

File Sharing Application

1 Introduction

1.1 Purpose

The File Sharing Application aims to provide a secure and efficient platform for two users to share files with each other.

1.2 Description

The home page will have 2 buttons: "Join Room" and "Create Room". A file can be shared in the room created by one of the users. Another user can join the room by inserting the room number in the "Join Room" page. Each room will have a unique ID, and only two users can join a single room.

After clicking the "Create Room" button, the room address will be displayed, which can be shared with another user. The second user can join the room, and both users will be redirected to the room.

The room will have a chat-app-like interface where users can send text messages or files. File sharing will be done as follows:

1. The user will select the file they want to share, then the file will be divided into 10 MB data chunks and encrypted.
2. When the user sends the first 10 MB data chunk to the server, the server will forward it to the receiver.
3. The receiver will collect all the 10 MB data chunks and then request the next 10 MB data chunk.
4. After the server gets the request, it will forward the request to the sender, and the sender will again send the next 10 MB until the whole file is shared.
5. After the whole file is shared with the receiver, it will be decrypted and saved on the receiver's device.

1.3 Safety Mechanisms

- A user can be present in a single room at a time.

- A room can only share one file at a time.
- Only two users can join a single room.
- Every 10 seconds, the status of both users will be checked. If either user fails to respond within a minute, the room will be discarded, and all file-sharing utilities will be deleted to free up memory.

2 Overall Description

2.1 Product Perspective

The File Sharing Application is a web-based platform designed to facilitate secure and efficient file sharing between two users. It operates within a client-server architecture, where users interact with the application through their web browsers (clients), and the server manages file transfers, room management, and messaging.

2.1.1 Client-Side

On the client-side, users access the application through a web browser. The user interface provides a dashboard for managing rooms, initiating file transfers, and communicating via text messages within a room. The client-side application utilizes HTML, CSS, and JavaScript to create an intuitive and user-friendly experience.

2.1.2 Server-Side

The server-side of the application is responsible for handling the core functionalities of the File Sharing Application. It consists of several components:

- **File Transfer Manager:** Manages the division of files into 10 MB data chunks, encryption of these chunks, and their transfer between clients.
- **Room Manager:** Handles the creation of new rooms, joining existing rooms, and monitoring the status of users within each room.
- **Message Broker:** Facilitates the exchange of text messages between users within the same room.
- **Database:** Stores essential information such as user credentials, room details, and file transfer statuses.

2.2 User Interaction

2.2.1 Creating and Joining Rooms

To initiate a file sharing session, a user can create a new room by providing a unique room name or number. The room manager assigns a unique ID to

the room, and the user can then share this room identifier with another user. Alternatively, a user can join an existing room by entering the room number provided by another user.

2.2.2 File Sharing Process

Once users are in the same room, they can start sharing files. The sender selects a file from their local drive, which is then divided into 10 MB data chunks and encrypted. The first chunk is sent to the server, which forwards it to the receiver. The receiver decrypts and saves the chunk, then requests the next chunk until the entire file is received and reconstructed.

2.2.3 Chat Functionality

Within a room, users can also exchange text messages in real-time. These messages are sent via the message broker to ensure timely delivery and are displayed in the room's chat interface. Users can type messages, press "Enter" to send, and view the message history.

2.3 Product Features

2.3.1 Secure File Transfer

The application ensures secure file transfers by encrypting data chunks before transmission. This encryption prevents unauthorized access to shared files during transmission.

2.3.2 Efficient File Sharing

File sharing is optimized by dividing files into manageable 10 MB chunks. This approach minimizes network congestion and ensures smoother and faster transfers, particularly for large files.

2.3.3 Real-time Communication

The chat functionality allows users to communicate with each other in real-time within the same room. This feature enhances collaboration and coordination during file sharing sessions.

2.3.4 Room Management

Users can create new rooms for specific file sharing sessions or join existing rooms using unique room identifiers. The room manager ensures that only two users can be present in a room at a time, maintaining privacy and focused collaboration.

2.4 Constraints

2.4.1 Network Availability

The File Sharing Application relies on network connectivity for users to access the platform and transfer files. Users must have a stable internet connection to use the application effectively.

2.4.2 Client-Side Compatibility

The client-side application is compatible with modern web browsers such as Chrome, Firefox, and Safari. Users may encounter compatibility issues with older browsers or outdated versions.

2.4.3 File Size Limitations

While the application supports efficient file sharing through 10 MB data chunks, users should be aware of file size limitations based on available storage space and network bandwidth.

2.5 Assumptions and Dependencies

2.5.1 User Authentication

The application assumes that users have valid credentials to log in and access the platform. User authentication is necessary to maintain security and privacy within rooms.

2.5.2 Reliable Server Operation

The File Sharing Application relies on the server's continuous operation and availability. Users' ability to create, join, and manage rooms, as well as transfer files, depends on the server's reliability.

2.5.3 JavaScript Enabled

Users should have JavaScript enabled in their web browsers to ensure proper functionality of the client-side application. JavaScript is essential for interactive features and real-time updates.

2.6 Future Enhancements

2.6.1 Enhanced Security Measures

Future versions of the application may include additional security measures such as two-factor authentication (2FA) to further protect user accounts and data.

2.6.2 File Preview Feature

Integrating a file preview feature would allow users to preview shared files within the application before downloading them fully. This feature enhances user experience and saves time.

2.6.3 Mobile Application

Developing a mobile application for the File Sharing Application would extend its accessibility, allowing users to share files and communicate on-the-go using their smartphones or tablets.

2.7 Glossary

- **File Sharing Application:** The web-based platform designed for secure and efficient file sharing between two users.
- **Room Manager:** The server-side component responsible for creating, managing, and monitoring rooms within the application.
- **File Transfer Manager:** The server-side component that handles file division into data chunks, encryption, and transfer between clients.
- **Message Broker:** The server-side component facilitating real-time text message exchange between users within the same room.

2.8 Product Functions

The File Sharing Application offers several key functions and features to facilitate secure and efficient file sharing between users. These functions include:

2.8.1 Room Creation and Joining

- **Create Room:** Users can create a new room by providing a unique room name or number. The room is assigned a unique identifier, and users can share this identifier with another user to invite them to the room.
- **Join Room:** Users can join an existing room by entering the room number provided by another user. Once in the same room, users can start sharing files and communicating.

2.8.2 File Sharing

- **Share File:** Within a room, users can share files with each other. To share a file, the sender selects the desired file from their local drive, and the application automatically divides it into 10 MB data chunks.

