

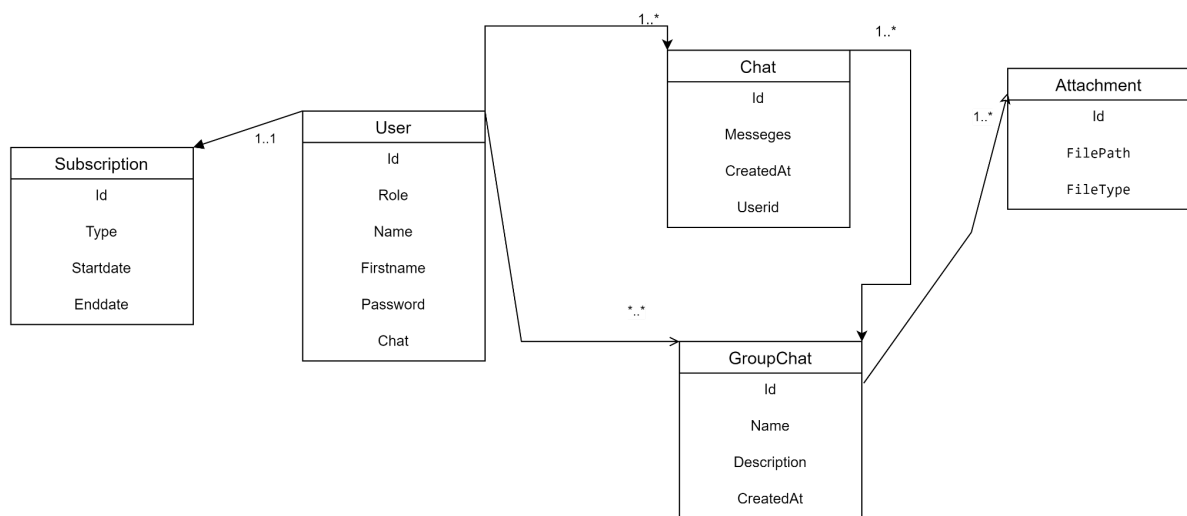
Project Analysis

Product Description

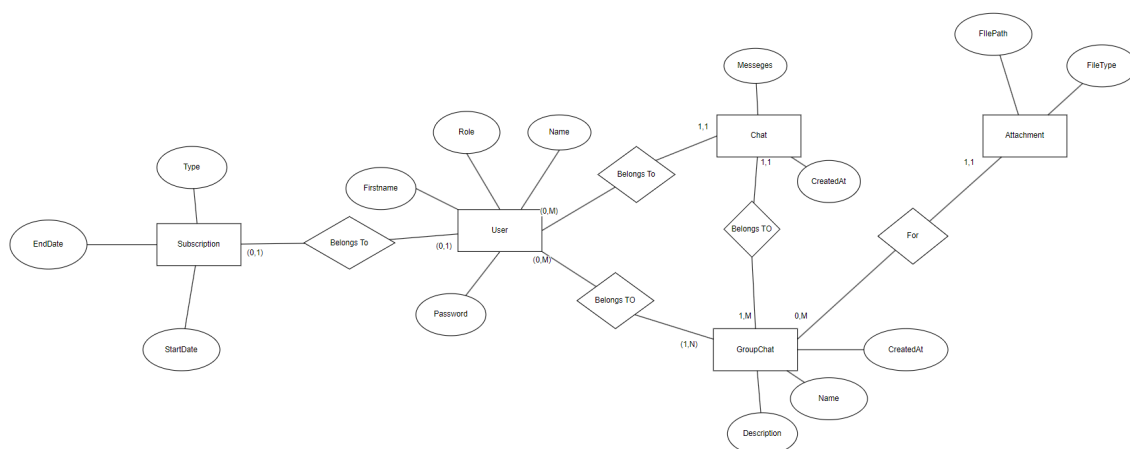
Title: ClassConnect

description:A secure and intuitive chat app for students and teachers to communicate inside and outside of class. ClassConnect provides organized channels for different courses and activities to streamline discussions and resource sharing.

