
SOFTWARE REQUIREMENTS SPECIFICATION

for

WebMAve

Version 1.0

Prepared by

Sardar Abdul Moiz Khan	CS22049	s.a.moizkhan988@gmail.com
------------------------	---------	---------------------------

Instructor: Mr. Kashif Asrar

Date: November 3, 2024

Contents

1	Introduction	5
1.1	Purpose	5
1.2	Document Conventions	5
1.3	Intended Audience and Reading Suggestions	5
1.4	Project Scope	5
1.5	References and Acknowledgments	5
2	Overall Description	7
2.1	Product Perspective	7
2.2	Product Functions	7
2.3	User Classes and Characteristics	9
2.4	Operating Environment	10
2.5	Design and Implementation Constraints	10
2.5.1	Technology Stack and Design Choices	10
2.5.2	Constraints	10
2.5.3	User Documentation	11
2.6	Assumptions and Dependencies	11
2.6.1	Assumptions	11
2.6.2	Dependencies	12
3	External Interface Requirements	13
3.1	User Interfaces	13
3.1.1	Hardware Interfaces	16
3.1.2	Software Interfaces	16
3.1.3	Communications Interfaces	16
3.2	Functional Requirements	16
3.2.1	User Authentication	16
3.2.2	Real-Time Communication	17
3.2.3	Session Management	17
3.2.4	Database Management	17
3.2.5	Security and Privacy	17
3.2.6	User Interface	17
3.3	Behavior Requirements	18
3.3.1	Use Case View	18
4	Other Nonfunctional Requirements	19
4.1	Performance Requirements	19
4.2	Safety and Security Requirements	19
4.3	Safety and Security Requirements	19
4.4	Safety and Security Requirements	19
4.4.1	Safety Requirements	20
4.4.2	Security Requirements	20
4.5	Software Quality Attributes	20
4.5.1	Reliability	20
4.5.2	Scalability	20

4.5.3	Maintainability	21
4.5.4	Usability	21
4.5.5	Adaptability	21
5	Appendix A: Glossary	22

Revision History

Name	Date	Reason For Changes	Version
------	------	--------------------	---------

1 Introduction

1.1 Purpose

This document is created to provide a detailed overview of the requirements and features which would be provided by our web software. The purpose of this document is to present a detailed description of the web video calling software WebMAve. It will explain the purpose and features of the software, along with the interfaces of the software, what the software will do and the constraints under which it must operate. This document is intended for users of the software and also potential developers.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

- Programmers who are interested in working on the project by further developing it or fix existing bugs.
- Professor or Researchers who are interested in reviewing or suggestions based on their experiences, for further growth of the project.
- Clients who wants to use our software to solve their video calling operations.

1.4 Project Scope

WebMAve is a web application that allows its users to host meetings and conferences with other users. Allow webcams and media devices to interact during the meetings. It will also allow the Users to Host the meetings in an Audience Mode (Only the Host Can interact with everyone while others can only interact with the host). This feature isn't available in similar applications in the current Market. Providing solid Market Entry for the Business.

1.5 References and Acknowledgments

This Software Requirements Specification (SRS) for WebMAve refers to the following documents and web addresses:

1. *IEEE Recommended Practice for Software Requirements Specifications*, IEEE Std 830-1998. [Online]. Available: <https://doi.org/10.1109/IEEESTD.1998.1011530>.

2. Mozilla Developer Network, “WebRTC API,” [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API.
3. Socket.IO, “Socket.IO Documentation,” [Online]. Available: <https://socket.io/docs/v4/>.
4. Mozilla Developer Network, “JavaScript,” [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
5. European Commission, “Data Protection Reform,” [Online]. Available: https://commission.europa.eu/law/law-topic/data-protection/reform/rights-citizens/my-rights_en.
6. Docker, “Get Started with Docker,” [Online]. Available: <https://docs.docker.com/>.

2 Overall Description

2.1 Product Perspective

WebMAve is designed as a new, self-contained video conferencing solution aimed at the personal as well as education and corporate sectors. Other than its basic use for one-to-one calling, WebMAve's Audience Mode allows only hosts to communicate with all users, while others are limited to host interactions, making it ideal for lectures, large seminars, and one-to-many presentations.

