

Adding Different Preview Options To Designer In MIT App Inventor

By
Hemani Kaul

Introduction:

Before the adoption of **What You See Is What You Get (WYSIWYG)** techniques, text appeared in editors using the system standard typeface and style with little indication of layout (margins, spacing, etc.). Users would find it difficult to see how their work would actually look like in reality. Soon the problem of developers or creators was solved by WYSIWYG techniques. Also called as a content editing tool. In WYSIWYG the edited content whether text or graphics, appears in a form close to a final product. So instead of manually writing source code, you deal with a convenient rich text editor in which you manipulate design elements. The WYSIWYG editor allows us to see approximately what our work will look like. Even though these types of editors have made it easier to see what we will get, they are not perfect. What we see may not be exactly what we will get 100% of the time.

My project this summer will be focussing on the above idea of WYSIWYG (what you see is what you get) giving users the view of how their apps will look like on different devices supporting different software versions. For my project I have chosen three categories of devices which support Android Holo(Android 3.0- 4.4.2), Android Material(Android 5+) and general iOS theme. For designing apps we currently have MockForm(design editor) which gives us the basic picture of how our app will look like but during actual execution it may slightly change or variate on different devices. So keeping all such points in mind, I will work on providing the actual view of how the designed apps will look like so that users get to preview their apps and make changes accordingly rather than testing again and again on different devices.

User Scenario:

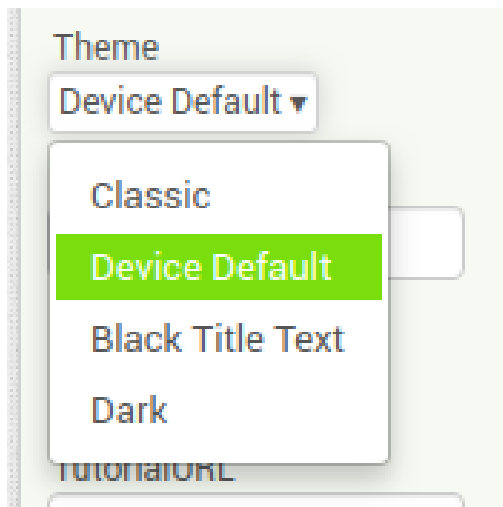
To understand better let's meet Carrie. Carrie's mother loves designing clothes and to display her talent to the world she decides to design an app which would help her mother to do an online fashion show. Carrie had recently learned to develop apps with the help of MIT App Inventor so she decided to create an app called "FashionWorld" where she would display all the clothes designed by her mother so that users can purchase or give special orders and thus motivate her mother's talent. The main thing in Carrie's mind was to create a beautiful UI so that users would love to explore more.

Carrie starts designing the FashionWorld app by adding a purple color status bar. To see how this color would match with Primary Color and Color Accent she has to build apk again and again just to make the perfect color combination. Currently she can preview how her app would

