

## Project Overview

This document outlines the specification of mobile application template.

**Disclaimer:** This is a shortened demo document intended for the University of Cincinnati's Capstone project presentation. The full, proprietary UI specification is not publicly available.

## Document Revisions

Revision	Description	Date
00	Initial document.	3/28/2024
01	Refactoring the PE template app & widgets collection.	6/19/2024
02	Making necessary adjustments according to meeting notes from 6/28/2024.	7/2/2024
03	Small bugfixes & inconsistencies with documentation found while programming them into the project. Added directional button widget (RT).	7/24/2024

---

## **Methodology - Templated App Architecture**

- The templated app architecture emphasizes maintaining a generic template app, along with a set of modular widgets which can be “dragged & dropped” in wherever needed.
- Functionality is logically split across multiple different libraries (submodules). This allows for individual modifications to the various libraries without needing to change the app itself. Moreover, apps simply dictate what version of the library is loaded, rather than requiring a completely specialized codebase for their own project.
- The template app itself is the frame/layout of the app, with placeholders or empty space where widgets may be populated. Throughout the course of this document, many of these will be labeled with colors or text, indicating that a widget may go there.

**Disclaimer: Images are included for illustration only.**

## Document Details - Arrows Color Key

- This document utilizes a large set of colored arrows & outline boxes to point out specific UI elements & attributes. Different colors have different distinctions:
  - **Green** – UI elements
  - **Purple** – Color attributes

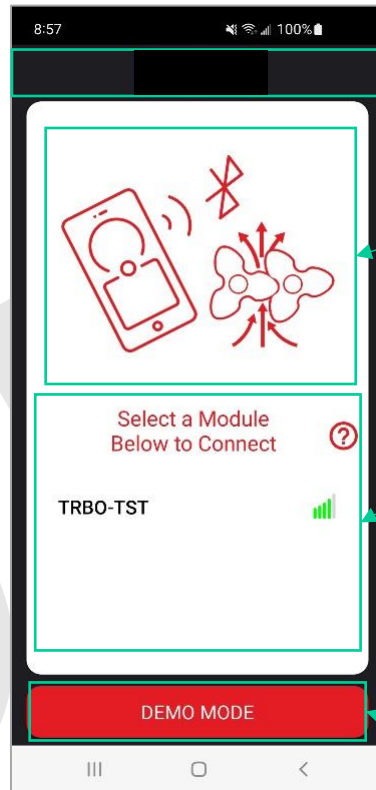
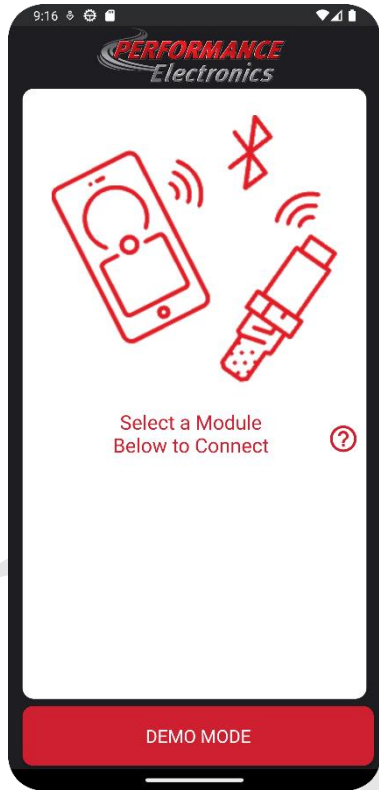
Template App

## Instructions & Scan Screen - Requirements

- The app will first launch on the instructions & scan screen. This screen is statically populated with the following components:
  - Header bar with company logo
  - Large central graphic
  - Connect to module label & help button
  - Scan fragment where scanned devices appear
  - Demo mode button for launching in demo mode
- Optionally, the following sections are customizable via widgets:
  - Label for important information (terms & conditions, user agreement, privacy statement, etc.)