- **Encrypted Transmission:** Before transmission, each data chunk is encrypted to ensure secure transfer between users. This encryption prevents unauthorized access to the shared files during transmission over the network.
- **Progressive File Transfer:** The first 10 MB data chunk is sent to the server, which forwards it to the receiver. Upon receiving the chunk, the receiver decrypts and saves it, then requests the next chunk. This process continues until the entire file is reconstructed on the receiver's device.
- **File Download:** Once all data chunks are received and reconstructed, the receiver can download the complete file to their local drive.

2.8.3 Real-time Messaging

- **Chat Functionality:** Users within the same room can communicate via real-time text messages. The chat interface allows users to type messages, send them, and view the message history.
- **Message Broker:** Text messages are sent via the message broker, ensuring timely delivery and synchronization between users in the same room.
- **Message Timestamps:** Each message in the chat history is timestamped, providing users with a reference for when messages were sent.

2.8.4 Room Management

- **User Status Monitoring:** The application regularly checks the status of users within a room. If a user fails to respond within a certain time frame, the room is discarded, and file-sharing utilities are deleted to free up memory.
- **Single Room Limitation:** Users can only be present in one room at a time. This limitation ensures focused collaboration and prevents user confusion across multiple rooms.
- **File Limitation:** Each room can only have one file shared at a time. If a new file needs to be shared, the previous file must be fully received and downloaded first.

2.9 User Roles

The File Sharing Application does not have distinct user roles. However, it operates on a simple two-user model within each room:

- **Sender:** The user who initiates the file sharing process by selecting and sharing a file within a room.
- **Receiver:** The user who receives the shared file chunks, decrypts them, and reconstructs the file on their device.

2.10 Use Case Example

Consider the following use case scenario to better understand the functionality of the File Sharing Application:

2.10.1 Scenario: Sharing a Presentation

Sarah needs to share a large presentation file with her colleague, John. Here's how she uses the File Sharing Application:

1. Sarah logs into the application and creates a new room named "Presentation Room."
2. The application generates a unique room number (e.g., Room 123456) for the "Presentation Room."
3. Sarah shares the room number (Room 123456) with John via email.
4. John receives the room number and logs into the application.
5. John enters the room number (Room 123456) to join the "Presentation Room."
6. Once both Sarah and John are in the room, Sarah selects the presentation file from her local drive and clicks "Share File."
7. The File Transfer Manager divides the presentation file into 10 MB data chunks, encrypts them, and sends the first chunk to the server.
8. The server forwards the first chunk to John, who receives, decrypts, and saves it on his device.
9. John's application automatically requests the next chunk, and the process continues until the entire presentation file is reconstructed on John's device.
10. Once the file transfer is complete, John can download the entire presentation file to his local drive for viewing and editing.
11. Throughout the process, Sarah and John can also exchange real-time text messages in the chat interface to discuss the presentation or ask questions.

This use case demonstrates how the File Sharing Application facilitates the seamless sharing of large files between users while ensuring security, efficiency, and real-time communication within a room.

2.11 User Classes and Characteristics

The File Sharing Application is designed to cater to the following user classes with their respective characteristics:

2.11.1 General Users

- **Description:** General users are individuals or professionals who need to share files securely and efficiently with specific individuals.
- **Characteristics:**
 - Have basic computer literacy and familiarity with web browsers.
 - Require a simple and intuitive platform for file sharing.
 - Often collaborate with colleagues, clients, or partners on projects or documents.
 - Value security and privacy when sharing sensitive files or information.
 - May work in various industries such as business, education, health-care, etc.
- **Example:** Sarah, a marketing manager, needs to share campaign materials with her team members for review and collaboration.

2.11.2 Collaborative Teams

- **Description:** Collaborative teams consist of multiple individuals working together on projects or tasks requiring file sharing and real-time communication.
- **Characteristics:**
 - Members may be located in different geographical locations.
 - Need a centralized platform for file sharing and discussion.
 - Work on shared documents, presentations, design files, etc.
 - Require efficient and secure ways to exchange files and ideas.
- **Example:** A design team working on a new product line needs to share design files, images, and specifications for collaboration and feedback.

2.11.3 Educational Institutions

- **Description:** Educational institutions such as schools, colleges, and universities can use the File Sharing Application for sharing study materials, assignments, and resources among students and faculty.
- **Characteristics:**
 - Students and teachers who need to share study materials, lecture notes, and assignments.
 - Faculty members sharing research papers, academic articles, and course materials.
 - Remote learning scenarios where students access resources from home.

- Need for a secure and efficient platform for educational content distribution.
- **Example:** A professor sharing lecture slides and supplementary materials with students enrolled in an online course.

2.11.4 Small Businesses and Freelancers

- **Description:** Small businesses, startups, and freelancers often require a cost-effective solution for file sharing and collaboration with clients and partners.
- **Characteristics:**
 - Small teams or individuals working on projects with external collaborators.
 - Need to share project files, contracts, proposals, and other business documents.
 - Limited IT resources and budgets for complex collaboration tools.
 - Focus on efficiency, ease of use, and professional presentation.
- **Example:** A freelance graphic designer sharing design concepts and project files with a client for feedback and approval.

2.12 User Characteristics Summary

- **Technical Proficiency:** Users are generally expected to have basic computer skills and familiarity with web browsing.
- **Collaboration Needs:** Target users often collaborate with others on projects or tasks, requiring efficient file sharing and real-time communication.
- **Privacy Concerns:** Security and privacy are essential, especially when sharing sensitive or confidential files.
- **Diverse Industries:** Users come from various industries such as business, education, healthcare, design, and more.
- **Remote Work:** With the rise of remote work and online collaboration, users may need to access and share files from different locations.

Understanding these user classes and their characteristics helps tailor the File Sharing Application to meet their specific needs, ensuring a user-friendly and efficient platform for secure file sharing and collaboration.

