alexwhite

# portfolio

## Contents


UX Research 1

Design Ops 2

Design 3

Development 4

alexwhite

# portfolio

## UX Research

Over the last year, I have been given time to develop a UX process from it's early stages. In the past, I have been involved with other systems of UX, defined by the companies I used to work for.

I started by mapping down the standard process and began to overlay it with the tasks and methods I have already been taught.

### Overview

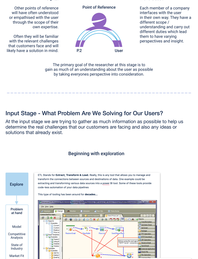
The UX team typically has three questions to answer:

- **What problem are we solving for our users? (input)**
- **How do we solve it? (collaborate / investigate)**
- **How do we know that we have solved it? (output)**

A user is defined as someone that uses the product to achieve their goals, ideally a customer.

In all three stages, the process is **collaborative**

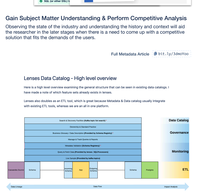
Everyone involved comes together through what I see as the "Two arms of UX"




## Input Stage - What Problem Are We Solving for Our Users?

At the input stage we are trying to gather as much information as possible to help us determine the real challenges that our customers are facing and also any ideas or solutions that already exist.

### Beginning with exploration


Explore
Problem at hand
Model
Competitive Analysis
State of Competition
Market Fit
Solutions & IA
ITI, Stands for Behind, Transition & Look. Really, this is only tool that allows you to manage and organise your research. It allows you to create a hierarchy of your research, tag and filter, and add/edit/delete/copy/move data across the page. Some of these tools provide code ready examples of your data structure.
The type of coding that best should be Metadata.
Data Catalog - Jan Jan Jeremy Kim
Metadata Attributes
Metadata Attributes
Metadata Attributes
Dynamic Data Page
Dynamic Data Page
Dynamic Data Page

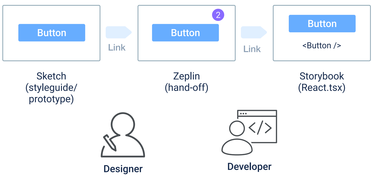
alexwhite

# portfolio

## Design Ops

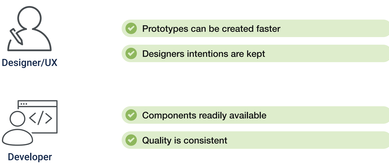
The goal of Design Ops is to **bridge the gap betwen design and development.** Traditionally a design is rendered in programs like Sketch and are handed off to developers.

In my past roles I have been given the opportunity to develop bespoke component libraries. This was later expanded into a wider design system that covers both design and development.



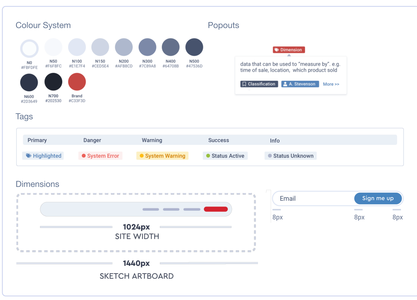

## Design Infrastructure

The diagram above gives a birds eye view of the most recent design system that both me and my team have constructed.



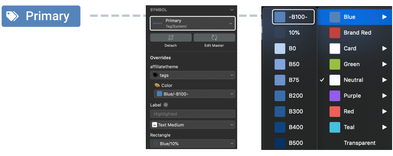

## Values

Having such an infractructure in place creates the following values for Designers & Devs:



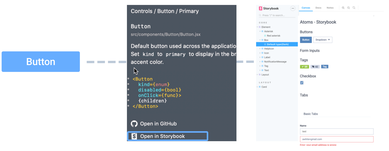

## Sketch Styleguide

I have been given the chance to define and revamp several styleguides in the past. Usually I start by defining the atoms and working my way up to creating reuseable patterns.