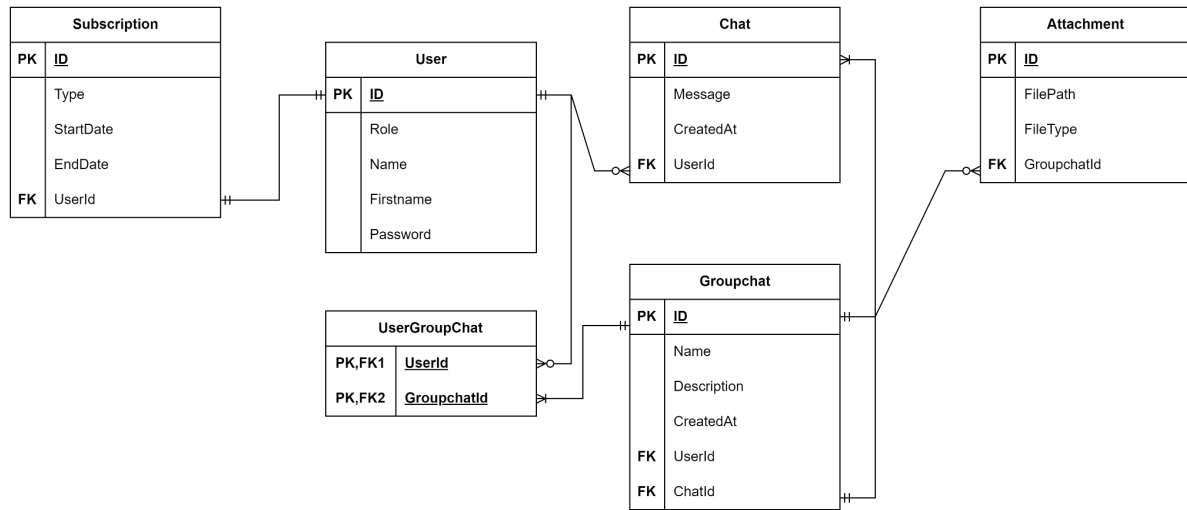
Domain model



Conceptual model



Logic model



User Stories

1. User Registration (as Student or Lecturer)

Story:

As a user, I'm able to register as a student or lecturer so that I can access the platform with the correct role.

Wireframe:

Register

ID

Naam

Voornaam

Password

Enter

Acceptance Criteria:

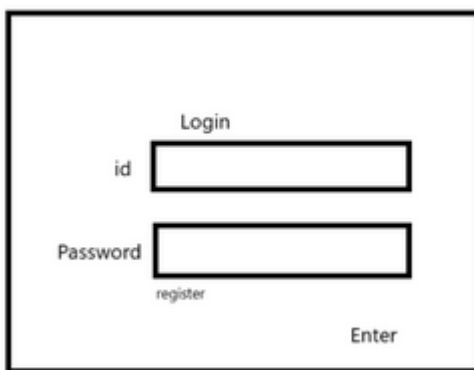
- If a user registers successfully, they should receive a confirmation email.
- The registration form should include validation for required fields (e.g., valid email, password length).
- The user must be able to choose between student and lecturer roles, and this role is stored.

2. User Login

Story:

As a user, I'm able to login so that I can access the platform using my credentials.

Wireframe:



A wireframe diagram of a login form. It consists of a large rectangular container. Inside, at the top, is the word "Login". Below it are two input fields. The first input field is preceded by the label "id". The second input field is preceded by the label "Password". Below the "Password" label, there is a small, faint word "register". At the bottom right of the container, there is a button labeled "Enter".

