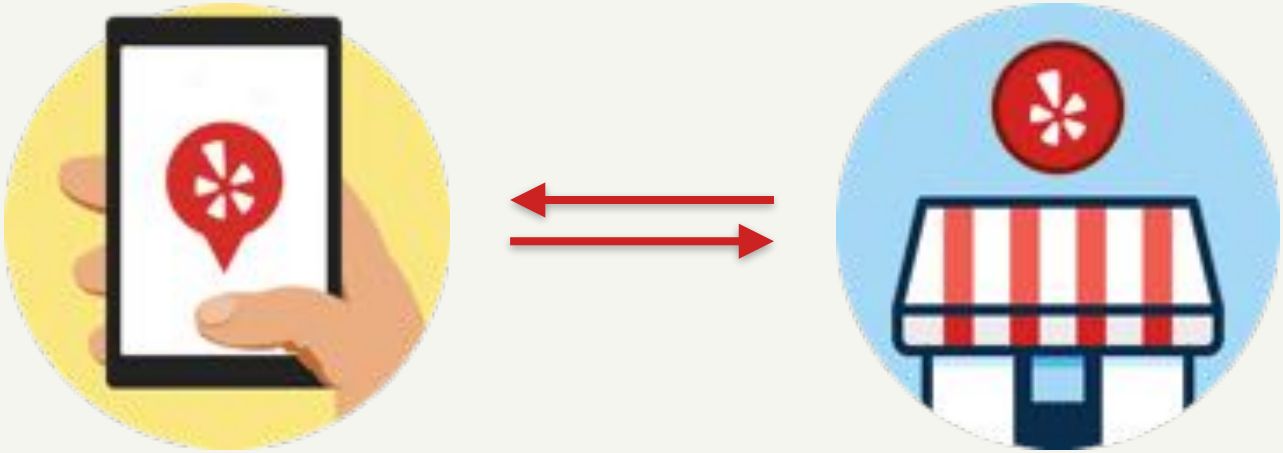
Building UI Consistent Android Apps

Nicola Corti
@cortinico
nco@yelp.com

yelp
WELCOME
TO YELP

Yelp Mission



Connecting people with great local businesses.

About me

Nicola Corti

Android Dev

nco@yelp.com

@cortinico

🇮🇹🍕🕹️🎤📷✈️🏞️

What is Consistency?