2.13 Operating Environment

The File Sharing Application is designed to run in a web-based environment, providing users with flexibility and accessibility across different devices and platforms. Below are the specifications for the operating environment:

2.13.1 Client-Side Requirements

Users can access the File Sharing Application through modern web browsers on various devices, including desktop computers, laptops, tablets, and smartphones. The following are the recommended specifications for the client-side environment:

- **Web Browsers:** The application is compatible with the following web browsers:
 - Google Chrome (latest version)
 - Mozilla Firefox (latest version)
 - Apple Safari (latest version)
 - Microsoft Edge (latest version)
- **Operating Systems:** The File Sharing Application supports the following operating systems:
 - Windows 10 or later
 - macOS (latest version)
 - iOS (latest version)
 - Android (latest version)
- **Internet Connection:** A stable internet connection is required for accessing the application and transferring files. Users with low bandwidth connections may experience slower file transfer speeds.
- **JavaScript:** Users should have JavaScript enabled in their web browsers to ensure the proper functioning of the client-side application. JavaScript is essential for interactive features and real-time updates.
- **Screen Resolution:** The application is designed to be responsive and adapt to different screen sizes. However, a minimum screen resolution of 1024x768 pixels is recommended for an optimal user experience.

2.13.2 Server-Side Requirements

The File Sharing Application's server-side components are hosted on a dedicated server or cloud platform. The following are the recommended specifications for the server-side environment:

- **Operating System:** The server can run on Linux (e.g., Ubuntu, CentOS) or Windows Server.
- **Web Server:** The application can be deployed on a web server such as Apache, Nginx, or Microsoft IIS.
- **Programming Language and Framework:** The server-side components are developed using [Specify the programming language(s)] and [List any relevant frameworks or libraries].
- **Database:** The application uses a [Specify the database technology] database for storing user data, room information, and file transfer details.
- **Network Security:** The server should have proper firewall configurations and SSL/TLS encryption to ensure secure data transmission between clients and the server.
- **Memory and Storage:** Adequate memory and storage space should be available on the server to handle file transfers and user data storage.

2.13.3 Third-Party Integrations

The File Sharing Application may integrate with third-party services for additional functionality or authentication. These integrations may include:

- **Authentication Services:** Integration with OAuth providers (e.g., Google, Facebook) for user authentication.
- **Cloud Storage:** Integration with cloud storage services (e.g., Google Drive, Dropbox) for direct file access and storage.
- **Email Services:** Integration with email services (e.g., SMTP) for sending notifications and invitations to users.

Users and administrators should ensure that the operating environment meets these requirements for optimal performance and functionality of the File Sharing Application.

2.14 Design and Implementation Constraints

The design and implementation of the File Sharing Application are subject to certain constraints that impact its development and functionality. These constraints include:

2.14.1 File Size Limitations

Due to the nature of file sharing and transmission over networks, the File Sharing Application imposes a file size limitation for individual file uploads. The application divides files into 10 MB data chunks for efficient transfer. While

this approach optimizes file sharing, it also means that very large files may take longer to transfer, and users may need to wait for all chunks to be received and reconstructed.

2.14.2 Network Bandwidth

The speed and efficiency of file transfers in the File Sharing Application are dependent on the users' network bandwidth. Users with limited bandwidth may experience slower upload and download speeds, potentially affecting the overall performance of the application. It's recommended that users have a stable and reasonably fast internet connection for optimal file sharing.

2.14.3 Client-Side Compatibility

While the File Sharing Application is designed to be compatible with modern web browsers (Chrome, Firefox, Safari, Edge), users may encounter compatibility issues with older browser versions. JavaScript must also be enabled for proper functionality. Users should ensure they have the latest browser versions and JavaScript enabled for the best user experience.

2.14.4 Server Capacity

The server-side components of the File Sharing Application require sufficient capacity to handle concurrent file transfers, room management, and messaging. Adequate memory and storage resources are essential to ensure smooth operation, especially during peak usage periods. Administrators should monitor server capacity and scale resources as needed to maintain optimal performance.

2.14.5 Security Considerations

The File Sharing Application places a strong emphasis on security to protect user data and files during transmission. However, the security measures implemented are subject to constraints such as:

- **Encryption/Decryption Overhead:** The encryption and decryption processes for file chunks can introduce some overhead, particularly for very large files. This overhead may impact file transfer speeds and overall performance.
- **Third-Party Integrations:** If the application integrates with third-party services for authentication or cloud storage, it must adhere to the security protocols and constraints of those services. This may limit certain security configurations or introduce additional vulnerabilities.
- **User Authentication:** The application relies on user authentication for access control and privacy. Users must have valid credentials, but the strength of passwords and authentication methods may vary, affecting overall security.

2.14.6 Database Scalability

The database used by the File Sharing Application needs to be scalable to accommodate growing user bases and increasing file transfer activity. As the number of users and files in the system grows, the database must handle the load efficiently to maintain responsiveness and performance. Administrators should plan for database scaling strategies as the application usage grows.

2.14.7 Regulatory Compliance

Depending on the industries and jurisdictions where the File Sharing Application is used, there may be regulatory requirements related to data privacy and security. The application must adhere to these regulations, which may impose constraints on data handling, encryption standards, and user consent mechanisms.

2.15 Conclusion

These design and implementation constraints guide the development and operation of the File Sharing Application. By understanding and addressing these constraints, developers and administrators can ensure that the application functions optimally, meets user expectations, and complies with relevant security and regulatory standards.

2.16 User Documentation

The File Sharing Application provides comprehensive user documentation to guide users through its features, functionalities, and usage. The following user documentation will be provided:

2.16.1 User Manual

A detailed User Manual will be available to users, providing step-by-step instructions on how to use the File Sharing Application. The User Manual covers the following topics:

- **Getting Started:** Instructions on creating an account, logging in, and navigating the dashboard.
- **Room Creation and Joining:** Steps to create a new room, join an existing room, and share room numbers with other users.
- **File Sharing:** Guidelines on sharing files, downloading received files, and monitoring file transfer progress.
- **Chat Functionality:** How to use the real-time chat feature within rooms for communication.

- **Room Management:** Instructions on managing rooms, including checking user status, room deletion, and limitations.
- **Troubleshooting:** Common issues users may encounter and troubleshooting steps to resolve them.

2.16.2 FAQ Section

The File Sharing Application includes a Frequently Asked Questions (FAQ) section to address common queries and concerns users may have. The FAQ section covers topics such as:

- **Account Management:** Questions related to account creation, password resets, and account settings.
- **File Sharing:** Common queries about file size limits, file types supported, and download speeds.
- **Chat Functionality:** Questions about using the chat feature, message history, and notifications.
- **Room Management:** Information on creating, joining, and managing rooms, as well as room deletion.
- **Technical Issues:** Troubleshooting tips for network connectivity, browser compatibility, and error messages.

