

Chatt applikation

Designdokument

FÖRFATTARE

Kolehmainen, Mikael

DATUM

07.02.2023

Innehållsförteckning

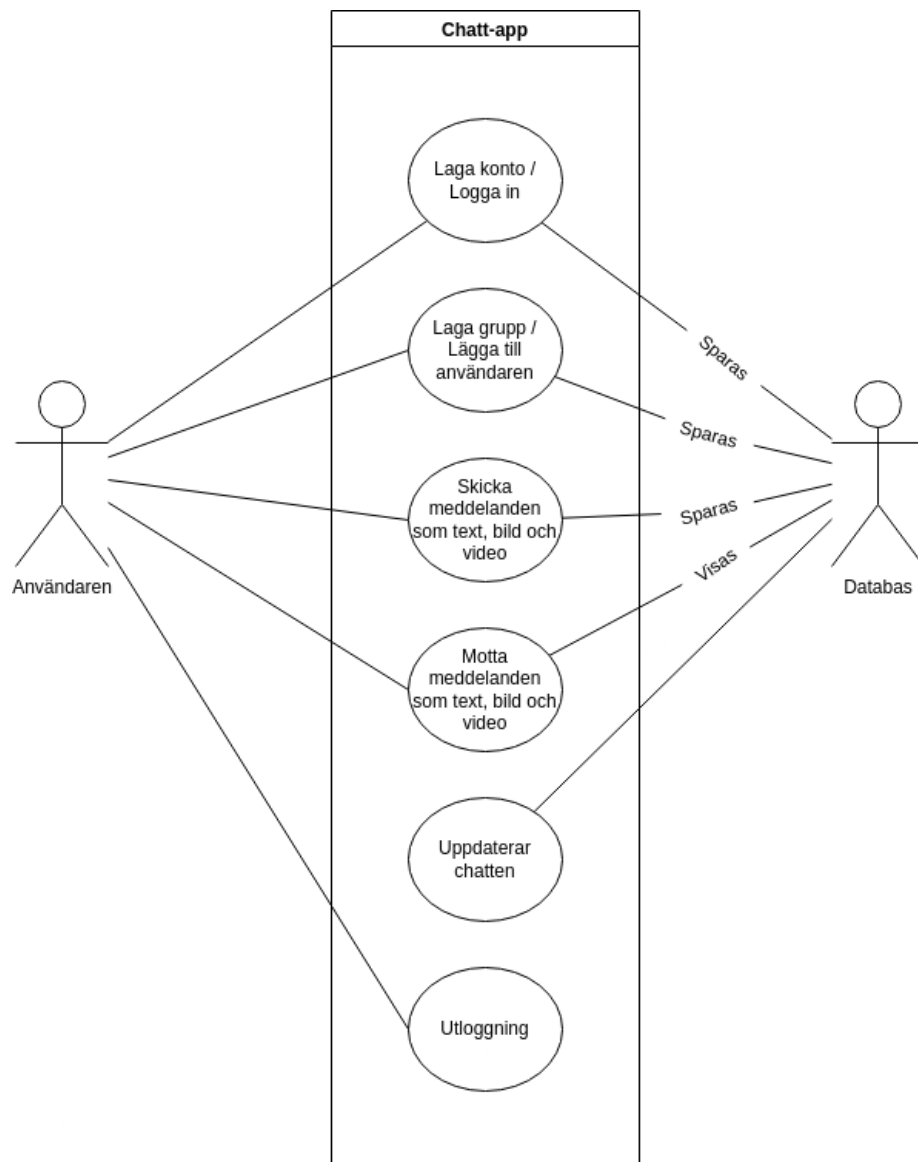
| | |
|--------------------------------|---|
| 1 Beskrivning av systemet | 2 |
| 2 Användarfall – Use Case | 2 |
| 3 Design av användargränssnitt | 3 |
| 4 Databasdesign | 3 |
| 5 Mappstruktur | 4 |
| 6 API planering | 5 |
| 7 Plan för testning | 6 |
| 8 Referenser | 7 |
| 9 Framtida förbättringar | 7 |
| 10 Bilagor | 7 |

1 Beskrivning av systemet

En chatt applikation där användaren kan laga ett konto (användarnamn, användarbild, lösenord) och logga in. Användaren kan sedan laga en chattgrupp med minst en annan användare, i en chattgrupp kan man skicka text och bild. Chatt applikationen är byggd med PHP, HTML, JavaScript, CSS / SASS och MySQL. Projektet finns på [Github](#) och [Vlab](#).