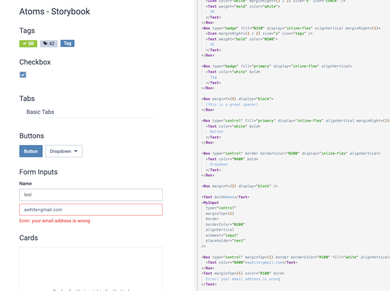

## Sketch - Library

To make life easier for designers we consolidated the styleguide into a Sketch library which had an organised set of typography, colours, symbols and patterns.




## Zeplin - Connected Components

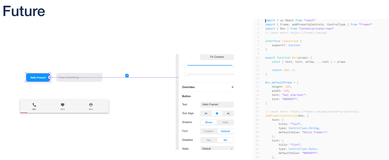
Zeplin has allowed us to synchronise our Sketch Library to our code components and also to Storybook.




## Storybook - React Library (atoms)

At the foundational level, all atoms exist as code components. The approach that I took here was to develop a general purpose framework to help developers render screens faster and with more consistency.

## Future



## Framer

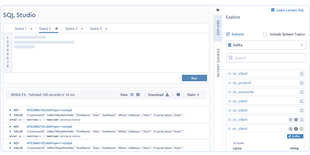
Right now I am investigating the possibility of being able to create prototypes with real React components. Framer can import components from your React component library, which can then be hooked up via Framers interface. Framer also supports private packages, developer hand-off, animations and all the prototyping facilities you would find in sketch.

You can find more information about both the React component library and Framer in the Develop section of my portfolio

alexwhite

# portfolio

## Design




### SQL studio

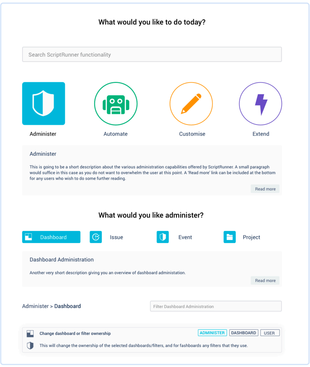
Fetching live data from Kafka & other data sources through SQL.
Notable features include a minified data catalog experience on the side panel.

### Data Catalog

Designed to house many different storage technologies, including HDFS.
Tagging & metadata management were key aspects here. Usability testing revealed that all tags on the prototype must be clickable and that the filtering must be faceted.

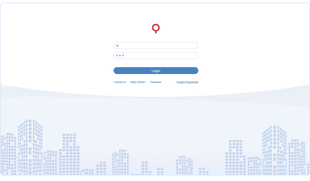
### TruRating Dashboard

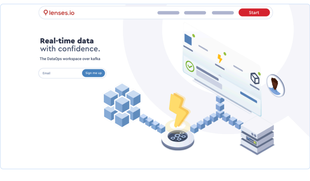
Concept was created as an initiative to revamp what was already in place. This also tied in well with the design system I was creating at the time (with stencil JS)

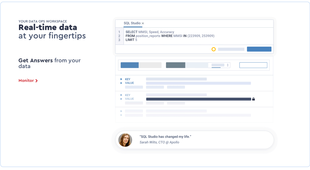


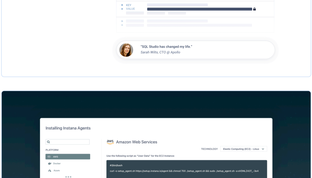

### Scriptrunner Onboarding

The UX here was predetermined, the addition added was that the presentation would unfold as the user steps through the options. In the end, it was consolidated as a single Browse component. We had great usage metrics with Segment - the information we were able to track on Tagging was used as the basis for the content found in Adaptaxist Script Library.




