 5.6 User Requirement

User-Friendly Interface:

Users often prefer an intuitive and easy-to-navigate interface for a seamless chatting experience.

Real-Time Messaging:

The ability to send and receive messages instantly is a fundamental requirement for any chat application.

Multimedia Sharing:

Users expect the option to share images, videos, and other multimedia files within the chat.

Group Chat Functionality:

Group chat features are essential for users who want to communicate with multiple people simultaneously.

## 6. Other Requirements

### 6.1 Data and Category Requirement

User Profile Data:

Requirement: The application should collect and store user profile data, including usernames, profile pictures, and optional bio information.

Message Data:

Requirement: Store message data securely, including text messages, multimedia files, timestamps, and sender/receiver information.

Group Chat Data:

Requirement: Store information related to group chats, including group names, member lists, and group settings.

### 6.2 Appendix

API Documentation:

If the chat application provides an API for developers, include a comprehensive appendix with detailed API documentation, endpoints, request-response formats, and authentication requirements.

### 6.3 Glossary

### Provide a glossary of terms used throughout the SRS to ensure a common understanding among stakeholders. Include definitions for technical terms, acronyms, and domain-specific terminology.

### 6.4 Class Diagram