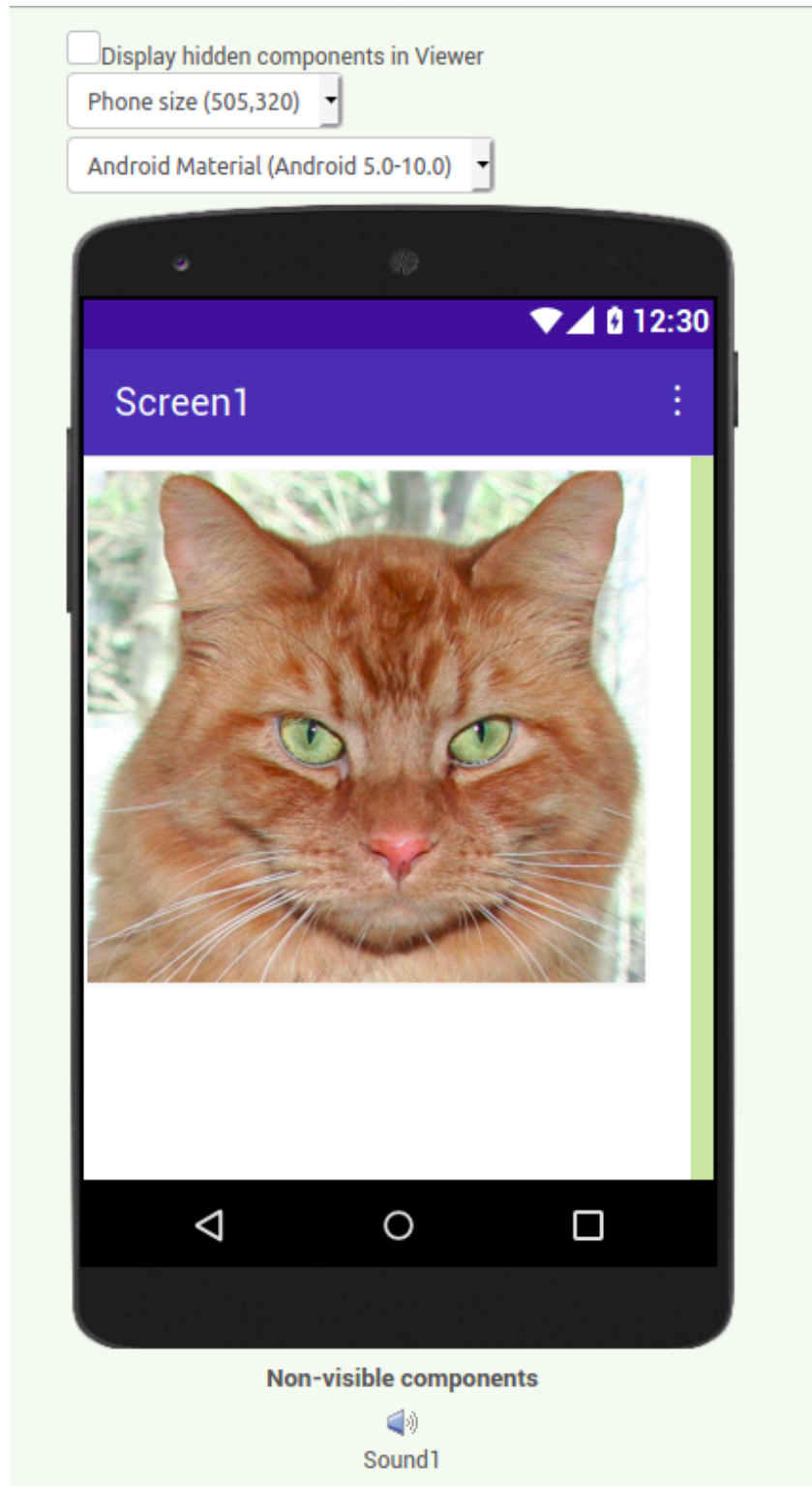
look like on devices with different sizes but she doesn't know how it would look on different android versions and even iOS devices. Carrie owns a Samsung Galaxy M31 which supports android 10 which she is using for her app testing. She does not have any older android phones or an iPhone, so she is not sure how her customers phone would portray her designed app. My project this summer will address Carrie's problem by adding different preview styles which will help her develop "FashionWorld" and reach more users to promote her mother's business.

Solution:

- To enable the preview mode one has to select the "**Device Default**" theme option from the properties panel (Classic mode is always the same regardless of Android version). The drop-down will appear above the MockForm. As seen in the below images.



- Since my project is based on Designer Changes I would like to represent my work with the help of images which would give you a better idea about what I am going to do this summer.
- Below is the image of the drop-down which would appear on the screen as soon as Device Default is clicked. Currently, it is representing the Material UI preview which would target devices with Android (5+). Here I have added a basic image of how I am going to implement the Material UI preview.



- So coming back to Carrie we will see how her “Fashion World” login screen will look on different devices.