### Visual Design

lenses.io

CULINARY ARTS MASTERS

Business Plan bit.ly/33Ortgc

APPLE OF EDEN

### Logo

PLAYLIST

1 **Lenses.io**
*At The Heart of Data* bit.ly/2GUFTT2

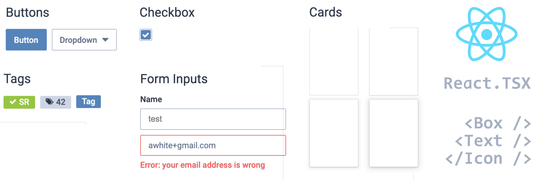
2 **Lenses.io**
*Audio Jingle* bit.ly/34UXmmS

### Musical Composition - Adverts & Audio Branding

alexwhite

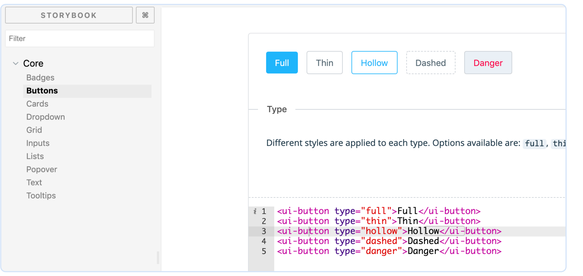
# portfolio

## Develop




### React Component Library

I have also created a small but effective library in React. There was more careful thought on the underlying API this time round. The example above is rendered using only 3 basic but highly customisable building blocks. Currently, I plan to release this as an open-source library called 'Neutronic'. What I have built here are 'neutrons', as they can be used as a neutral starting point to build atoms!

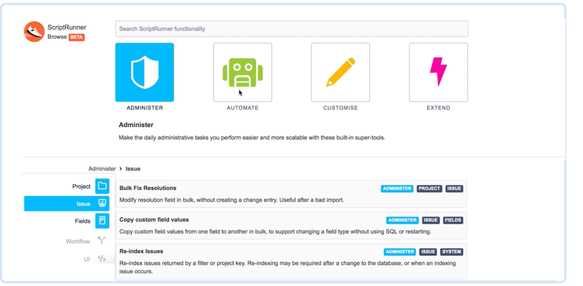



### Stencil Component Library

bit.ly/2GNYMHC

Made as a 'future-proof' library which would act as a stepping stone towards a framework-agnostic design system. Stencil does this by compiling it's components down to the web-component spec, allowing you to import them into any framework. The api that is provided by Stencil is very similar to React and both use JSX with TS.

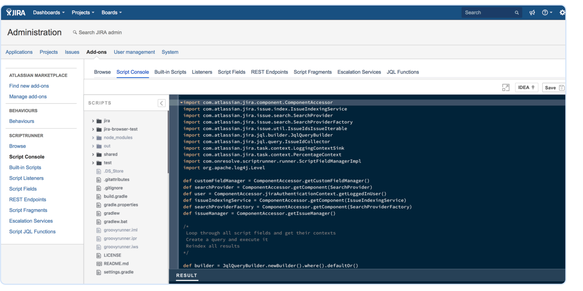
As well as the library, I had also hooked it up to storybook to create live documentation.




### Browse Page

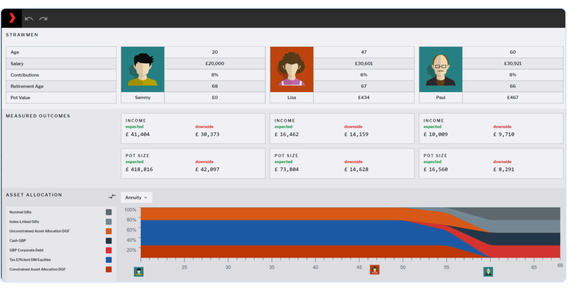
bit.ly/30XBgyW

This was the consolidated 'onboarding / browse' component for ScriptRunner. It featured different top-level categories which were colour-coded according to the brand, and also sub-categories. Gathering user metrics were also apart of the development effort and we managed to gather great usage data which helped the company decide which areas of Jira their scripts should focus on the most.




### Code Editor

Allows users to write scripts inside of Jira. The editor included features such as debugging, intellisense and bespoke FileTree which matched elements of the brand and also offered helpful micro-interactions on-drag.




### Analytics Dashboard

One of my first React.js projects - a data visualisation app for pension scheme plans. Bespoke components with Victory charts used as the underlying charting library.