

Project Documentation

1. Requirements Analysis

Functional Requirements

1. User Authentication

- Users can sign up and will be assigned an email address in the format **userName@myApp.com**.
- Users can then log-in using their credentials.

2. Chat System

- After login, users are redirected to the chat page.
- A panel on the left displays all of the online users.
- Users can select an online user to start a private chat.
- Users can switch between chats.

3. Email System

- Users can send emails to other users who are registered to the domain.
- Users can view their inbox for incoming emails.
- Emails are restricted to the domain of the application (i.e. **userName@myApp.com**).

Non-Functional Requirements

1. The application will be hosted on our personal laptop, and will be accessible via IP address and port number.
2. Responsiveness for only desktop devices (optional for mobile).
3. Basic usability; security is not a priority.

2. System Design

High-Level Architecture

1. Frontend

- Technologies: HTML, CSS, JavaScript, Nginx, and C#.
- Features: Chat UI, Email UI, user login/signup pages.

2. Backend

- Framework: ASP.NET Core MVC.
- Modules:
 - Authentication and User Management
 - Chat System
 - Email System

3. Database

- SQL Server hosted locally.
- Tables:
 - Users:** UserId, UserName, Password, Email, OnlineStatus.
 - Messages:** MessageId, SenderId, ReceiverId, MessageContent, TimeStamp.
 - Emails:** EmailId, SenderId, ReceiverId, Content, TimeStamp.

System Workflow

1. Sign-Up Process

- User enters credentials and is assigned an email address.
- User data is stored in the User's table.

2. Login Process

- Credentials are validated.
- Online Status is set to true.
- User is redirected to the chat page.

3. Chat Process

- Online users are fetched and displayed.
- Messages are stored in the Messages table.

4. Email Process

- Emails are stored in the Emails table.
- Inbox and sent emails are fetched based on SenderId and ReceiverId.

3. Task Breakdown

Modules and Assignments

1. **Authentication and User Management**
 - Task: Implement signup, login, and online status tracking.
 - Assigned to: Backend Developer.
2. **Chat System**
 - Task: Real-time messaging using SignalR and message storage.
 - Assigned to: Chat Developer.
3. **Email System**
 - Task: Email backend, including inbox and sending emails.
 - Assigned to: Email Developer.
4. **Frontend Development**
 - Task: Build and enhance UI for all modules.
 - Assigned to: Frontend Specialist.