2 Användarfall – Use Case

Applikationens funktioner är följande: Skapelse av konto och inloggning, skapelse av grupp och tilläggning av andra användaren, funktion att skicka meddelanden som text/bild, funktion att motta meddelanden som text/bild och utloggning. Chatten uppdateras automatiskt utan att uppdatera webbläsaren med hjälp av AJAX.



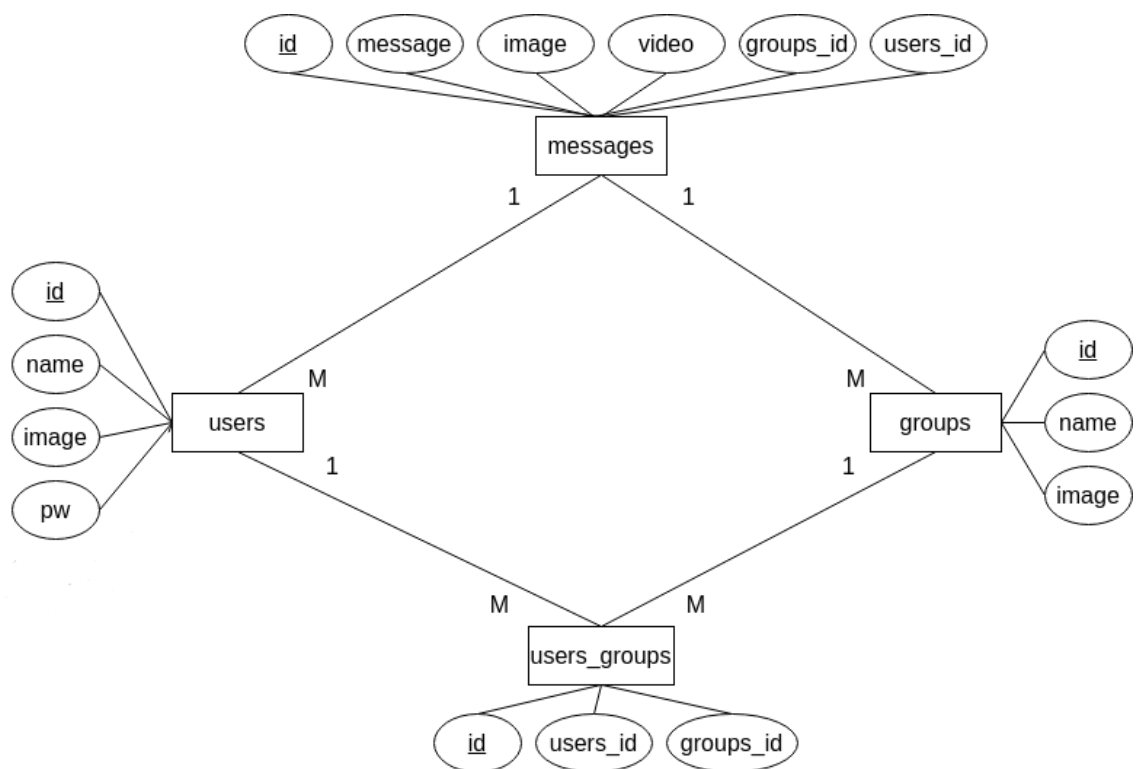
Figur 1. Figuren visar applikationens funktioner.

3 Design av användargränssnitt

Användargränssnittet använder fonten Roboto och huvudfärgerna är #5BC0F8 och #006197. Sidorna som finns i applikationen: Logga in, skapa konto, huvudmeny med grupper, chatten, gruppskapelse, lägg till användare i grupp och felsida som berättar åt användaren vad som gick fel i programmet och omdirigerar användaren, t.ex. Om man får till chatt vyn utan att logga in. En stilguide och wireframes finns som bilaga i dokumentet.

