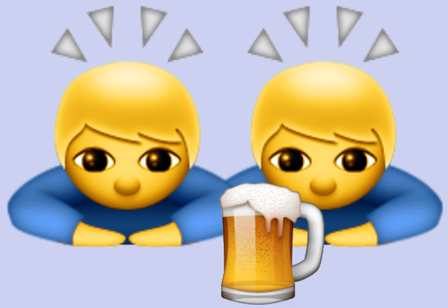
React Native

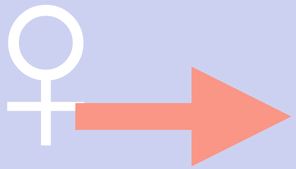
Basic intro and Stuff


Hollr

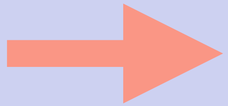
The Team

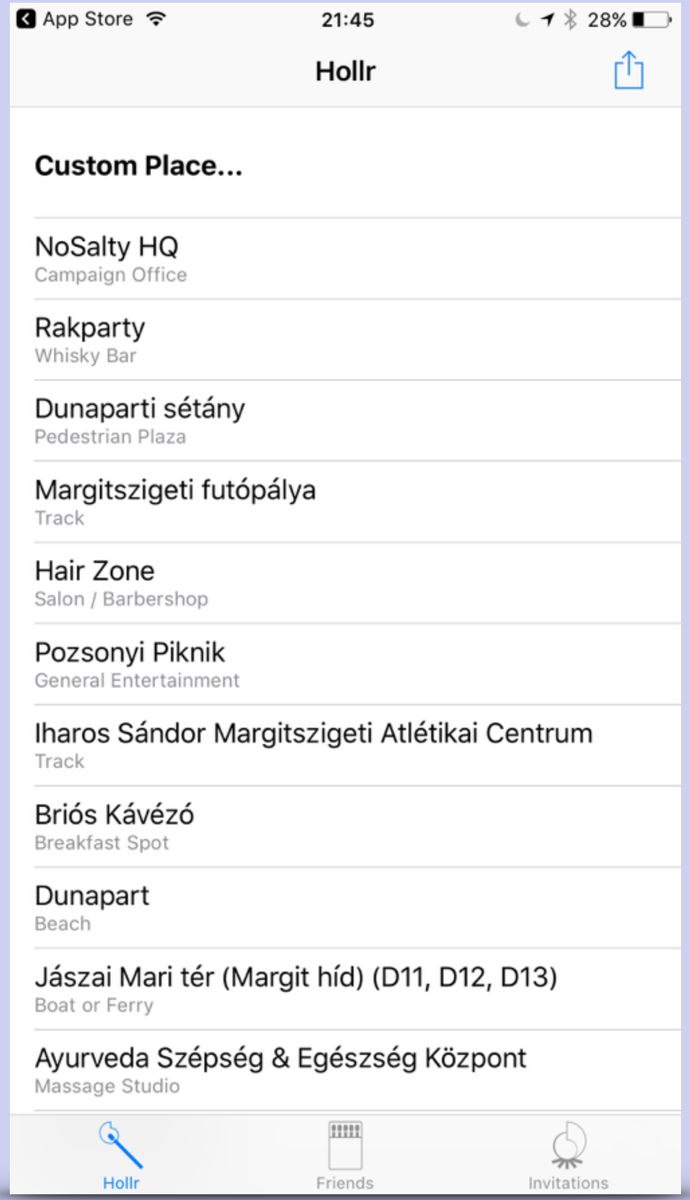
@ Daniel Falus
@ Balazs Bajorics
@ Maja Szakadat
@ Andras Balogh
@ Peter 'rozmy' Polgar



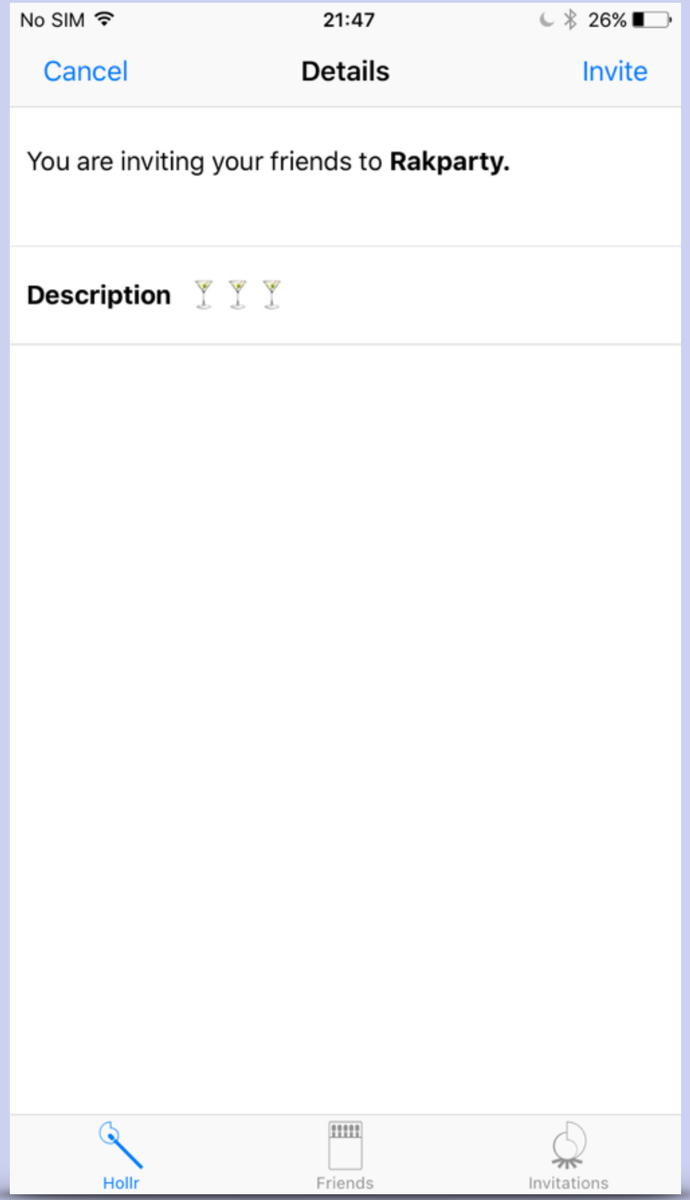



Hollr

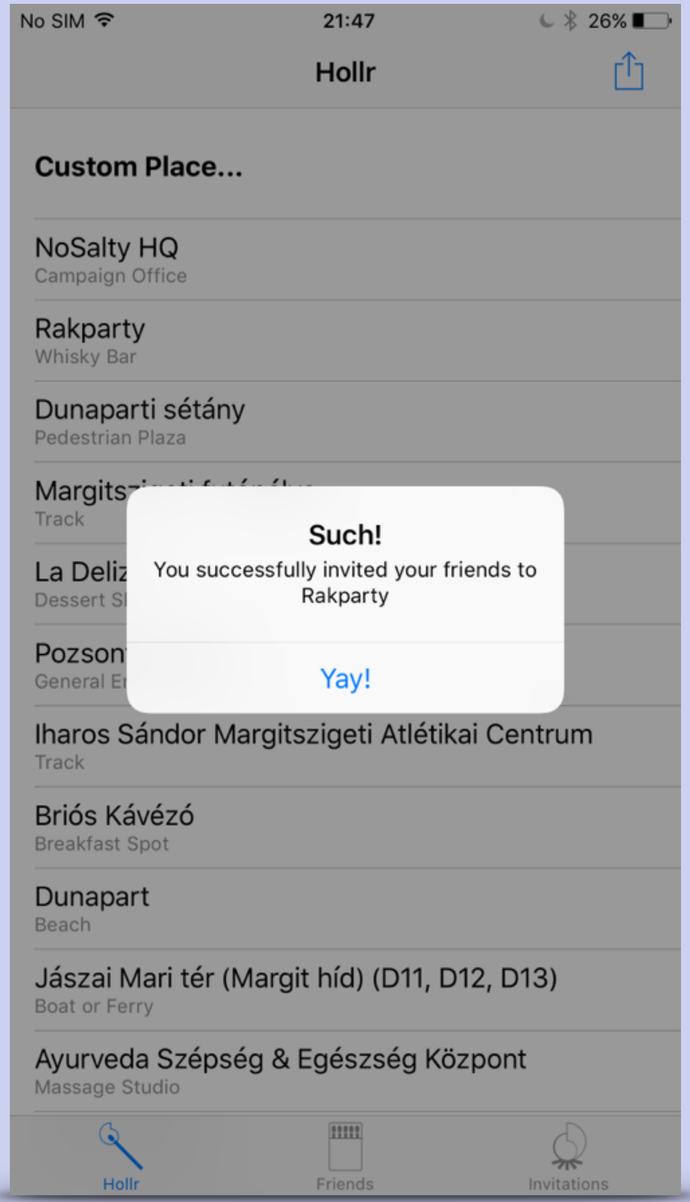



App Store
21:45
28%
Hollr
Custom Place...
NoSalty HQ
Campaign Office
Rakparty
Whisky Bar
Dunaparti sétány
Pedestrian Plaza
Margitszigeti futópálya
Track
Hair Zone
Salon / Barbershop
Pozsonyi Piknik
General Entertainment
Iharos Sándor Margitszigeti Atlétikai Centrum
Track
Briós Kávézó
Breakfast Spot
Dunapart
Beach
Jászai Mari tér (Margit híd) (D11, D12, D13)
Boat or Ferry
Ayurveda Szépség & Egészség Központ
Massage Studio
Hollr
Friends
Invitations

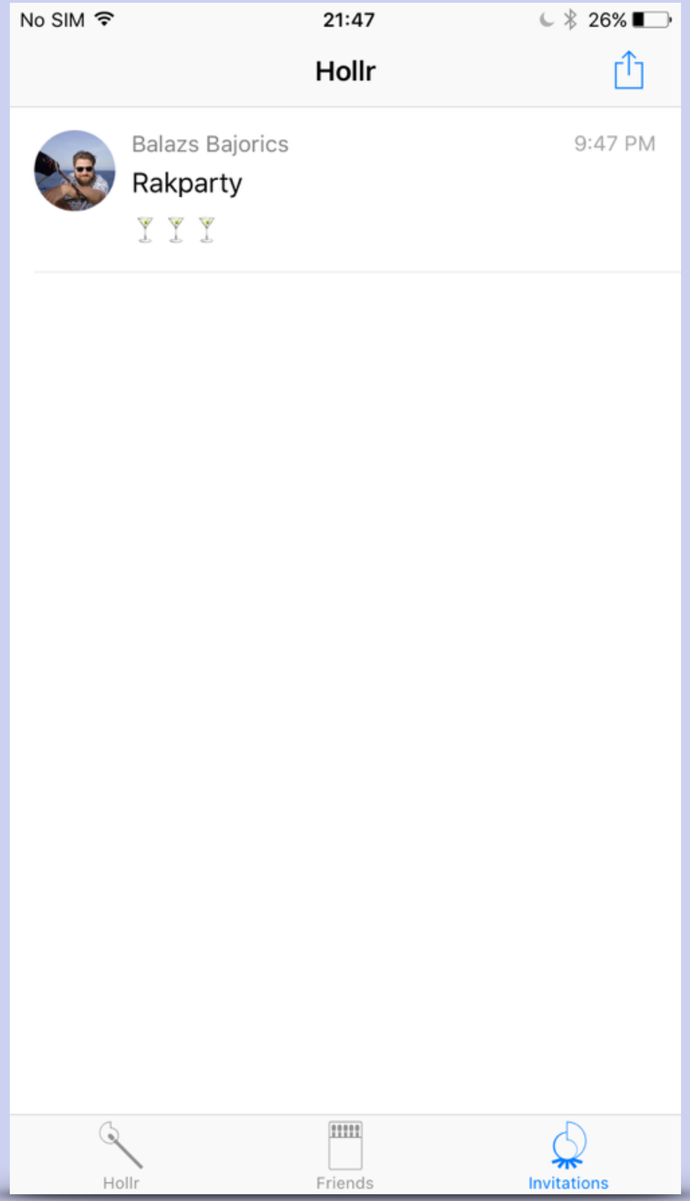

Hollr


No SIM
21:47
26%
Cancel
Details
Invite
You are inviting your friends to **Rakparty.**
Description
Hollr
Friends
Invitations


Hollr


No SIM
21:47
26%
Hollr
Custom Place...
NoSalty HQ
Campaign Office
Rakparty
Whisky Bar
Dunaparti sétány
Pedestrian Plaza
Track
Such!
You successfully invited your friends to Rakparty
Yay!
Track
Briós Kávézó
Breakfast Spot
Dunapart
Beach
Jászai Mari tér (Margit híd) (D11, D12, D13)
Boat or Ferry
Ayurveda Szépség & Egészség Központ
Massage Studio
Hollr
Friends
Invitations


Hollr


No SIM
21:47
26%
Hollr
Balazs Bajorics
9:47 PM
Rakparty
Hollr
Friends
Invitations


Hollr

In The App Store!

https://hollr.social

App Store Experience

@ Great!