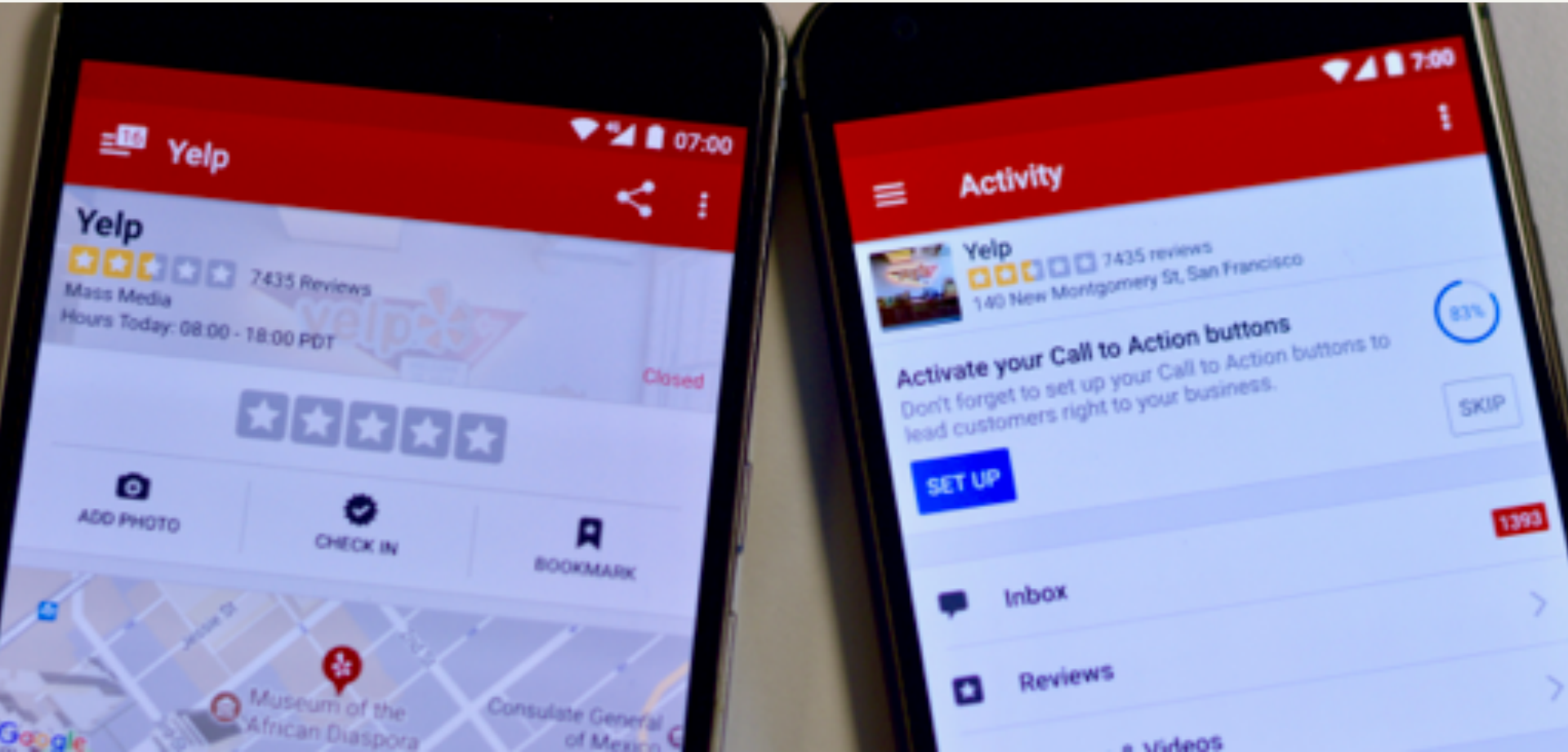
Unified use of Design Elements, such as color, typography, spatial layout and behaviors.

Internal Consistency

Visual

Functional

External Consistency - Across **Product**



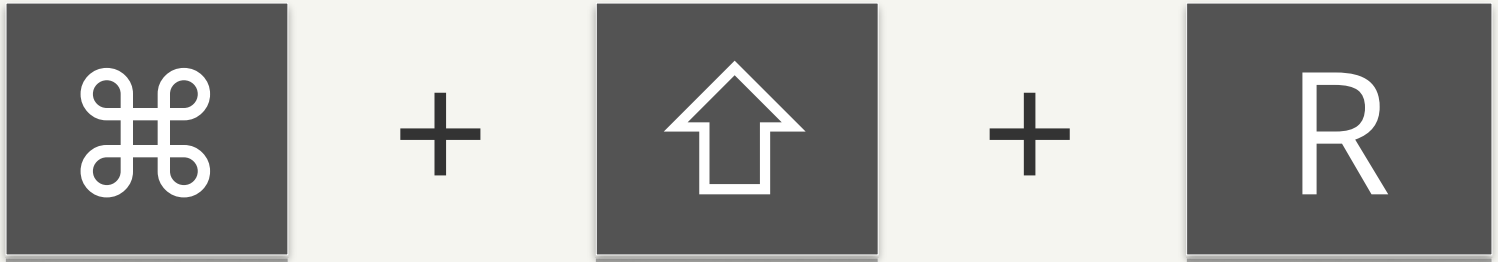
External Consistency - Across **Platform**