Acceptance Criteria:

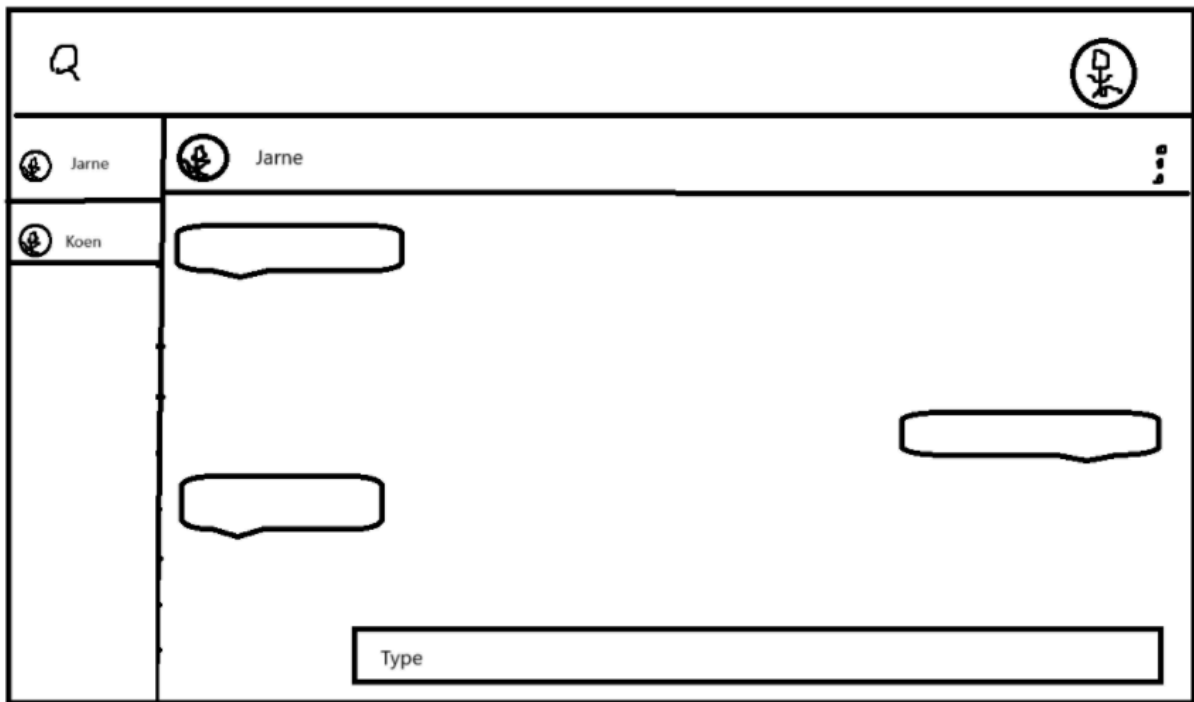
- If login is successful, the user is redirected to the dashboard (based on their role).
- Invalid credentials result in an error message ("Invalid email or password").
- Users should be able to reset their password using the "Forgot Password?" feature.

3. Chat

Story:

As a student/teachers, I'm able to chat with other students/teachers so that I can collaborate and communicate within the platform.

Wireframe:



Acceptance Criteria:

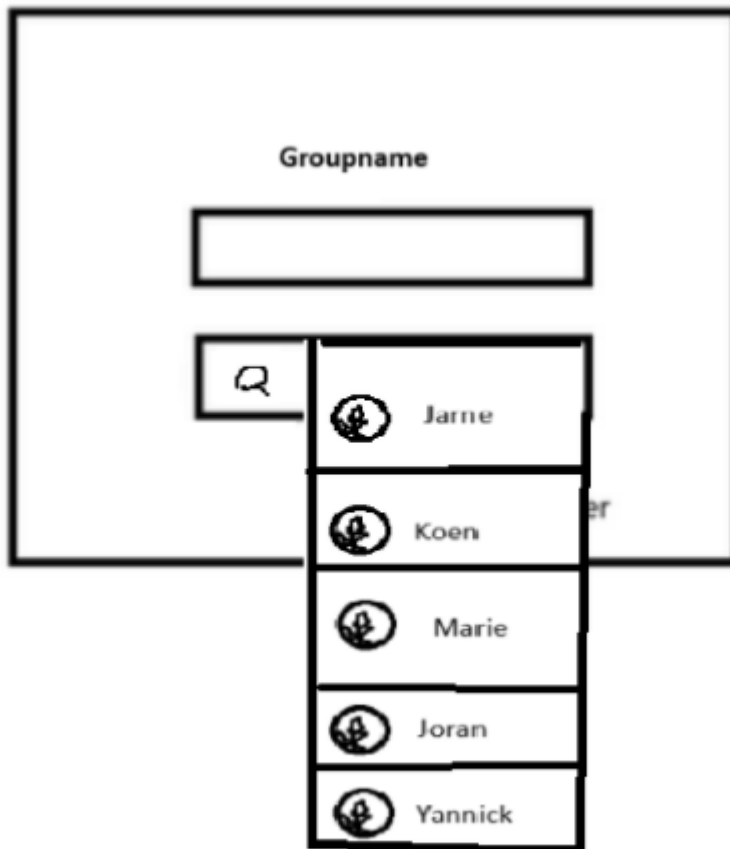
- Students should be able to send and receive messages in real time.
- The chat should show a message history for each conversation.
- Students can search for and start chats with other students.