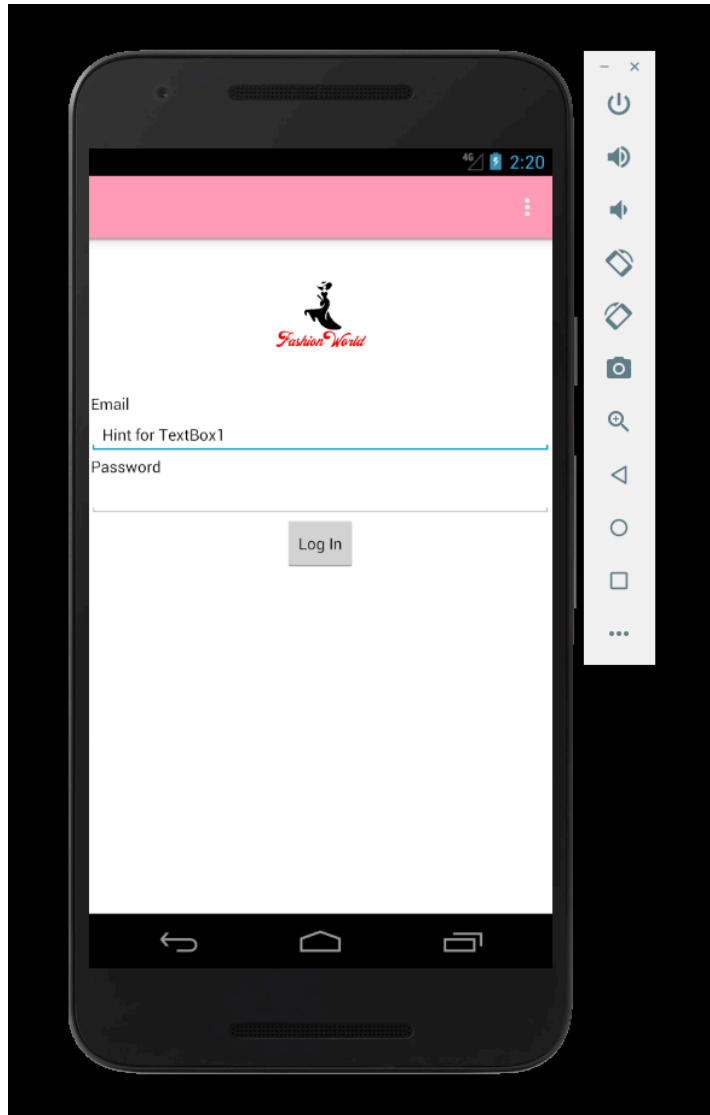
Email

Password

Log In

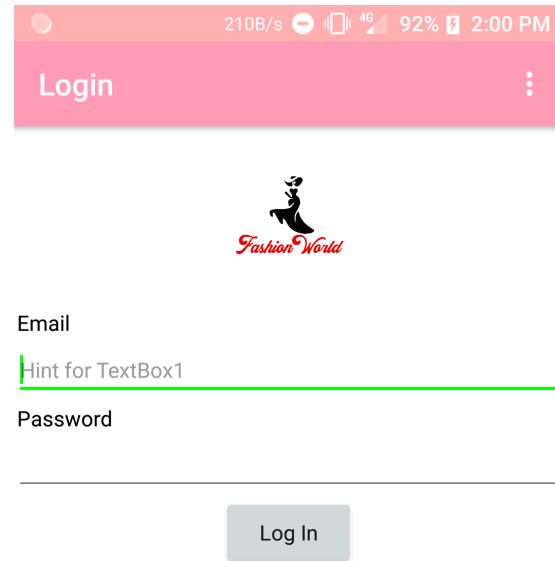
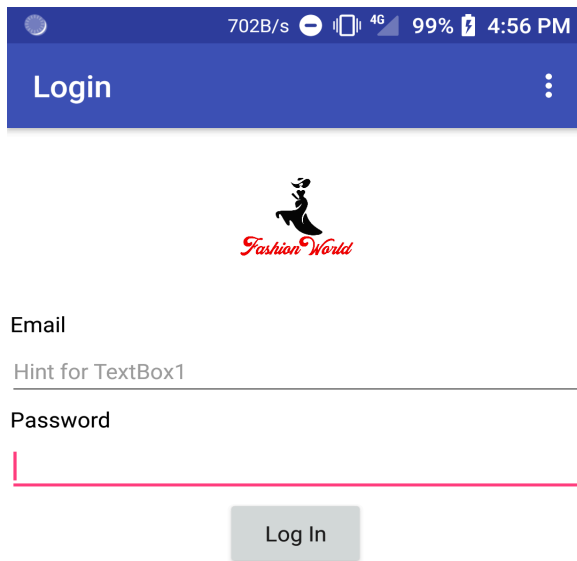