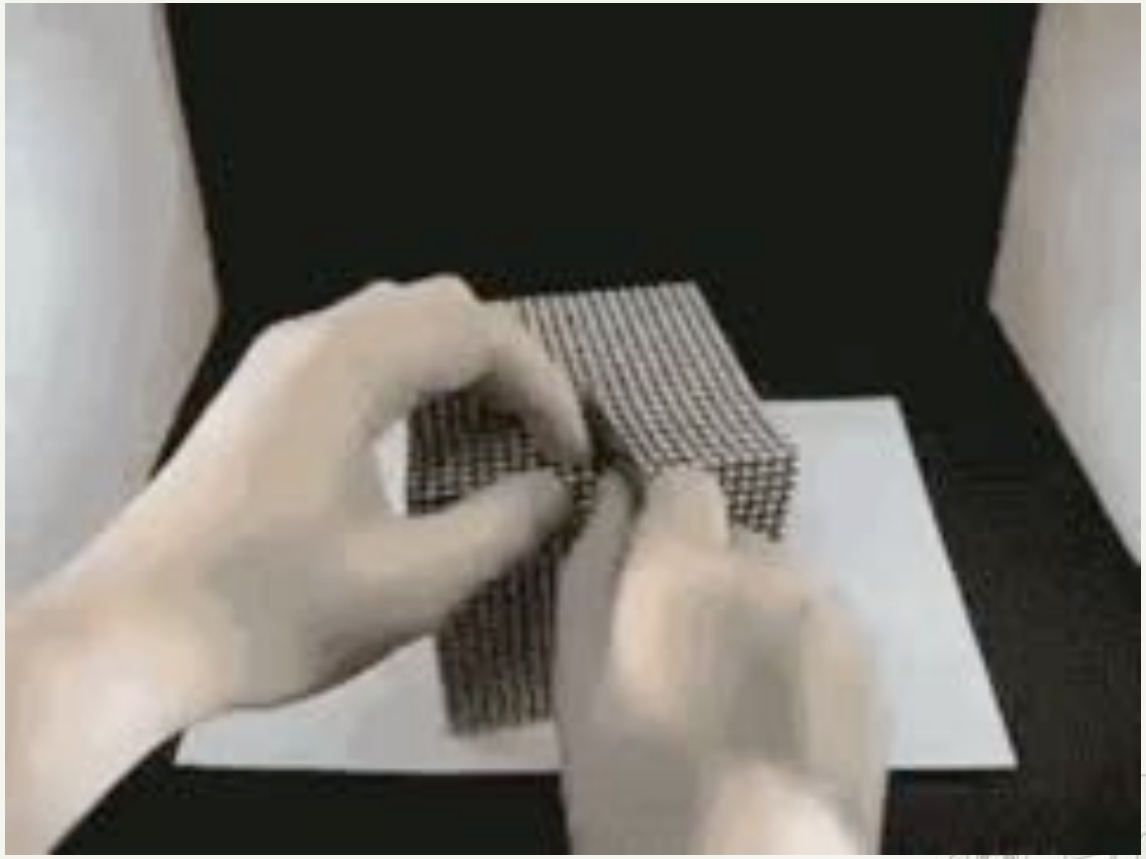
Benefits

- **Learnability**
- Reduce **frustrations**
- Save **money/time** 💰

Photo by davelawler/CC BY - NC

How to tackle Consistency?