## Instructions & Scan Screen - Illustrations

### PE Template App



### Example Apps

#### Header Bar

1. Company logo

#### Central Graphic

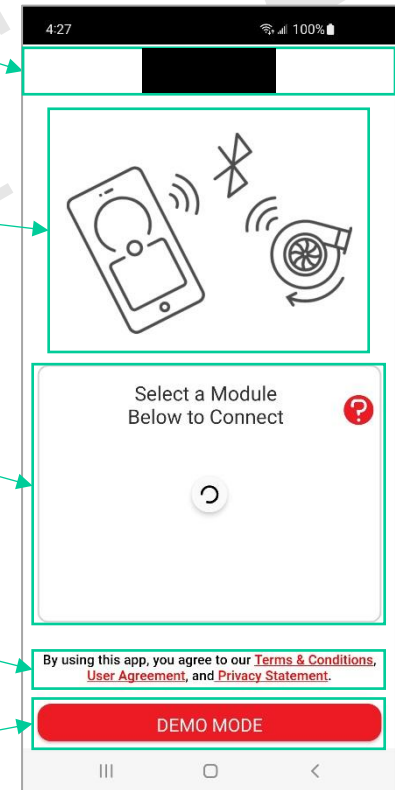
#### Scan Fragment

1. Available devices

#### Optional Info Label

1. Important info
2. Links
3. Announcements

#### Demo Mode Button



## Home Screen - Requirements

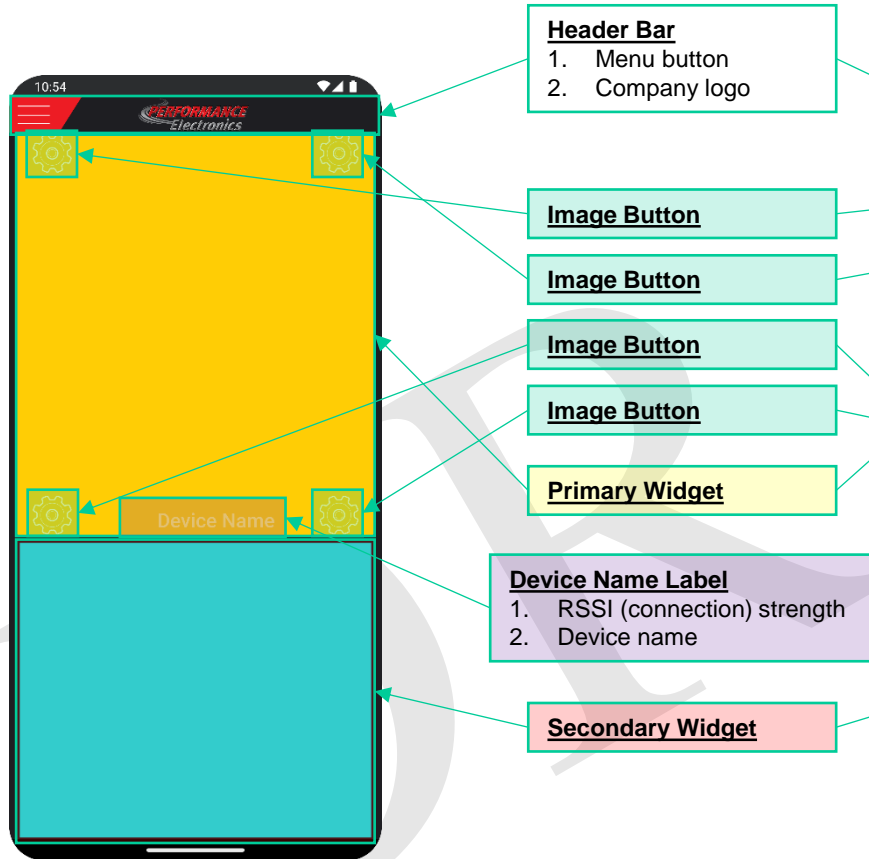
- Once connected to a module or in demo mode, the home screen of the application is pulled up. This screen is statically populated with the following components:
  - Header bar with menu button & company logo
  - Device name & RSSI (connection strength) label
- Optionally, the following sections are customizable via widgets:
  - Image button widget
    - Image button widget that can be used for status updates, alerts, responding to click events, etc. One of these can be placed in each of the corners of the main screen. See the “widgets” section of this document for details.
  - Primary widget
    - The bigger widget on the top half of the screen, with the images mentioned above overlaying the corners. This can be any widget the customer desires.
  - Secondary widget
    - The smaller widget on the bottom half of the screen. This can be any widget the customer desires.



