Essentials:

1. <http://facebook.github.io/react/docs/getting-started.html>
2. <https://github.com/arkency/reactjs_koans>
3. <http://tutorialzine.com/2014/07/5-practical-examples-for-learning-facebooks-react-framework/>
4. <http://facebook.github.io/jsx/>
5. <https://christianalfoni.github.io/react-webpack-cookbook/index.html>
6. <http://gaearon.github.io/react-hot-loader/>
7. <https://github.com/rackt/react-router>
8. <http://facebook.github.io/flux/docs/overview.html>
9. <http://reactfordesigners.com/labs/reactjs-introduction-for-people-who-know-just-enough-jquery-to-get-by/>

Additional:

1. <http://rackt.github.io/redux/>
2. <https://github.com/FormidableLabs/radium>
3. <https://github.com/facebook/immutable-js>
4. <http://conf.reactjs.com/schedule.html#immutable-data-and-react>
5. <https://scotch.io/tutorials/getting-to-know-flux-the-react-js-architecture>
6. <https://scotch.io/tutorials/creating-a-simple-shopping-cart-with-react-js-and-flux>
7. <https://www.youtube.com/watch?v=DgVS-zXgMTk>
8. <http://react-bootstrap.github.io>
9. <http://jaketrent.com/post/send-props-to-children-react/>
10. <https://egghead.io/series/getting-started-with-redux>

**Practical task**

The name of the application is “Note Manager”. The application must help users create, edit, and delete their personal notes, and maintain the structure of folders and tags.

For starting the project, we strongly recommend you to use the [create-react-app](https://github.com/facebookincubator/create-react-app) tool so that you don't need to perform build configuration.

**Functional Requirements:**

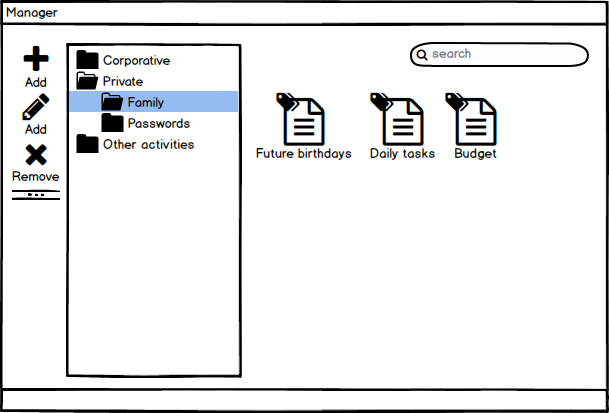
1. The “Search” function can be switched to “simple” and “advanced” modes. In “simple” mode the search goes only through the names of notes, and in “advanced” mode can search through the content of notes and tags.
2. “Search” must support autocomplete (suggestion - [React Autocomplete](https://github.com/rackt/react-autocomplete)).
3. The order of the notes inside a folder can be changed by using drag-and-drop (suggestion - [React DnD](https://github.com/gaearon/react-dnd)).
4. Insightful tooltips should appear when a user hovers over any element of the interface.
5. Names of notes and folders must be editable inline.
6. After clicking on a note icon, the details of that note be displayed with full text and tags.

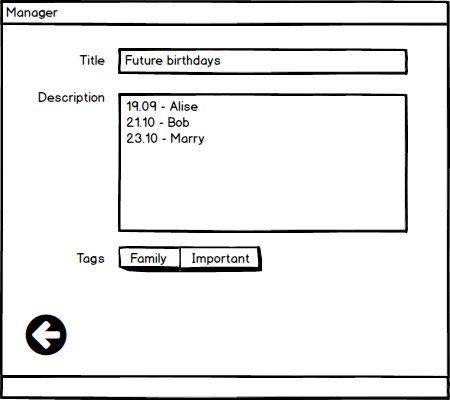
**Other Requirements:**

1. The solution must utilize [ES6](https://babeljs.io/docs/learn-es2015/#ecmascript-6-features) and have two build types: for production and development respectfully.
2. [React Hot Loader](https://github.com/gaearon/react-hot-loader) or [react-transform-webpack-hmr](https://github.com/gaearon/react-transform-webpack-hmr) should be used for development. It’s more convenient and it will burst your productivity.
3. The solution must use an intelligent routing system in order to save the state of opened directories, search results and note details. It is necessary to handle 404 and 500 statuses (suggestion - [React Router](https://github.com/rackt/react-router)).
4. Minimize the usage of `state` in components by adding a Flux implementation to the solution. Provide a description of every action or action creator and the tasks they perform (suggestion - [Redux](https://github.com/rackt/redux)).

The use of third-party components is permitted. Using new approaches to writing styles in React.js applications ([CSS Modules](https://github.com/css-modules/css-modules) / [Radium](https://github.com/FormidableLabs/radium) (inline styles) / [PostCSS](https://github.com/postcss/postcss)) is a plus.

**Mockups**





**Backend**

You can download mockup server [here](https://dastream.dataart.com/Internal_DataArt_Courses_and_Seminars/Training.Coordination/React.js/react-practice-backend.zip).

**API**

**Get index page**

GET     /

Returns index.html

**Get directories**

GET     /directories

Returns directories list

**Create Directory**

POST    /directories

Create a new directory and return it along with its ID. You must specify the parent ID and name fields

**Update Directory**

PUT     /directories/:id

Update directory

**Delete Directory**

DELETE  /directories/:id

Delete directory

**Get Notices**

GET     /notices

Return notices list

**Create Notice**

POST    /notices

Create new notice and return it with an ID and position. You must specify the directory ID, title, description and tags fields

**Update Notice**

PUT     /notices/:id

Update notice (can be used for changing a position)

**Delete Notice**

DELETE  /notices/:id

Delete notice