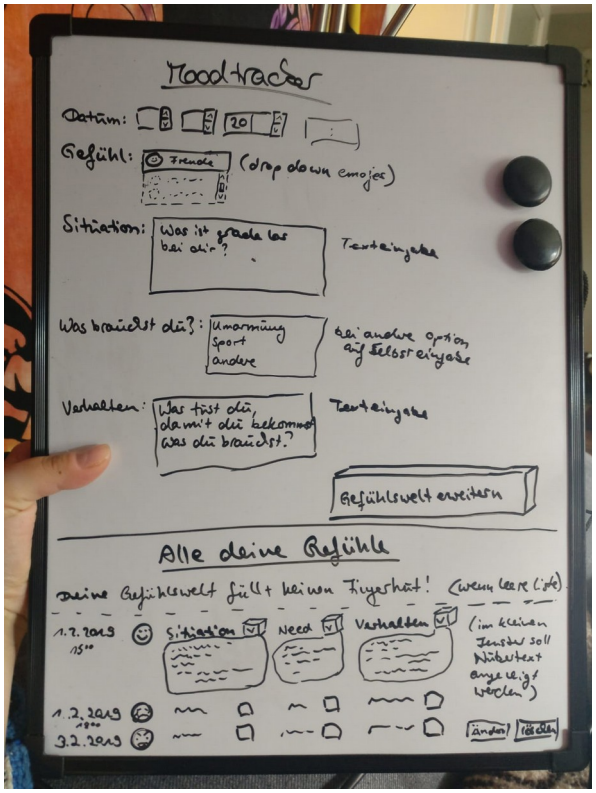
Dokumentation der Erstellung unserer WebApp mit React

Modulverantwortliche: Franziska Uhing
Dozent: Michael Jonathan Lee

Ein Projekt von:
Julia Noroschat 931052
Nadja Bender 931038

Vorgehensweise beim Erstellen unserer WebApp

Für unser Projekt haben wir uns zunächst überlegt was wir als ListenApp umsetzen wollen. Ein privates Ereignis hat uns inspiriert zu einem Moodtracker. Wir zeichneten uns auf, wie die grundlegende Funktionalität im UserInterface aussehen soll.
Man soll mit der Anwendung seine Gefühle in einer Situation aufschreiben können, unsere Version ist noch ausbaubar, aber basiert auf Gefühlstagebüchern die auch von Therapeuten genutzt werden, um Patienten ihre Gefühle näher zu bringen. Der User kann nun auswählen, an welchem Tag er diesen Eintrag machen will und füllt im Folgenden die Eingabefelder aus. Wenn er den Button „Gefühlswelt erweitern“ betätigt, wird der Gefühlseintrag der Liste hinzugefügt. In den Feldern selbst stehen Placeholder, die dem User helfen sollen zu erkennen, was er dort eintragen soll.



Nachdem uns klar war, wie wir die App aufbauen wollten, recherchierten wir verschiedene Ansätze der anfangs für uns nachvollziehbarste gilt als unsere Basis, er entstammt aus den Beispielen der Übungen zum Modul Mobile Anwendungen.

In der Realität mussten wir uns mangels KnowHow von manchen Ideen trennen und haben zum Beispiel das geplante Dropdown-Auswahlmenü für die Gefühle als simples Texteingabefeld belassen. Auch die geplanten Emoticons wurden weggelassen. Unsere App kann bis zu diesem Zeitpunkt, einen Status speichern, einen Eintrag hinzufügen und löschen. Die verschiedenen Möglichkeiten, die wir im Internet zum Bearbeiten fanden, konnten wir jedoch nicht mit unserem Code in Einklang bringen.

```
//ButtonDefinition und Ausgabe
@connect()
class Elem extends React.Component {

    deleteFeeling = () => {
        this.props.dispatch({type: DELETE_ELEM, payload: this.props.id})
    }
    changeEditMode =() => {
         this.props.dispatch({type: EDIT_ELEM, payload: this.props.id, isInEditMode: !this.state.feeling.isInEditMode})

    }
    renderEditView =() => {
        return <div>
            <input type="text" defaultValue={this.props.id.value}  >
    }

    renderDefaultView =() => {
        return <div onDoubleClick={this.changeEditMode}>
        {this.props.date}
            {this.props.feel}
            {this.props.situation}
            {this.props.need}
            {this.props.behavior}
            <button onClick={this.deleteFeeling}>Gefühl verdrängen</button>
        </div>
    }
    render() {
       return this.props.id.isInEditMode ?
       this.renderEditView():
       this.renderDefaultView()
    }
}
```

Diese Idee zum Bearbeiten funktioniert nicht, ist jedoch unserer Meinung nach jene, die uns zum gewünschten Ergebnis gebracht hätte. Unser Plan war es, wenn der Button betätigt wird, soll der Eintrag des Arrays mittels der ID in ein Eingabefeld umgewandelt werden, wo die zu bearbeitenden Einträge als Value vorgegeben sind. Es soll auf der selben ID gespeichert werden, wenn die Bearbeitung abgeschlossen ist. Angelehnt zu diesem Tutorial (http://www.js-craft.io/blog/03-How-to-build-a-React-Edit-Cancel-Text-Input-component/) haben wir zunächst versucht herauszufinden, wie wir den Value übergeben und an dieser Stelle sind wir an unsere Grenzen gestoßen.

```
case EDIT_ELEM: {
    const id = action.payload
    const newFeeling = [...state.feeling, {id: action.payload.id, date: action.payload.date, feel: action.payload.feel, situation: action.payload.situation, need: actic
    return {...state, feeling: newFeeling}

    for (let elem of state.feeling) {
        if (elem.id === id) {
            newFeeling.push(elem)
        }
    }
    return {...state, feeling: newFeeling}
```

In diesem Versuch war die Idee, dass man das Bearbeitenfeld, genauso nutzt, wie das Feld für den neuen Eintrag, in das die vorhandenen Daten mittels ID übergeben werden. Das Abspeichern soll dann ähnlich funktionieren, wie beim Löschen, dass der neue Status gerendert wird und dann die Daten neu eingegeben werden können.

Es erscheint uns nur plausibel, die ID als Value zu übergeben, aber unser Wissen reicht nicht aus um dies zu programmieren.

Besonderheiten und Probleme

Wie funktioniert React (componenten, wie funzen sie und wozu sind sie da?, daten in der app und der updatemechanismus, UI in zusammenhang mit Bäumen, wie funktionieren minimale updates, virtualDOM vs DOM

Wie funktioniert die Modularisierung

Screenshots und Infografiken