5. Student/teachers Chat Groups Creation

Story:

As a student, I'm able to create chat groups so that I can collaborate with multiple students.

Wireframe:



Acceptance Criteria:

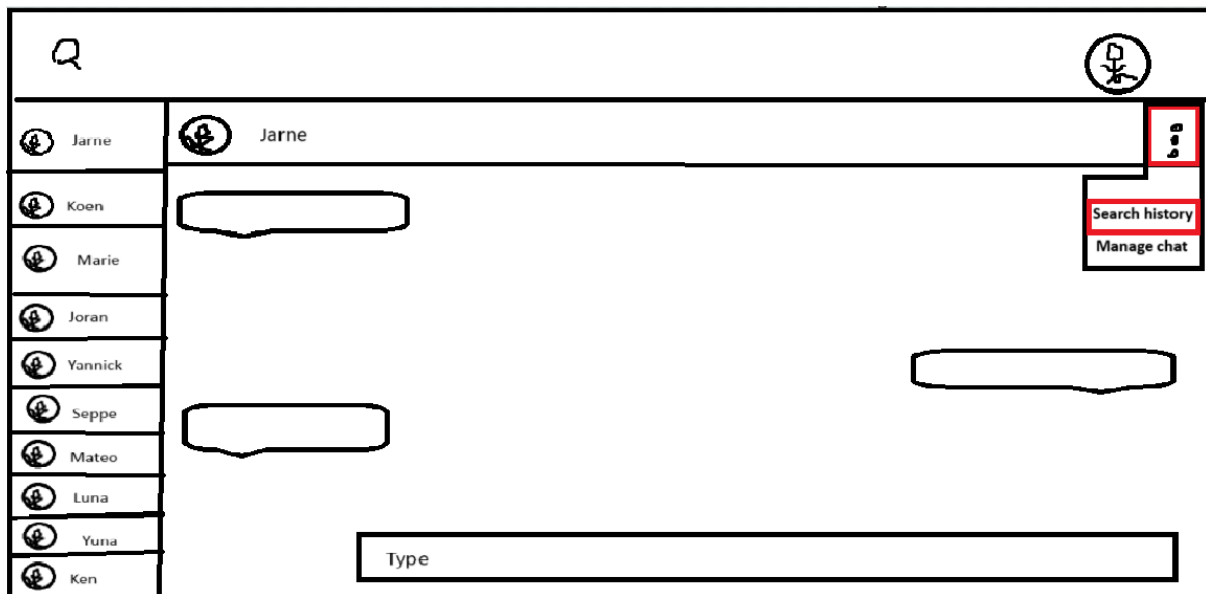
- The student should be able to create a group with other students.
- Group chat should allow all members to send messages.
- The group creator should have the option to add/remove participants.

6. Student Chat History Search

Story:

As a student, I'm able to search in the chat history so that I can find past messages easily.