4 Databasdesign

Databashanteringsverktyget i projektet är MySQL. Tabellerna är *users*, *groups*, *messages* och *users_groups*. Tabellerna är uppbyggda på följande sätt: *users*(*id*, *name*, *image*, *pw*), *groups*(*id*, *name*, *image*), *messages*(*id*, *message*, *image*, *video*, *groups_id*, *users_id*) och *users_groups*(*id*, *users_id*, *groups_id*). Relationen mellan grupper och användare är många till många. Relationen mellan grupper och meddelanden är ett till många alltså ett meddelande kan bara finnas i en grupp men en grupp kan ha flera meddelanden.



Figur 2. Figuren visar relationerna mellan databastabellerna

```
CREATE DATABASE chat_app_db;

CREATE TABLE groups (
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL,
    image VARCHAR(255) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE users (
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL,
    image VARCHAR(255) NOT NULL,
    pw VARCHAR(255) NOT NULL,
    identifier VARCHAR(20),
    PRIMARY KEY (id)
);

CREATE TABLE users_groups (
    id INT NOT NULL AUTO_INCREMENT,
    users_id INT NOT NULL,
    groups_id INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (users_id) REFERENCES users(id),
    FOREIGN KEY (groups_id) REFERENCES groups(id)
);

CREATE TABLE messages (
    id INT NOT NULL AUTO_INCREMENT,
    message VARCHAR(1000) DEFAULT NULL,
    media VARCHAR(255) DEFAULT NULL,
    dateofmessage DATE,
    timeofmessage TIME,
    groups_id INT NOT NULL,
    users_id INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (groups_id) REFERENCES groups(id),
    FOREIGN KEY (users_id) REFERENCES users(id)
);
```

Figur 3. Figuren visar databasdesignen med MySQL.

5 Mappstruktur

Mappstrukturen är uppdelad i back-end och front-end. Sidan visas från index.php där man instansierar olika klasser i controller mappen beroende på hur URI:n ser ut, t.ex. index.php/create-user anropar klassen UserController.

```

/
src
  public (front-end)
    controller
    styles
      css
        main.css
      sass
        abstract
        base
        layout
        main.scss
    media
      icons
      chat
    font
  api (back-end)
    model
    manager
  misc
index.php
autoloader.php
README.md

```

Figur 4. Figuren visar applikationens mappstruktur.

6 API planering

| URI | Klass | Funktion av sidan |
|---------------------------------------|-----------------|---|
| index.php/user/create | controller/User | Visa ett formulär där man kan laga en användare. |
| index.php/user/log-in | controller/User | Visa ett formulär där man kan logga in. |
| index.php/user/insert (POST) | controller/User | Instansiera model/User som insätter användaren till databasen, sedan förs till chattgrupper vyn. |
| index.php/user/select (POST) | controller/User | Instansiera model/User som kollar om användaren finns och att lösenordet är rätt, sedan förs till chattgrupper vyn. |
| index.php/user/log-out | controller/User | Tar bort sessionen och dirigerar användaren till index.php/user/log-in. |

| | | |
|--|--------------------|---|
| index.php/group/create | controller/Group | Visa ett formulär där man kan laga en chattgrupp. |
| index.php/group/insert (POST) | controller/Group | Instansierar model/Group som insätter gruppen till databasen, sedan förs till chattgruppens vyn. |
| index.php/group/add-user/{group.id} | controller/Group | Visa ett formulär där man kan lägga till användaren till gruppen. |
| index.php/group/insert-user/{group.id} (POST) | controller/Group | Instansiera model/Group som lägger till användaren till gruppen. |
| index.php/groups | controller/Group | Visar alla grupper som användaren är medlem i. |
| index.php/group/chat/{group.id} | controller/Group | Visar gruppens chat och tar datat från model/Group. |
| index.php/group/camera/{group.id} | controller/Camera | Visar kameran. |
| index.php/ajax/send-media | controller/Camera | Instansierar model\Message och skickar bilden till databasen. |
| index.php/message/insert/{group.id} (POST) | controller/Message | Instansierar model\Message som insätter meddelandet till databasen, sedan för användaren tillbaka till chatt vyn. |
| index.php/ajax/get-messages/{group.id} | controller/Group | Tar meddelanden och uppdaterar dem från model\Message. |
| index.php/error | controller/Error | Hanterar fel och visar användargränssnitt om det händer fel. |