@ <24h Review

@ ~300 downloads ;)

@ Side project - limited time available

@ Some prior experience with React for the web

@ Hollr is very simple

React Native

React Native?

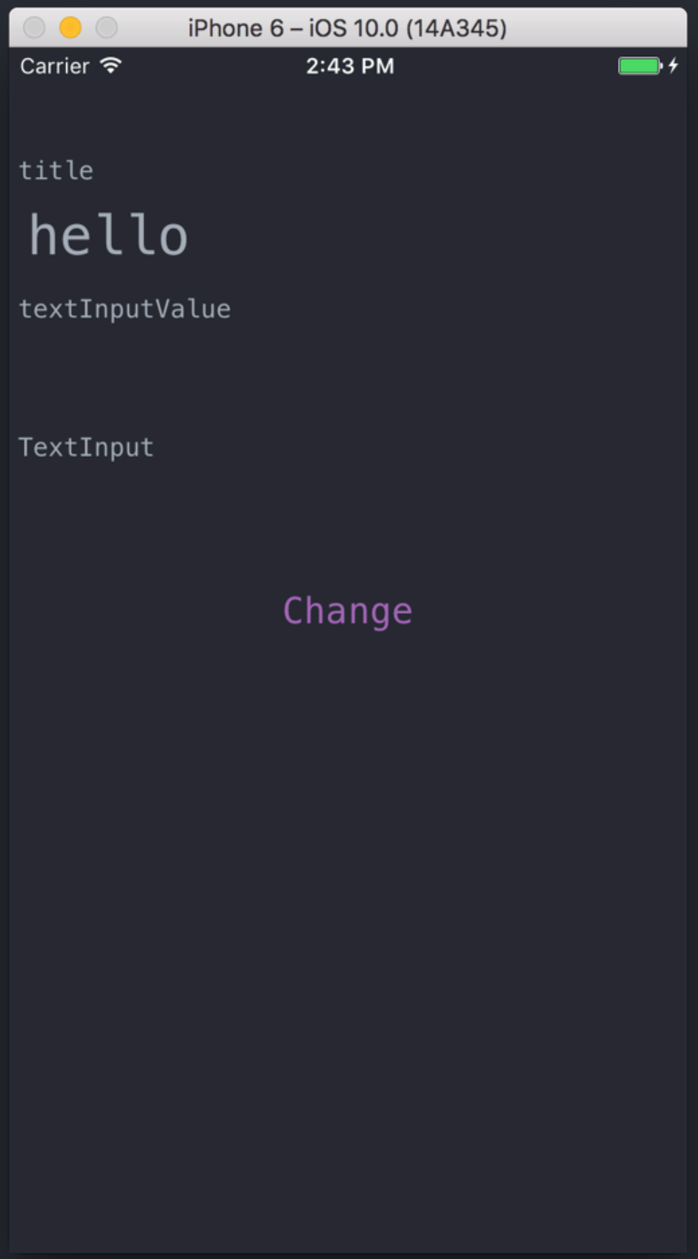
```
const Hello = ({style}) => {
 return (
  <View style={style}>
    <Text>hello</Text>
  </View>
 )
}
```

React Native?

```
interface HelloProps {
 style: ViewStyle,
}

const Hello = ({style}: HelloProps) => {
 return (
  <View style={style}>
    <Text>hello</Text>
  </View>
 )
}
```

Stylesheets

```
const styles = StyleSheet.create<Styles>({
    container: {
        flex: 1,
        justifyContent: 'flex-start',
        alignItems: 'flex-start',
        backgroundColor: '#282c34',
        paddingTop: 60,
    },
    mono: {
        fontFamily: fontFamily,
        color: '#A8AFBB',
        fontSize: 14,
    },
})
```



```
const Hello = ({style}: HelloProps) => {
```

ES6

@Arrow Function

```
const Hello = ({style}: HelloProps) => {}

const Hello =  function({style}: HelloProps) {}
```

ES6

@Arrow Function

```
const Hello = ({style}: HelloProps) => {}
```

```
const Hello = (function({style}: HelloProps) {}).bind(this)
```

ES6

@Arrow Function

```
const Hello = ({style}: HelloProps) => {}

const Hello = (function({style}: HelloProps) {}).bind(this)

const Curry = (a: string) => (b: string) => `Hello ${a} ${b}`
```

ES6

@Object Destructuring Syntax

```
const stuff = {
  a: 5,
  b: 15,
  c: 'cica',
}

const a = stuff.a
const c = stuff.c

const { a, c } = stuff
```

ES6

@Object Destructuring Syntax

```
// w/ destructuring
const Hello = ({style}: HelloProps) => {
  ...
}

// without destructuring
const Hello = (props: HelloProps) => {
  const style = props.style
  ...
}
```

React Native?

```
interface HelloProps {
 style: ViewStyle,
}

const Hello = ({style}: HelloProps) => {
 return (
  <View style={style}>
    <Text>hello</Text>
  </View>
 )
}
```

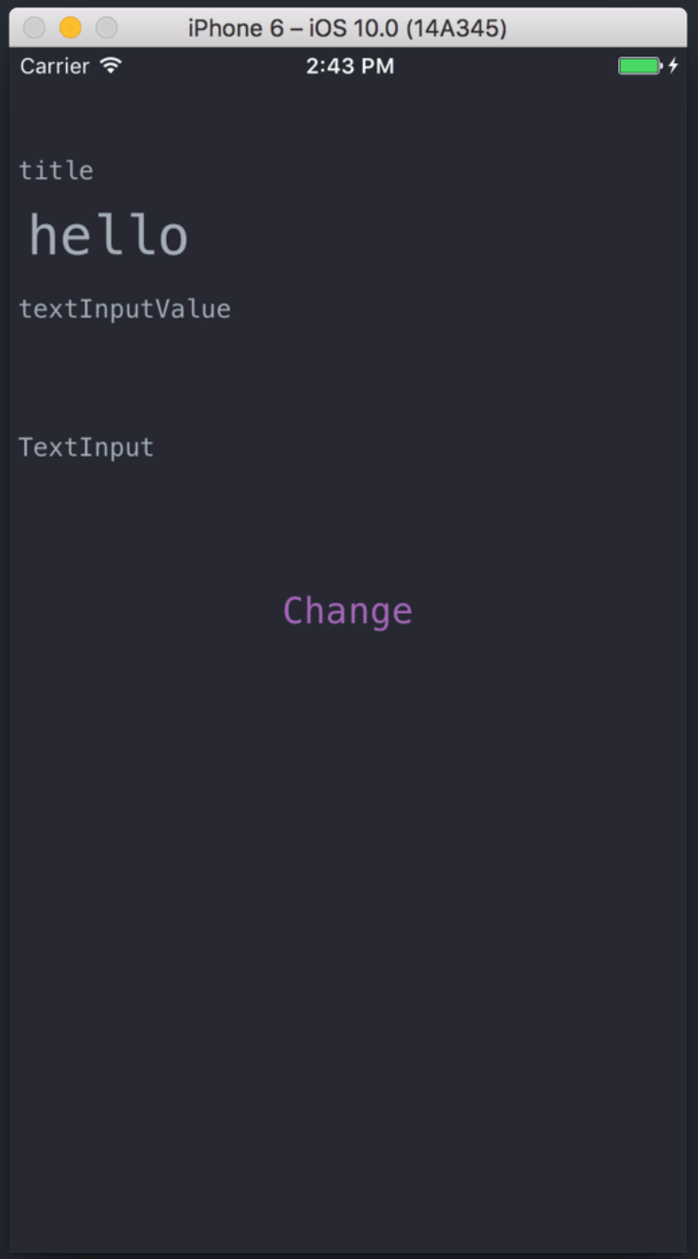
JSX

```
<View style={{ backgroundColor: 'black' }} />
```

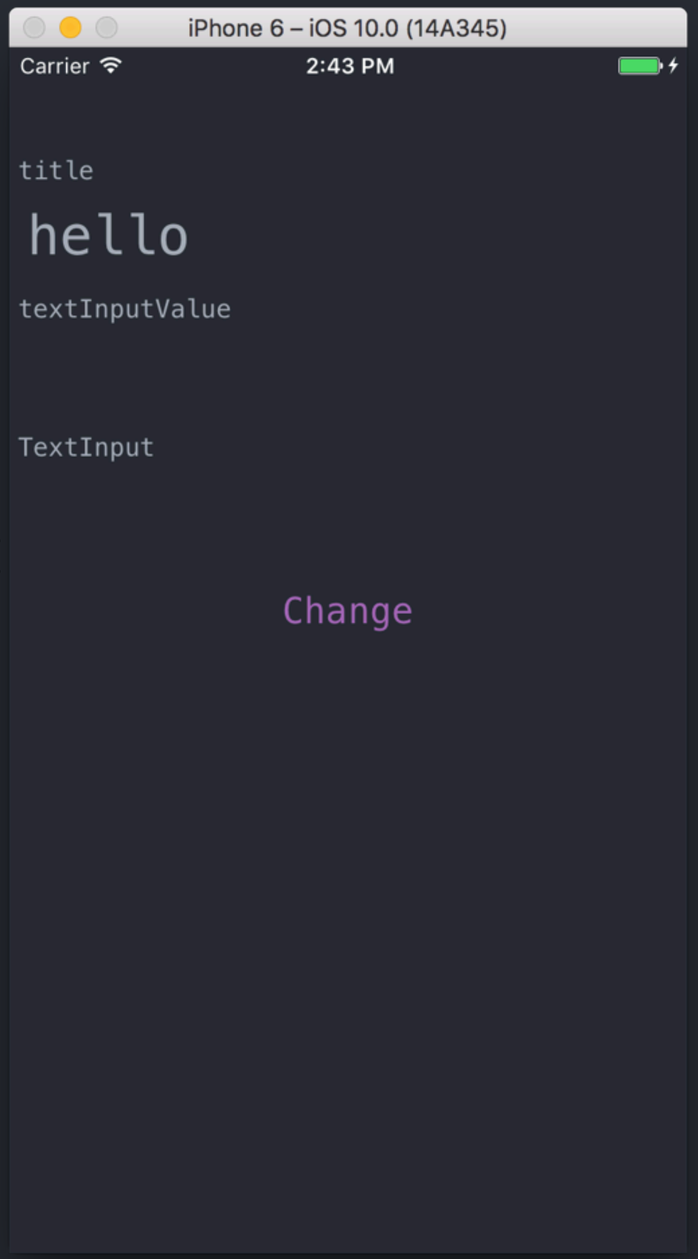
⬇ *JSX Transform*

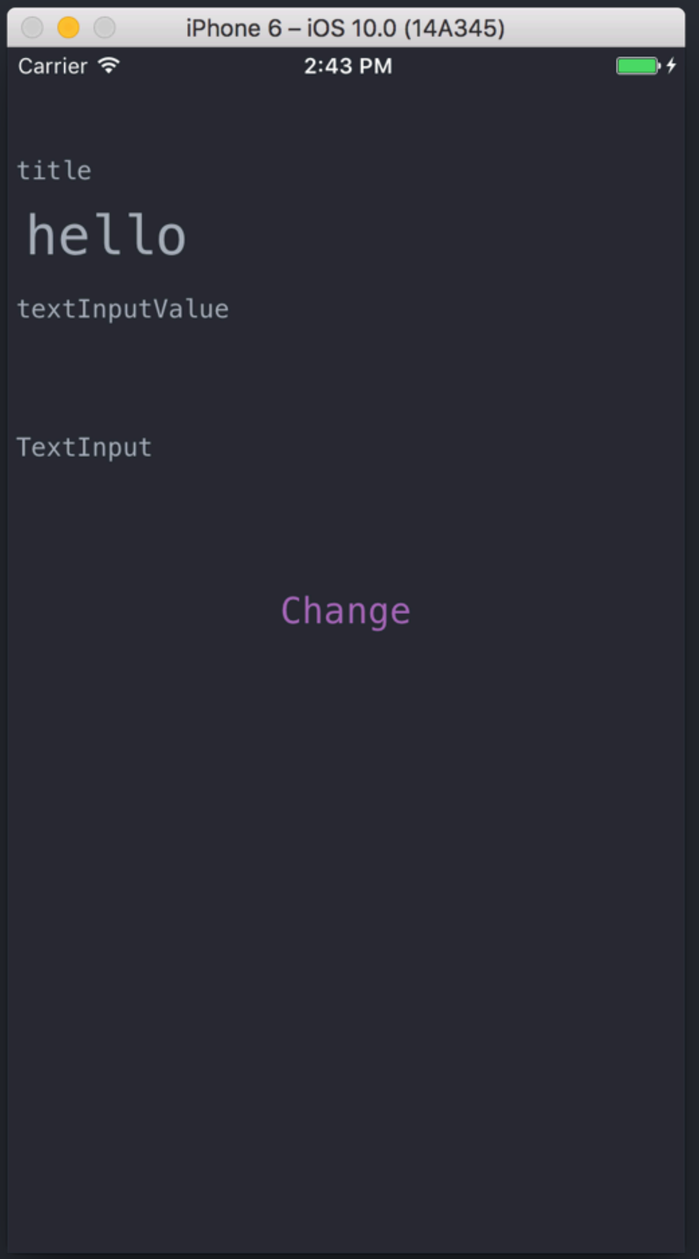
```
React.createElement(
  View,
  {style: { backgroundColor: 'black' } }
)
```

