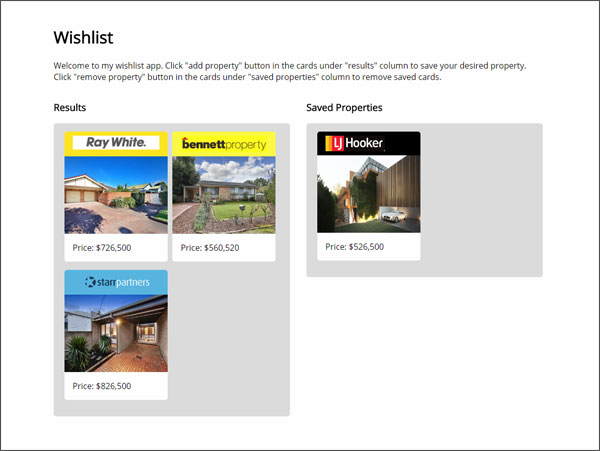
**Overview**

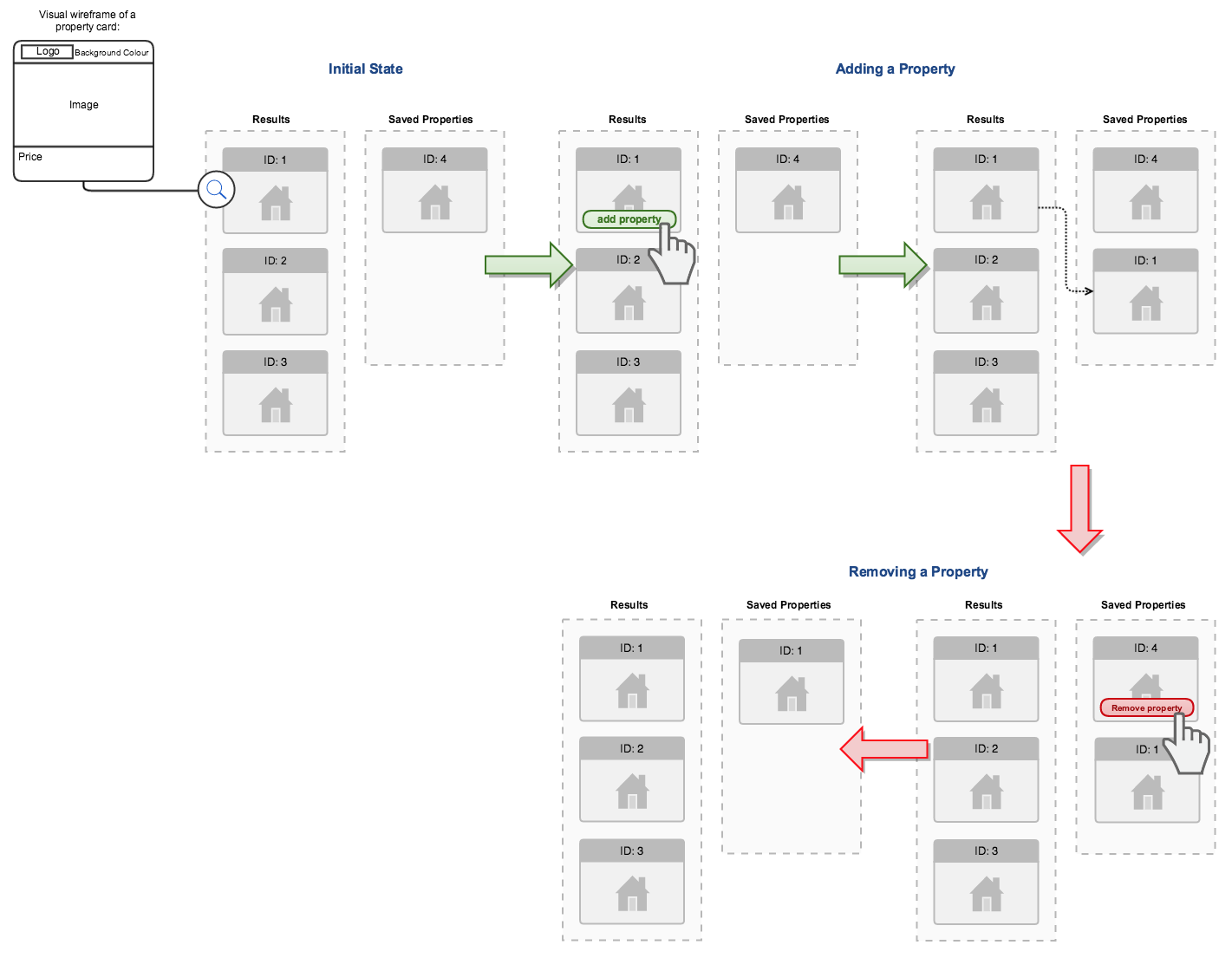
Welcome to my property wish list app. It allows you to save and remove property cards. Screenshot below is an overview of the app.