2.16.3 Help Section within the Application

The File Sharing Application includes an integrated Help section accessible within the application interface. Users can access this Help section for quick assistance while using the application. The Help section provides:

- **Contextual Guidance:** Help tips and tooltips within the application interface to guide users through specific tasks.
- **Quick Links:** Direct links to relevant sections of the User Manual and FAQ for more detailed information.
- **Contact Information:** Information on how to contact customer support or technical assistance for further help.

2.16.4 Updates and Notifications

Users will receive notifications within the application regarding updates, new features, and any changes to the user documentation. The application will also provide prompts to users to review the User Manual or FAQ when relevant, ensuring that users stay informed about the latest changes and best practices.

2.16.5 Conclusion

The File Sharing Application is accompanied by robust user documentation to support users in effectively using the platform. From detailed step-by-step instructions in the User Manual to quick reference in the FAQ section and integrated Help within the application, users have access to the information they need to navigate and utilize the File Sharing Application efficiently.

2.17 Assumptions and Dependencies

The development and operation of the File Sharing Application are based on certain assumptions and dependencies that play a crucial role in its functionality and performance. These include:

2.17.1 User Authentication

- **Assumption:** The application assumes that users have valid credentials (username and password) to log in and access the platform.
- **Dependency:** User authentication is essential for maintaining the security and privacy of user data within the application. The authentication process relies on secure storage of user credentials and encryption methods to protect sensitive information.

2.17.2 Network Connectivity

- **Assumption:** Users are assumed to have a stable and reliable internet connection to access the File Sharing Application.
- **Dependency:** The application's functionality, including file sharing, real-time messaging, and room management, depends on network connectivity. Users with poor or intermittent internet connections may experience delays or disruptions in file transfers and communication.

2.17.3 Client-Side Technology

- **Assumption:** Users are assumed to have modern web browsers with JavaScript enabled for accessing the File Sharing Application.
- **Dependency:** The client-side functionality of the application relies on JavaScript for interactive features, real-time updates, and efficient file handling. Users must have compatible web browsers (Chrome, Firefox, Safari, Edge) to ensure optimal performance.

2.17.4 Third-Party Integrations

- **Assumption:** The application may integrate with third-party services for authentication (e.g., OAuth providers) or cloud storage (e.g., Google Drive, Dropbox).

- **Dependency:** Integration with third-party services introduces dependencies on their availability and compatibility. The application must adhere to the APIs and security protocols of these services for seamless integration and data exchange.

2.17.5 File Size Limitations

- **Assumption:** The application divides files into 10 MB data chunks for efficient transfer.
- **Dependency:** Users are expected to comply with the file size limitations set by the application. Very large files may need to be split manually before sharing to fit within the 10 MB chunk size.

2.17.6 Server-Side Infrastructure

- **Assumption:** The application assumes a reliable and scalable server-side infrastructure to handle file transfers, room management, and messaging.
- **Dependency:** The server-side components of the application rely on adequate server capacity, memory, and storage to ensure optimal performance. Server maintenance and monitoring are necessary to address potential issues and ensure uptime.

2.17.7 Regulatory Compliance

- **Assumption:** The application assumes compliance with relevant data privacy and security regulations based on the industries and jurisdictions where it is used.
- **Dependency:** Compliance with regulations such as GDPR, HIPAA, or others may impact the application's design, data handling practices, and security measures. The application must adhere to these regulations to protect user data and ensure legal compliance.

2.18 Conclusion

The File Sharing Application operates under these assumptions and dependencies to provide users with a secure, efficient, and reliable platform for file sharing and collaboration. Understanding these factors is crucial for both users and administrators to ensure the application's proper functionality and compliance with relevant standards and regulations.

3 External Interface Requirements

3.1 User Interfaces

The File Sharing Application provides a user-friendly and intuitive interface designed to facilitate seamless file sharing and communication between users. The user interfaces (UIs) of the application include the following components:

3.1.1 Login Page

- The Login Page is the entry point for users to access the application.
- Users are prompted to enter their credentials (username and password) to log in.
- Forgot password link: Allows users to reset their passwords if needed.
- Sign-up link: New users can create an account by accessing the sign-up page.

3.1.2 Dashboard

- Upon successful login, users are directed to the Dashboard, the central hub of the application.
- The Dashboard provides an overview of the user's rooms, ongoing file transfers, and recent chat messages.
- Room Management: Users can create new rooms, join existing rooms, and view their active rooms.
- File Transfer Status: Displays the status of ongoing file transfers, including progress bars and completion status.
- Chat Interface: A real-time chat interface allows users to communicate with others within the same room.

3.1.3 Room Interface

- Each room in the application has its dedicated interface for file sharing and communication.
- File Upload: Users can upload files from their local drive to share within the room.
- File List: Displays the files shared within the room, including file names, sizes, and download options.
- File Transfer Progress: Shows the progress of file uploads and downloads, indicating completion percentage.

- **Chat Window:** A chat window within the room interface allows users to send and receive real-time text messages.
- **User Status:** Displays the online/offline status of other users in the room.

3.1.4 File Preview

- When a file is selected for download, users can preview the file before downloading it.
- Supported file types (PDF, images, documents) can be previewed within the application.
- **Download Button:** Users can choose to download the file to their local device from the preview window.

3.1.5 Settings Page

- The Settings Page allows users to customize their account and application preferences.
- **Account Settings:** Users can update their profile information, change passwords, and manage notifications.
- **Theme and Layout:** Options to change the theme, layout, and language preferences.
- **Help and Support:** Access to user documentation, FAQ, and contact information for customer support.

3.1.6 Notifications

- The application provides notifications to users for various events and activities.
- **New Room Invitation:** Users receive notifications when invited to join a new room.
- **File Transfer Updates:** Notifications for file upload/download progress and completion.
- **Chat Messages:** Real-time notifications for new messages in active rooms.

3.2 Mockups

Below are mockups illustrating the user interfaces of the File Sharing Application:

These user interface components and mockups provide users with a clear and intuitive way to interact with the File Sharing Application, making file sharing, communication, and management tasks straightforward and efficient.