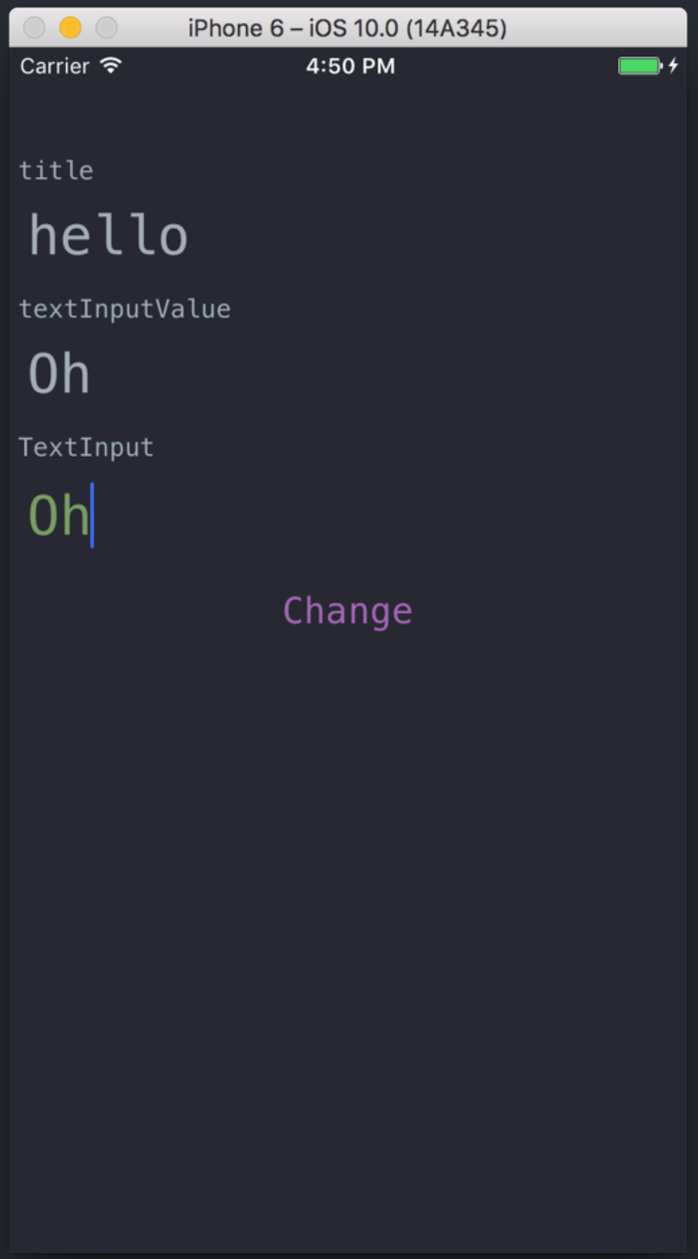
```tsx
interface MainScreenProps {
  title: string
  editedTextValue: string
  onTextInput: (text: string) => void
  onSetTitle: (text: string) => void
}

const MainScreen = ({
  title, editedTextValue, onTextInput, onSetTitle,
}: MainScreenProps) => {
  return (
    <View style={{ flex: 1 }}>
      <StatusBar
        backgroundColor='#282c34'
        barStyle='light-content'
        />
      <View style={styles.container}>
        <LabelAndTitle label='title' value={title} />
        <LabelAndTitle label='textInputValue' value={editedTextValue} />
        <TextInputField textInput={editedTextValue} onTextInput={onTextInput} />
        <ChangeButton onPress={() => onSetTitle(editedTextValue)} />
      </View>
    </View>
  )
}
```

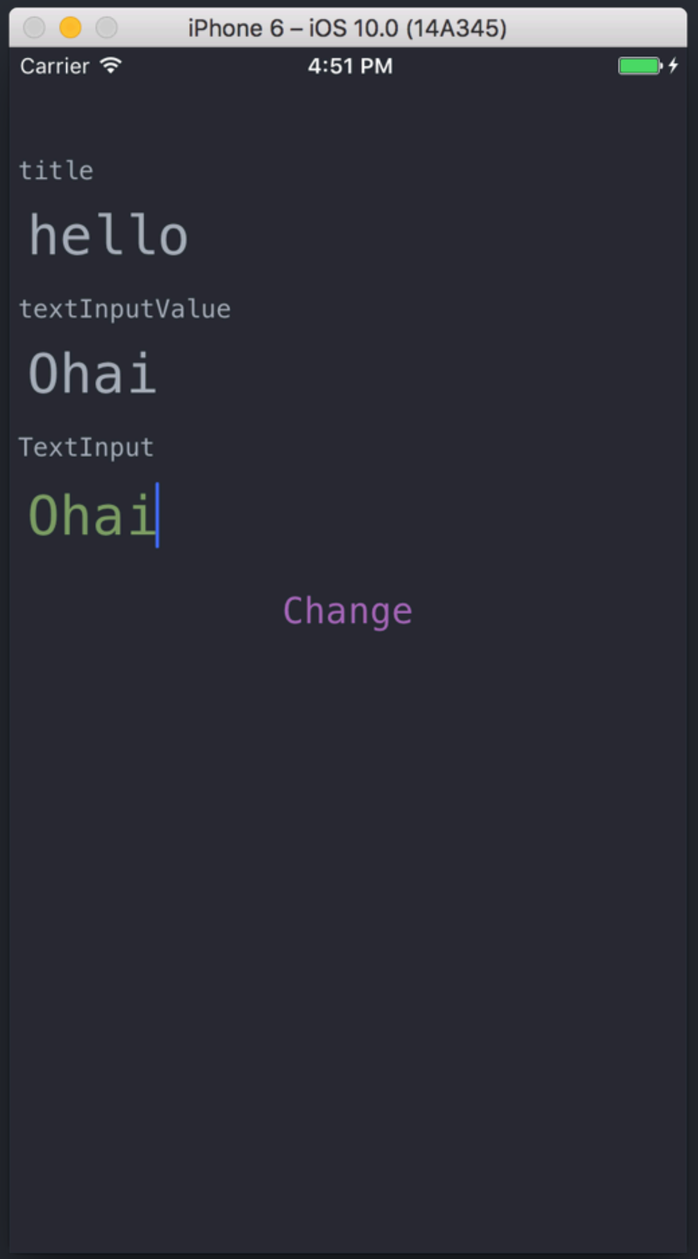


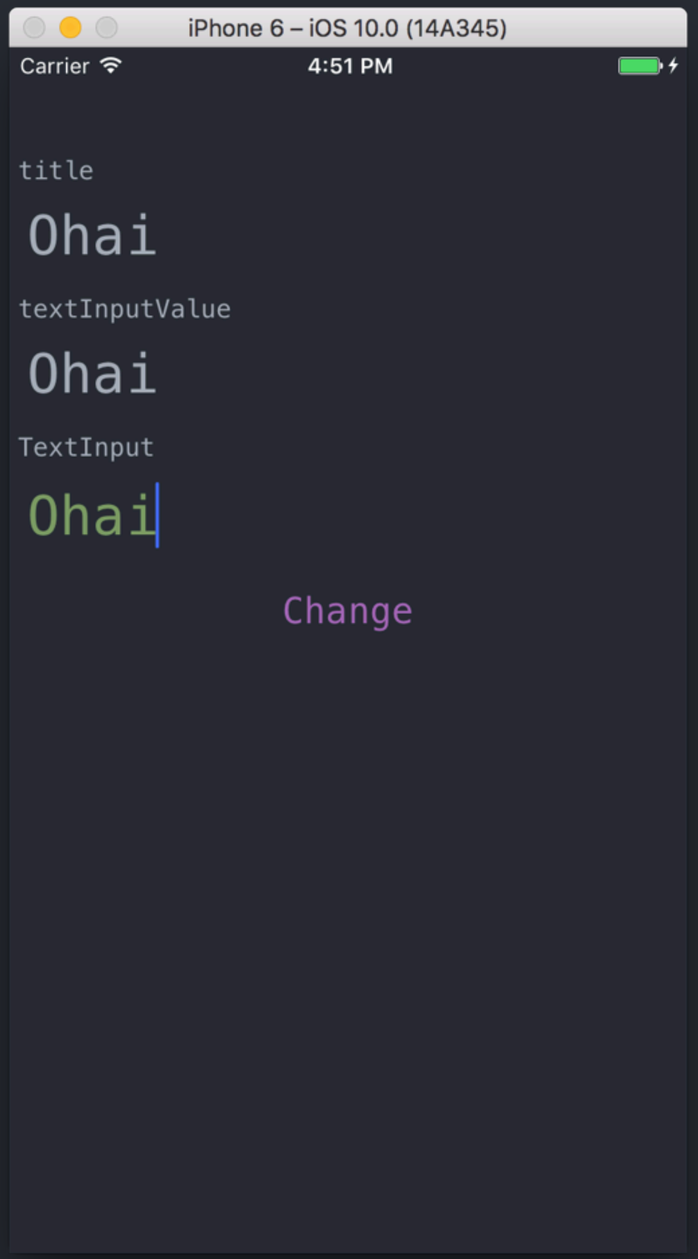
```tsx
interface TextInputFieldProps {
  textInput: string
  onTextInput: (text: string) => void
}

const TextInputField = ({textInput, onTextInput}: TextInputFieldProps) => {
  return (
    <View style={{ alignSelf: 'stretch' }}>
      <Text style={[styles.mono, styles.label]}>
        TextInput
      </Text>
      <TextInput
        style={[styles.mono, styles.title, styles.textInput]}
        onChangeText={onTextInput}
        value={textInput}
        />
    </View>
  )
}
```



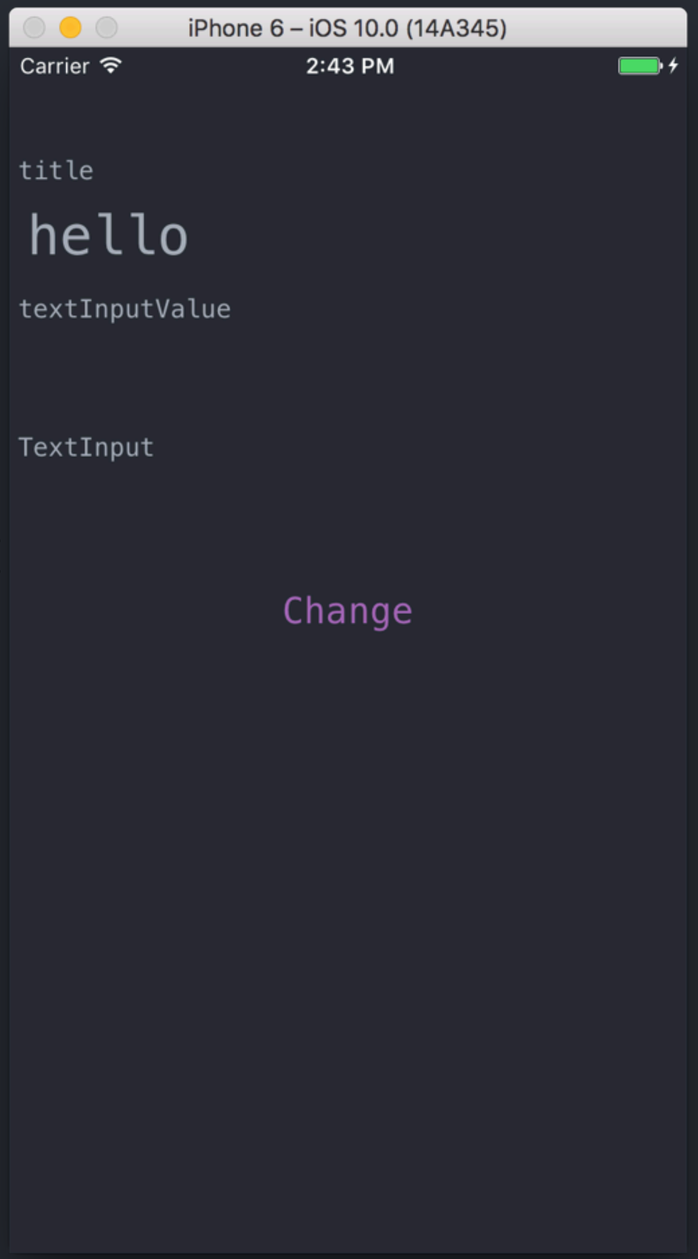

iPhone 6 – iOS 10.0 (14A345)
Carrier
2:43 PM
title
hello
textInputValue
TextInput
Change


iPhone 6 – iOS 10.0 (14A345)
Carrier
4:50 PM
title
hello
textInputValue
Oh
TextInput
Oh
Change

