Pogo Gakkō Style Guide

Introduction and Overview

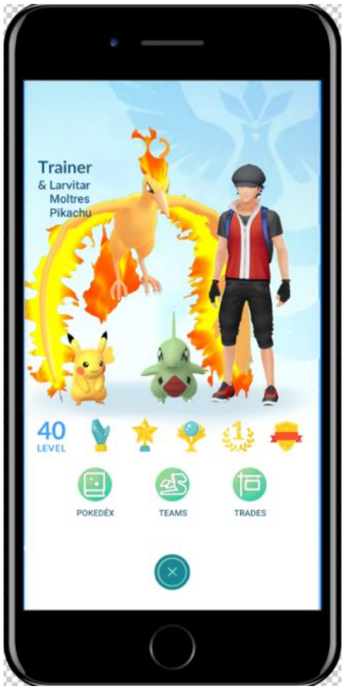
Our goal for Pogo Gakkō is to capture the look and feel of Pokémon Go as closely as possible while also meeting the functional requirements of the application. We are aiming for a mobile first design that should function responsively as well. We hope to match the general layout of Pokémon Go such that Pogo Gakkō almost appears to be an extension of the Pokémon Go itself.

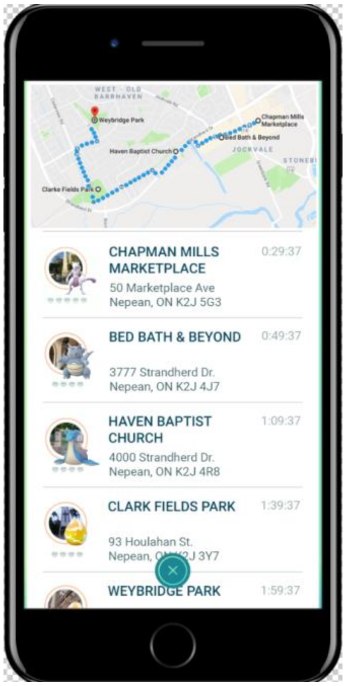
We are aiming to create two main pages for this application. The page is the “Profile” page, which displays dynamic player information that the user can set up themselves. The second page is the “Raids” page, which displays a map along with a list of available raids to participate in for Pokémon Go.

To meet these needs, we are using Ionic 4.0 as a UI toolkit and front-end framework. This framework provides a lot of the necessary theming and layout options for us. Ionic allows for the creation of a very dynamic and responsive layout and design, and it is used to create progressive web applications that function well and look good on any platform or device.

Layout

For both the “Profile” and “Raids” page, we have created mock-ups that we are aiming to match by using Ionic’s theming and layout options:





CHAPMAN MILLS MARKETPLACE 0:29:37
50 Marketplace Ave
Nepean, ON K2J 5G3

BED BATH & BEYOND 0:49:37
3777 Strandherd Dr.
Nepean, ON K2J 4J7

HAVEN BAPTIST CHURCH 1:09:37
4000 Strandherd Dr.
Nepean, ON K2J 4R8

CLARK FIELDS PARK 1:39:37
93 Houlahan St.
Nepean, ON K2J 3Y7

WEYBRIDGE PARK 1:59:37

Wherever possible, we are aiming to rely on Ionic's grid system for placement of graphics and elements. Not only does this streamline development time, but this helps to ensure that the application remains responsive. Ionic components are very responsive, and so they can be depended on to ensure that the application functions and displays correctly on many devices.

For more information on Ionic's grid system, see:

