Chat application

The system is a multiuser chat application. The system mainly works in 2 modes. One to one chat mode & Broadcast mode. Users can create multiple channels, and can share the name of channel with multiple users, and by joining the channel they can chat with multiple users at a time. The system also comes with rich features like using emoji in chat, media sharing support, update profile, forgot password.

## **Purpose**

The purpose of this document is to develop this system. This document provides a detailed overview of the system, details of the functionalities provided to users, target audience of the system and their user interface

## **Technology/Platform/Tools used**

**Technology:**

* Dot net core 3.1
* EF Core
* Signal - R
* SQL Server database.
* JavaScript
* Bootstrap
* Knockout js
* jQuery

## **Functional Requirements**

### 1.User module

**1.1 Login**

Description: User can login to the system

Exception Flow: If credentials are incorrect or insufficient data is provided

Input: User data

Output: Redirects to Dashboard

**1.2 Register**

Description: User can login to the system

Exception flow: If username is already taken or insufficient data is provided, user will be asked to fill the details again.

Description: On successful registration of a new users, the system will send a greeting email to the registered email

Input: User data

Output: Redirects to Login page with success message

**1.3 Update profile**

Exception flow: If username is already taken or insufficient data is provided

Input: User profile data

Output: Success message

**1.4 Forgot Password**

Input: username, old password, new password

Output: Response message

**1.5 Logout**

Input: User selection

Output: Redirects to Login page with success message

**1.6 Delete my account**

Input: User selection

Output: Redirects to Login page with success message

**1.7 Search user**

Input: Username to be searched

Output: List of users matching search criteria

### 2 One to one chat module

**2.1 Send a message to particular user**

Input: Message to be sent

Output: Upon successful delivery of a message, the message will be listed onto the chat screen

Description: Users selects the name of the user he want to chat with, and enters the message

### 

### 3 Broadcast chat module

**3.1 Create channel / Room**

Exception flow: If channel exists, error message will be displayed

Input: Channel name

Output: Success message

**3.2 Join channel / Room**

Exception flow: If channel does not exist, error message will be displayed

Input: Channel name

Output: Success message

**3.3 Send a message in channel**

Input: Message to be sent

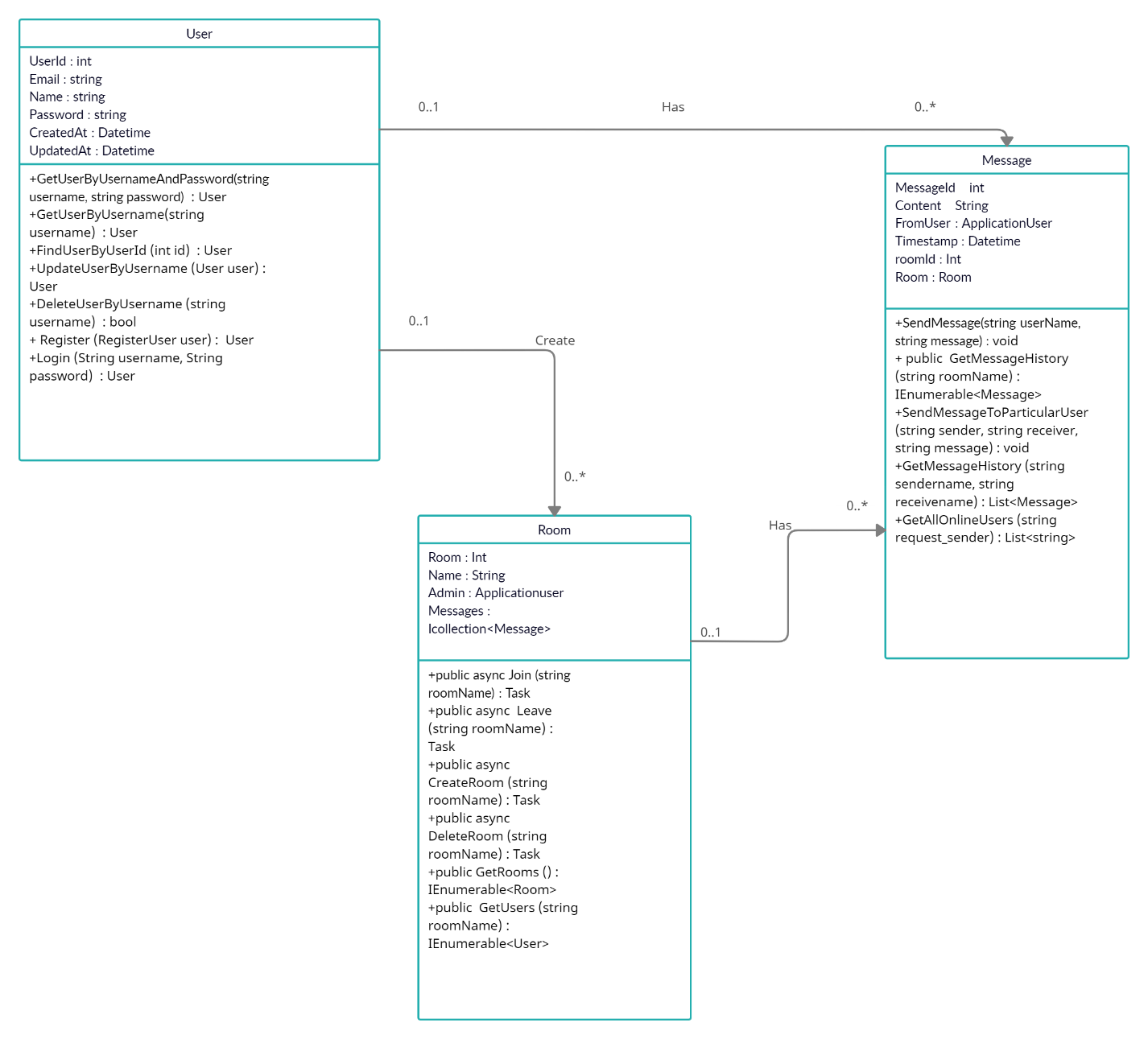
Output: Upon successful delivery of a message, message will be listed onto the chat screen