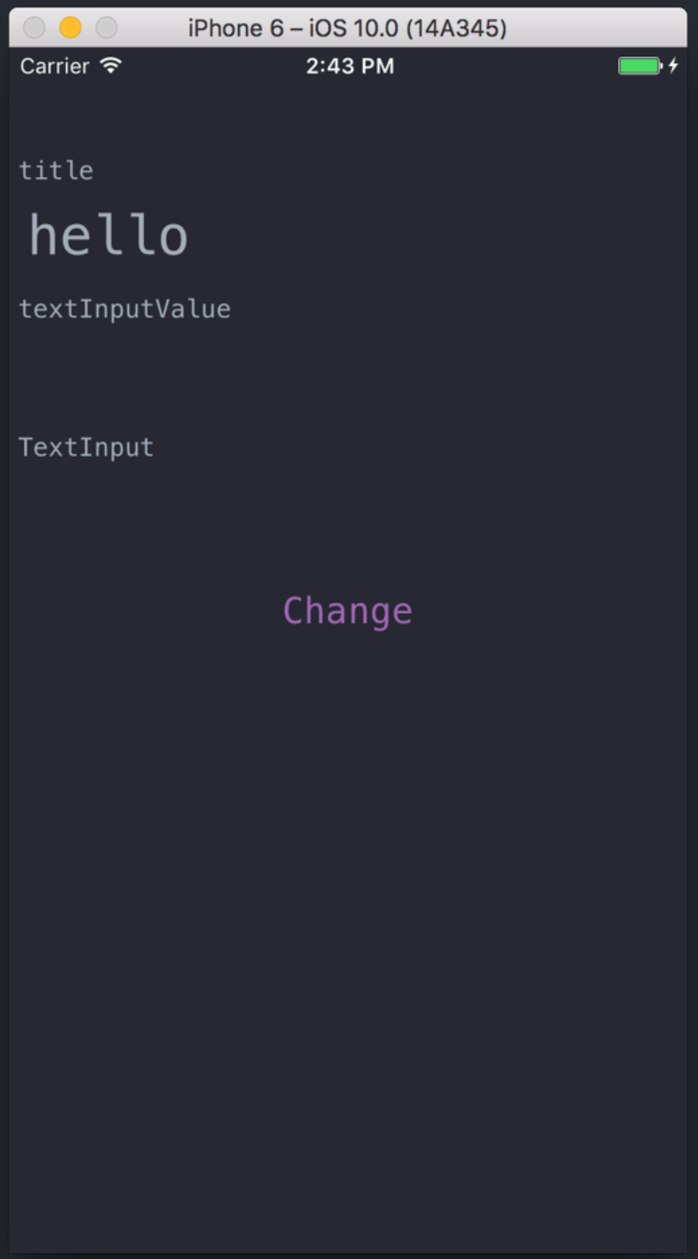

iPhone 6 – iOS 10.0 (14A345)
Carrier
4:51 PM
title
hello
textInputValue
Ohai
TextInput
Ohai
Change

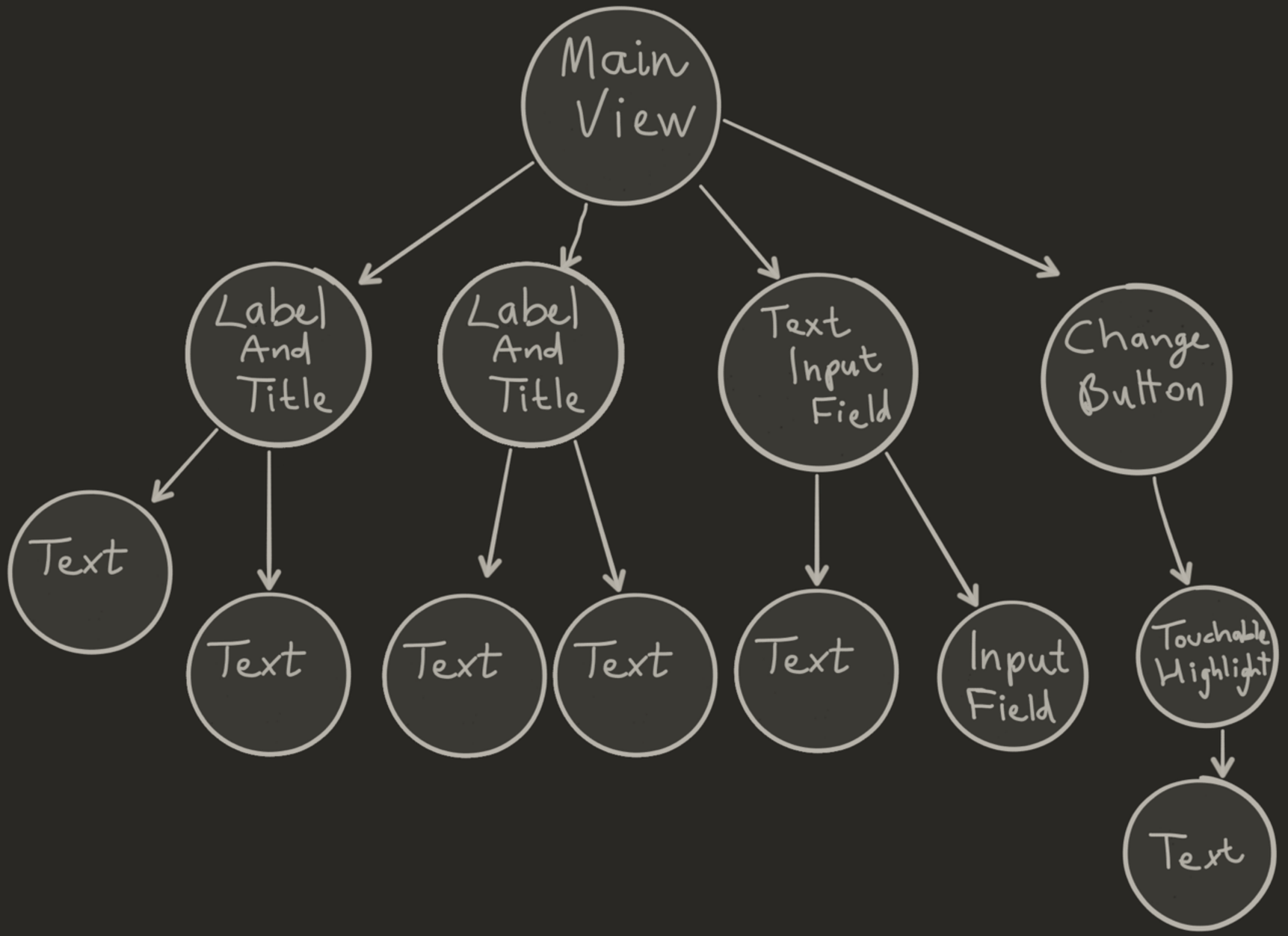

iPhone 6 – iOS 10.0 (14A345)
Carrier
4:51 PM
title
Ohai
textInputValue
Ohai
TextInput
Ohai
Change

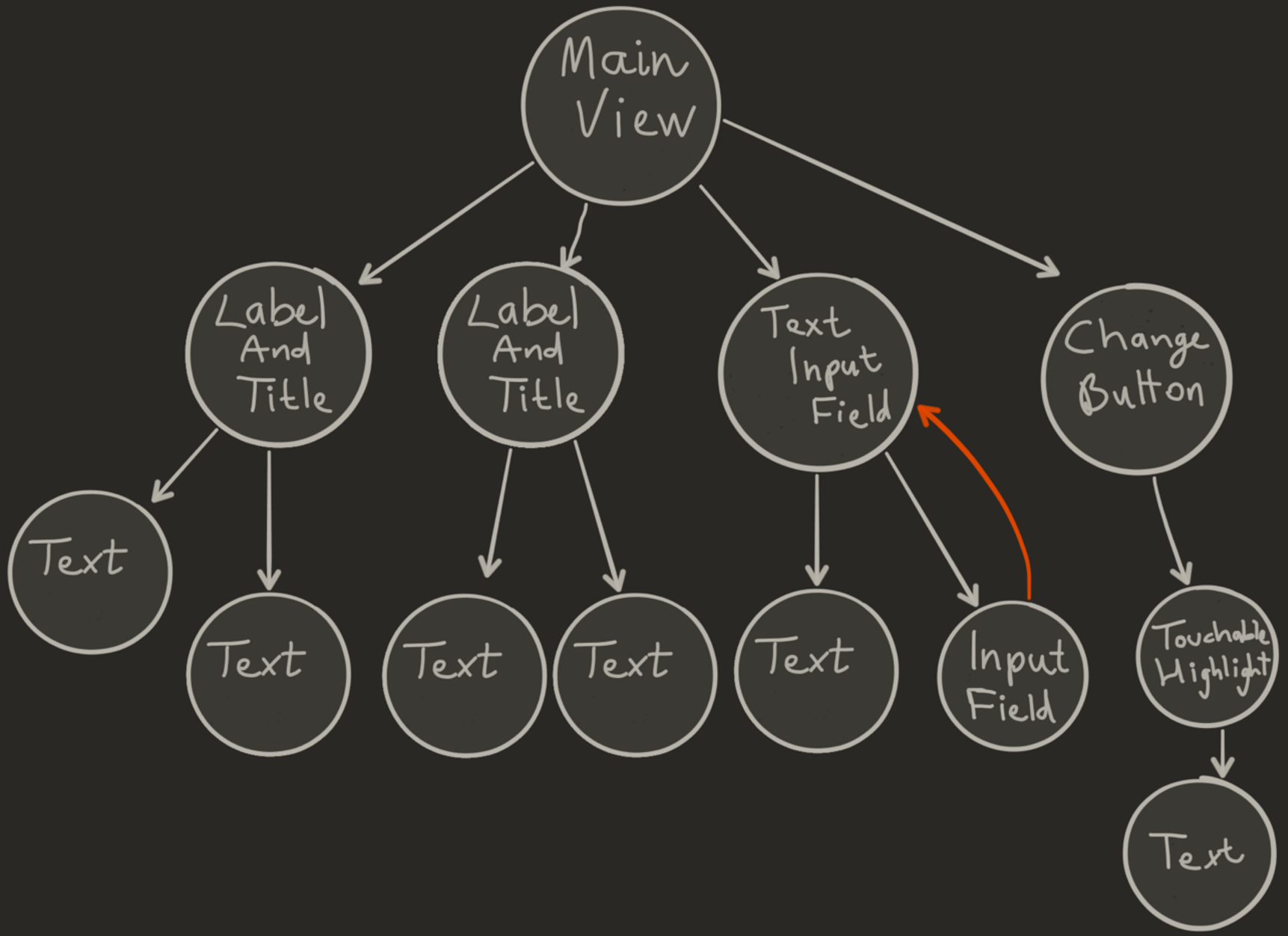
```tsx
interface TextInputFieldProps {
  textInput: string
  onTextInput: (text: string) => void
}

const TextInputField = ({textInput, onTextInput}: TextInputFieldProps) => {
  return (
    <View style={{ alignSelf: 'stretch' }}>
      <Text style={[styles.mono, styles.label]}>
        TextInput
      </Text>
      <TextInput
        style={[styles.mono, styles.title, styles.textInput]}
        onChangeText={onTextInput}
        value={textInput}
        />
    </View>
  )
}
```

