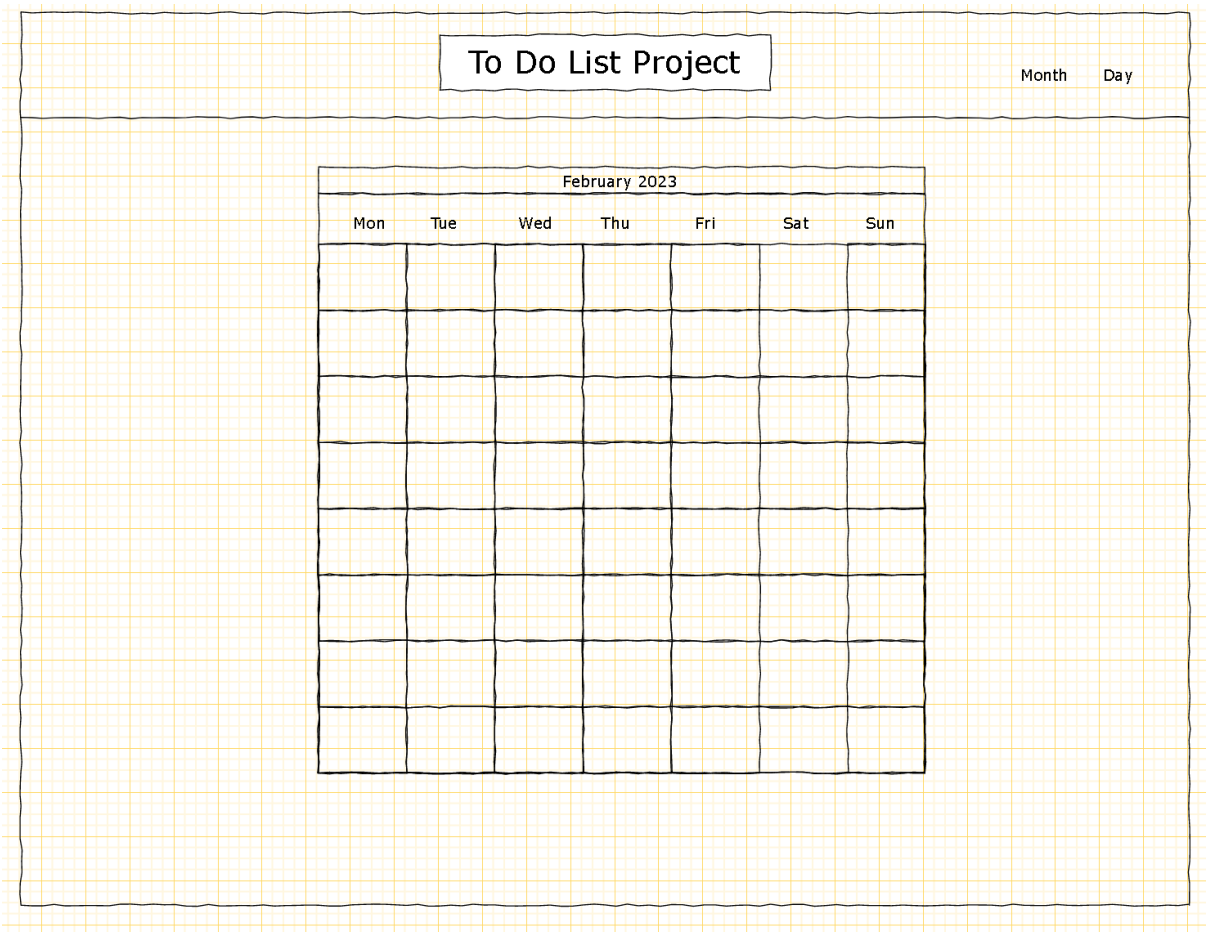
Dokumentation

Matr. Nr.: 2571020