7 Plan för testning

Detta är en planering för testning och hur applikationen skall testas.

| Testfall | Resultat |
|--|-----------|
| Kontoskapelse | Fungerar! |
| Inloggning på ett befintligt konto | Fungerar! |
| Gruppskapelse | Fungerar! |
| Lägga till en och flera användare till en grupp. | Fungerar! |

| | |
|---|-----------|
| Skicka text / motta text | Fungerar! |
| Skicka bild / motta bild | Fungerar! |
| Utloggning | Fungerar! |
| Testa layouten på sidan på olika enheter (telefon & dator) och på olika webbläsaren (Chrome, Firefox, Edge) | Fungerar! |

8 Referenser

<http://www.databasteknik.se/webbkursen/index.html>, 16.01.2023, ER-modellering.

<https://stackoverflow.com/questions/4459379/preview-an-image-before-it-is-uploaded>,

19.01.2023, Kollade hur man kan visa en bild från file input med JavaScript.

9 Framtida förbättringar

Framtida förbättringar skulle kunna vara möjligheten för användaren att ringa / videosamtala med chattgruppen. Också möjligheten att skicka video i chatten. Möjligheten att ta bort gruppmedlemmar.

10 Bilagor

Bilaga 1, Stilguide.

Bilaga 2, Wireframe logga in.

Bilaga 3, Wireframe skapa användare.

Bilaga 4, Wireframe skapa grupp.

Bilaga 5, Wireframe lägg till användare till grupp.

Bilaga 6, Wireframe chattgrupper (desktop).

Bilaga 7, Wireframe chattgrupper (mobile).

Bilaga 8, Wireframe chatt (mobile).

Style guide

| | |
|--------------|---------|
| Primary | #5BC0F8 |
| Background | #0081C9 |
| Box | #006197 |
| Font-dark | #000000 |
| Font-light | #FFFFFF |
| Secondary | #FFC93C |
| Secondary bg | #B88917 |

Font-family: Roboto

HEADING 1

Size: 48px; weight: light

paragraph

size: 16px; weight: regular;
line-height: 1.4

Lorem ipsum dolor sit

amet, consectetur

adipiscing elit. Quisque

posuere aliquet arcu vitae

hendrerit.

[link@example.fi](#)

[hover_link@example.fi](#)

[link2@example.fi](#)

[hover_link@example.fi](#)