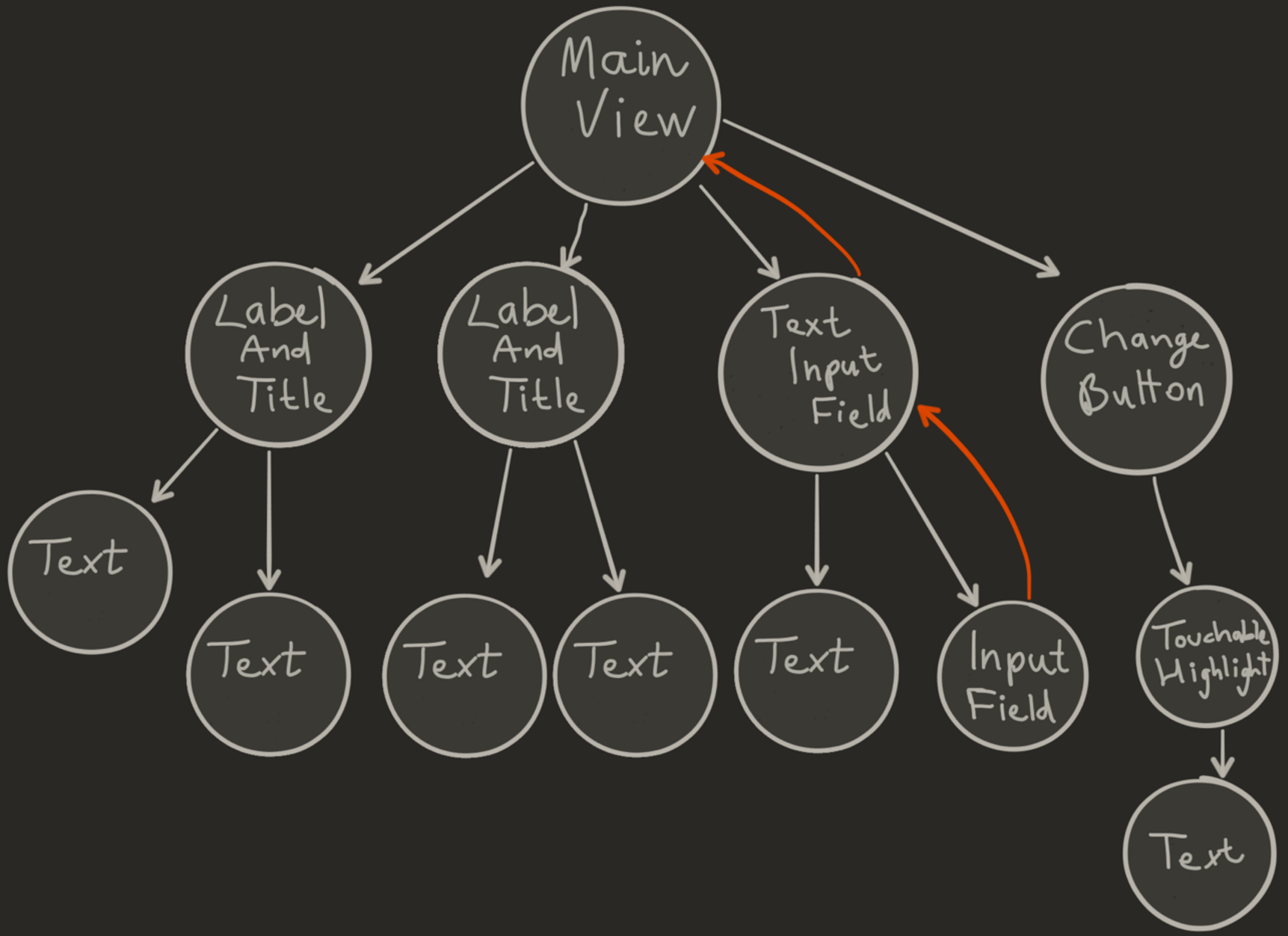


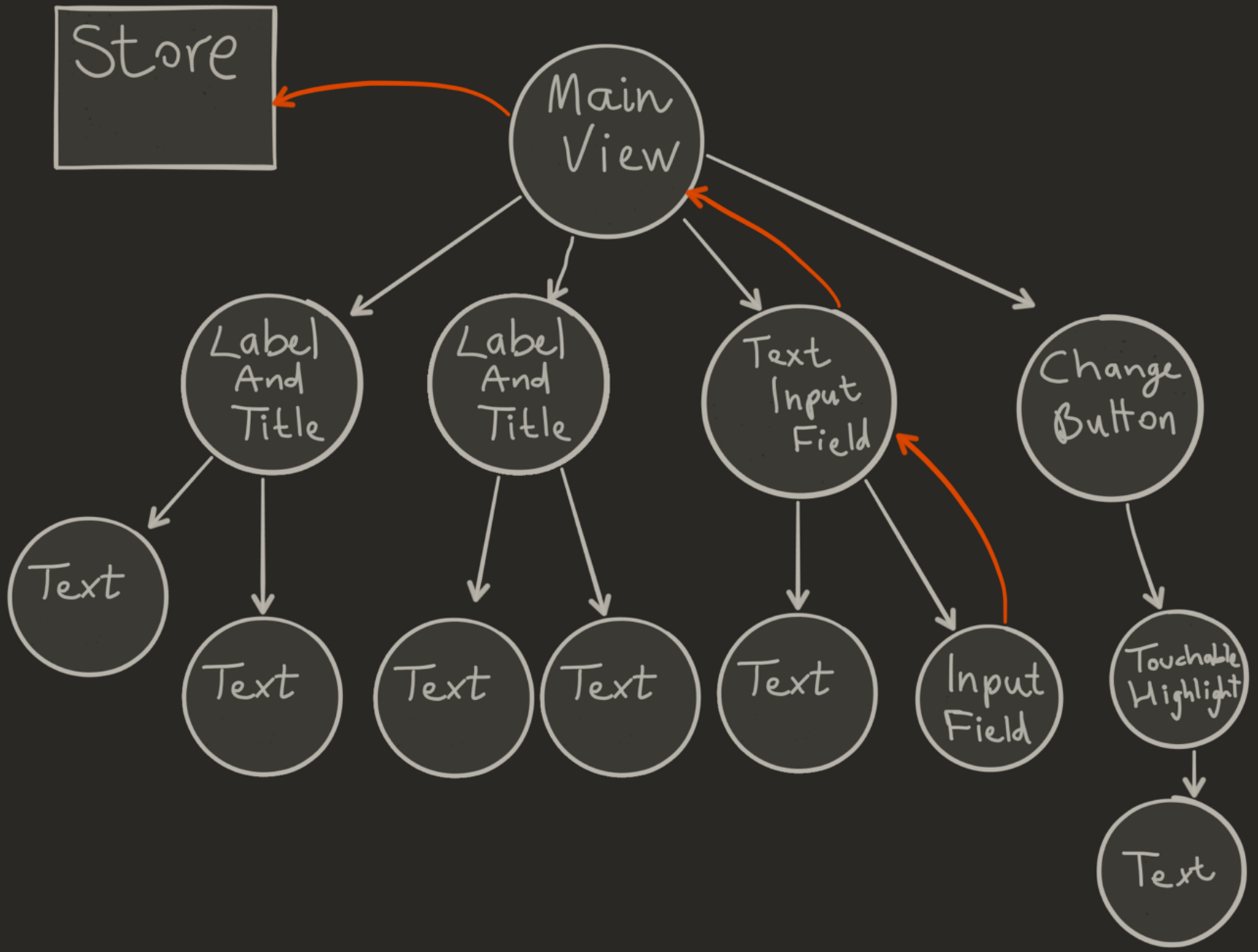
```tsx
interface MainScreenProps {
  title: string
  editedTextValue: string
  onTextInput: (text: string) => void
  onSetTitle: (text: string) => void
}

const MainScreen = ({
  title, editedTextValue, onTextInput, onSetTitle,
}: MainScreenProps) => {
  return (
    <View style={{ flex: 1 }}>
      <StatusBar
        backgroundColor='#282c34'
        barStyle='light-content'
        />
      <View style={styles.container}>
        <LabelAndTitle label='title' value={title} />
        <LabelAndTitle label='textInputValue' value={editedTextValue} />
        <TextInputField textInput={editedTextValue} onTextInput={onTextInput} />
        <ChangeButton onPress={() => onSetTitle(editedTextValue)} />
      </View>
    </View>
  )
}
```