Wireframe:



Acceptance Criteria:

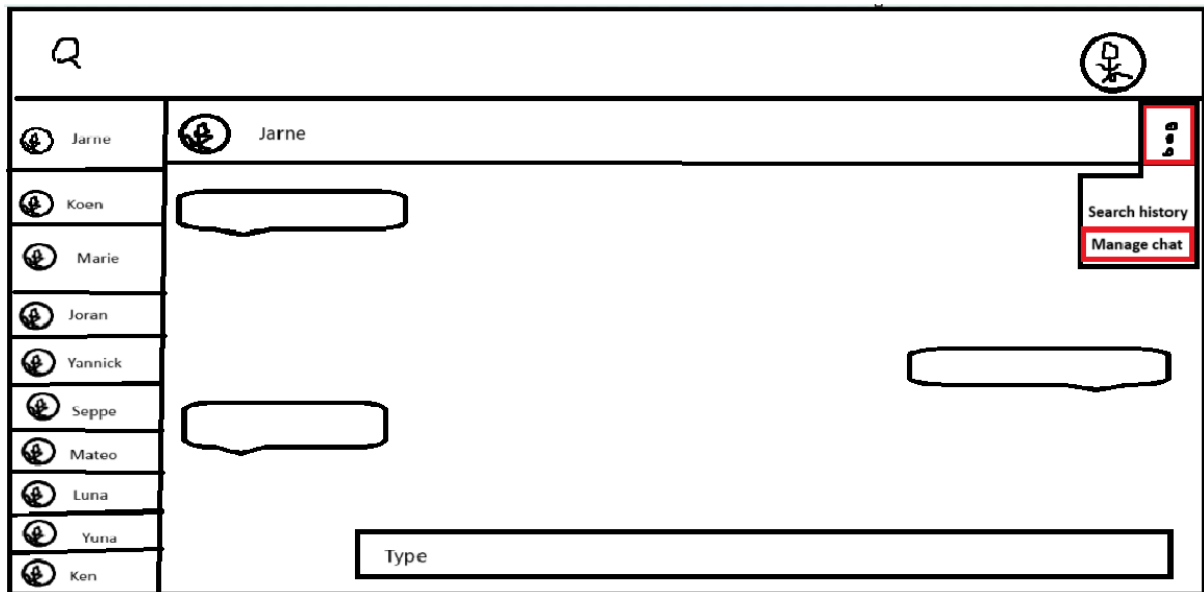
- The search should return messages that match the query, highlighting the results.
- The search should be able to filter by date and message content.

11. Admin Chat Management

Story:

As an admin, I'm able to manage chats so that I can oversee the platform's communication system.

Wireframe:



Acceptance Criteria:

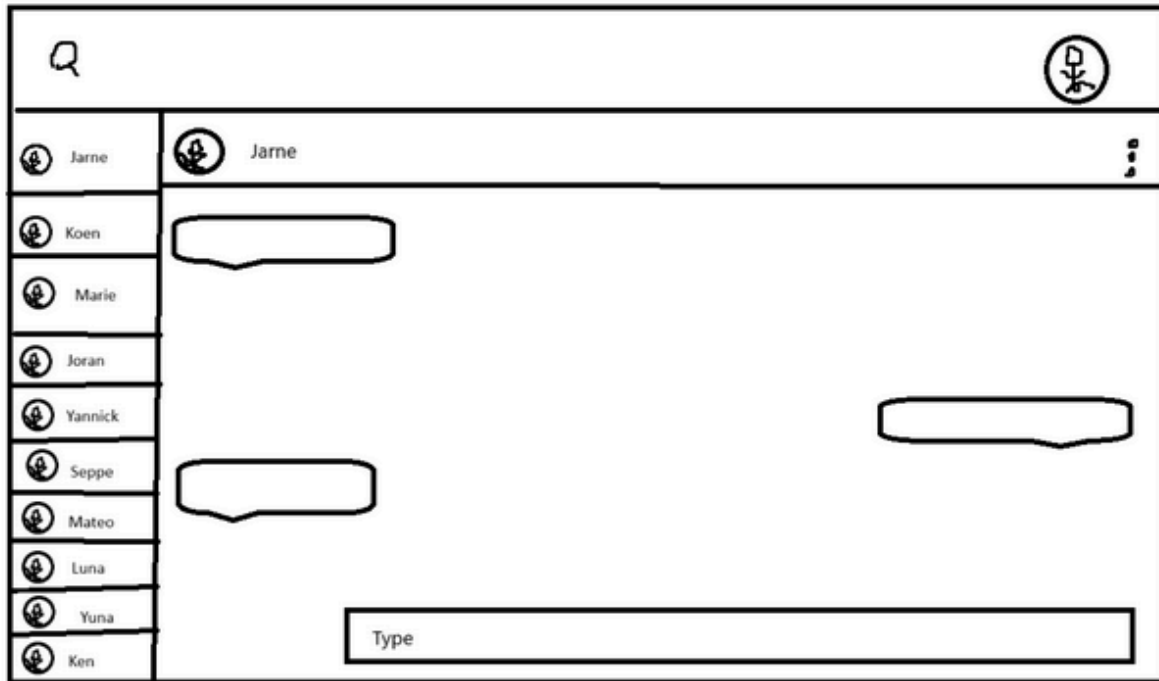
- Admins should be able to view and manage chats (e.g., archive, resolve conflicts).
- They can remove inappropriate messages or users from chats.