Hovering over a card in the “results” column will display an “add property” button. When clicked, that property card will be duplicated and saved in the “saved properties” column. Each property card can only be saved once.

Hovering over a card in the “saved properties” column will display a “remove property” button. When clicked, that property card will be removed from the “saved properties” column.

Diagram below illustrates a user’s potential journey.



**Requirements**

There are a few requirements to install and run this project:

* [Node js](https://nodejs.org/en/)
* and or [Yarn](https://yarnpkg.com/lang/en/)

**Installation**

To install packages, run:

**Npm** or **yarn**

**Scripts**

To run the app in development mode viewable at <http://localhost:3000/>, run:

**Npm start** or **yarn start**

To launch test runner, run:

**Npm test** or **yarn test**

To build production version of app in “build” folder, run:

**Npm run build** or **yarn build**

**Data flow**

Diagram below illustrates top down data flow from the Json data(file path: “data/Data.js”) into components on load.

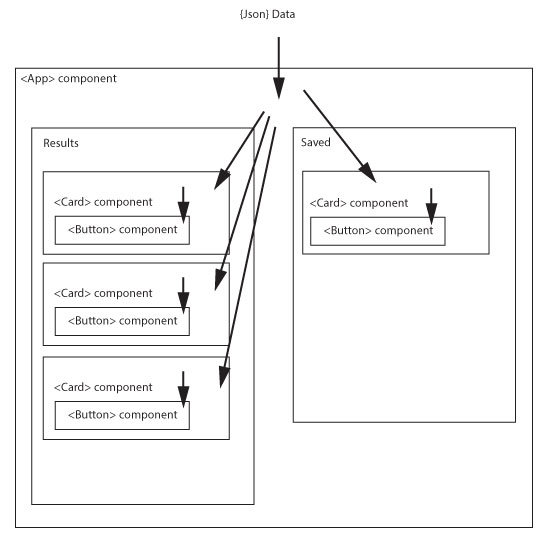


Diagram below illustrates data flow when a card in “results” column is clicked.

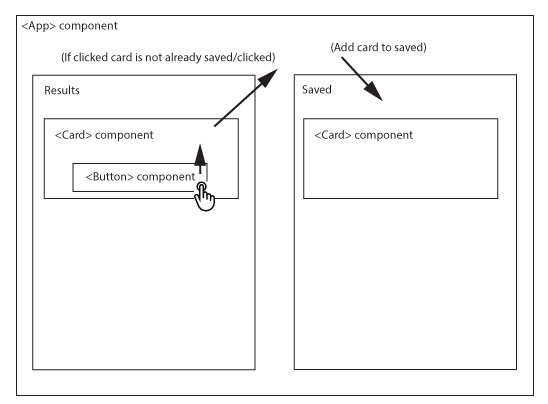
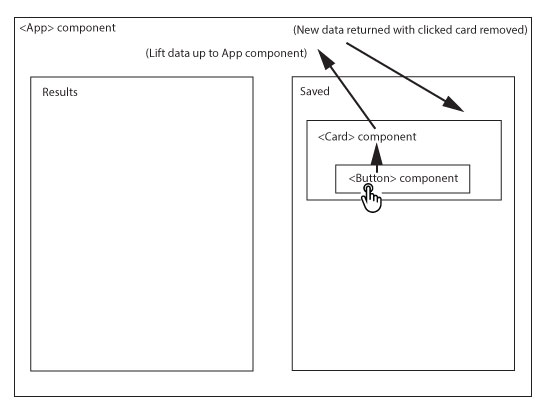