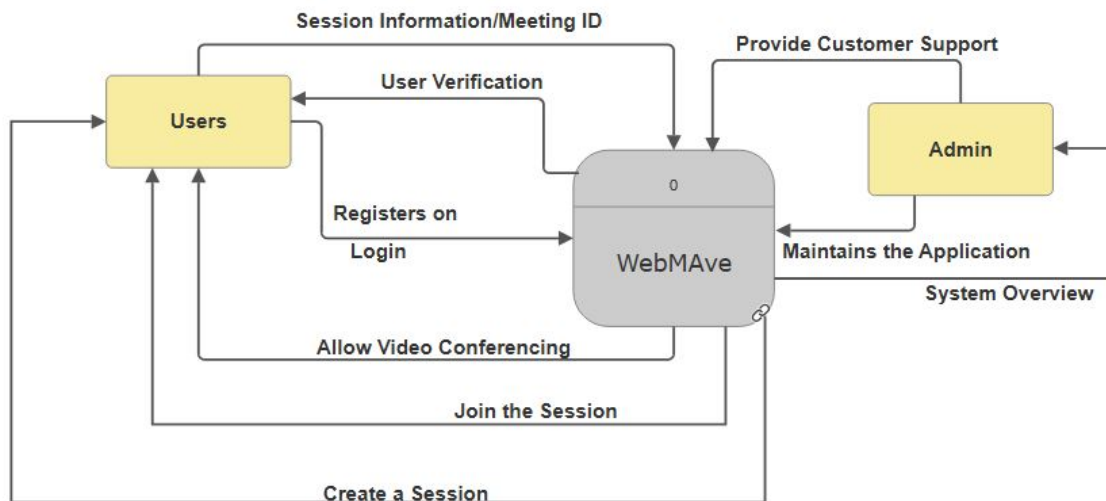


Figure 2.1: General Diagram of WebMAve System

2.2 Product Functions

The major functions of the WebMAve system are summarized as follows:

- User Registration and Authentication
 - Allow new users to register and existing users to log in.
 - Verify user credentials to grant access to the video conferencing platform.
- User Management
 - Manage user profiles, allowing updates and modifications of user details.
 - Maintain a record of all registered users for tracking and access control.
- Session Management
 - Enable users to create and join video conferencing sessions.

- Generate unique session/meeting IDs and provide session information to participants.
- Manage session details such as session status, and meeting times.
- Audience Mode and Host Interaction
 - Provide special privileges to hosts, enabling them to communicate with all users in “Audience Mode.”
 - Limit participants’ interactions to the host only, making it suitable for large presentations and lectures.
- Admin Functions
 - Allow administrators to perform system maintenance and manage overall platform health.
 - Enable administrators to view a system overview, including current sessions and user activity.
- Customer Support
 - Provide users with access to customer support for resolving issues or answering questions.
 - Record support interactions for tracking and improvement purposes.

These functions represent the features and capabilities of WebMAve, allowing for a structured understanding of the system’s main roles and operations.

