



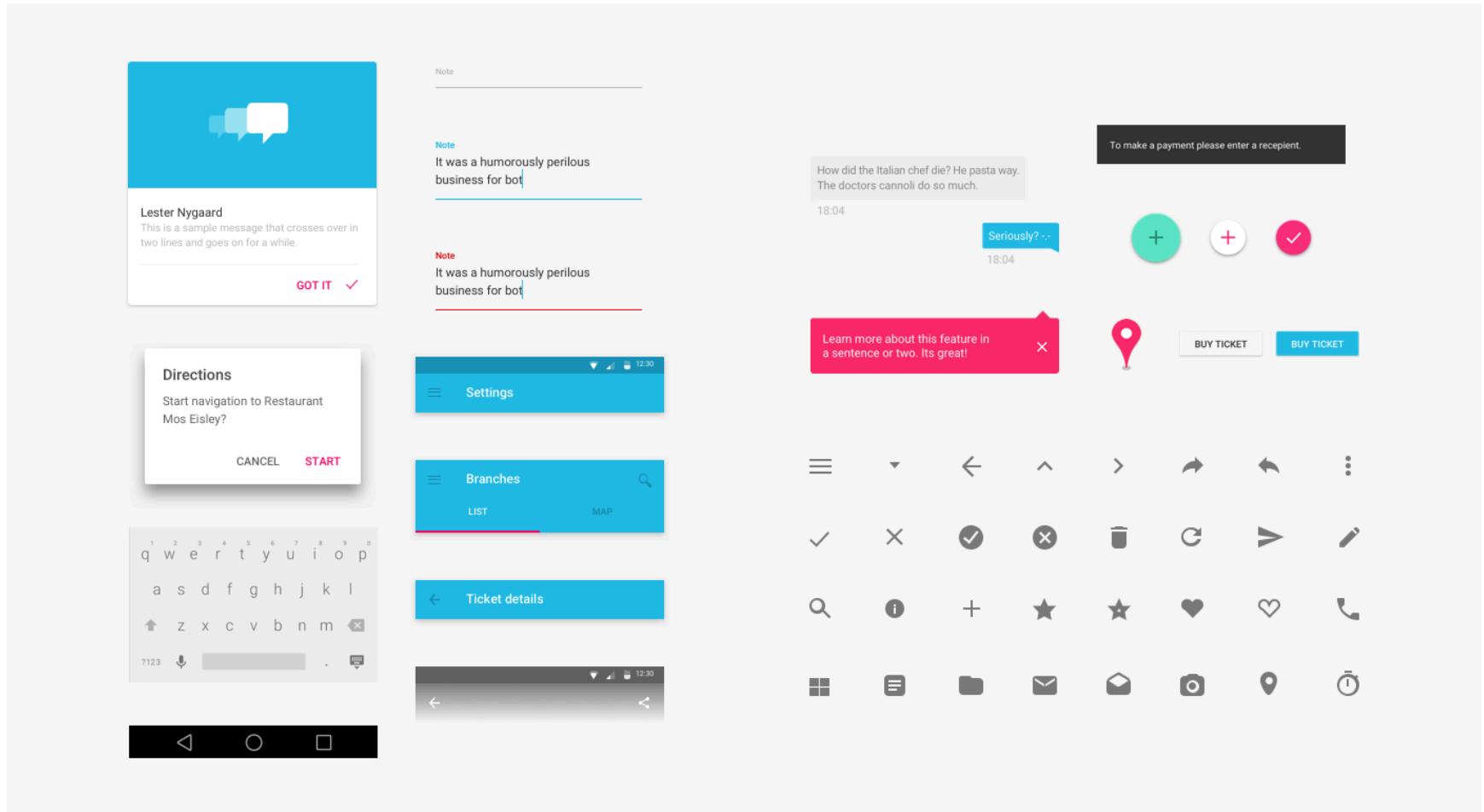
INTRO TO ANDROID UI

Android UI Intro

- GUI Elements
- Layouts
- Views
- UI Editor
- R.java



GUI Elements

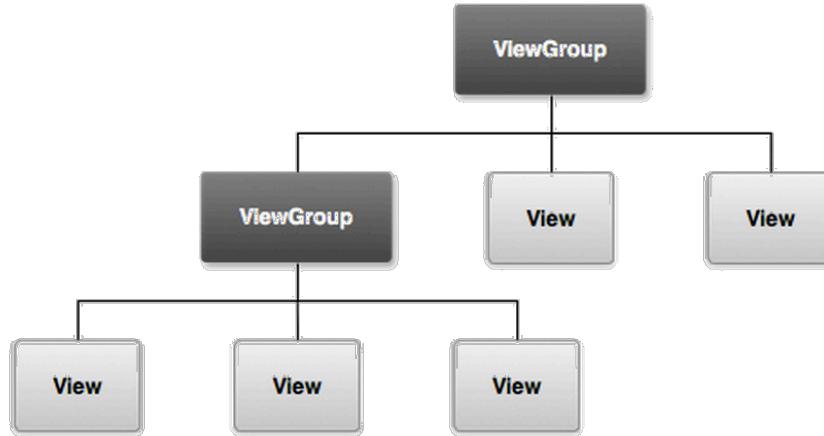


<https://www.infinum.co/the-capsized-eight/articles/android-lollipop-material-design-ui-kit-for-sketch>



Views

- View is the base class for all the stuff that is drawn in the interface
- View Groups organize views and their position