Diagram below illustrates data flow when a card in “saved” column is clicked.



**Component structure**

**App component**

File path: “App.js”

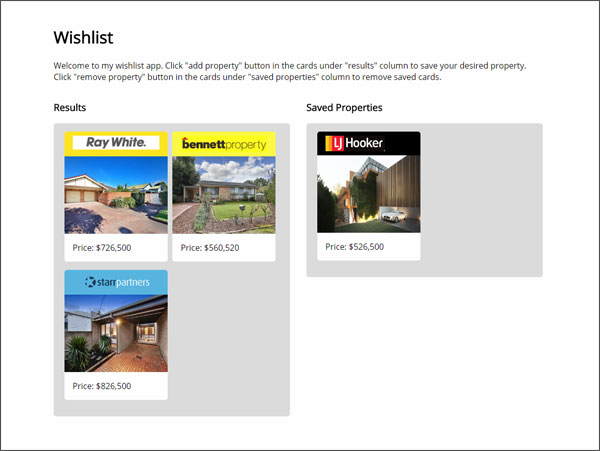
The App component is the master component that contains the “results” cards and “saved” cards. The main json data gets passed into this App component as props which are then split up into “results” and “saved” states/columns. Each row of data in the “results” and “saved” state/column gets rendered as a Card component. This App component also houses the logic to determine if a card gets saved or removed in the “results” column.

Props:

* results: Initial cards data
* saved: initial saved cards data

State:

* results: copied from props initial cards data. This does not change
* saved: copied initial saved cards data. “handleAddProperty” and “handleRemoveProperty” functions update this state to render saved cards



Example of props and state below.



**Card component**

File path: “components/Card.js”

The card component receives individual house property data as props and renders it. The card contains a Button component. It also receives these props:

* isButtonDisabled: to determine if Button component inside will be disabled. This occurs if the card has already been saved.
* type: to determine if the Button component inside will be “add” or “remove”
* onHandleClick: When the Button component gets clicked, data gets lifted up to the App component.



Example props of Card component in “results” column below



Example props of Card component in “saved” column below

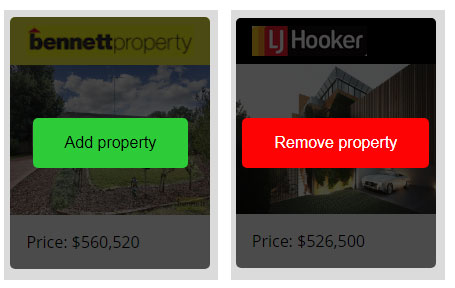


**Button component**

File path: “components/Button.js”

The Button component inside the Card component receives these props:

* type: to determine text and class name to apply to button tag
* disableButton: to determine disabled attribute on button tag
* onClick: to trigger and return/lift click event back to Card component



Props of the two different button types below  


**Testing**

To launch test runner, run:

**Npm test** or **yarn test**

This runs the testing script in “App.test.js”. The test uses “Jest” and “enzyme”.

List of tests done:

* On start, render the App without crashing
* There are 4 cards loaded at the start
* Total of 4 cards becomes 3 after a “remove property” button is clicked
* Cards get saved to “saved” column after “add property” button is clicked
* The “add property” button gets disabled after getting clicked
* Single card gets rendered with data passed in