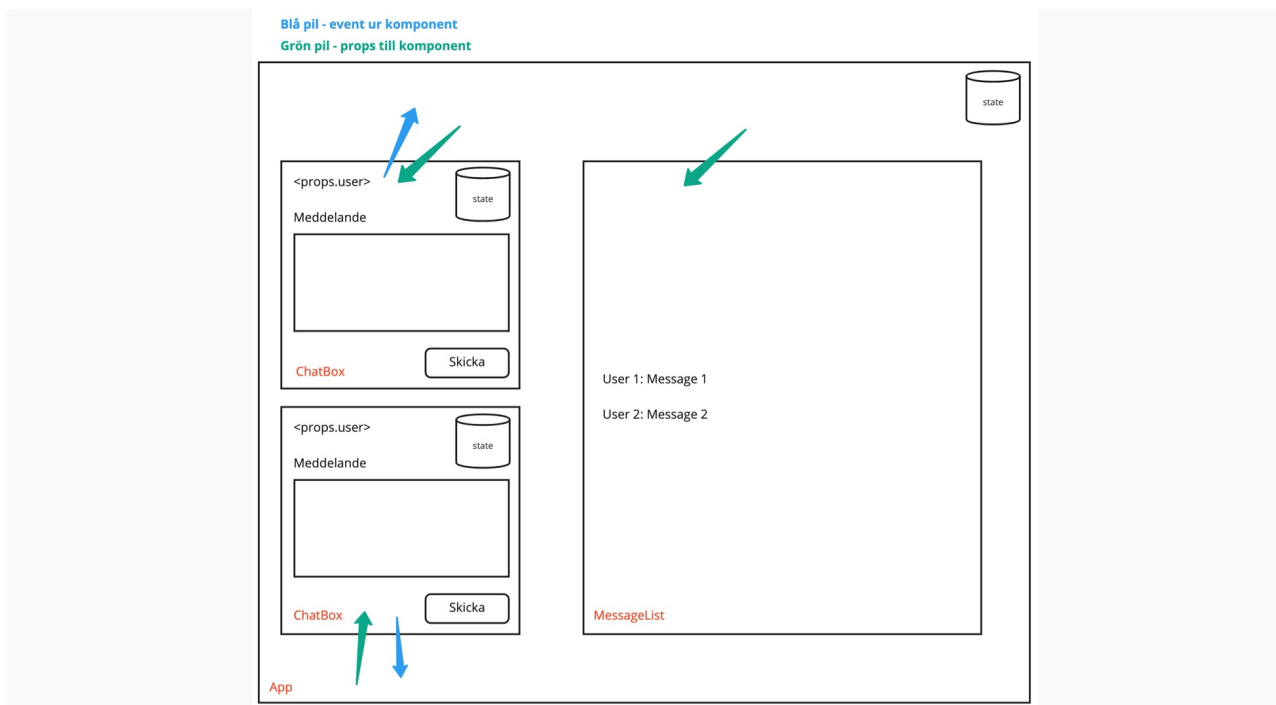


## Övningsuppgift 2 - Chatten

I denna övningsuppgift ska vi utveckla lite olika komponenter. Målsättningen med övingen är att skapa klasskomponenter, funktionskomponenter och att arbeta med events.

Börja denna övningen med att skapa ett nytt react projekt. Du kommer att arbeta med tre olika komponenter App, ChatBox och MessageList.



### ChatBox

Skapa en **klasskomponent** som du kallar för ChatBox. Den skall se ut på följande sätt:

The visual representation of the **ChatBox** component shows a large text input field labeled "Meddelande" and a button labeled "Send Message". Above the input field, the text "User1" is displayed, indicating the user's name.

#### Implementera så att

1. "User1" ska komma från en "props" till komponenten som heter "name".

2. Din komponent ska ha en state. Denna ska innehålla texten i rutan - dvs stoppa din state-text i text-rutans value och stoppa in förändringar från rutan i din state-text.
3. Din komponent ska ha en "props" som heter "onMessage" - du kan förutsätta att den kommer att innehålla en funktion. Din komponent skall skicka användarens namn och meddelandet i text-rutan (du bör ha den texten i din state) till denna funktionen när knappen "Send Message" trycks ner. (onMessage är en callback-funktion).
4. När "Send Message" trycks ner, töms ditt meddelande-fält genom att du tömmer din state-text.

## MessageList

Skapa en **funktionskomponent** som du kallar för MessageList. Denna komponent behöver ingen egen state men den kommer att behöva ett "props" - den kan vi kalla för "messages".

I denna props förväntar vi oss en array av objekt - dvs [ { user: 'Kalle', text: 'Hej' }, { user: 'Stina', text: 'Hej på dig!' } ]. Denna array ska renderas som en lista i komponenten.

# Chat-history

- User1: Hej!
- User2: Hej på dig!

Svårigheten med denna är att vi vill göra en loop i vår renderering. Här kan man använda sig av array funktionen "map". T.ex.

```
let minLista = ['Ett', 'Två', 'Tre'];  
  
return (  
  { minLista.map((item, index) => {  
    return (<p key={index}>{item}</p>);  
  }  
);
```

key={index} behövs av react för att vet vad som skall uppdateras i DOM:en vid en förändring. Det är inget som påverkar komponenten.

## App

Den här komponenten kommer vi att använda för att knyta ihop ChatBox-komponenterna (ja, vi skall använda den två gånger) och MessageList.

**User1**

Meddelande

Send Message

**Chat-history**

- User1: Hej!
- User2: Hej på dig!

**User2**

Meddelande

Send Message

Skapa App-komponenten som en **funktionskomponent**. Denna kommer att behöva en state. I din state kommer du att spara en array av objekt (som ser ut på samma sätt som i MessageList-komponenten). Skapa en metod som du skall använda som callback-funktion till dina ChatBox-komponenter - den behöver kunna ta emot namnet på användaren och meddelandet.

Detta betyder ChatBox (se punkt 3 för ChatBox) kommer att vara denna metoden.

Ta datan som kommer från ChatBox:en och spara den till din array i din state. Kom ihåg; du får inte ändra staten direkt. Du måste använda useState-hook:en.

I denna komponent kan du:

- rendera ChatBox-komponenten två gånger - de har olika name (name props), men samma callback kan användas (onMessage props).
- rendera MessageList-komponenten - du behöver ge din listan med användare och meddelanden som props - den har du i din state.