12. Admin View All Chats

Story:

As an admin, I'm able to see all the chats so that I can monitor communication between users.

Wireframe:



Acceptance Criteria:

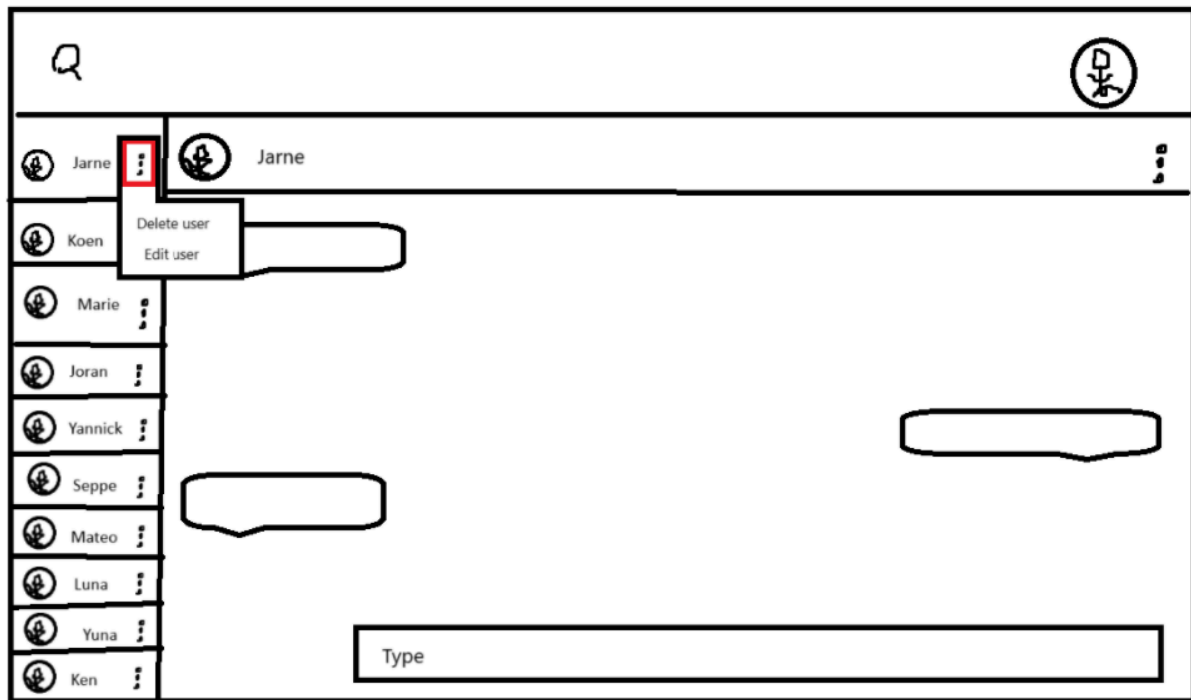
- Admins should be able to view all active chats, with the ability to filter and search.
- They should be able to view chat history and active participants.

13. Admin Delete Students

Story:

As an admin, I'm able to delete students so that I can manage the user base of the platform.

Wireframe:



Acceptance Criteria:

- Admins should be able to delete any student account from the system.
- Once deleted, a student's chat history should no longer be visible, but may be archived.