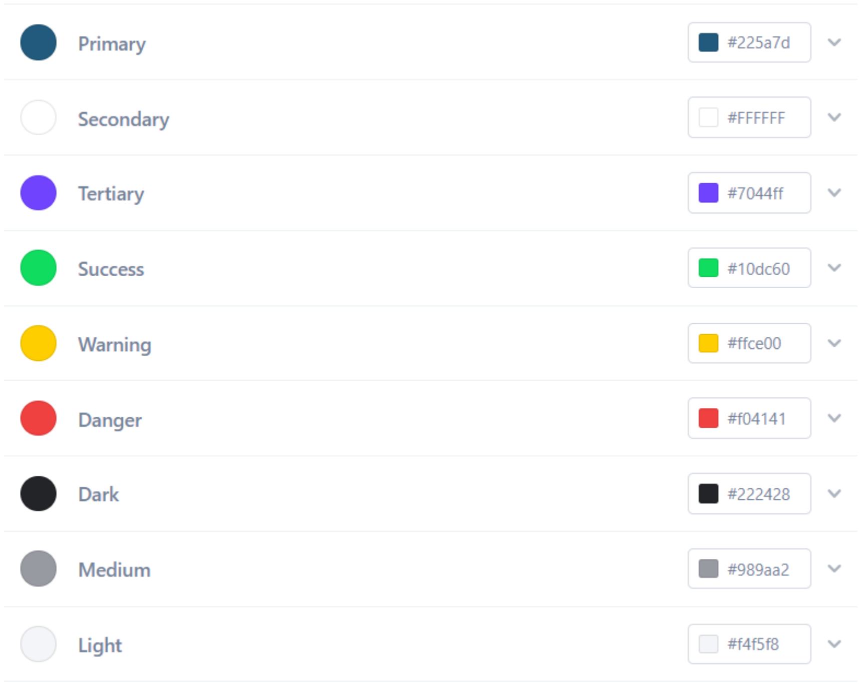
https://ionicframework.com/docs/layout/grid

For a full overview on Ionic component and layout options, see the documentation:

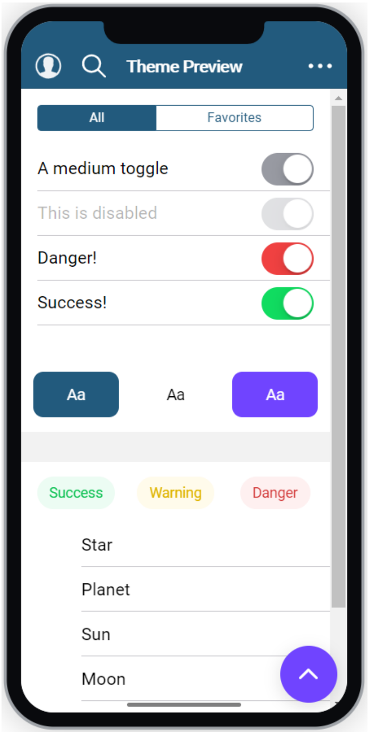
https://ionicframework.com/docs/

Colour Palette

Ionic allows for the creation of a colour palette which can be bound to CSS properties, which allows them to be easily accessed and used across the application. The scope of our application is quite small, and so we did not deviate far from the default Ionic colour scheme as most colour properties will likely go unused (i.e. in our scope there are no planned uses for a "warning" or "danger" colour). We have decided to use **#225A7D** as the primary colour, as it is the official Pokémon Go primary colour. We are using plain white as a secondary colour for contrast. See the full colour palette below, which can be configured using Ionic's colour picker.



Primary	#225a7d
Secondary	#FFFFFF
Tertiary	#7044ff
Success	#10dc60
Warning	#ffce00
Danger	#f04141
Dark	#222428
Medium	#989aa2
Light	#f4f5f8



For more information on Ionic's colour generator and theming options, please see:

https://ionicframework.com/docs/theming/color-generator
https://ionicframework.com/docs/theming/advanced

Fonts & Iconography

Our font selection is *Lato* for heading text, and *Roboto Condensed* for body text. Both fonts are available on Google fonts and are the official fonts of Pokémon Go. These can be found here:

https://fonts.google.com/specimen/Lato
https://fonts.google.com/specimen/Roboto+Condensed

For iconography we are using Ionicons, which are open source icons built into Ionic. We are using some of these icons as placeholders or approximations of what we hope future icons should look like. Ionic also allows the creation of custom icons that can be used across the app.

For more information on Ionic icons, see:

https://ionicons.com/
https://ionicons.com/usage

Assets

We have pulled existing Pokémon Go assets from here: https://github.com/ZeChrales/PogoAssets. We are using these wherever possible instead of creating our own. Given the large number of assets we may need (for example, to display the large selection of available Pokémon), using an existing pool of assets helps with development time. This also helps us to create an application with the same look and feel as the original Pokémon Go app.

For the “Raids” page map, we are using default styling, map tile design, and marker images provided by openstreetmap and leaflet.js. You can view more on these here:

https://leafletjs.com/reference-1.4.0.html#marker

https://wiki.openstreetmap.org/wiki/Use_OpenStreetMap