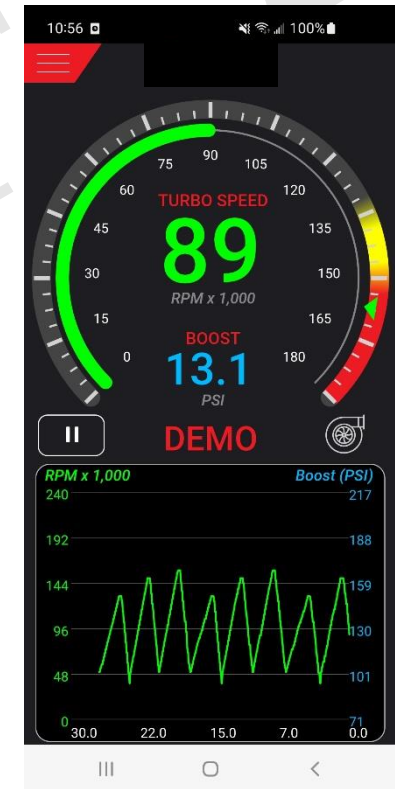
## Home Screen - Illustrations

*\*\* Important Note: Tinted fill added to help distinguish UI elements. \*\**

### PE Template App



### Example Apps



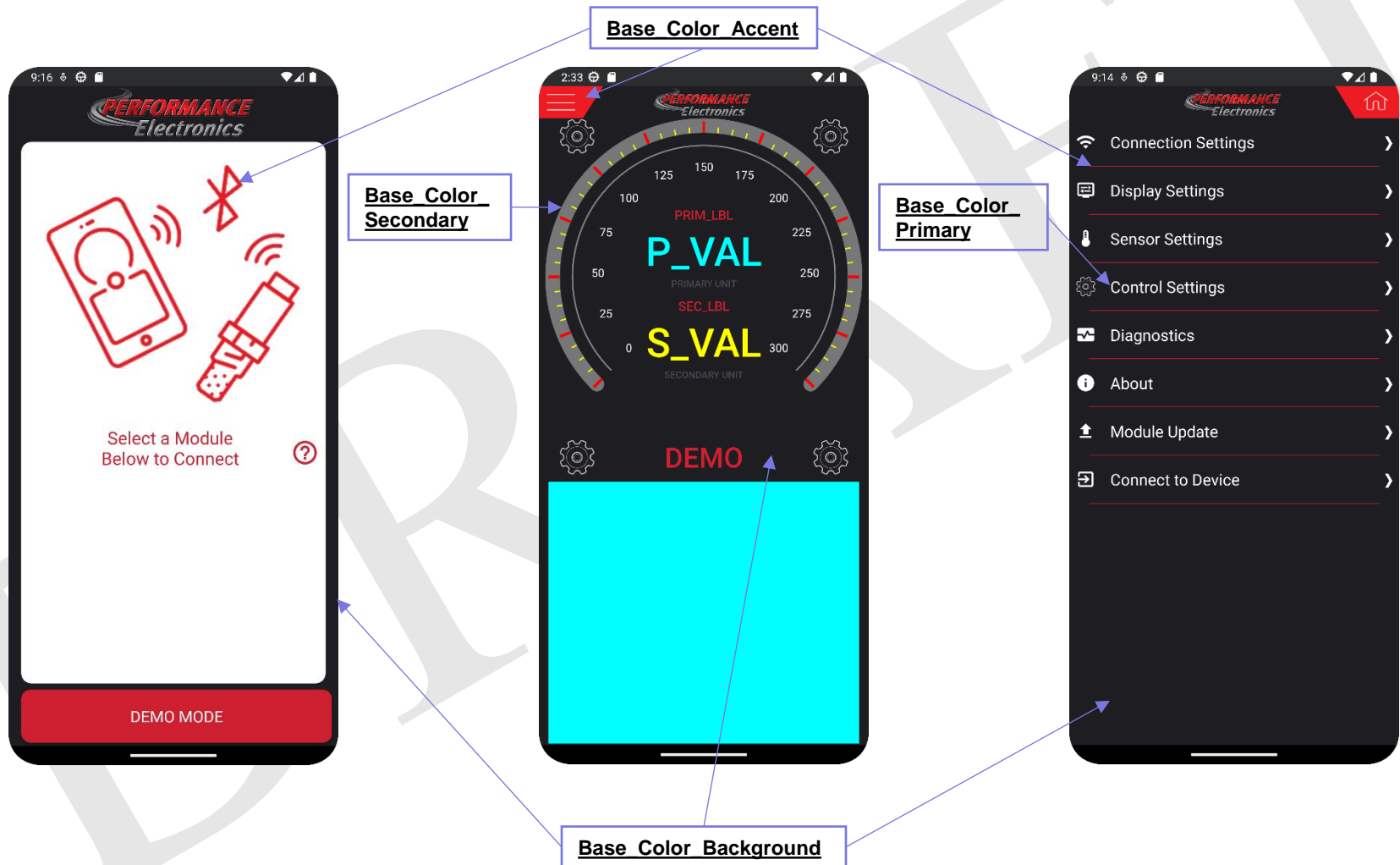
Theming

### Theming - Overview

- The templated app architecture introduces the ability to rather quickly apply themes to the entire app. This allows for configurable, project-based UI themes with only a few minor changes necessary. The base color values required for any given theme are as follows:
  - `base_color_primary`
    - Often used for foreground text & UI elements.
  - `base_color_secondary`
    - Often used for shaded regions, section headers, etc.
  - `base_color_accent`
    - Often used for interactive UI elements or borders.
  - `base_color_background`
    - Used as the default background for every screen and widget.
- Optionally, the following colors can be set to override defaults & extend the reach of the theme:
  - **IN PROGRESS**

*Images with some of the uses of these theme attributes can be seen on the following pages.*

## Theming - Illustrations



## Widget Theme Inheritance

### Widget Theme Inheritance - Overview

- The templated app architecture widgets are all customizable. To do so, a series of color attributes must be configured for each widget. To ensure that all the necessary attributes are set, these inherit from the base colors of the application by default.
- For example, if an attribute which inherits from the *base\_color\_primary* isn't set, then the *base\_color\_primary* value will be used instead.
- To concisely document which attributes inherit from which base colors, the following color index table has been created:

Color	Index
Transparent	0
base_color_primary	1
base_color_secondary	2
base_color_accent	3
base_color_background	4

- The “parent color” for each attribute of a given widget will be listed directly after the attribute name, as exemplified below:
  - Widget background color - 4
  - Widget border color - 3

Widgets

## Widgets - Overview

- The following widgets are available to be used anywhere throughout the app. Many of these are already implemented as part of the base template:
  - Confirmed Action button
  - Data table
  - Dialog
    - Positive/negative/neutral
    - Text input
  - Directional button
  - Display table
  - Dropdown
  - Edit device name
  - Gauge
  - Graph
  - Header bar
  - Image button
  - Radial button
  - Settings cell view
    - Content only
    - Title content
  - Slider
    - Title content image
    - Title dropdown
  - Labeled Range Slider
  - Labeled Single Slider
  - Range Slider
  - Single Slider
  - Snack bar
  - Value display label