3.3 Hardware Interfaces

The File Sharing Application primarily operates in a web-based environment, requiring minimal hardware interfaces. However, certain hardware components and requirements are necessary for users to access and use the application effectively:

3.3.1 User Devices

- **Desktop/Laptop Computers:** Users can access the File Sharing Application using desktop or laptop computers with the following specifications:
 - Processor: Dual-core CPU or higher
 - RAM: 4 GB or more
 - Storage: Sufficient free storage space for browser cache and downloaded files
 - Network: Ethernet or Wi-Fi connection for internet access
 - Display: Monitor with at least 1024x768 resolution for optimal viewing
- **Mobile Devices:** The application is also accessible on mobile devices such as smartphones and tablets with the following requirements:
 - Operating System: iOS (latest version) or Android (latest version)
 - Screen Size: Suitable screen size for comfortable interaction (e.g., 5 inches or larger for smartphones)
 - Network: Cellular data or Wi-Fi connection for internet access

3.3.2 Web Browsers

- The File Sharing Application is compatible with modern web browsers on various devices, including:
 - Google Chrome (latest version)
 - Mozilla Firefox (latest version)
 - Apple Safari (latest version)
 - Microsoft Edge (latest version)
- Users should ensure that their web browsers are up-to-date to experience optimal performance and compatibility with the application.

3.3.3 Network Infrastructure

- Users need a stable and reliable internet connection to access and use the File Sharing Application effectively.
- Minimum recommended internet speed:
 - Download: 5 Mbps or higher
 - Upload: 2 Mbps or higher
- The application's performance, especially during file transfers and real-time messaging, can be impacted by network latency and bandwidth limitations.

3.4 Conclusion

The File Sharing Application's hardware interfaces are primarily centered around user devices such as desktop computers, laptops, and mobile devices. Users are required to have devices with adequate specifications and modern web browsers for accessing the application. Additionally, a stable internet connection with sufficient bandwidth is essential for optimal performance and responsiveness while using the application.

3.5 Software Interfaces

The File Sharing Application interacts with various software components, libraries, and APIs to provide its functionality. These software interfaces are essential for file management, encryption, network communication, and user interface development. The key software interfaces include:

3.5.1 Programming Languages and Frameworks

- **Programming Languages:** The application is developed primarily using [Specify the programming language(s)] for both client-side and server-side components.
- **Web Framework:** [Specify the web framework], a [MVC/RESTful] framework, is used to build the application's web-based user interface and server-side logic.
- **Client-Side Framework:** [Specify the client-side framework/library], such as React.js or Angular, is utilized for dynamic and responsive user interfaces.

3.5.2 Database

- **Database Technology:** The application uses [Specify the database technology] as the backend database for storing user data, room information, and file metadata.

- **Database Management System (DBMS):** [Specify the DBMS], such as MySQL or PostgreSQL, is used to interact with the database.
- **Object-Relational Mapping (ORM):** [Specify the ORM], such as Sequelize or Hibernate, is employed for mapping database entities to application objects.

3.5.3 File Management

- **File Upload/Download:** The application interacts with a file upload/download library or API to handle file transfer operations.
- **File Chunking:** [Specify the file chunking library], such as Dropzone.js or Fine Uploader, is used to divide files into 10 MB data chunks for efficient transfer.
- **File Encryption:** The application utilizes [Specify the encryption library], such as CryptoJS or OpenSSL, for encrypting and decrypting file data chunks before transmission.

3.5.4 Network Communication

- **HTTP/HTTPS Protocol:** The application communicates with the server using the HTTP/HTTPS protocol for transmitting data between the client and server.
- **WebSocket:** [Specify the WebSocket library], such as Socket.IO or WebSocket API, is used for real-time messaging and notifications within rooms.
- **RESTful API:** The server exposes a RESTful API for client-server communication, allowing clients to perform operations such as room creation, file sharing, and user authentication.

3.5.5 Third-Party Integrations

- **OAuth Integration:** The application integrates with OAuth providers (e.g., Google, Facebook) for user authentication and authorization.
- **Cloud Storage API:** [Specify the cloud storage API], such as Google Drive API or Dropbox API, is used for integrating with cloud storage services for direct file access and storage.
- **Email Service API:** [Specify the email service API], such as SMTP or SendGrid API, is utilized for sending notifications and invitations to users via email.

3.5.6 Dependencies

- The File Sharing Application relies on the availability and compatibility of the aforementioned software interfaces.
- Third-party APIs and libraries must be properly integrated and configured for seamless functionality.
- Updates or changes to software interfaces may impact the application's performance and behavior, requiring version compatibility checks.
- Developers and administrators should stay informed about updates and best practices for the software interfaces used in the application.

3.6 Conclusion

The File Sharing Application interacts with various software interfaces, libraries, and APIs to provide its core functionalities. These interfaces include programming languages, web frameworks, databases, file management tools, network communication protocols, and third-party integrations. Understanding and properly configuring these software components are crucial for the application's performance, security, and user experience.

3.7 Communications Interfaces

The File Sharing Application utilizes various communication interfaces and protocols for efficient data exchange and real-time messaging between clients (users) and the server. These communication interfaces ensure reliable and secure transmission of data within the application. The key communication interfaces include:

3.7.1 HTTP/HTTPS Protocol

- The File Sharing Application uses the HTTP/HTTPS protocol for communication between clients (web browsers) and the server.
- **Purpose:** HTTP/HTTPS is used for standard web-based communication, including sending and receiving HTTP requests and responses.
- **Functionality:** Clients send HTTP requests to the server to perform actions such as user authentication, room creation, file upload/download, and messaging.
- **Security:** When using HTTPS, data is encrypted during transmission, ensuring confidentiality and integrity.

3.7.2 WebSocket Protocol

- The File Sharing Application employs the WebSocket protocol for real-time messaging and notifications within rooms.
- **Purpose:** WebSocket enables full-duplex communication channels over a single TCP connection, allowing for efficient and low-latency messaging.
- **Functionality:** Clients establish WebSocket connections with the server to send and receive real-time messages, such as chat messages and file transfer status updates.
- **Advantages:** WebSocket provides advantages over traditional HTTP for real-time applications, including reduced latency, improved performance, and lower overhead.

3.7.3 RESTful API

- The File Sharing Application exposes a RESTful API for client-server communication, providing a set of endpoints for performing operations.
- **Purpose:** The RESTful API allows clients to interact with the server using standard HTTP methods (GET, POST, PUT, DELETE) for various functionalities.
- **Functionality:** Clients can make API requests to the server for tasks such as creating rooms, sharing files, retrieving file lists, and updating user information.
- **Security:** The API endpoints are secured using authentication mechanisms (e.g., OAuth tokens) to ensure authorized access to sensitive operations.