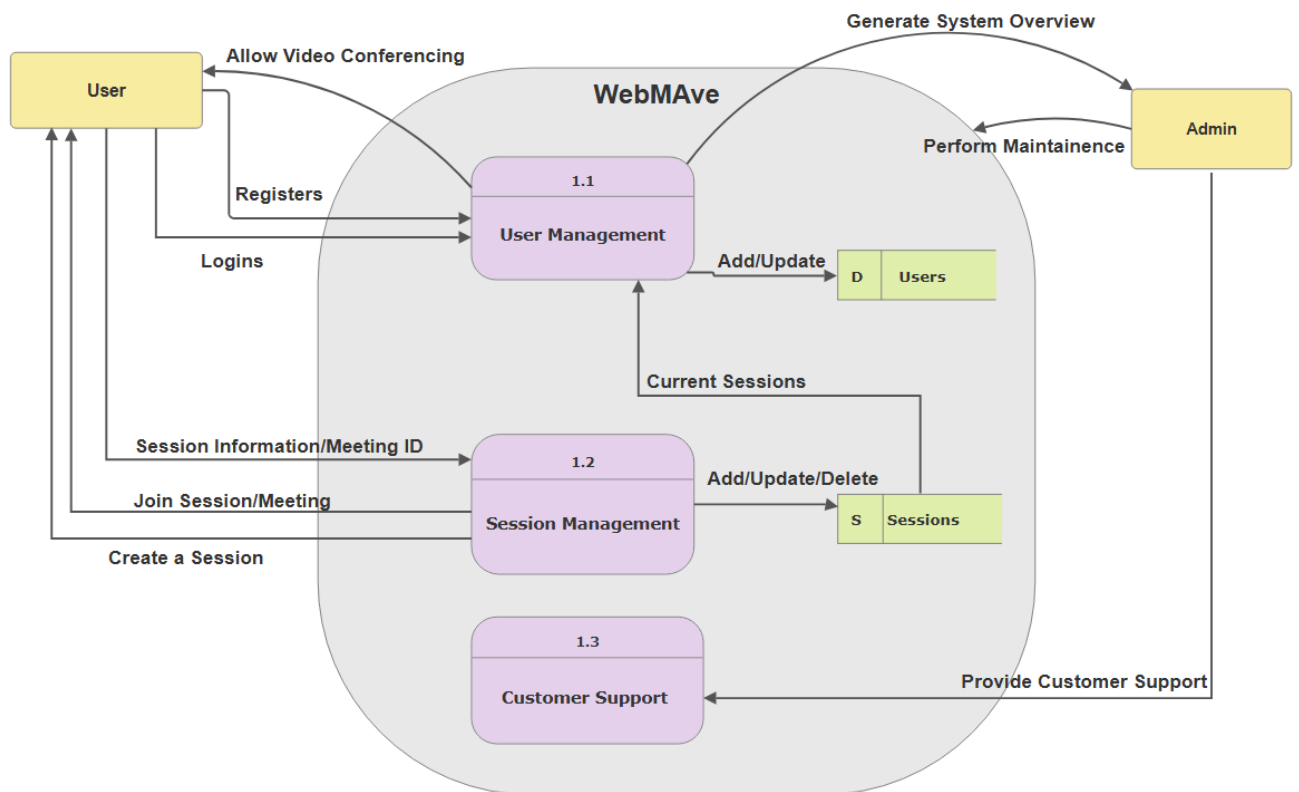


Figure 2.2: Level:1 Data-Flow Diagram of WebMAve System

2.3 User Classes and Characteristics

- **Regular Users:**
 - **Description:** Users who utilize the WebMave application for personal meetings and video calls.
 - **Typical Usage:** Engaging in casual discussions or planning social events with friends or family.
 - **Average Number of Users:** 2-4 participants per meeting.
 - **Characteristics:**
 - * **Frequency of Use:** Moderate; may use the application weekly or bi-weekly.
 - * **Technical Expertise:** Low to moderate; familiar with basic video conferencing features but may require guidance for advanced settings.
 - * **Device Usage:** Primarily smartphones or laptops.
- **Professional Users:**
 - **Description:** Users who use the WebMave application for business meetings and collaboration with colleagues or clients.
 - **Typical Usage:** Conducting team meetings, client presentations, or project collaborations.
 - **Average Number of Users:** 5-10 participants per meeting.
 - **Characteristics:**
 - * **Frequency of Use:** High; may use the application daily or several times a week.
 - * **Technical Expertise:** Moderate to high; comfortable with advanced features such as screen sharing, recording, and managing participant roles.
 - * **Device Usage:** Primarily laptops and desktop computers for enhanced functionality.
- **Educational Users:**
 - **Description:** Users who use the WebMave application for educational purposes, such as delivering lectures or seminars.
 - **Typical Usage:** Engaging students through virtual classrooms, webinars, and workshops.
 - **Average Number of Users:** 10-15 participants per session, potentially more for larger lectures.
 - **Characteristics:**
 - * **Frequency of Use:** High; may use the application several times a week, particularly during the academic year.
 - * **Technical Expertise:** Moderate; familiar with educational technology but may require training for specific features.
 - * **Device Usage:** Often utilize a mix of laptops for teaching and students using various devices (smartphones, tablets, and laptops).