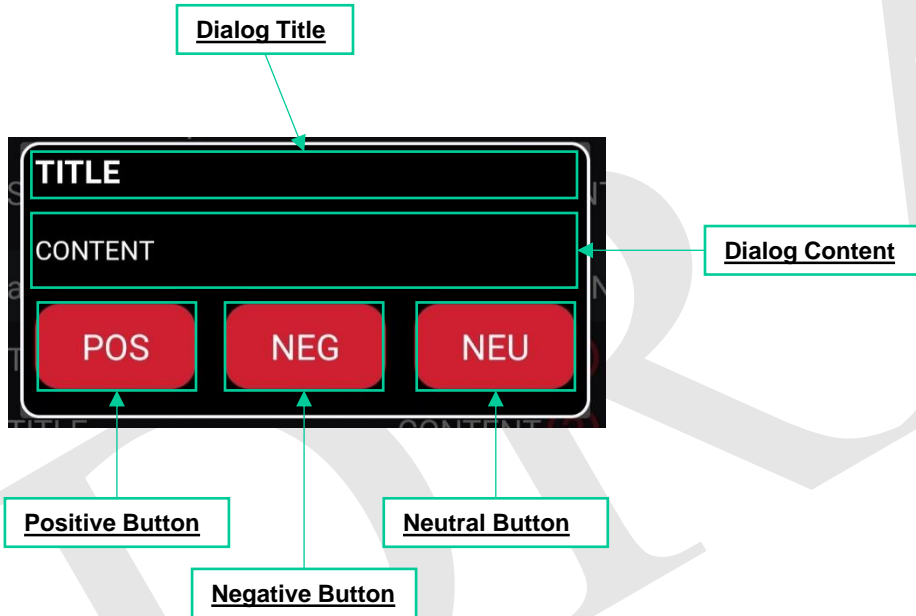


## Dialog Widget (Positive/Negative/Neutral) - Requirements

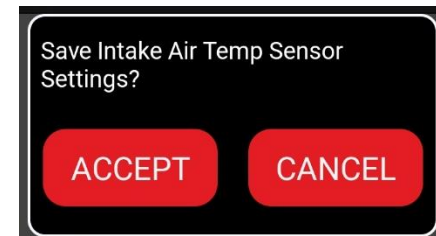
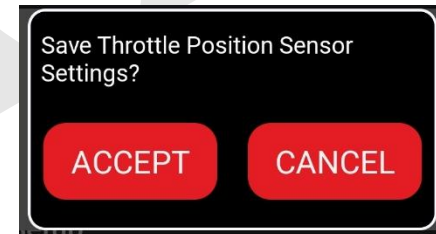
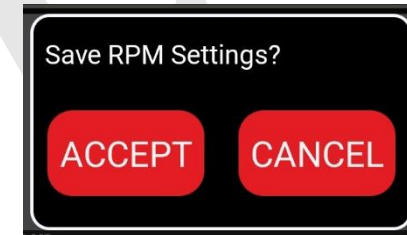
- This widget contains the following components:
  - Dialog content
  - Dialog title
  - Negative button
  - Neutral button
  - Positive button
- Widget Information:
  - **IN PROGRESS**
- Widget Attributes:
  - Dialog content text color - 1
  - Dialog title text color - 1
  - Widget background color - 4
  - Widget border color - 3