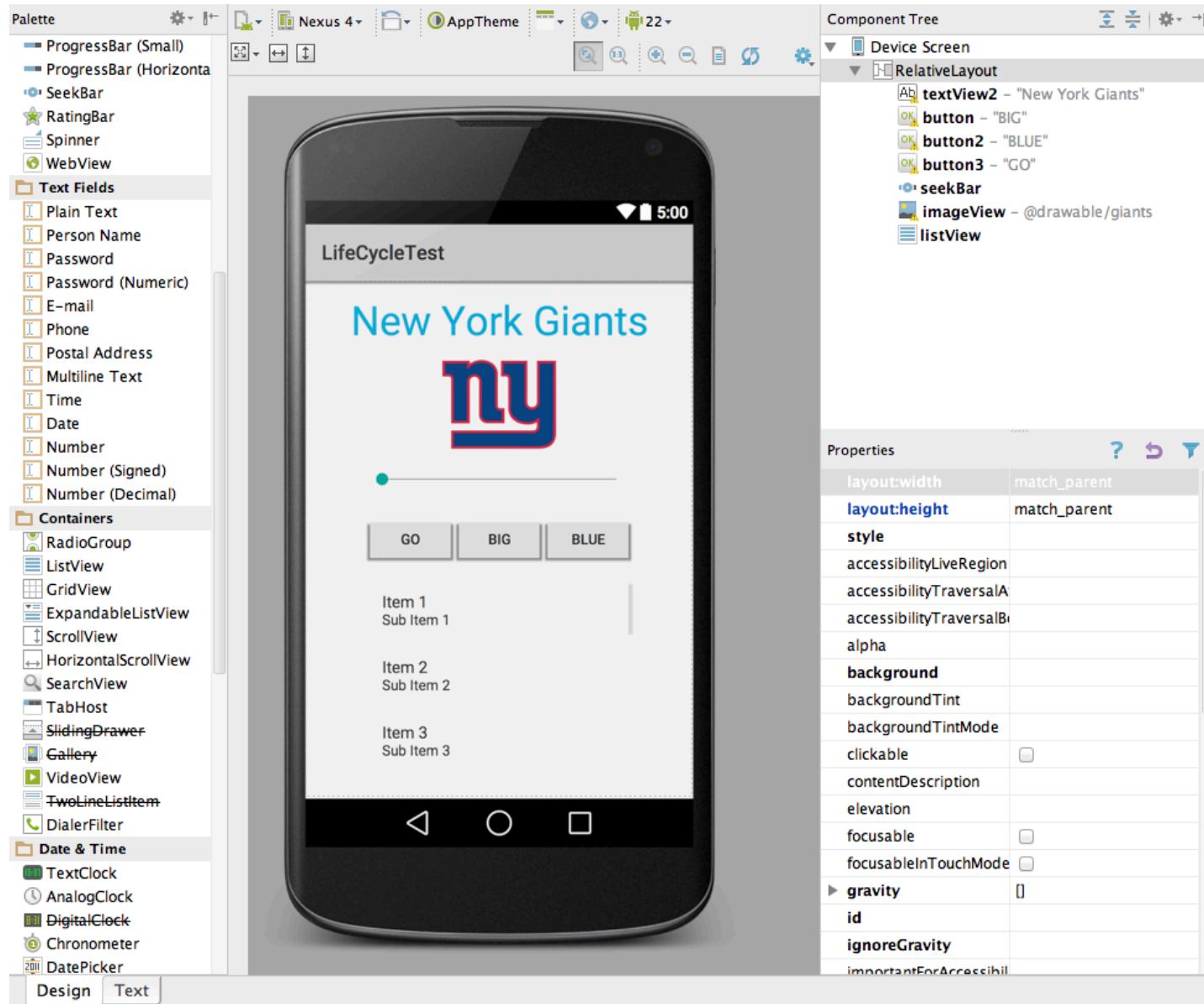
<http://developer.android.com/guide/topics/ui/overview.html>



Android Layouts

- Mostly the UI is defined and designed in XML files using the built-in Layout Editor
 - Drag'n'drop interface
 - Text/XML based interface
- XML object name ~= Android class name
- Behaviors are implemented other places, e.g. in Activities that use the layouts at runtime.





Common Input Controls/Widgets

Control Type	Description	Related Classes
Button	A push-button that can be pressed, or clicked, by the user to perform an action.	Button
Text field	An editable text field. You can use the AutoCompleteTextView widget to create a text entry widget that provides auto-complete suggestions	EditText , AutoCompleteTextView
Checkbox	An on/off switch that can be toggled by the user. You should use checkboxes when presenting users with a group of selectable options that are not mutually exclusive.	CheckBox
Radio button	Similar to checkboxes, except that only one option can be selected in the group.	RadioGroup RadioButton
Toggle button	An on/off button with a light indicator.	ToggleButton
Spinner	A drop-down list that allows users to select one value from a set.	Spinner
Pickers	A dialog for users to select a single value for a set by using up/down buttons or via a swipe gesture. Use a DatePicker widget to enter the values for the date (month, day, year) or a TimePicker widget to enter the values for a time (hour, minute, AM/PM), which will be formatted automatically for the user's locale.	DatePicker , TimePicker

<http://developer.android.com/guide/topics/ui/controls.html>



Runtime

- A **View** is instantiated as the given Layout is inflated e.g. an **Activity** has:
`setContentView(R.layout.<layout name>);`
- Can be retrieved by knowing the id of the **View** as defined in XML
`findViewById(R.id.<view id>);`
- Can also be created dynamically from your code or adaptors



R.java

- Automatically generated on build
- Holds id for all resources
 - (layouts, drawables, colors, strings, etc.)
- Essentially a range of static nested classes with static final ints as ids.



Event Listeners

- Writing the code that reacts to user input
- Listen for UI events from views, e.g.
 - `onClick()`
From `View.OnClickListener`
 - `onLongClick()`
From `View.OnLongClickListener`



Creating Android UIs



EXERCISE 2

