# AG's Agworld Accessory Design

**DATA**

const title = `Agworld Accessory`;

let currentCompany, ~~currentFarm,~~ currentField;

let currentPage = 'login';

let lastViewedWeatherStation;

let company, farms, fields;

+ scratch storage for each field, whether is it expanded in the weather pane (defaults to false).

**Login page**, like <https://au.agworld.com/> but with stubs.

Clicking "LOG IN>"...

=>

Sets company using a stub function, to an object for company like at [https://us.agworld.co/user\_api/v1/docs#company](https://us.agworld.co/user_api/v1/docs" \l "company) => [Simulated Response](https://us.agworld.co/user_api/v1/docs/simulate/company/get_a_company) ~~but with key id = 1 and company name = "My company" and change the Lat/Long [etc]~~.

sets currentCompany ~~to company 1~~:

Farms: Get this from a function that takes the company id (but just console logs that), an object containing 3 farm objects like at https://us.agworld.co/user\_api/v1/docs#farms and they all have ~~company\_id = 1, and storage for weather station for each field~~.

Fields: Get this from a function that takes the company id (but just console logs that), an object containing multiple field objects like at [https://us.agworld.co/user\_api/v1/docs#fields](https://us.agworld.co/user_api/v1/docs" \l "fields) including at least 1 or 2 fields for each farm object (and farm\_id set accordingly). Give e*ach field record storage for weather station.*

=>

...and then displays the Weather page. currentPage = 'weather'

FOR ALL OTHER PAGES:

**Green title bar** with AG logo on the left, then title, then "Weather" Clicking that: currentPage = 'weather' and

on the right {company name}, a person icon button and then "By Andrew Greenhill". White text.

Clicking on the person icon button reveals a menu with options for Help & Support and Logout.

Clicking Logout goes back to the Login page with currentPage = 'login'.

**Weather page left pane**

Tall rectangle on LHS (with rounded corners) containing:

Have an X in the top right corning. Clicking that: currentPage = 'none'

> Farm names.

Clicking on them changes to "v Farm names" and lists Fields.

Each field has a ~~weather icon~~ button ~~on the RHS~~. Clicking that sets currentField and opens weather page **right** pane. currentPage = 'weather+obs'

Clicking anything else hides the weather page right pane.

Should have a vertical scroll bar if insufficient height for all farms & fields to be displayed.

=>

**Weather page right pane**

Pre-opening: Look at currentField, and which weather station that uses (if any).

If a weather station is configured for that field then

{

If that is different to the *last-viewed weather station* then blank the Weather Observation data.

Display fields for data like in Agworld's New Actual Weather Record but

Weather Observation instead of Weather Record.

Request data from the weather service

When that arrives, the weather station name is shown at the top, ✔

or if data doesn't arrive then display the error.

}

Have a way to change units between metric and USA or at least declare "metric". [Also date format]

Display a button to configure the weather station. Pressing that: currentPage = 'weather+obs+config'

=>

**Weather station configuration**

Display at the bottom (or top?) of the pane a field(s) for getting the closest weather station.

Initially have that take, for example IDW60801/IDW60801.95622

Need to store that in the field record… <= have a stub for writing that.

Later, ideally it'd nice to use [package/weather-au-api](https://www.npmjs.com/package/weather-au-api)'s to get a list of locations…

Have an OK button Pressing that: currentPage = 'weather+obs'.

**None**

If currentPage = 'none' then display `Press 'Weather'`.

Then:

**CSS!**

**Remove QQQ**

**Add internal documentation**

**README.MD**

**Add ………………………..**

Change "Loading…" to show progressive updating.

**Put issues in README and/or GitHub**

c.f <https://www.agworld.com/au/new/weather-records-agworld-actuals/>

[[React Router]]

**[Add [Jest] tests] <= research the files that create-react-app provided.**

**Create postgresQL database...**

**GitHub Pages**

Put on GH pages.

+ Extend my Heroku proxy to allow this.