

# Styling Components Lab

Now that we know the better and worse ways of styling React Native components, let's apply some of those to (finally) getting our app to look good. This lab will have fewer explicit instructions because we want you to express yourself in the look and feel of your own app.

And since it is a simpler topic, this would be a great time to erase any technical debt you might have built up. As you look through your components, take any poor designs or antipatterns and refactor them so that your code is stronger, more robust, more encapsulated, more abstract, or simply easier to understand.

## Working with fonts

1. Experiment with a few different fonts. We generally like to stick to the standard fonts throughout, but you may have an eye for design and want a different look.
2. Choose a `<Text>` element somewhere. Apply a style to it. Then edit that style and add this:
3. `fontFamily: 'Papyrus'`,
4. Run and test on both platforms. It should work great on iOS but throw on Android.
5. Now change `fontFamily` to `'Roboto'`. It should work great on Android but throw on iOS.
6. Make these fonts work on the same label, choosing the supported font for each platform.
7. Once you do, feel free to experiment with a few fonts until you find one you like.

## Making a styled element

8. Notice that we have the movie title in several places in the app. We want that to be consistent, so ...
9. Create a new component called `Title.js`
10. It should do something like ...

```
export const Title = props =>
  <Text style={{fontSize:30,fontWeight:Bold}}>{props.children}</Text>
```
11. Now, go through each screen and change the movie titles to use `<Title>` instead of `<Text>`. Don't forget to import!
12. Run and test. Adjust as needed.
13. Maybe use your `Title` for the company's business name. It's up to you whether to do that or create an independent style.

## Nesting `<Texts>`

14. Look at the ratings of the film at the bottom of the scene. It says "X.Y/10". Notice that our UX designers wanted the "X.Y/" portion to be large and the "10" portion to be small. If you haven't already done so, use Text nesting. Write it something like this:

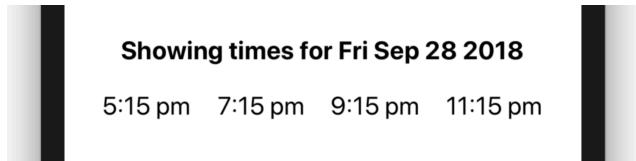
```
<Text style={bigger}>X.Y/<Text style={smaller}>10</Text></Text>
```
15. Then, of course, improve the style names.

## Scaling images

16. Make your movie posters resize nicely. Right now, you probably have them scaled, but parts of them are cut off. Change the `resizeMode` property until you like how it looks better.
17. Adjust the sizes until you like their look

# Showing Times

18. In ShowingTimes.js, make the headline bigger and centered.
19. Make each of the times large enough to be tapped easily. Something like this:



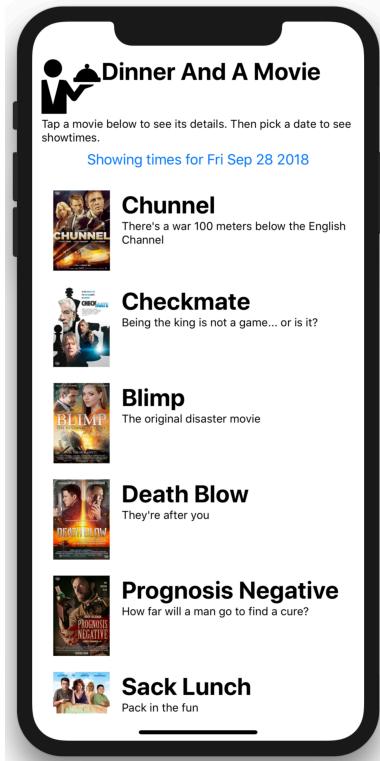
Showing times for Fri Sep 28 2018

5:15 pm 7:15 pm 9:15 pm 11:15 pm

# Perfecting the styles

Below you'll notice some screen shots. These can serve as inspiration to you. Your mission for each of them is to make your design as good or better. ('Better' should not be too much of a challenge). You'll also see that there are some specific instructions on a few of the scenes to improve the operation.

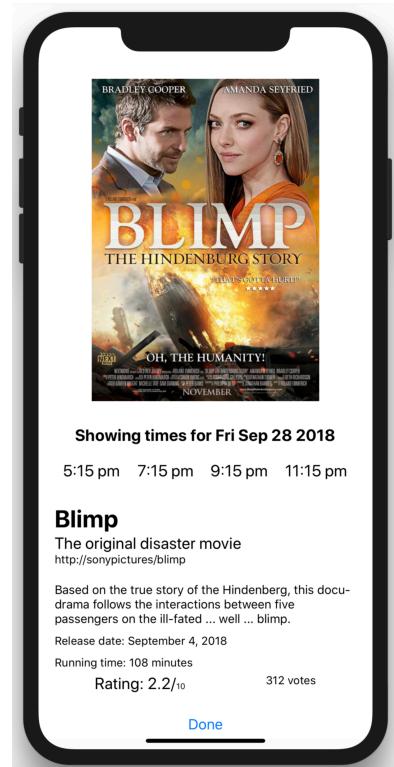
20. For each one, adjust styles as needed to make the scenes look more impressive!



## Landing Scene

21. When the user chooses a film, change that film's style slightly to let the user know they've chosen it. You can change background color, increase the margins or put a border around it. Whatever looks good to you.

## Film Details

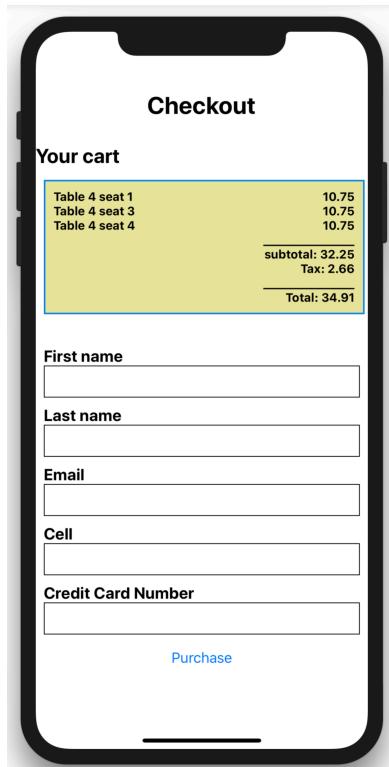




## Pick Seats

22. We're going to use styling to let the user know the status of a seat.
23. Alter the data in your JSON file to add another property to each seat object, the 'status' property. Throughout, give it either 'seatIsTaken', 'seatIsSelected', or remove the status property altogether.
24. Then make the styling of a seat conditional. If the seatIsTaken, make the text lighter and red. If the seatIsSelected, make it darker and orange. If the status is falsey, it is available. Make it bold and dark.

## Checkout



25. Bonus!! If you get finished early, note that PickSeats is getting pretty heavy. You may want to extract Table into its own component and Seat into its own.