3.7.4 WebSockets Over HTTPS (WSS)

- For WebSocket communication, the application uses the secure WebSocket protocol (WSS) over HTTPS for encrypted and secure messaging.
- **Purpose:** WSS provides secure WebSocket connections by encrypting data using SSL/TLS, ensuring confidentiality and integrity of messages.
- **Functionality:** Clients establish WSS connections with the server for real-time messaging within rooms, including chat messages and file transfer updates.
- **Security:** WSS ensures that all data transmitted between clients and the server is encrypted, protecting against eavesdropping and tampering.

3.8 Conclusion

The File Sharing Application utilizes multiple communication interfaces and protocols to facilitate efficient and secure data exchange between clients and the server. HTTP/HTTPS is used for standard web-based communication, while WebSocket and RESTful API enable real-time messaging and client-server interactions. The application ensures data confidentiality and integrity by employing secure WebSocket (WSS) connections over HTTPS. These communication interfaces play a vital role in delivering a seamless and responsive user experience within the application.

4 System Features

4.1 System Feature 1: User Management

The User Management feature of the File Sharing Application provides functionalities for creating user accounts, authentication, and managing user interactions within the system. These features ensure secure access to the application and proper handling of user-related operations. The key aspects of the User Management feature include:

4.1.1 User Registration

- **Purpose:** Allows new users to create accounts and gain access to the application.
- **Functionality:** Users can register by providing necessary information such as username, email address, and password.
- **Validation:** Input validation is performed to ensure the uniqueness of usernames and valid email formats.
- **Confirmation:** Users may need to confirm their email addresses through a verification link sent to the provided email.

4.1.2 User Authentication

- **Purpose:** Verifies the identity of users to grant access to their accounts.
- **Functionality:** Users authenticate by entering their credentials (username and password) on the Login Page.
- **Session Management:** Upon successful authentication, the server creates a session or issues a token for subsequent requests, maintaining the user's logged-in state.
- **Password Security:** Passwords are securely hashed and stored in the database to protect user accounts from unauthorized access.

4.1.3 User Profile Management

- **Purpose:** Allows users to view and update their profile information.
- **Functionality:** Users can access the Settings Page to modify their profile details such as name, email, and profile picture.
- **Password Reset:** Users can request a password reset if they forget their password, receiving a password reset link via email.
- **Preferences:** Users can customize their application preferences, such as notification settings and theme selection.

4.1.4 User Roles and Permissions

- **Purpose:** Assigns specific roles and permissions to users based on their access level.
- **Roles:** The application may have roles such as Admin, Regular User, or Guest, each with different privileges.
- **Permissions:** Admin users may have additional privileges such as room creation, user management, and system configuration.
- **Access Control:** Role-based access control (RBAC) ensures that users can only perform actions permitted by their role.

4.1.5 User Session Management

- **Purpose:** Manages user sessions to maintain security and control access to the application.
- **Session Expiry:** Sessions have a timeout period to automatically log out inactive users for security reasons.
- **Logout:** Users can manually log out from their accounts to end their session and prevent unauthorized access.
- **Multiple Device Support:** Users can access the application from multiple devices, and sessions are synchronized across devices.

4.2 Conclusion

The User Management feature of the File Sharing Application ensures secure access, authentication, and user interaction within the system. From user registration and authentication to profile management and role-based permissions, these functionalities provide users with a seamless and personalized experience while maintaining the security and integrity of the application.

4.3 System Feature 2: File Sharing

The File Sharing feature of the application enables users to share files securely and efficiently within rooms. Users can upload files, divide them into 10 MB data chunks, encrypt the chunks before transmission, and monitor the progress of file transfers. The key aspects of the File Sharing feature include:

4.3.1 File Upload

- **Purpose:** Allows users to upload files from their local devices to share within a room.
- **Functionality:** Users can select files through the interface or drag-and-drop files to initiate the upload process.
- **File Type Support:** Supports various file types such as documents, images, videos, and archives.
- **File Size Limitation:** Users are informed about the maximum file size allowed for upload (10 MB per chunk).

4.3.2 File Division and Encryption

- **Purpose:** Divides uploaded files into 10 MB data chunks for efficient transfer and encryption.
- **Functionality:** Upon upload, files are automatically divided into 10 MB chunks on the client-side.
- **Encryption:** Each data chunk is encrypted using [Specify the encryption algorithm or library], ensuring data security during transmission.
- **Client-Side Encryption:** Encryption occurs on the client-side before data is sent to the server, enhancing security.

4.3.3 File Transmission

- **Purpose:** Facilitates the transmission of encrypted file chunks between users within a room.
- **Functionality:** The sender initiates the file transfer by sending the first 10 MB data chunk to the server.
- **Chunk Request:** Upon receiving a data chunk, the receiver requests the next chunk until the entire file is received.
- **Progress Tracking:** Users can monitor the progress of file transfers with progress bars or indicators.
- **Resumable Transfers:** If a transfer is interrupted, users can resume from the last successfully received chunk.

4.3.4 File Download and Decryption

- **Purpose:** Allows users to download shared files and decrypt them for local use.
- **Functionality:** Users can select a file from the shared files list and initiate the download process.
- **Preview Before Download:** Users can preview supported file types (PDF, images, documents) before downloading.
- **Decryption:** Upon download completion, the received file is decrypted on the client-side using the same encryption algorithm.
- **Save Location:** Users can choose the destination folder to save the decrypted file on their local device.

4.3.5 File Management in Rooms

- **Purpose:** Ensures that only one file can be shared at a time within a room.
- **Functionality:** Users cannot initiate a new file share until the current file transfer is completed or canceled.
- **Room Capacity Control:** Limits the number of files shared in a room to prevent overcrowding and confusion.
- **Automatic Cleanup:** If a file transfer is not completed within a certain time frame, the system cleans up incomplete transfers to free up resources.

4.4 Conclusion

The File Sharing feature of the application provides users with a secure and efficient way to share files within rooms. From file upload and encryption to transmission, download, and decryption, these functionalities ensure that files are transferred safely and only accessible to authorized users. Users can easily track the progress of file transfers and manage shared files within rooms, enhancing collaboration and productivity.

4.5 System Feature 3: Chat Functionality

The Chat Functionality feature of the File Sharing Application enables users within a room to communicate in real-time via text messages. Users can send messages, view message history, and engage in discussions related to shared files. The key aspects of the Chat Functionality feature include:

4.5.1 Real-time Messaging

- **Purpose:** Allows users within a room to communicate in real-time through text messages.
- **Functionality:** Users can send and receive messages instantly, creating a dynamic and interactive environment.
- **Message Input:** Users type messages in the chat input box and press Enter or Send to send messages.
- **Message Display:** Sent and received messages are displayed in the chat window with timestamps.