*This is general **Classic Preview** which remains uniform on all devices*



*This is **Android Holo Preview** and the current emulator supports Android Jellybean*

- The main goal of this preview is to get an idea how the user's designed app would look like on older Android Versions and referring to this UI, I am working on implementing this in Android Holo Preview.



*This is the **Android Material Preview** and the current device supports Android Oreo*

- Here in the above images we can see the color combination of PrimaryColor, PrimaryDarkColor and ColorAccent. So here in the Material Theme I am focussing on this goal to achieve this color combination for all the respective components.
- Another goal I am focussing here is to style the components according to Material Design.



*The above image is the general introduction of the **iOS preview**. More features would be added during the course of summer.*

Technical Details:

To initiate the preview styles firstly we have to add some changes in the [SimpleVisibleComponentsPanel.java](#)

- Declare new variables:

```
private final List<Box> listboxThemePreview; // A List<Box> for Holo/Material/iOS
preview themes styles
```

```
private final String[] drop_list_preview_theme = { "Android 5.0-10.0", "Android
3.0-4.4.2", "iOS" };
```

- Only enable the `listboxThemePreview` when Classic Theme is set to false.

```
boolean classic = (themeStyle.equals("Classic"));
```

```
listboxThemePreview.setVisible(!classic);
```

- For sending the selected index to `MockForm.java` :

```
form.changeThemePreview(idx);
```

Further in the [MockForm.java](#) following changes are needed:

- Added a method which connects the chosen preview to its respective CSS File.

```
private void changePreview() {
```

```
    // making the preview changes
```

```
    if (idxThemeStyle == 0) {
```

```
        phoneBar = new PhoneBar(primaryDarkColor);
```

```
        formWidget.addStyleDependentName("AndroidMaterial");
```

```
    } else if (idxThemeStyle == 1) {
```

```
        phoneBar = new PhoneBar();
```

```
        formWidget.addStyleDependentName("AndroidHolo");
```



```

    } else if (idxThemeStyle ==2){

phoneWidget.setStylePrimaryName("ode-SimpleMockFormiPhonePotrait");

    formWidget.addStyleDependentName("iOS");

    }

```

- Added new constructor for Material UI PhoneBar class:

```

PhoneBar(String color) {

panel=new HorizontalPanel();

panel.add(phoneBarWifiImage);

panel.add(phoneBarSignalImage);

panel.add(phoneBarBatteryImage);

panel.add(time);

panel.setStyleName("AndroidMaterialIconsPanel");

bar = new DockPanel();

bar.setHorizontalAlignment(HorizontalPanel.ALIGN_RIGHT);

bar.add(panel, DockPanel.EAST);

initWidget(bar);

MockComponentsUtil.setWidgetBackgroundColor(bar, color);

setStylePrimaryName("ode-SimpleMockFormPhoneBarAndroidMaterial");

setSize("100%", HEIGHT + "px");

}

```

- To avoid the complexity three different css files are added [Android_Material.css](#), [Android_Holo.css](#) and [iOS.css](#) which will contain all the respective css changes.

Conclusion:

The goal of this project will be an attempt to improvise the user experience in designing their apps. Developers will get to preview how their work would look like on various devices just like we have preview options for different phone sizes.

Lastly I conclude that this is the basic idea of what I will be working on this summer, various things may change along the way but the main target will remain the same.

Thanks.