**3.4 Delete channel / Room**

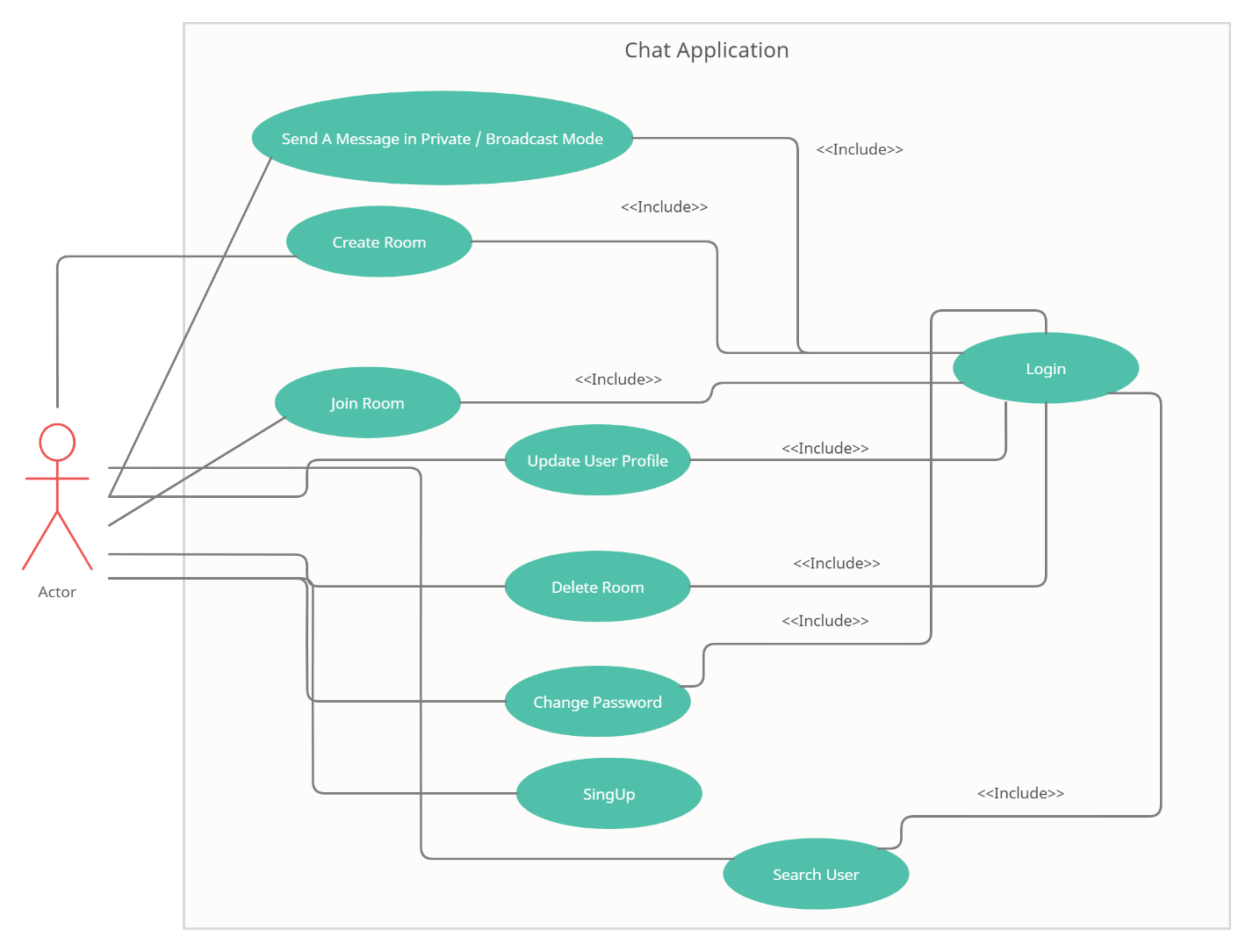
Input: User Selection

Output: Success message

### Design

Class Diagram

## **Use case Diagram**



## **ER Diagram**