2.4 Operating Environment

WebMAve is designed to operate across a variety of environments. The key components include:

- **Hardware Platform:**
 - Compatible with desktops, laptops, tablets, and smartphones.
 - Recommended: Dual-core processor, 4 GB RAM (8 GB preferred), and functional webcam/microphone.
- **Operating System:**
 - Compatible with major OS versions:
 - * Windows: 10 and above.
 - * macOS: Latest two versions.
 - * Linux: Distributions ≥ 20.04 .
 - * Mobile: iOS and Android.
- **Browser Compatibility:**
 - Optimized for the latest versions of:
 - * Chrome, Firefox, Safari, Edge.
- **Network Requirements:**
 - A stable internet connection is essential, with a minimum speed of 1 Mbps recommended for standard calls.

2.5 Design and Implementation Constraints

2.5.1 Technology Stack and Design Choices

WebMAve utilizes a modern technology stack that includes:

- **Frontend:** Next.js, chosen for simplified deployment and improved SEO, helps avoid issues associated with multi-layer application hosting.
- **Middleware:** Socket.io and WebRTC to enable real-time audio and video communication.
- **Backend:** Python frameworks, specifically Django or FastAPI, would be selected based on performance requirements.
- **Database:** MongoDB, a NoSQL database, is preferred for its flexibility in managing unstructured data.

2.5.2 Constraints

The following constraints impact the development process:

- **Middleware Dependencies:** Relying on Socket.io and WebRTC requires a stable internet connection; poor network conditions can degrade audio and video quality.

- **Backend Framework Decisions:** The choice between Django or FastAPI complicates development, necessitating careful assessment of compatibility and performance.
- **Resource Constraints from Free-Tier Deployment:** Limited resources in free-tier hosting restrict concurrent users, making it crucial to manage active call sessions effectively.
- **Cross-Browser Compatibility Challenges:** Ensuring consistent performance across browsers can be challenging and requires thorough testing to maintain a uniform user experience.
- **Scalability Limitations:** The application architecture must be capable of handling increased loads as user demand grows to prevent performance degradation during peak times.

2.5.3 User Documentation

The User documentation for WebMAve will include a recorded demo and a comprehensive user manual detailing common use-cases. The user manual will also provide instructions for various functionalities, ensuring users can navigate the application effectively. In-case of any user query or technical issue, a customer support team is there to assist and resolve any problems.

2.6 Assumptions and Dependencies

This section outlines the assumptions made during the development of WebMAve and the dependencies that could impact the project.

2.6.1 Assumptions

- **User Capacity:** It is assumed that the application will need to support simultaneous users during peak usage times. This influences the architectural around the server resources and load management.
- **Stable Internet Connection:** It is assumed that users will have access to a stable and reasonably decent internet connection, as the application relies on real-time audio and video streaming.
- **Browser Compatibility:** The software is expected to function seamlessly across major web browsers (e.g., Chrome, Firefox, Safari). Users are assumed to use updated versions of these browsers to ensure compatibility and security features.
- **User's Technical Proficiency:** It is assumed that users possess a basic level of technical proficiency, allowing them to navigate web applications without extensive guidance.
- **Availability of Third-Party Services:** The project assumes that third-party services such as Socket.io, WebRTC, and MongoDB, will remain available and supported throughout the development and operational phases.

2.6.2 Dependencies

- **Third-Party Libraries:** The project heavily relies on external libraries and frameworks such as Socket.io for real-time communication, WebRTC for media streaming, MongoDB for data storage, and FastAPI or Django are backend interactions. Changes or discontinuation of these libraries will impact the project's functionalities.
- **Hosting Services:** WebMAve is dependent on cloud hosting services that provides the necessary infrastructure to support video conferencing capabilities. Limitations of free-tier hosting restricts scalability.
- **Browser API Support:** WebMAve leverages various Browser APIs for media access, security, and real-time communication. Deprecations or changes in Browser API support could necessitate modifications to the application.
- **Network Quality:** WebMAve relies on stable network infrastructure for real-time media transmission. Variations in network latency, or bandwidth availability can impact video and audio quality, and these factors are outside the control of the application itself.