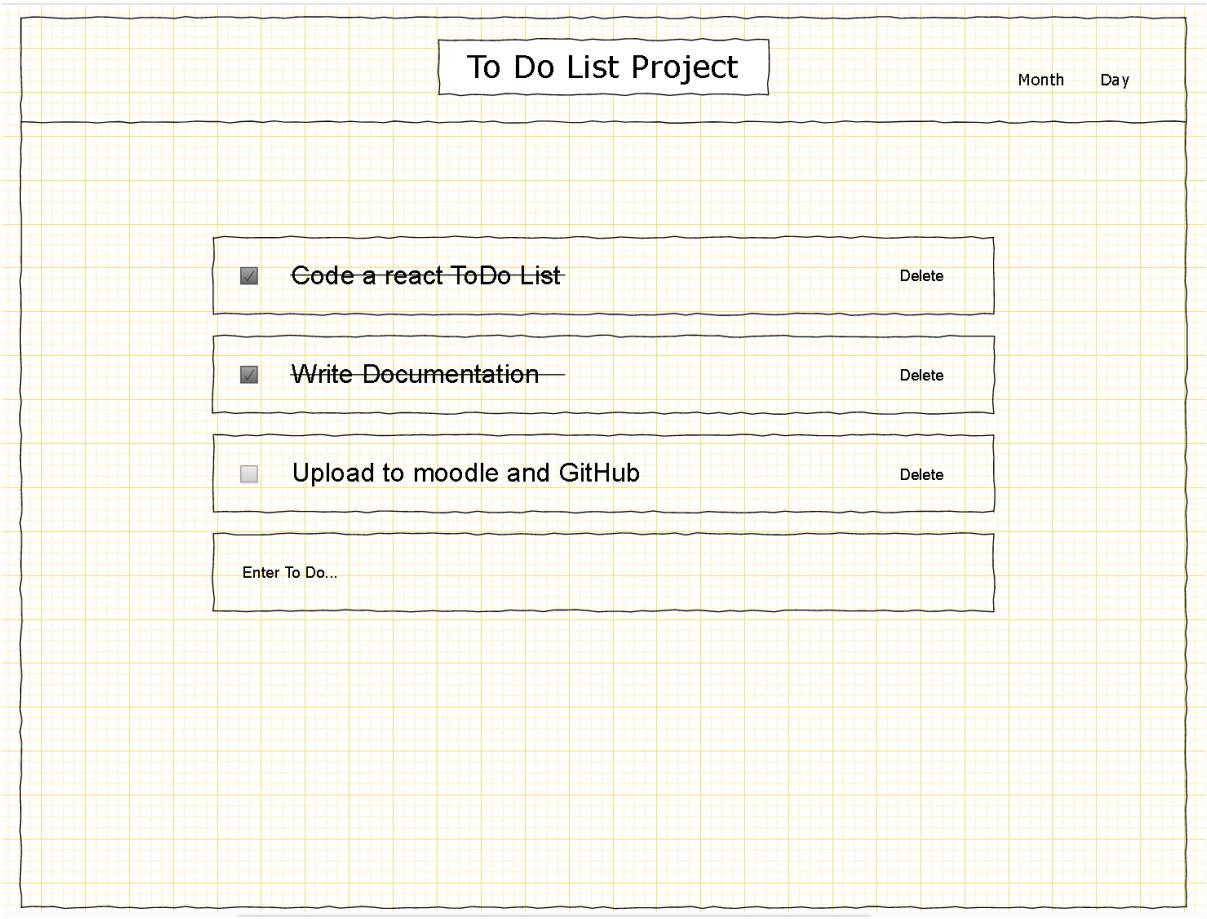
Project GitHub: https://github.com/adavidho/2571020