4.5.2 Message History

- **Purpose:** Provides users with access to the history of messages within a room.
- **Functionality:** Users can scroll through previous messages to view past discussions and information.
- **Message Search:** Users may have the ability to search for specific keywords or phrases within the message history.
- **Message Timestamps:** Each message includes a timestamp indicating when it was sent, helping users track the conversation timeline.

4.5.3 Message Notifications

- **Purpose:** Notifies users of new messages and activities within the room.
- **Functionality:** Users receive real-time notifications for new messages, file uploads, and other room activities.
- **Notification Sound:** Optionally, users can enable notification sounds for incoming messages.
- **Visual Notifications:** A visual indicator (e.g., blinking icon) alerts users to unread messages or updates.

4.5.4 Message Formatting

- **Purpose:** Supports basic message formatting for improved readability and emphasis.
- **Functionality:** Users can use formatting options such as bold, italic, and underline to format their messages.
- **Emojis and Stickers:** Provides a selection of emojis and stickers for users to express emotions and reactions.

- **Code Blocks:** Supports code blocks for displaying code snippets with proper formatting and syntax highlighting.

4.5.5 Message Actions

- **Purpose:** Enables users to perform actions on messages such as editing and deleting.
- **Functionality:** Users can edit their own messages for corrections or updates.
- **Message Deletion:** Users may have the ability to delete their own messages within a certain time frame.
- **Reply and Mention:** Users can reply to specific messages and mention other users in their messages for direct communication.

4.5.6 Moderation Tools (Optional)

- **Purpose:** Provides room moderators or admins with tools to manage messages.
- **Functionality:** Moderators can delete inappropriate messages, mute users, or restrict certain actions.
- **User Reporting:** Users can report inappropriate messages to moderators for review and action.

4.6 Conclusion

The Chat Functionality feature of the File Sharing Application enhances collaboration and communication among users within rooms. With real-time messaging, message history, notifications, formatting options, and message actions, users can effectively discuss shared files, exchange information, and engage in discussions. Optional moderation tools empower moderators to maintain a positive and respectful chat environment.

5 Other Functional Requirements

5.1 Performance Requirements

The performance of the File Sharing Application is crucial to provide users with a seamless and efficient experience while sharing files and communicating within rooms. The following performance requirements have been identified to ensure optimal system performance:

5.1.1 Responsiveness

- **Requirement:** The application shall respond to user actions (e.g., clicks, uploads, messages) within 1 second.
- **Justification:** Users expect a responsive interface to interact with the application smoothly and without delays.

5.1.2 File Transfer Speed

- **Requirement:** File transfers shall have an average speed of at least 5 MB/s.
- **Justification:** Faster file transfer speeds ensure efficient sharing of large files and reduce waiting times for users.

5.1.3 Room Creation Time

- **Requirement:** The time taken to create a new room shall not exceed 3 seconds.
- **Justification:** Users should be able to create rooms quickly to start file sharing and messaging without delays.

5.1.4 Message Delivery Time

- **Requirement:** Messages shall be delivered to all users in a room within 1 second of sending.
- **Justification:** Real-time communication is essential for effective collaboration, and messages should reach recipients promptly.

5.1.5 File Chunking and Encryption Time

- **Requirement:** The time taken to divide a file into 10 MB chunks and encrypt each chunk shall not exceed 1 second per 10 MB.
- **Justification:** Efficient chunking and encryption ensure that file sharing operations do not cause significant delays.

5.1.6 System Availability

- **Requirement:** The system shall have an uptime of at least 99.9
- **Justification:** Users rely on the application for important file sharing and communication tasks, necessitating high availability.

5.1.7 Scalability

- **Requirement:** The application shall be able to handle up to 100 simultaneous file transfers and messages in a single room.
- **Justification:** Scalability ensures that the application can accommodate increasing user load without performance degradation.

5.2 Conclusion

The File Sharing Application's performance requirements focus on responsiveness, file transfer speed, room creation time, message delivery time, chunking and encryption efficiency, system availability, and scalability. Meeting these requirements ensures that users experience fast and reliable file sharing, messaging, and room management capabilities, leading to enhanced productivity and user satisfaction.

5.3 Hardware Requirements

The File Sharing Application has minimal hardware requirements to run efficiently on various devices. The following hardware specifications are recommended for optimal performance:

5.3.1 Client-Side Hardware

- **Processor:** Dual-core processor (1.8 GHz or higher)
- **Memory (RAM):** 4 GB or higher
- **Storage:** 100 MB of available disk space for application installation
- **Network:** Stable internet connection (minimum 1 Mbps for smooth file transfers and messaging)
- **Display:** Minimum resolution of 1280x720 pixels for optimal user interface experience
- **Input Devices:** Keyboard and mouse (or touchpad) for navigation and interaction

5.3.2 Server-Side Hardware (for Deployment)

- **Processor:** Quad-core processor (2.4 GHz or higher)
- **Memory (RAM):** 8 GB or higher
- **Storage:** 500 GB of available disk space for file storage and database
- **Network:** Gigabit Ethernet for high-speed data transfer

- **Operating System:** Linux-based OS (e.g., Ubuntu Server) for stability and security
- **Web Server:** Apache or Nginx for serving web pages and handling HTTP requests
- **Database:** MySQL or PostgreSQL for storing user data, room information, and file metadata

5.4 Conclusion

The File Sharing Application’s hardware requirements are designed to ensure optimal performance and reliability. Clients should have a standard configuration with a dual-core processor, 4 GB of RAM, stable internet connection, and a minimum display resolution. For server-side deployment, a more robust setup with a quad-core processor, 8 GB of RAM, ample storage, and a Linux-based OS is recommended. These hardware specifications will support smooth file sharing, messaging, and room management operations within the application.

