## Задание

Разработать одностраничное приложение с одной экранной формой

Приложение должно предоставлять интерфейс для редактирования справочника сотрудников компании.

Каждый сотрудник имеет следующий список атрибутов

|  |  |  |
| --- | --- | --- |
| **Наименование атрибута** | **Тип значения** | **Обязательность заполнения** |
| ФИО | Строка | Да |
| Должность | Выбор из справочника(придумать самому) | Да |
| Дата рождения | Дата, выбор из календаря |  |
| Пол | Выбор с помощью радио переключателя с двумя значениями |  |
| Уволен | Булевский атрибут, редактируется с помощью чек бокса |  |
| [Дополнительно, не обязательно] Коллеги | Выбор из списка сотрудников. С возможностью мульти выбора |  |
|  |  |  |

Интерфейс должен состоять из панели инструментов с кнопками :

* «Сохранить изменения»
* «Обновить данные»
* «Удалить выбранного сотрудника»
* «Добавить нового сотрудника»

и двух областей:

* Слева список всех сотрудников
* Справа область для редактирования атрибутов сотрудника, выбранного в левой области

## Элементы управления

1. **«Сохранить»**

Как видно из перечня кнопок, все внесенные изменения не должны сразу отправляться на сервер, а должны сохраняться в памяти. Сохранение должно происходить при нажатии на кнопку «Сохранить». Если изменений не было, ток кнопка сохранения должна быть неактивной.

1. **«Обновить»**

При нажатии на кнопку «Обновить» должна обновлять данные с сервера. Если при нажатии на обновление есть несохраненные данные, нужно отобразить информационно окно с сообщением что все внесенные изменения будут потеряны и двумя вариантами ответа «Продолжить» и «Отмена»

1. **«Удалить»**

При нажатии на кнопку «Удалить» должно происходить удаление сотрудника выбранного в левой области. Если ни один сотрудник не выбран, кнопка «Удалить» должна быть неактивной.

1. **«Добавить»**

При нажатии на кнопку «Добавить» в левой области должен добавляться новый сотрудник, новый сотрудник автоматически должен выбираться в списке.

1. **«Список сотрудников»**

Список всех сотрудников, с отображением их ФИО. При клике на элемент списка он должен визуально помечаться как выбранный.

1. **«Карточка сотрудника»**

Область со списком атрибутов и их значений выбранного сотрудника.

## Хранение данных и взаимодействие с сервером

Данные необходимо хранить в localstorage браузера.

Методы по взаимодействию с хранилищем должны быть абстрагированы от реализации самого хранилища

## Стек

При реализации проекта необходимо использовать следующие технологии:

* yarn/npm в качестве менеджера пакетов
* webpack для сборки проекта
* TypeScript
* React

## Визуальное оформление

Особых требований к визуальному оформлению не предъявляется. Можно использовать библиотеки визуальных компонентов для ускорения реализации(<http://getbootstrap.com/> , <http://www.material-ui.com/#/> или другие)

## Результат

Исходный код тестового задания необходимо разместить на github

Так же необходимо опубликовать работающее приложение на хостинге и предоставить ссылку, например github-pages.

## Дополнительно

Если есть желание и возможность можно реализовать дополнительные требования. Их выполнение положительно повлияет на оценку результата выполнения тестового задания.

* Использовать mobx для управления состоянием приложения
* В карточку сотрудника добавить атрибут коллеги (см. таблицу атрибутов сотрудника)
* Реализовать возможность добавления дополнительных атрибутов каждому сотруднику. Т.е. в карточке сотрудника должны появиться отдельная область, в которой можно добавлять доп. атрибуты, с указанием типа их значение (строка, число, дата) и возможностью сохранения их значений. У каждого сотрудника перечень доп. атрибутов свой.