## Dialog Widget (Positive/Negative/Neutral) - Illustrations

### PE Template App

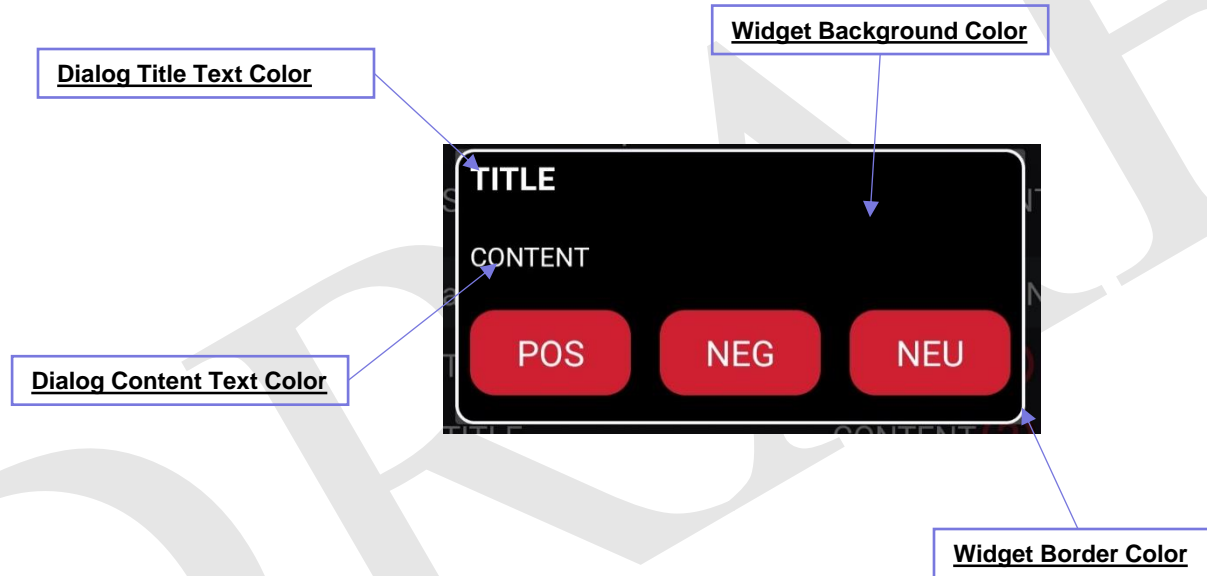


### Example Apps



*\*\* Elements of the dialog are shown/hidden depending on the implementation. In these examples, both the dialog title & negative button have been hidden. \*\**

## Dialog Widget (Positive/Negative/Neutral) - Illustrations Cont.



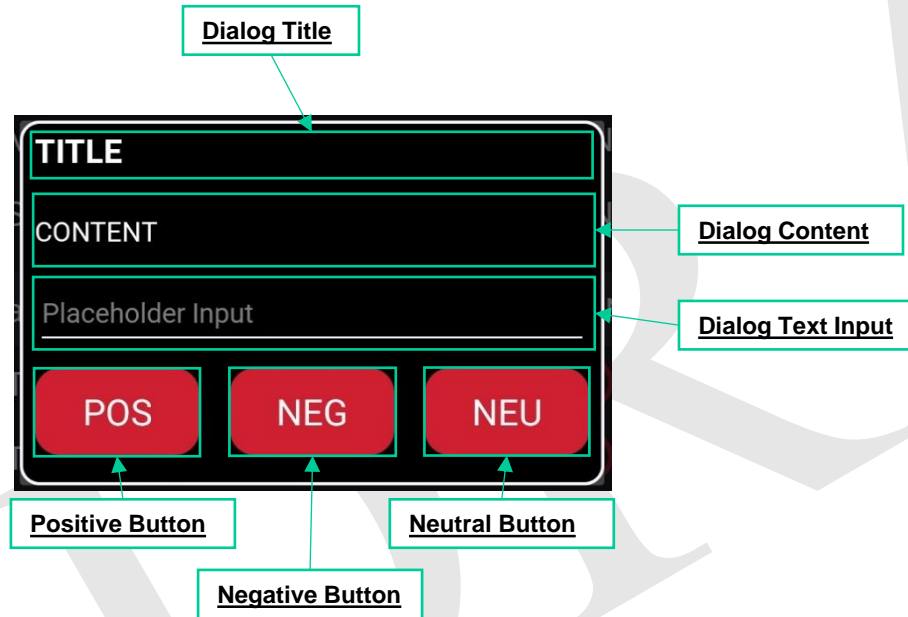
---

## **Dialog Widget (Text Input) - Requirements**

- This widget contains the following components:
  - Dialog content
  - Dialog text input
  - Dialog title
  - Negative button
  - Neutral button
  - Positive button
- Widget Information:
  - **IN PROGRESS**
- Widget Attributes:
  - Dialog content text color - 1
  - Dialog text input text color - 1
  - Dialog title text color - 1
  - Widget background color - 4
  - Widget border color - 3

## Dialog Widget (Text Input) - Illustrations

### PE Template App



## Dialog Widget (Text Input) - Illustrations Cont.