5.5 Software Requirements

The File Sharing Application has specific software requirements for both client-side devices and server-side deployment. The following software components are necessary for the application to function properly:

5.5.1 Client-Side Software

- **Operating System:** Windows 10, macOS Big Sur, Ubuntu 20.04 LTS, or newer versions
- **Web Browser:** Google Chrome (latest version), Mozilla Firefox (latest version), Safari (latest version), Microsoft Edge (latest version)
- **JavaScript:** Enabled in the web browser for dynamic content and interactivity
- **Network Connectivity:** Stable internet connection (minimum 1 Mbps)

5.5.2 Server-Side Software (for Deployment)

- **Operating System:** Linux-based OS (e.g., Ubuntu Server 20.04 LTS)
- **Web Server:** Apache HTTP Server (version 2.4.x) or Nginx (version 1.18.x)
- **Database:** MySQL (version 8.x) or PostgreSQL (version 13.x) for storing user data, room information, and file metadata
- **Programming Language:** Node.js (version 14.x) for server-side application logic

- **Node.js Framework:** Express.js (version 4.x) for building RESTful APIs
- **Messaging Protocol:** WebSocket (for real-time messaging)
- **Encryption Library:** OpenSSL (version 1.1.x) for secure data encryption

5.5.3 Additional Tools and Libraries

- **File Chunking Library:** If not built into the system, a file chunking library like ‘file-chunker’ (Node.js) for dividing files into 10 MB chunks
- **Encryption Library:** CryptoJS (JavaScript library) for client-side encryption if needed
- **Frontend Framework:** Vue.js (version 3.x) or React (version 17.x) for building dynamic user interfaces (optional)
- **Package Manager:** npm (Node.js Package Manager) for managing server-side dependencies

5.6 Conclusion

The File Sharing Application requires specific software components for both client-side devices and server-side deployment. Clients should have a compatible operating system and web browser with JavaScript enabled. For server-side deployment, a Linux-based OS with Apache or Nginx web server, MySQL or PostgreSQL database, Node.js for application logic, and WebSocket for real-time messaging are necessary. Additional tools and libraries such as file chunking and encryption libraries may be required for specific functionalities. These software requirements ensure that the application runs smoothly and securely, providing users with a seamless file sharing and messaging experience.

6 Other Nonfunctional Requirements

6.1 Performance Requirements

In addition to the previously mentioned performance requirements, the File Sharing Application has additional performance criteria to meet user expectations and ensure efficient operation. These requirements focus on responsiveness, scalability, and system stability:

6.1.1 User Interface Responsiveness

- **Requirement:** The user interface shall remain responsive during file uploads and downloads, with no noticeable lag or freezing.

- **Justification:** Users should be able to navigate the application and interact with elements without delays, even during intensive file transfer operations.

6.1.2 Server Response Time

- **Requirement:** The server shall respond to API requests within 500 milliseconds on average.
- **Justification:** Fast server response times ensure that client applications receive timely data and perform actions without significant waiting periods.

6.1.3 Scalability Under Load

- **Requirement:** The application shall handle up to 500 simultaneous active users without degradation in performance.
- **Justification:** As the user base grows, the application should scale gracefully to accommodate increased traffic and usage without impacting user experience.

6.1.4 Network Bandwidth Utilization

- **Requirement:** The application shall efficiently utilize available network bandwidth without causing network congestion.
- **Justification:** Efficient bandwidth usage ensures smooth file transfers and messaging while avoiding network slowdowns or disruptions for other applications.

6.1.5 Error Handling Performance

- **Requirement:** Error handling mechanisms shall not introduce significant overhead, with error responses generated within 100 milliseconds.
- **Justification:** Effective error handling ensures that issues are identified and communicated to users promptly without affecting overall system performance.

6.1.6 File Encryption and Decryption Speed

- **Requirement:** File encryption and decryption shall occur at a speed of at least 20 MB/s.
- **Justification:** Efficient encryption and decryption are crucial for secure file sharing operations without causing delays.

6.2 Conclusion

The File Sharing Application's additional performance requirements emphasize user interface responsiveness, server response time, scalability, network bandwidth utilization, error handling performance, and file encryption/decryption speed. These requirements aim to ensure a smooth and efficient user experience, even under heavy loads and intensive file transfer operations. By meeting these criteria, the application can provide reliable and responsive file sharing and messaging capabilities to users.

7 Other Requirements

In addition to functional and nonfunctional requirements, the File Sharing Application has additional requirements and constraints that are important for its development and operation:

7.1 Legal and Compliance

- **Data Privacy Compliance:** The application shall comply with data privacy regulations such as GDPR (General Data Protection Regulation) and ensure user data security and confidentiality.
- **Copyright Protection:** Users shall be notified of copyright regulations and restrictions.

7.2 Localization

- **Language Support:** The application shall support multiple languages for user interfaces, such as English (default), Spanish, French, and German.
- **Date and Time Format:** Date and time formats shall be localized based on user preferences (e.g., DD/MM/YYYY or MM/DD/YYYY).

7.3 Accessibility

- **Accessibility Standards:** The application shall comply with WCAG (Web Content Accessibility Guidelines) 2.1 Level AA standards for accessibility.
- **Keyboard Navigation:** Users shall be able to navigate and use the application using only a keyboard, without relying on a mouse.

7.4 Backup and Recovery

- **Automatic Backups:** The server shall perform regular automatic backups of user data, room information, and file metadata.

- **Recovery Plan:** In case of system failure or data loss, a recovery plan shall be in place to restore the application and user data.

7.5 Authentication and Authorization

- **Two-Factor Authentication:** Users shall have the option to enable two-factor authentication (2FA) for enhanced account security.
- **Password Complexity:** Passwords shall require a minimum length of 8 characters with a combination of letters, numbers, and special characters.

7.6 Integration with Third-Party Services

- **Email Integration:** The application shall integrate with an email service provider (e.g., SMTP) for sending account verification emails and notifications.
- **File Preview:** If feasible, integrate with a file preview service (e.g., Google Docs Viewer) to allow users to preview supported file types before downloading.

7.7 User Feedback and Improvement

- **Feedback Mechanism:** The application shall include a feedback form or option for users to provide feedback and suggestions.
- **Continuous Improvement:** Based on user feedback, regular updates and improvements shall be made to the application to enhance user experience.

7.8 Conclusion

The File Sharing Application has various additional requirements and constraints that are essential for its development, compliance, accessibility, backup, authentication, and integration. From legal and compliance considerations to localization, accessibility standards, and development constraints, these requirements shape the application's functionality and operation. By adhering to these requirements, the application can provide a secure, accessible, and user-friendly platform for file sharing and communication.

[width=0.8]login_page_mockup.png

Figure 1: Login Page Mockup

[width=0.8]dashboard_mockup.png

Figure 2: Dashboard Mockup

[width=0.8]room_interface_mockup.png

Figure 3: Room Interface Mockup

[width=0.8]file_preview_mockup.png

Figure 4: File Preview Mockup

[width=0.8]settings_page_mockup.png

Figure 5: Settings Page Mockup