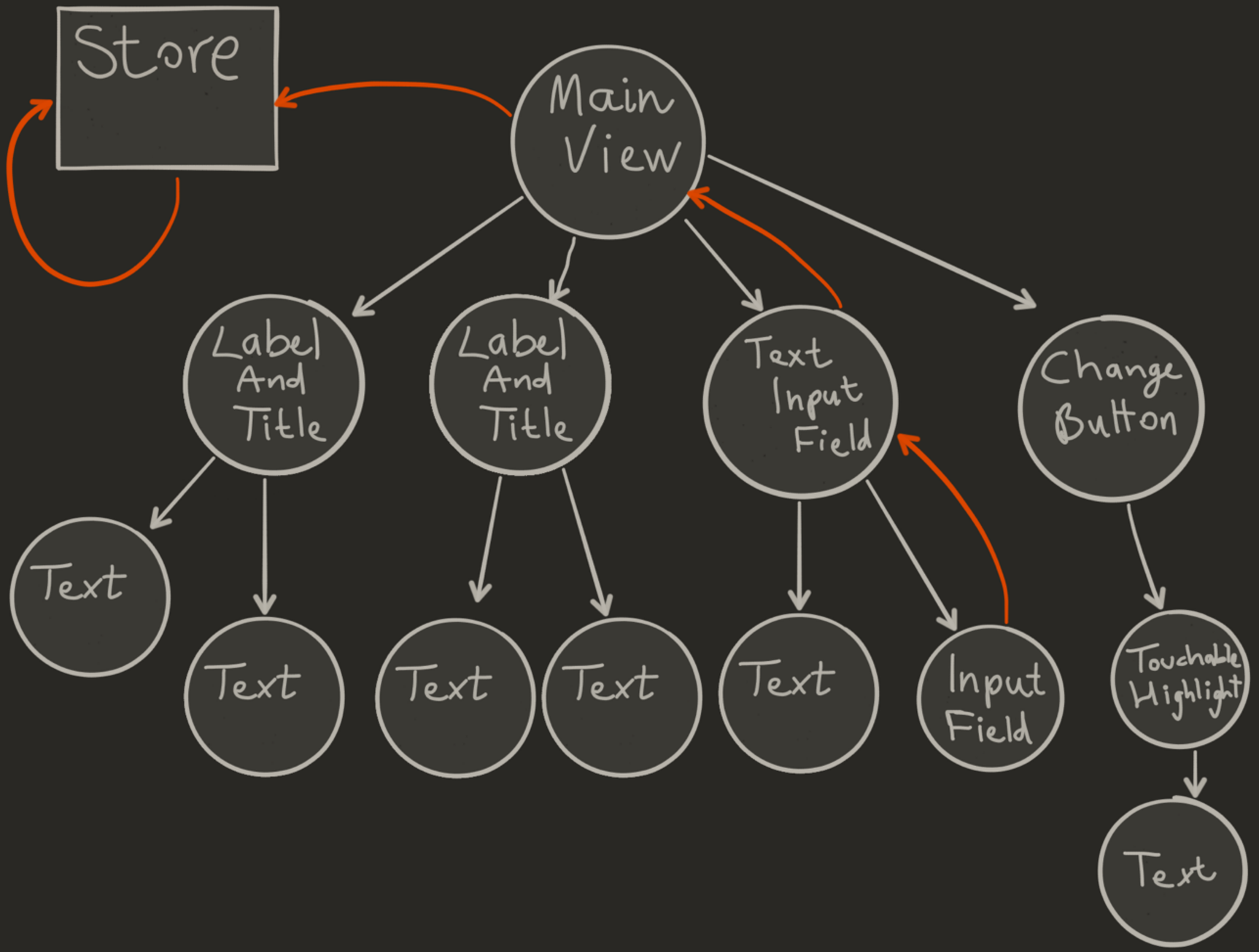


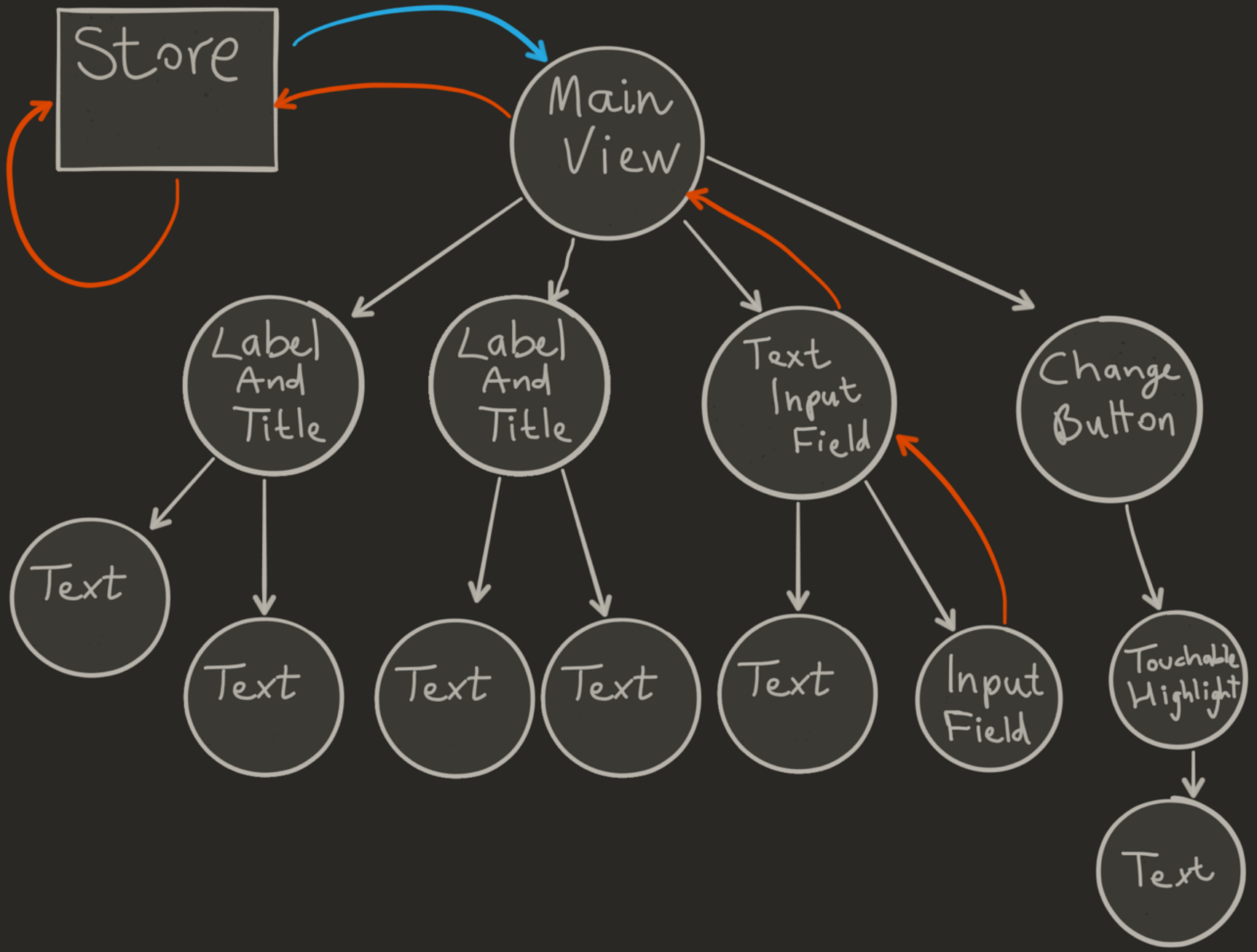

Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text

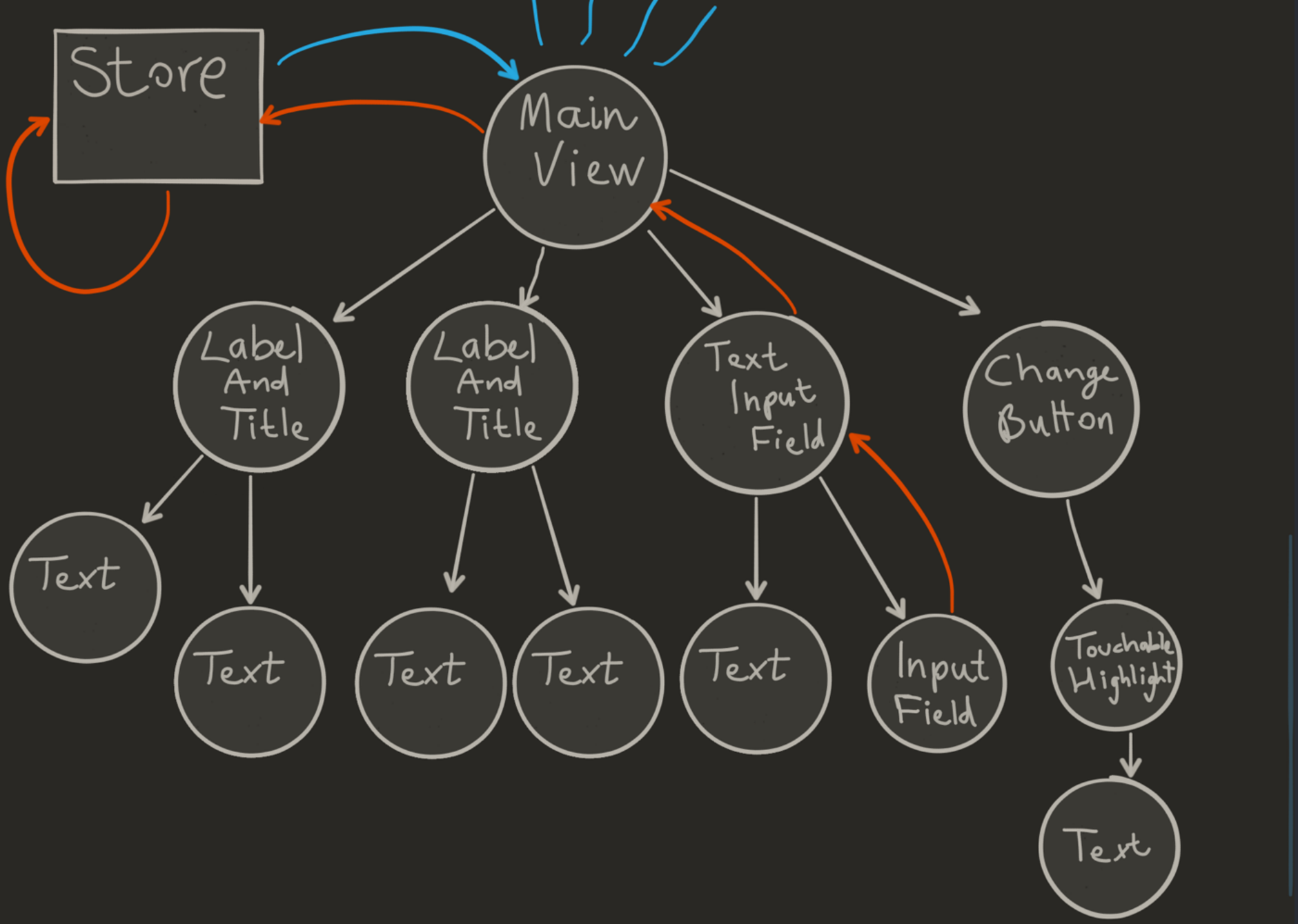

Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text

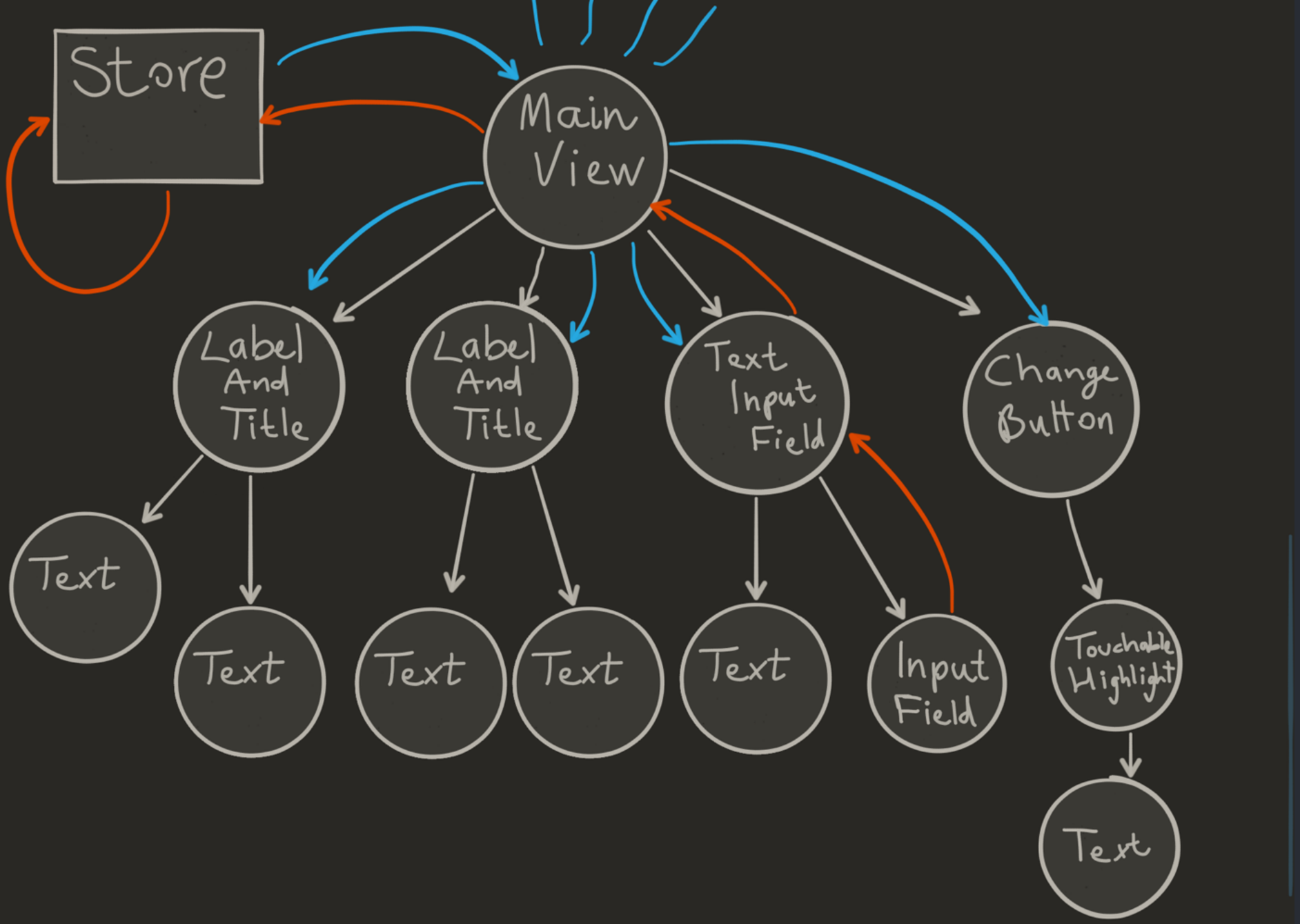

Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text

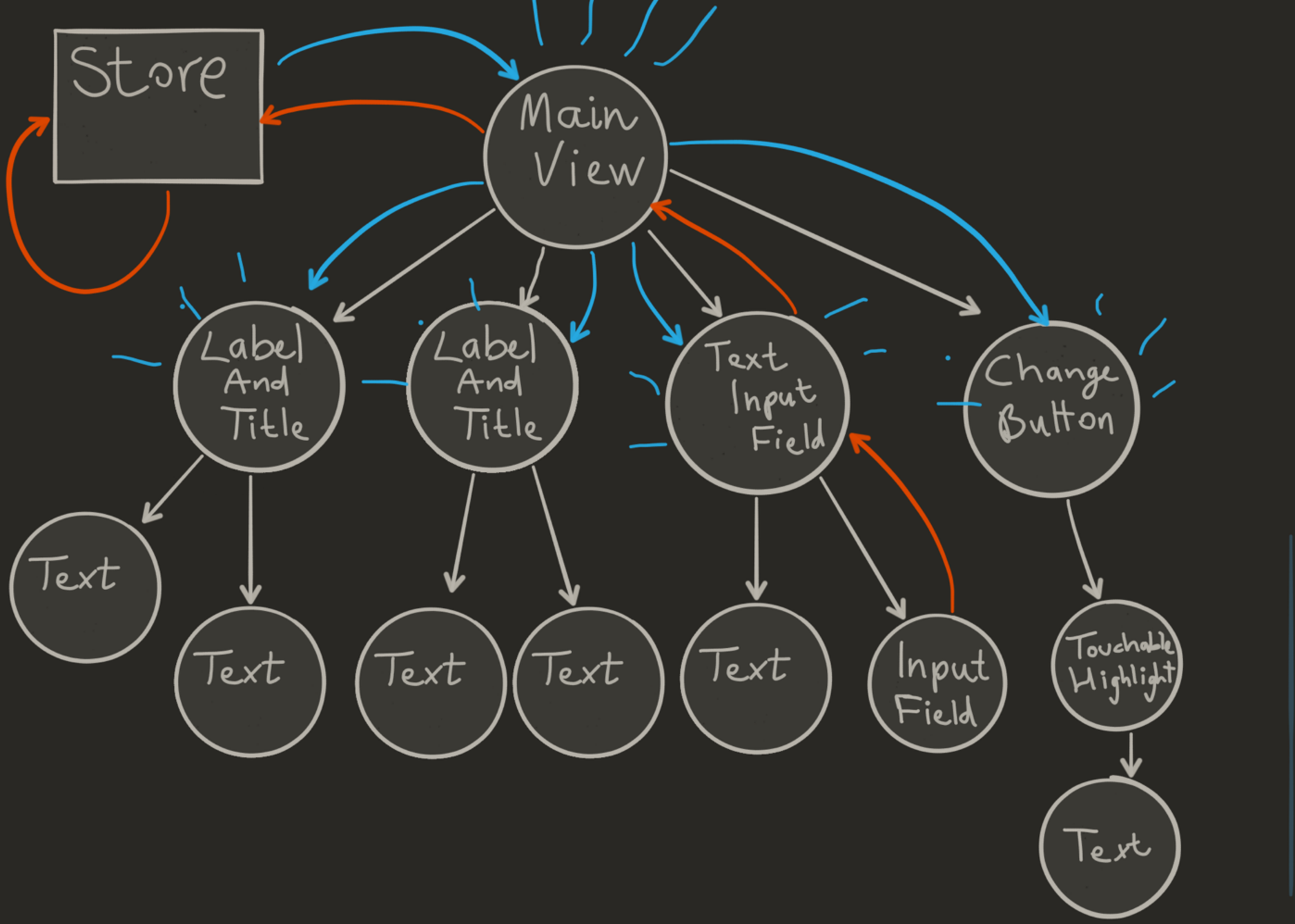

Store
Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text

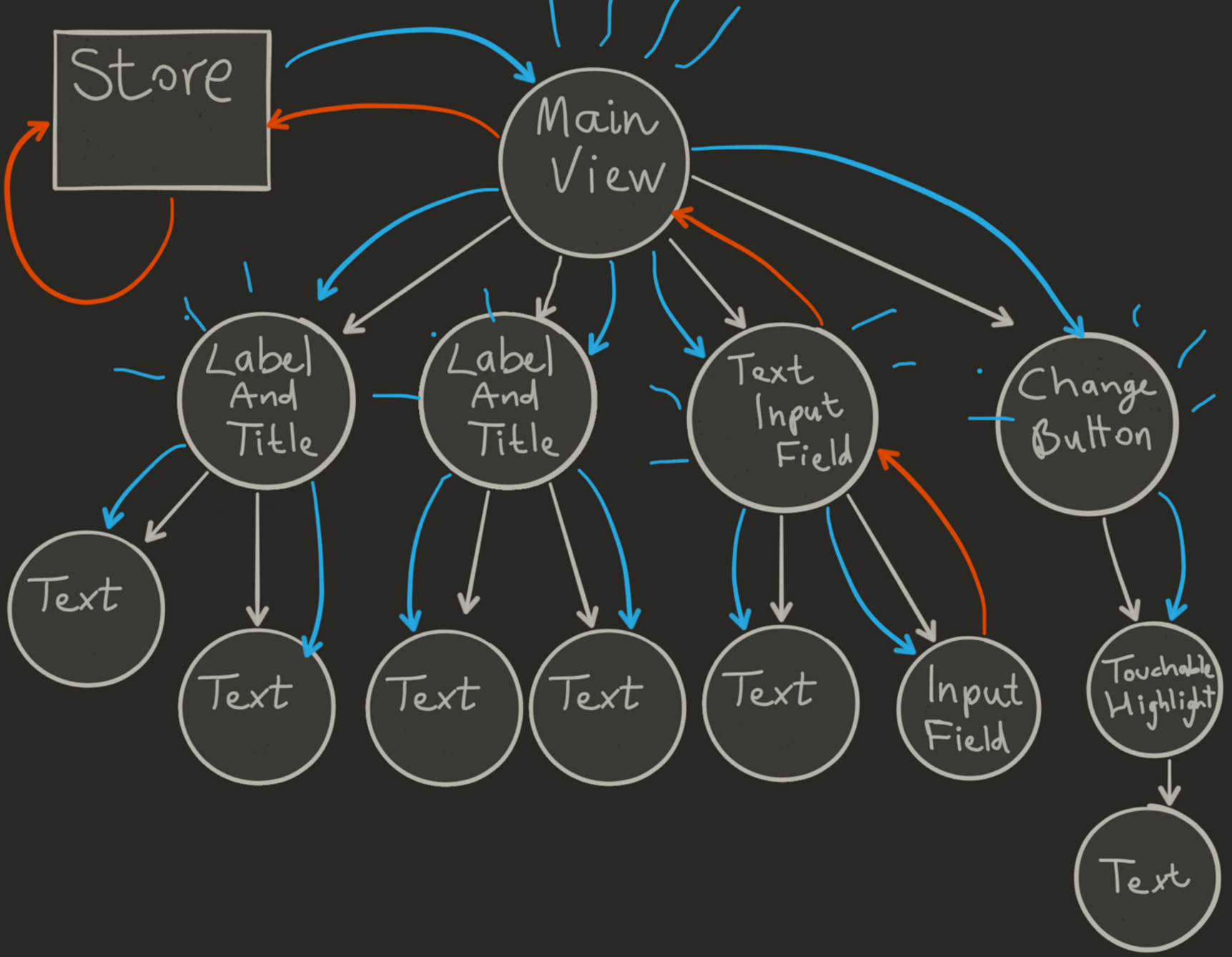

Store
Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text

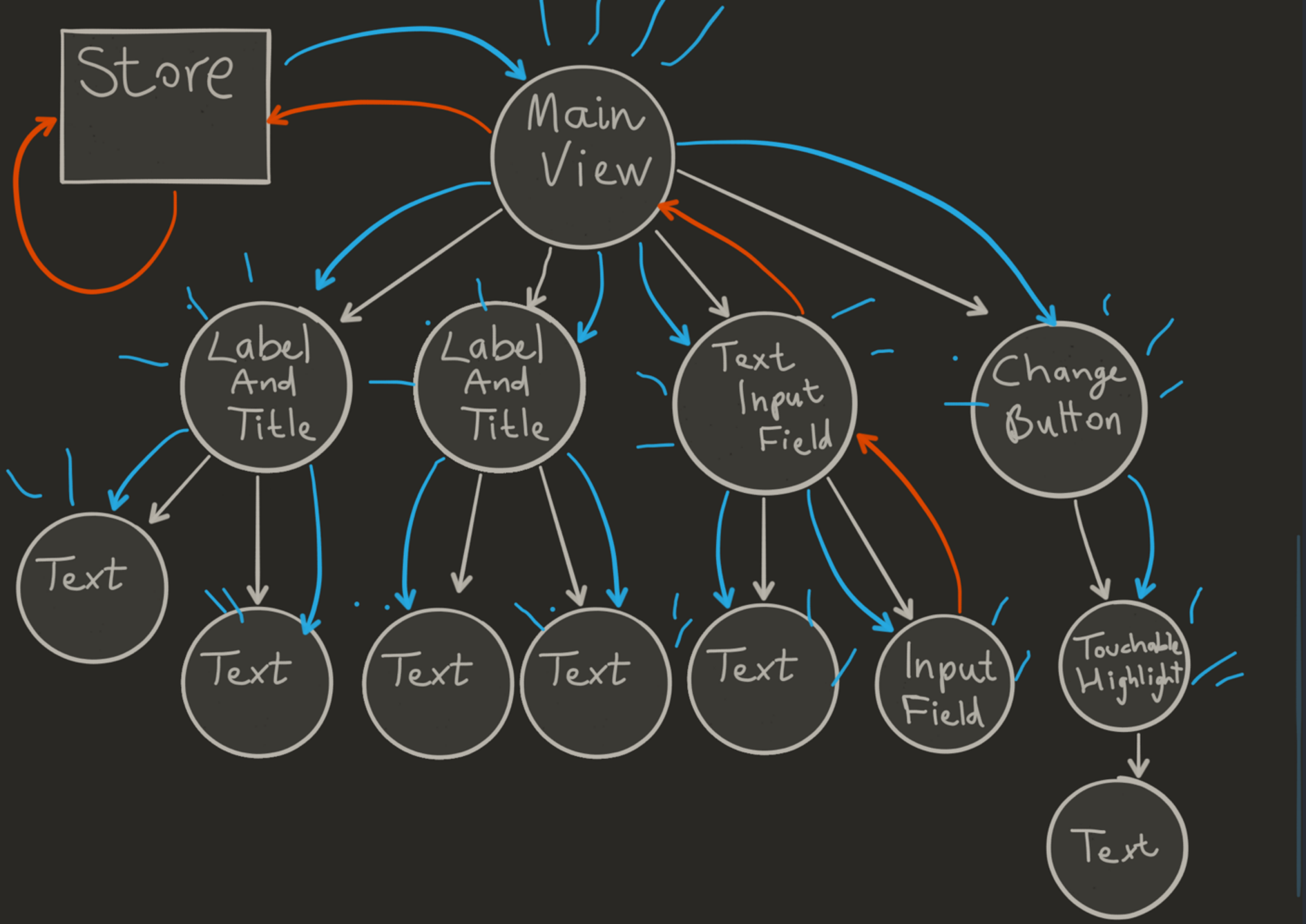

Store
Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text

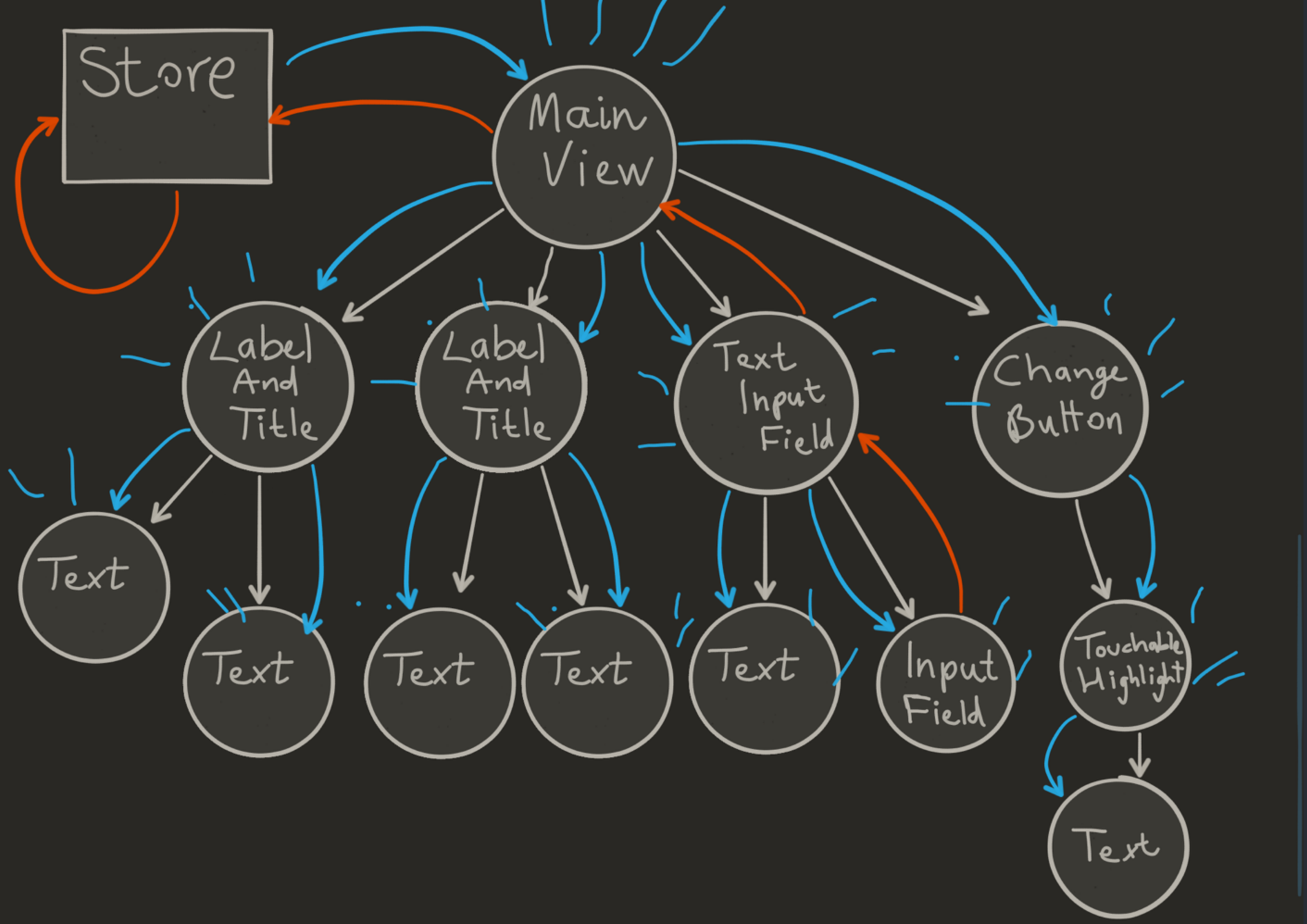

Store
Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text


Store
Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text

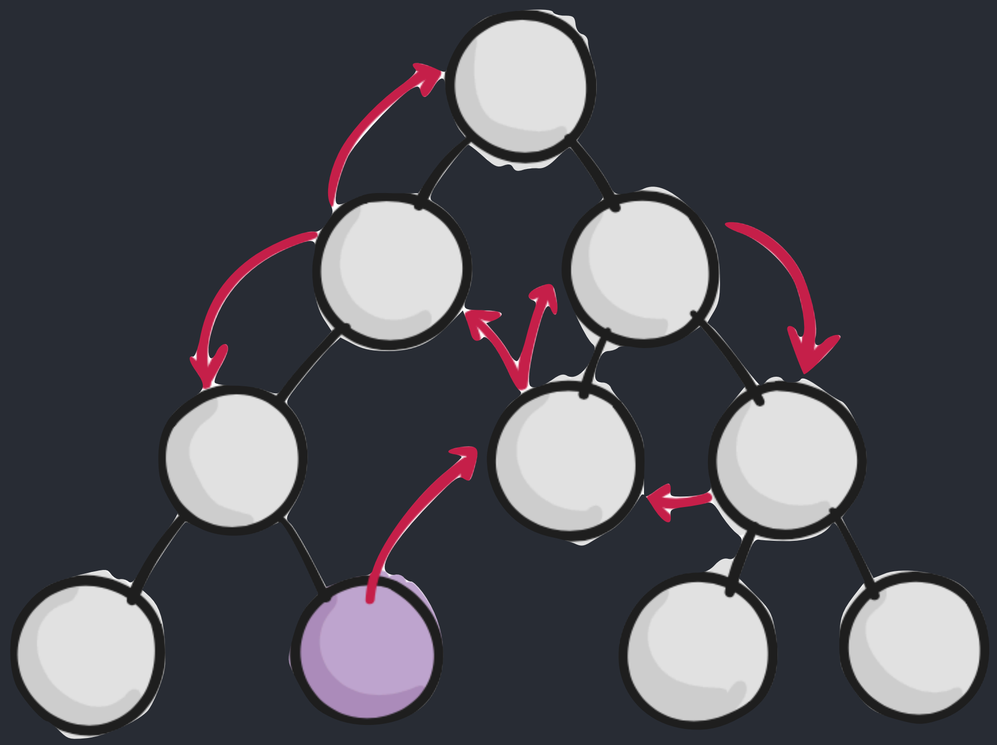

Store
Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text


Store
Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text

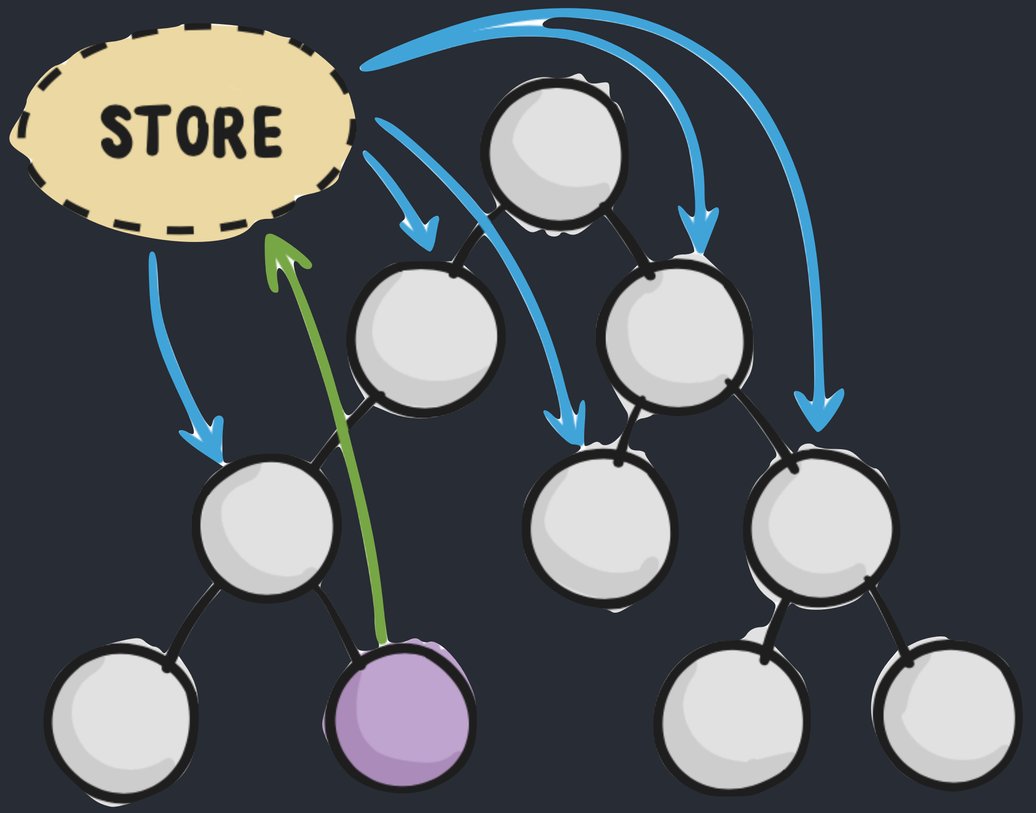

Store
Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text


Store
Main View
Label And Title
Label And Title
Text Input Field
Change Button
Text
Text
Text
Text
Text
Input Field
Touchable Highlight
Text

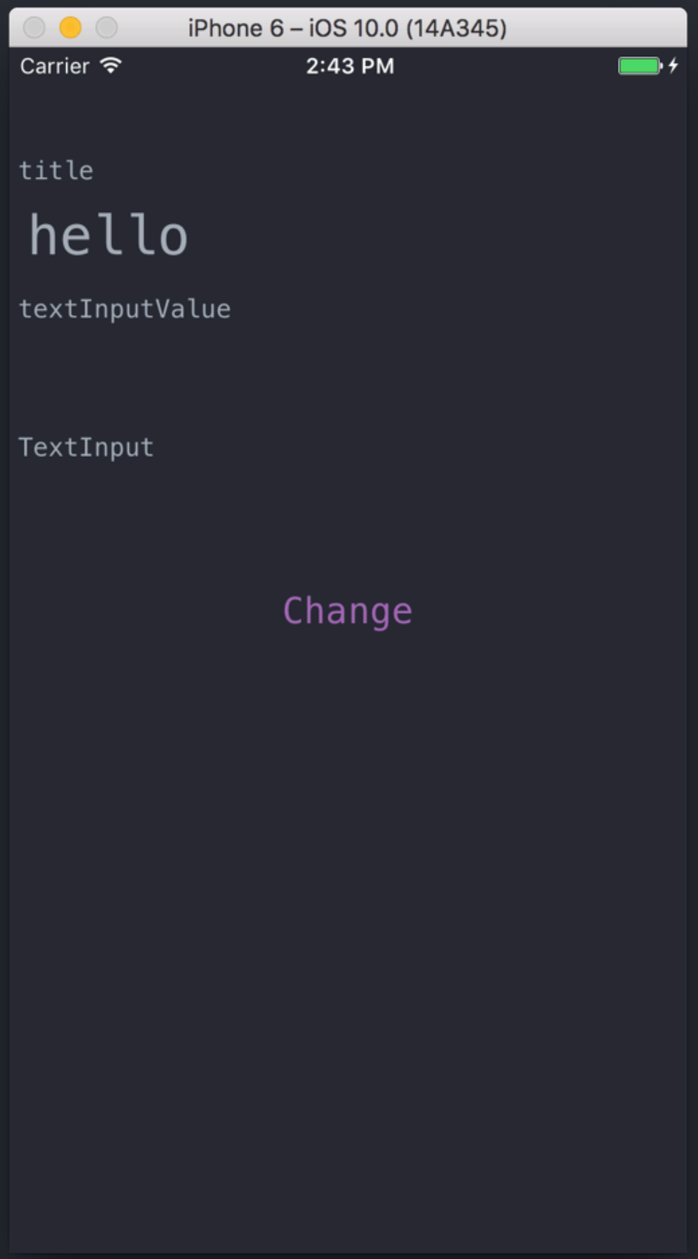
Without Redux



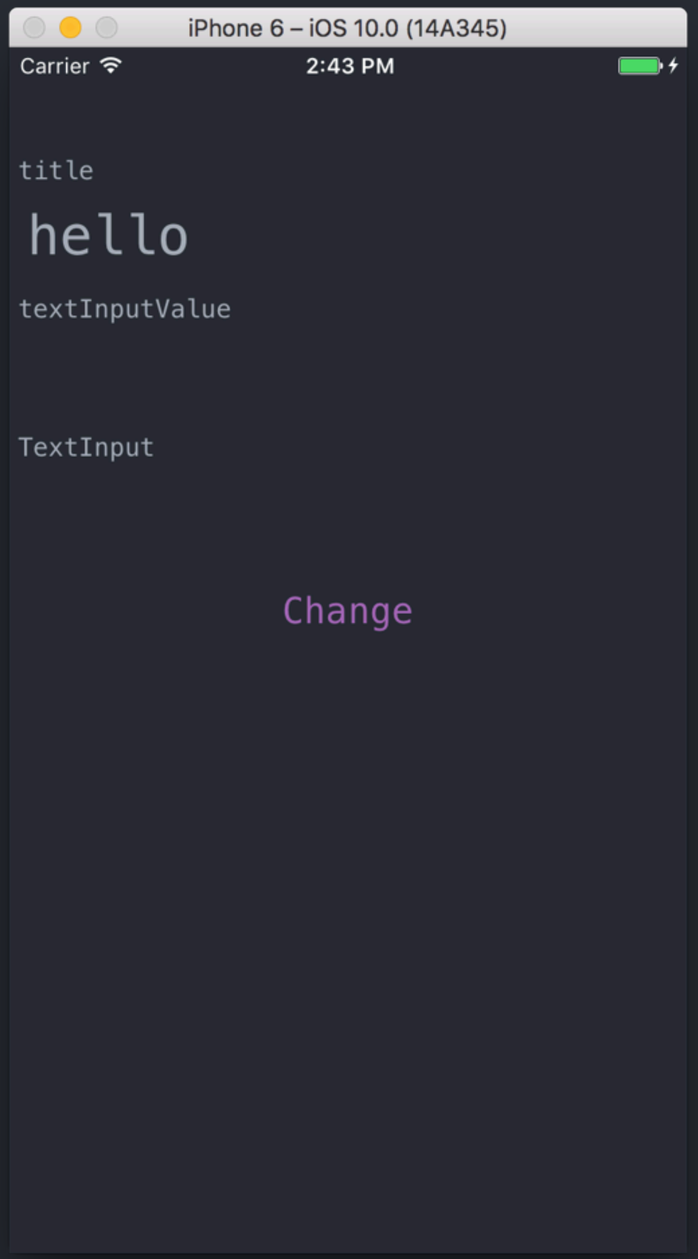
With Redux



```tsx
interface MainScreenProps {
  title: string
  editedTextValue: string
  onTextInput: (text: string) => void
  onSetTitle: (text: string) => void
}

const MainScreen = ({
  title, editedTextValue, onTextInput, onSetTitle,
}: MainScreenProps) => {
  return (
    <View style={{ flex: 1 }}>
      <StatusBar
        backgroundColor='#282c34'
        barStyle='light-content'
        />
      <View style={styles.container}>
        <LabelAndTitle label='title' value={title} />
        <LabelAndTitle label='textInputValue' value={editedTextValue} />
        <TextInputField textInput={editedTextValue} onTextInput={onTextInput} />
        <ChangeButton onPress={() => onSetTitle(editedTextValue)} />
      </View>
    </View>
  )
}
```



```
          <LabelAndTitle label='title' value={title} />
          <LabelAndTitle label='textInputValue' value={editedTextValue} />
          <TextInputField textInput={editedTextValue} onTextInput={onTextInput} />
          <ChangeButton onPress={() => onSetTitle(editedTextValue)} />
        </View>
      </View>
    )
}

const mapStateToProps = (state: State) => {
  return {
    title: state.title,
    editedTextValue: state.textInputValue,
  }
}
const mapDispatchToProps = (dispatch: Dispatch<any>) => {
  return {
    onTextInput: (text: string) => dispatch(textInput(text)),
    onSetTitle: (text: string) => dispatch(setTitle(text)),
  }
}
export const MainScreenContainer = connect(mapStateToProps, mapDispatchToProps)
(MainScreen)
```



Typescript

@ Good community support

@ Seems to have better IDE integration than Flow

@ always enable `noImplicitAny` and `strictNullChecks`

React Native

@ Try it!

@ Perfect for small projects, side projects

@ Production ready

React Native

@ Try it!

@ Perfect for small projects, side projects

@ Production ready

github.com/balazsbajorics/react-native-demo-app