BUTTON

:HOVER

BUTTON

:HOVER

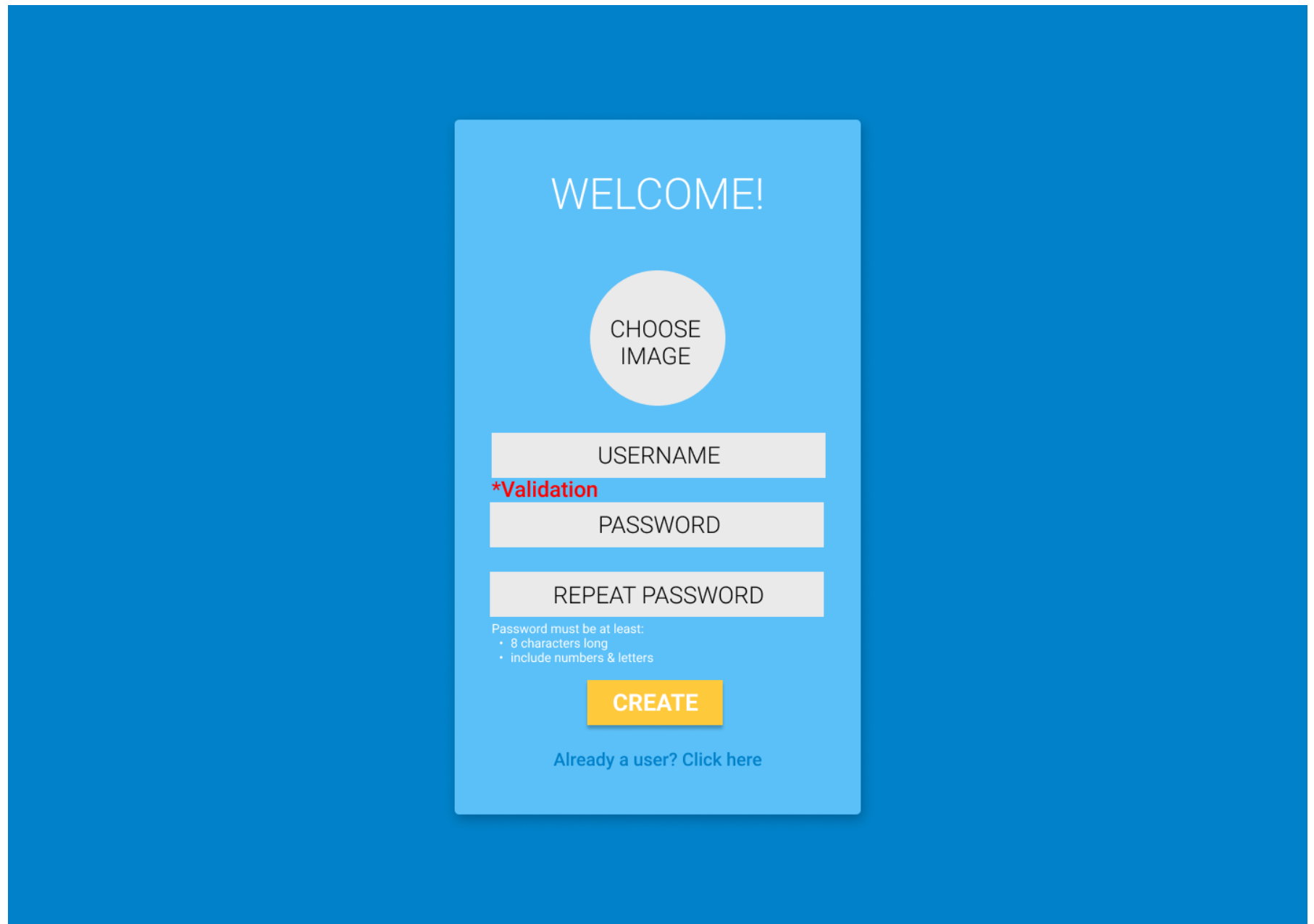
WELCOME BACK!

USERNAME

PASSWORD

LOGIN

[New? Click here](#)



WELCOME!

CHOOSE
IMAGE

USERNAME

***Validation**

PASSWORD

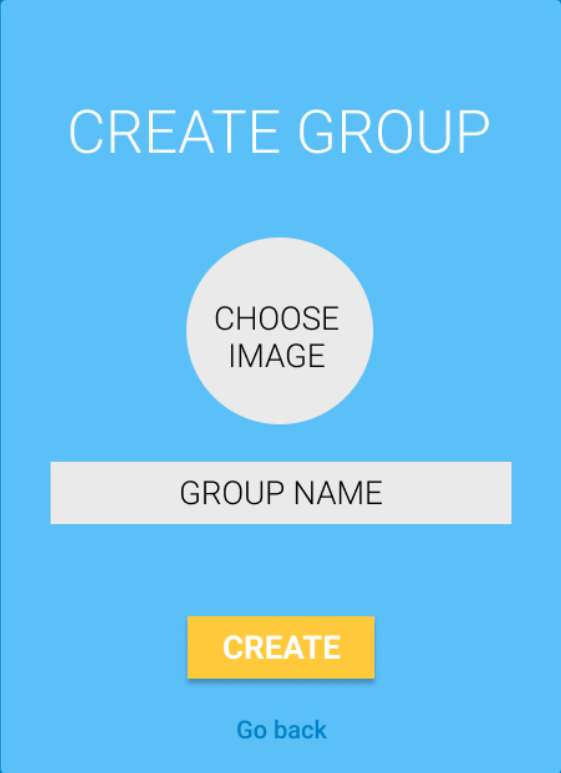
REPEAT PASSWORD

Password must be at least:

- 8 characters long
- include numbers & letters

CREATE

[Already a user? Click here](#)



CREATE GROUP

CHOOSE
IMAGE

GROUP NAME

CREATE

[Go back](#)

ADD USERS TO GROUPNAME

Search here (identifier)



IDENTIFIER



IDENTIFIER



IDENTIFIER



ADD

[Go back](#)