3 External Interface Requirements

3.1 User Interfaces



Figure 3.1: WebMAve WireFrame: Login Screen

Tagline

Welcome To WebMave

username

email

password

Or Sign up?

submit

Figure 3.2: WebMAve WireFrame: Signup Screen

Company Logo

Premium Video Meetings for Everyone!

Start a meeting or jump right into the action

New meetingJoin Meeting

Previously Joined Sessions

Meeting details

Meeting details

Meeting details

Figure 3.3: WebMAve WireFrame: User Dashboard Screen

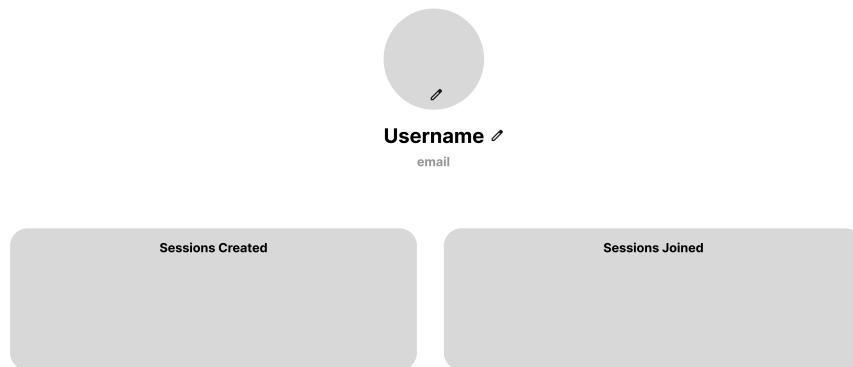


Figure 3.4: WebMAve WireFrame: Profile Screen



Figure 3.5: WebMAve WireFrame: Session Room Screen

3.1.1 Hardware Interfaces

WebMAve is a browser-based application, designed to be compatible across various devices that support modern web browsers. The following are the primary hardware interface requirements:

- **Device Compatibility:** The application can be run on any device that supports the latest versions of major web browsers.
- **Audio and Video Components:** WebMAve requires access to device's microphone and camera for real-time communication. These hardware components must be accessible via the browser's permissions for the application to function properly.
- **Network Interface:** As WebMAve is a web-based application, it requires a network interface, either Wi-Fi or cellular, to connect to the internet. The performance of video calls is dependent on the quality of this network connection.

3.1.2 Software Interfaces

WebMAve operates as a browser-based application, designed to be compatible across a range of operating systems. In WebMAve, all functionalities are enabled entirely through modern web browsers, leveraging WebRTC, HTML5, and JavaScript APIs to facilitate real-time communication.

- **Direct interaction with the operating system** is limited to standard browser permissions, such as those required for accessing the camera, microphone and screen sharing.

3.1.3 Communications Interfaces

WebMAve's communications primarily rely on WebSocket connections to enable real-time data transfer for audio, video, and chat functions. Additionally, interactions with the backend are conducted over CORS (HTTP, HTTPS) to facilitate data retrieval and management operations. Database transactions are managed securely through encrypted connections with MongoDB services, ensuring data integrity and security during transfer. Moreover, the Notifications/Emails sent would be using SSL and Microsoft's SMTP servers for communications.

3.2 Functional Requirements

This section outlines the detailed functional requirements, each area specifies the required behavior, services, and tasks that WebMAve is expected to perform.

3.2.1 User Authentication

- **User Registration:** WebMAve must provide a registration interface that allows users to create an account with unique credentials, including username, email, and password.
- **User Login and Logout:** The system must provide sign-in and sign-out functionalities, verifying user credentials while sign-in and managing active sessions.

- **Password Recovery:** Users should be able to reset their password via email if they forget their login credentials.

3.2.2 Real-Time Communication

- **Audio and Video Streaming:** WebMAve must support high-quality real-time audio and video communication.
- **Text Messaging:** Including audio and video, the platform must also offer real-time text chat during calls to support messaging.
- **Screen Sharing:** Users must be able to share their screens during a session, allowing other participants to view shared content.

3.2.3 Session Management

- **Session Creation and Joining:** WebMAve must enable users to create and join sessions.
- **Audience Mode::** Session host must have the ability to create a session in Audience Mode.
- **Participant Management:** The system must support functions for session Hosts to add, remove, or mute participants within a session.

