New UI Design Documentation

VPI is responsive HTML design based on the CSS framework Bootstrap 3. It utilizes all of the Bootstrap components in its design and re-styles many commonly used plugins to create a consistent design that can be used as a user interface for backend applications such as the VPI Platform. Design of VPI is based on a modular design, which allows it to be easily customized and built upon. This documentation provides a brief introduction to the various components that have been designed as part of the VPI New UI.

The VPI design webapp uses Bootstrap to power the proposed design for the new interface. Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

Original structure of the New UI Design webapp:

```
VPI/
├──── dist/
│    ├──── CSS/
│    ├──── JS
│    ├──── img
├──── build/
│    ├──── less/
│    │    ├──── VPI's Less files
│    └──── Bootstrap-less/ (Only for reference. No modifications have been made)
│        ├──── mixins/
│        ├──── variables.less
│        ├──── mixins.less
└──── plugins/
    ├──── All the customized plugins CSS and JS files
```

Dependencies

The following two frameworks have been used:

1. Bootstrap 3
2. jQuery 1.11+

The New UI folder already contains both of these libraries in their latest upgrades.

Plugins

Various plugins have been installed using the available javascript and css frameworks on the internet. The following plugins have been used to extend the features beyond Bootstrap:

Charts	Form elements	Editors	Others
ChartJS	Ion Slider	Bootstrap WYSIHTML5	Data Tables
Flot	Date Picker	CK editor	Full Calendar
Morris.js	Date Range Picker		jQuery UI
Sparkline	Color Picker		jQuery Knob
	Time Picker		jVector Map
	Select2		Slim Scroll
			Pace

Layout

The layout consists of four major parts:

1. Wrapper .wrapper. A div that wraps the whole site.
2. Main Header .main-header. Contains the logo and navbar.
3. Sidebar .sidebar-wrapper. Contains the user panel and sidebar menu.
4. Content .content-wrapper. Contains the page header and content.

A set of options can be applied to the main layout. Each on of these classes can be added to the body tag:

1. Fixed: use the class .fixed to get a fixed header and sidebar.
2. Collapsed Sidebar: use the class .sidebar-collapse to have a collapsed sidebar upon loading.
3. Boxed Layout: use the class .layout-boxed to get a boxed layout that stretches only to 1250px.
4. Top Navigation use the class .layout-top-nav to remove the sidebar and have your links at the top navbar.

Note: you cannot use both layout-boxed and fixed at the same time. Anything else can be mixed together.

Components

The main All of Bootstrap 3 components have been used. To name a few, Glyphicons, Dropdowns, Button groups, Button dropdowns, Input groups, Navs, Navbar, Breadcrumbs, Pagination, Labels, Badges, Jumbotron, Page header, Thumbnails, Alerts, Progress bars, Media object, List group, Panels and much more.

Bootstrap uses reusable components built to provide iconography, dropdowns, input groups, navigation, alerts, etc. Refer Bootstrap Documentation.
http://getbootstrap.com/components/

Main Sidebar

The sidebar has been used to provide a context menu for quick and easy navigation around the different Domains of the platform and also navigation around the current Domain itself. The VPI Logo at the top left corner is used to toggle between the narrow style and wide styles of the main sidebar.

Control Sidebar/Module Selector

The control sidebar has been used as a Module Selector. Control sidebar is the right sidebar. The sidebar has been designed to provide two different show/hide styles. The first allows the sidebar to slide over the content. The second pushes the content to make space for the sidebar. Either of these methods can be set through the Javascript options. The sidebar can be controlled with a toggle button to open/close it. By adding the attribute data-toggle="control-sidebar" to any button, it will automatically act as the toggle button.
Info Box
Info boxes are used to display statistical snippets. There are two types of info boxes.

Box

Info Box can used to display some snippets of textual and graphic information. Info boxes have been designed to present the projects. Project name, Date created, Date Updated and AUthor etc can be presented in a modular form.
The **Box** component is the most widely used component throughout the proposed design. It can be used for anything from displaying charts to just blocks of text. It helps in encapsulating the different elements of the view. The Box comes in different styles namely, Default, Primary, Warning, Success, Danger etc which provide color headers to indicate the type of content. **Solid Boxes** are alternative ways to display boxes. They can be created by simply adding the box-solid class to the box component.
Also **labels, badges, pagination, tooltips, inputs** etc can be added to the box tools. The data-widget attribute provides boxes with the ability to collapse or be removed. The buttons are placed in the box-tools which is placed in the box-header.
Loading state feature is also available. To simulate a loading state, simply place this code before the .box closing tag.

Charts

KPI Dashboard of the Analyst pages heavily uses the various charting design elements. **ChartJS** among other charting frameworks has been used for the data visualisations because it provides a modular, interactive and responsive design. Different types of charts such as Line, Bar, Donut and Area charts along with legends have been designed to help visualise the data.

Form Elements

Form elements have been extensively designed to provide the interface needed for the ‘Create new project’ view etc. Most of the commonly used form elements have been designed. Interesting feature is the inclusion of **Select2** framework that provides a customizable select box with support for searching, tagging, remote data sets, infinite scrolling, and many other highly used options.

There also features such as the **breadcrumbs** etc for making the interface as usable as possible.

The code itself is extensively documented with relevant comments everywhere.

Make sure all CSS and JS files are updated that are related to VPI. Otherwise, the layout will not function properly. Most important files are VPI.css, skins CSS files, and app.js.

Modifying the Javascript Options of VPI's app.js can be done using one of the following ways.

- Editing app.js
  Within the main Javascript file, modify the $.VPI.options.
- Defining VPIOptions
  Alternatively, you can define a global options variable named VPIOptions and initialize it before loading app.js.