Paper Prototype

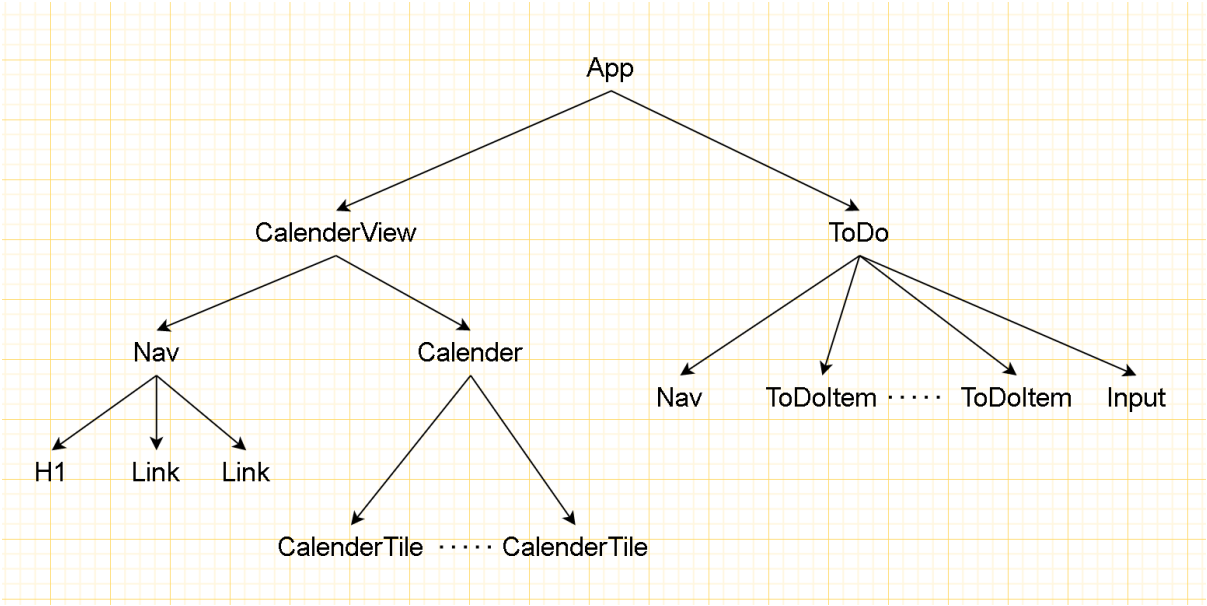
Calendar Page:



To Do Page:



Class Diagram



Requirements

The To Do List Project web app has two pages. A start page with a calendar of one month and a day page with a day specific to-do list. Both pages have a navigation bar on the top.

Start Page: this page shows a calendar view of one month. Each day has the number of to-dos stored in the app listed below the date this includes all to-dos, even those that are already checked. A click on any day opens the corresponding to-do list in the same tab (links to the day page). The start page has a navbar at the top as described in the navbar section of the requirements.

Day Page: this page shows the to-dos of the selected day in a list format. It allows the user to add new to-dos to the list and check of or delete individual to-do items. The day page also has a navbar at the top as described in the navbar section of the requirements.

Nav Bar: the navigation bar is shown at the top of both pages. It consists of three components: (1) A heading showing the title of the page and (2) a button to navigate to the start page (month view) and a button to navigate to todays to do list (day view).

Concept

This section describes the interpretation of the basic requirements and their implementation.

Start Page: The start page uses the react-calendar package for the month view. The calendar component is parametrised using props in a way to restrict it’s functionality to only showing the month February. This month was chosen since I expect the project will be graded then.

The tileContent prop is used to add the number of to-dos for each day to the default calendar tile, by calling a function that returns a custom component (CalenderTile) parametrised with day and toDoStore (data object). CalendarTile uses these props to look up the number of to-dos for the given day and returns a paragraph with this information.

Further the calendar uses the onClickDay property to call a function that navigates to the day page for the day the user clicked on. This function uses react router and its Link component to navigate to /todo/id where id ist the day the user clicked on.

Day Page: The Day / To Do Page (/todo/id) shows a to-do list for each day and an input field for new to-dos. The given day is extracted from the URL using the useParams hook. Each to-do item has the following components: (1) A text section that displays the text of each given task. (2) A checkbox to mark a to do as completed. When checked the text of a given to do is greyed out and crossed out using conditional rendering. (3) A delete button to remove the item from the to do list.

Data Handling: Each to-do item is stored in a JS Object with two items: (1) text - storing the task text and (2) checked - storing if an item is checked or not. All to-do items for a given day are stored in a list. The list of items is stored as a value in another object that has items for each day of February, (keys being the date (e.g. 1 for the first of a month)). This object is stored as a useState hook in App.jsx and passed down to the necessary sub components and updated using props and two-way-bindings.

Summary

To keep it short. After some trial and error everything worked out great.