3.2.4 Database Management

- **User Data Storage:** User profiles must be securely stored in the system.
- **Session Data Logging:** Information regarding session activities and participants should be logged in the database.
- **Database Operations:** WebMAve must provide efficient retrieval and updating of user and session data, maintaining quick response times for database transactions.

3.2.5 Security and Privacy

- **End-to-End Encryption:** Communication data, including audio, video, and text, should be encrypted during transmission to maintain user privacy.
- **User Data Protection:** Sensitive user information, including login credentials and session data, must be securely stored and protected from unauthorized access.
- **Session Authentication:** Only authenticated users should be able to join a session, with access rights/permission managed by the session host.

3.2.6 User Interface

- **Responsive Design:** The interface should be responsive and compatible, adjusting the layout and controls according to the device.
- **User Notifications:** The system must provide notifications for incoming calls, new messages, and session invitations, ensuring that users remain informed of ongoing interactions.

- **Settings and Preferences:** Users should have access to a settings section/panel, from it they can adjust various options including audio, and video preferences during the session.

3.3 Behavior Requirements

3.3.1 Use Case View

This UseCase Diagram represents the functionalities/features accessible to new as well as existing user, including the interactions between different system components

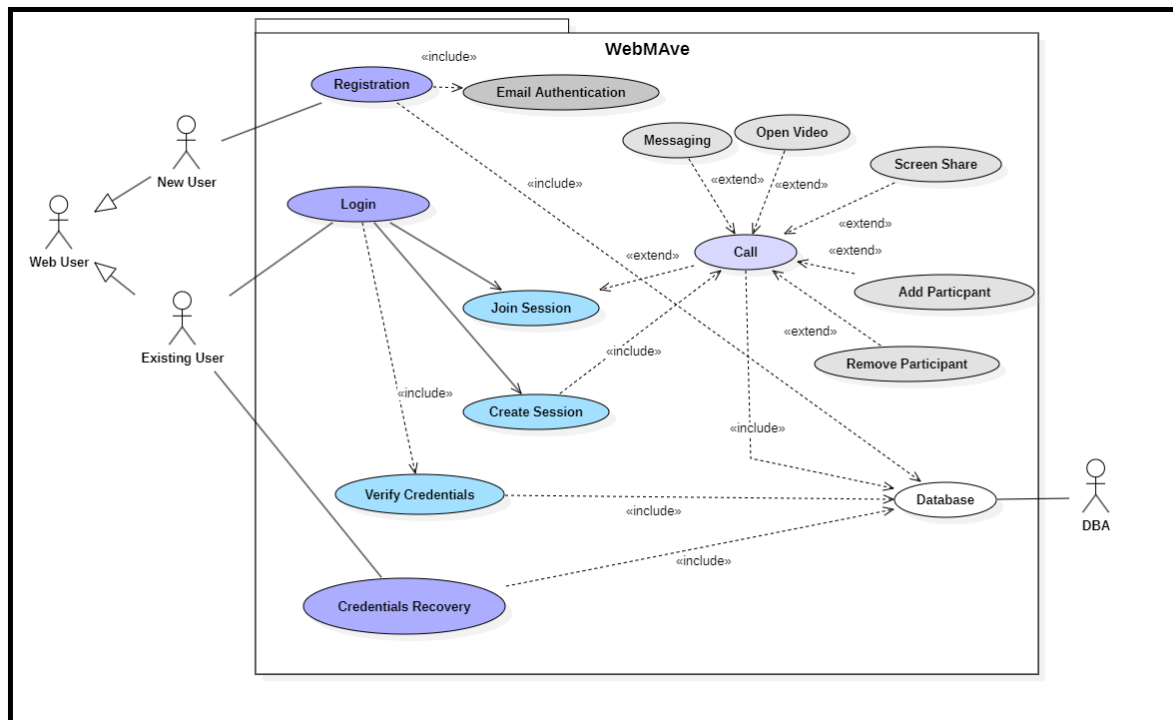


Figure 3.6: WebMAve: UseCase Diagram

4 Other Nonfunctional Requirements

4.1 Performance Requirements

WebMAve will account for the following performance requirements:

- **Authentication Speed:** Users should be able to access their accounts at all times, with authentication and account accessibility in less than 10 seconds.
- **Video and Audio Quality:** Video conferencing and Audio should maintain high quality, minimizing disturbances and noise during transmission.
- **Feature Accessibility During Sessions:** Users must have access to a variety of options during video sessions, allowing them to utilize different features without delays.
- **Database Logging Speed:** Database logging operations should complete within 10 seconds to maintain efficient data handling.
- **Application Load Time:** The application should load in under 12 seconds, even during peak usage times.
- **System Scalability:** The application should support concurrent sessions with minimal impact on performance, ensuring consistent quality as the number of users increases.

4.2 Safety and Security Requirements

This section defines the safety and security measures necessary for WebMAve to ensure data integrity, user privacy, and safe interactions within the application. WebMAve prioritizes adherence to data protection regulations and implements robust security standards to prevent unauthorized access and misuse.

4.3 Safety and Security Requirements

This section defines the safety and security measures necessary for WebMAve to ensure data integrity, user privacy, and safe interactions within the application. WebMAve prioritizes adherence to data protection regulations and implements robust security standards to prevent unauthorized access and misuse.

4.4 Safety and Security Requirements

This section defines the safety and security measures necessary for WebMAve to ensure data integrity, user privacy, and safe interactions within the application. WebMAve prioritizes adherence to data protection regulations and implements robust security standards to prevent unauthorized access and misuse.

4.4.1 Safety Requirements

WebMAve will have the following safety features:

- **Backup and Recovery:** To safeguard against data failure/loss, WebMAve will implement regular data backup protocols. In the event of a system failure, essential user data would be recovered, ensuring minimal disruption.
- **User Protection During Video Sessions:** Safety measures will be in place. During session, to avoid any inappropriate or harmful behavior an option to report would be available. Reporting the session would immediately notify the support team and trigger the session to end immediately.

4.4.2 Security Requirements

WebMAve will provide the following security requirements:

- **Authentication and Authorization:** Secure User authentication processes, such as email verification, is required to prevent unauthorized access to accounts.
- **Data Encryption:** All sensitive user data, including transmitted and stored information, will be encrypted. WebMAve will use secure and end-to-end encryption for video and audio streams.
- **Compliance with Data Privacy Regulations:** WebMAve must comply with data protection laws, including GDPR, to protect user privacy and handle data appropriately.

4.5 Software Quality Attributes

This section will outline WebMAve's functionality, usability, and sustainability features. Justifying, each quality attribute with clear objectives and planned approaches for implementation.

4.5.1 Reliability

To ensure continuous and dependable operation, WebMAve will be designed to achieve a high level of reliability. This will be accomplished through:

- **Error Handling and Recovery:** The system will incorporate error-handling mechanisms that detect and respond to common failures.
- **Redundant Data Storage:** User data will also be stored in a Backup Database to prevent losses, ensuring that any failures do not affect data integrity or availability.

4.5.2 Scalability

WebMAve must be able to adapt to increased user demands, especially during peak usage times. Scalability will be achieved by:

- **Flexible Architecture:** The application's architecture will support horizontal scaling, enabling additional servers or cloud instances to be added as user numbers grow.

- **Load Balancing:** A load balancer integrated in WebRTC, will distribute traffic efficiently across available resources, reducing latency and ensuring consistent performance during high traffic periods.

4.5.3 Maintainability

To facilitate ease of maintenance and adaptability to future changes, WebMAve will be designed with maintainability in mind. Approaches include:

- **Modular Structure:** The application's codebase will be organized into modules, allowing isolated updates or changes to individual components without affecting the entire system.
- **Clear Documentation and Coding Standards:** Detailed documentation and consistent coding standards will be maintained, making it easier for developers to understand and modify the code.
- **Version Support:** A Version history would be maintained on Github, to reflect the changes and rollback in case of issues.

