**Properties Description**

1. Native Drawer Component Properties
   1. openDrawer: () => void;
   2. closeDrawer :() => void;
   3. tapToClose:boolean;
   4. openDrawerOffset : 0.2;
   5. panCloseMask : 0.2;
   6. closedDrawerOffset : 0;
   7. onClose? : () => void;
2. Html Drawer Component Properties
   1. closeButtonContainerClassName: string;
   2. onOverlayClick: () => void,
   3. slideDuration: number;
   4. width: string,
3. Native NativeDecimalTextBox Component
   1. placeholder?: string;
   2. placeholderTextColor?: string;
   3. autoCapitalize?: "none" | "sentences" | "words" | "characters";
   4. autoCorrect?: boolean;
   5. autoFocus?: boolean;
   6. keyboardType?: "default" | "email-address" | "numeric" | "phone-pad" | "ascii-capable" | "numbers-and-punctuation" | "url" | "number-pad" | "name-phone-pad" | "decimal-pad" | "twitter" | "web-search";
   7. allowFontScaling?: boolean;
   8. maxLength?: number;
   9. multiline?: boolean;
   10. secureTextEntry?: boolean;
   11. style: {};
   12. inputAccessoryViewID?: string;
   13. textContentType?: string;
   14. returnKeyLabel?: string;
   15. editable?: boolean;
   16. defaultValue?: string;
   17. onChangeText : void
4. Html DecimalTextBox
   1. emptyIconClassName: string;
   2. nonEmptyIconClassName: string;
   3. containerClassName: string;
   4. placeholder: string;
   5. onTouched?: () => void;
   6. fractionalDigits: number;
   7. groupDigits: boolean;
5. NativeTextBox Props
   1. placeholder?: string;
   2. placeholderTextColor?: string;
   3. autoCapitalize?: "none" | "sentences" | "words" | "characters";
   4. autoCorrect?: boolean;
   5. autoFocus?: boolean;
   6. keyboardType?: "default" | "email-address" | "numeric" | "phone-pad" | "ascii-capable" | "numbers-and-punctuation" | "url" | "number-pad" | "name-phone-pad" | "decimal-pad" | "twitter" | "web-search";
   7. allowFontScaling?: boolean;
   8. maxLength?: number;
   9. multiline?: boolean;
   10. secureTextEntry?: boolean;
   11. style: {};
   12. inputAccessoryViewID?: string;
   13. textContentType?: string;
   14. returnKeyLabel?: string;
   15. editable?: boolean;
   16. defaultValue?: string;
6. HtmlText Box Props
   1. emptyIconClassName: string;
   2. nonEmptyIconClassName: string;
   3. containerClassName: string
   4. placeholder: string;
   5. onTouched?: () => void;
7. NativeText Props
   1. keyboardAppearance?: string;
   2. spellCheck?: boolean;
   3. textContentType?: string;
   4. scrollEnabled?: boolean;
   5. enablesReturnKeyAutomatically?: boolean;
   6. clearTextOnFocus?: boolean;
   7. clearButtonMode?: string;
   8. numberOfLines?: string;
   9. underlineColorAndroid?: string;
   10. lineBreakMode?: "head" | "middle" | "tail" | "clip";
   11. ellipsizeMode?: "head" | "middle" | "tail" | "clip";
   12. allowFontScaling?: boolean;
   13. disabled?: boolean;
   14. value?: string;
   15. onPress?: (event: ReactNative.GestureResponderEvent) => void;
   16. onLongPress?: (event: ReactNative.GestureResponderEvent) => void;
8. HtmlText Props  
   1. text : string
   2. tagName : string
9. NativeToast (consider this as Notification)   
   1. position: string,
   2. textStyle: Object,
   3. positionValue: number,
   4. fadeInDuration: null,
   5. fadeOutDuration: number,
   6. opacity: number
10. Html Notification Props:
    1. contents : Object
       1. type : string;
       2. text : string;
       3. title : string
11. NativePopOver
    1. children?: any;
    2. isVisible: boolean;
    3. arrowColor?: string;
    4. arrowWidth?: number;
    5. arrowHeight?: number;
    6. placement: string;
    7. pointerEvents?: string;
    8. offset? : {};
    9. registerPopover: PropTypes.func;
    10. unregisterPopover: PropTypes.func;
12. HtmlPopOver  
    1. iconClassName: string,
    2. itemClassName: string,
    3. itemContainerClassName: string,
    4. itemKey: any,
    5. itemsSource: Array<any>,
    6. onClick: (e: any) => void,
13. NativeValidationErrorMessage  
    1. bindingMode: string;
    2. keyboardAppearance?: string;
    3. spellCheck?: boolean;
    4. textContentType?: string;
    5. scrollEnabled?: boolean;
    6. enablesReturnKeyAutomatically?: boolean;
    7. clearTextOnFocus?: boolean;
    8. clearButtonMode?: string;
    9. numberOfLines?: number;
    10. underlineColorAndroid?: string;
    11. lineBreakMode?: "head" | "middle" | "tail" | "clip";
    12. ellipsizeMode?: "head" | "middle" | "tail" | "clip";
    13. allowFontScaling?: boolean;
    14. disabled?: boolean;
    15. value?: string;

1. ValidationErrorMessage HTML
   1. TagName
   2. Text
2. NativeReadOnlyFieldProps
   1. bindingMode: string;
   2. keyboardAppearance?: string;
   3. spellCheck?: boolean;
   4. textContentType?: string;
   5. scrollEnabled?: boolean;
   6. enablesReturnKeyAutomatically?: boolean;
   7. clearTextOnFocus?: boolean;
   8. clearButtonMode?: string;
   9. numberOfLines?: number;
   10. underlineColorAndroid?: string;
   11. lineBreakMode?: "head" | "middle" | "tail" | "clip";
   12. ellipsizeMode?: "head" | "middle" | "tail" | "clip";
   13. allowFontScaling?: boolean;
   14. disabled?: boolean;
   15. value?: string;
3. ReadOnlyFieldProps HTML
   1. bindingMode: string,
   2. Value
4. DropDownListProps HTML
   1. value:string
   2. className:string
   3. placeholder:string
   4. itemValue:string
   5. openSource:Array()
5. NativeSelect
   1. enabled?: boolean;
   2. mode?: string;
   3. prompt?: string;
   4. itemStyle?: {}
   5. Items ? : []
   6. selectedValue?: string;
6. NativeControl
   1. testId?: string;
   2. accessibilityHint?: string;
   3. accessibilityLabel?: string;
   4. nativeID?: string;
   5. accessibilityRole?: string;
   6. Key?:string;
7. NativeControl
   1. style: {};
   2. onLayout?: (event: ReactNative.LayoutChangeEvent) => void;