4.5.4 Usability

To maximize user engagement and ease of use, WebMAve will focus on usability with the following:

- **User-Centered Interface Design:** The user interface will be intuitive, with clearly labeled controls and accessible design elements, ensuring a smooth and enjoyable user experience.
- **Cross-Device Compatibility:** WebMAve will be optimized for various devices, including desktops, tablets, and mobile phones, enabling users to access the application seamlessly from any device.

4.5.5 Adaptability

To ensure WebMAve remains relevant in a fast-evolving technological landscape, the system will be designed with adaptability in mind. Strategies include:

- **Component-Based Design:** The system will use interchangeable components, making it easier to incorporate new technologies or replace outdated modules without major rework.
- **Containerized Development:** WebMAve will be Dockerized, allowing the application to be deployed on different servers with only minor changes.

5 Appendix A: Glossary

- **WebMAve:** The name of the web based video conferencing software designed for personal, educational, and corporate use.
- **Web Application:** A software application that runs on a web server and is accessed via a web browser, allowing users to interact with it over the internet.
- **Video Calling Software:** Software that enables users to communicate through video and audio over the internet in real-time.
- **Host:** The user who initiates and manages a meeting, controlling interactions with other participants.
- **Audience Mode:** A specific mode in WebMAve where only the host can interact with participants, while others can only listen or view.
- **Meeting:** A scheduled or impromptu gathering of users via the web application for discussion or collaboration.
- **Conference:** A larger-scale meeting that may involve multiple hosts and numerous participants, often used for webinars or larger events.
- **WebRTC (Web Real-Time Communication):** An open-source project that enables real-time communication capabilities in web browsers, allowing audio, video, and data sharing.
- **Socket.IO:** A JavaScript library for real-time web applications that enables bidirectional communication between web clients and servers.
- **Media Devices:** Hardware components such as webcams, microphones, and speakers used to facilitate audio and video communication.
- **User Registration:** The process by which new users create an account on the WebMAve platform to access its services.
- **Authentication:** The verification of user credentials (such as username and password) to grant access to the video conferencing platform.
- **Session Management:** The processes involved in creating, joining, and maintaining video conferencing sessions, including the generation of unique meeting IDs.
- **Admin Functions:** Features that allow administrators to oversee the platform's health, perform maintenance tasks, and monitor user activity and sessions.
- **User Management:** The functions related to managing user profiles, including updates, modifications, and record-keeping of user details.
- **Responsive Design:** A design approach that ensures a web application's interface adjusts seamlessly to different screen sizes and orientations, enhancing usability across devices.

- **User Notifications:** Alerts or messages sent to users regarding significant events within the application, such as incoming calls, new messages, or session invitations.
- **SSL (Secure Sockets Layer):** A standard security technology for establishing an encrypted link between a web server and a browser.
- **HTTP (Hypertext Transfer Protocol):** The foundational protocol used for transmitting data on the web.
- **HTTPS (HTTP Secure):** An extension of HTTP that uses SSL/TLS to encrypt data for secure communication over a computer network.
- **CORS (Cross-Origin Resource Sharing):** A security feature implemented in web browsers that allows web applications to make requests to domains other than the one that served the web page.
- **NEXTJS:** A React-based framework that enables server-side rendering and static site generation for web applications.
- **Middleware:** Software that acts as a bridge between different systems or applications.
- **Frontend:** The client-side technology used to build the user interface of the application.
- **Backend:** The server-side technology used to handle application logic and data management.
- **Sockets:** Communication endpoints used for sending and receiving messages over a network.
- **SMTP (Simple Mail Transfer Protocol):** A protocol used for sending emails across networks.
- **Server:** A system that provides data, services, or resources to other computers, known as clients.
- **APIs (Application Programming Interfaces):** Sets of protocols and tools that allow different software applications to communicate with each other.
- **Retrieval:** The process of obtaining data from a database or storage system.
- **MongoDB:** A NoSQL database used for storing and retrieving unstructured data.
- **FASTAPI:** A modern, fast web framework for building APIs with Python based on standard Python type hints.
- **DJANGO:** A high-level Python web framework that encourages rapid development and clean, pragmatic design.
- **Credentials:** Information used to authenticate a user's identity, typically consisting of a username and password.
- **End-to-End Encryption:** A method of data transmission where only the communicating users can read the messages, providing privacy and security.