+
+
R

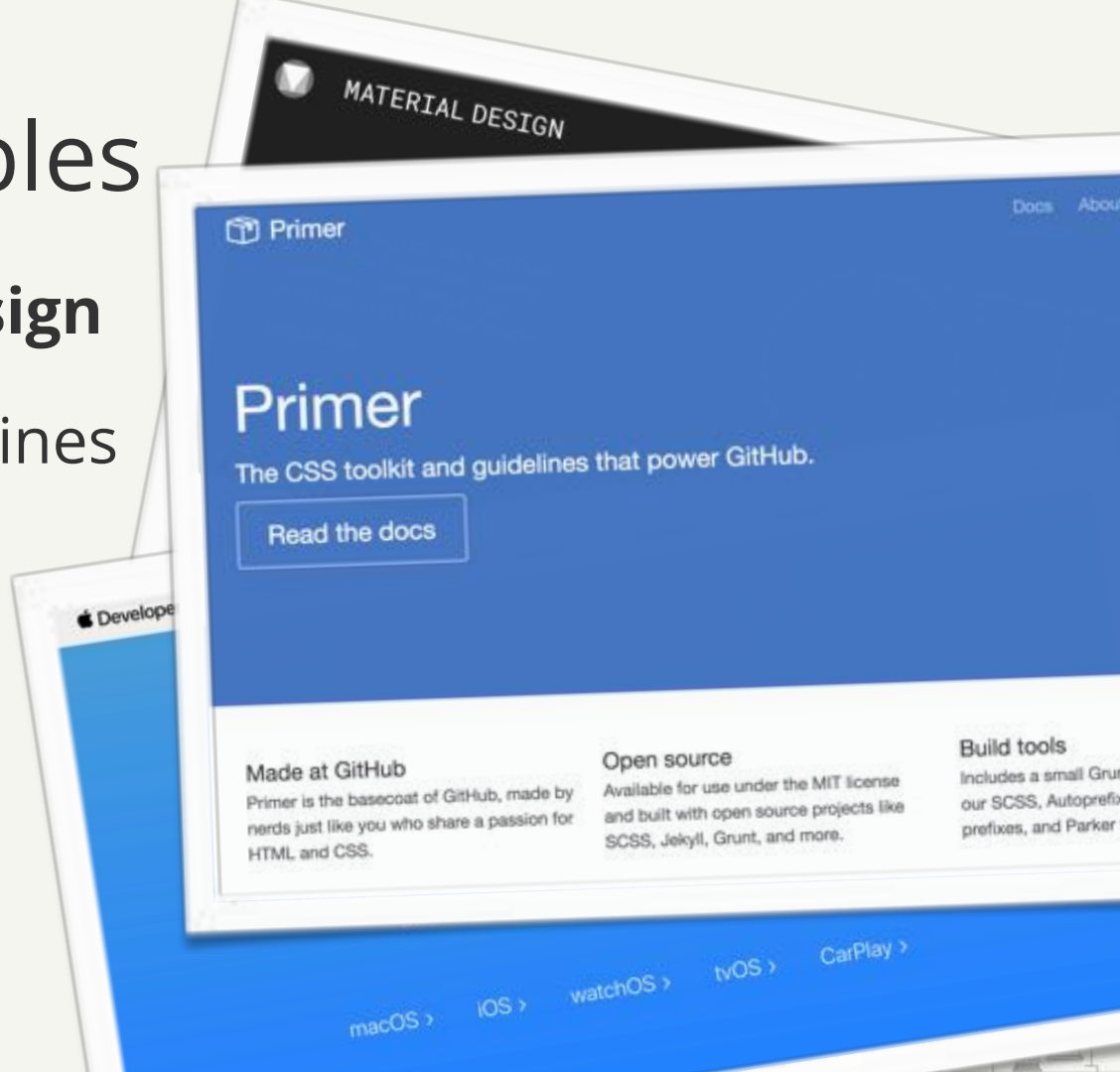


Source: GIPHY

Users
Colors
UX
Scenario
Metrics
Apps
Processes
Experience
Palette
Guides
Palette
Principles
Defining
Android
Components
Style
Easily
Voice
Icons
Tone
Mobile
iOS
Brand
Accessibility
Guide
Ensuring
UI
Design
Visual
Iterations
Grid
Teams
Assets
Web

External Examples

- Google **Material Design**
- **Apple Design** Guidelines
- Github **Primer**



http://styleguides.io/

github.com/alexpate/awesome-design-systems

[illegible]

> A design system is a collection of documentation on principles and best practices that helps guide a team in their digital presence. They are often confused for UI libraries and pattern libraries, but can expand to include guides on other areas such as voice and tone.

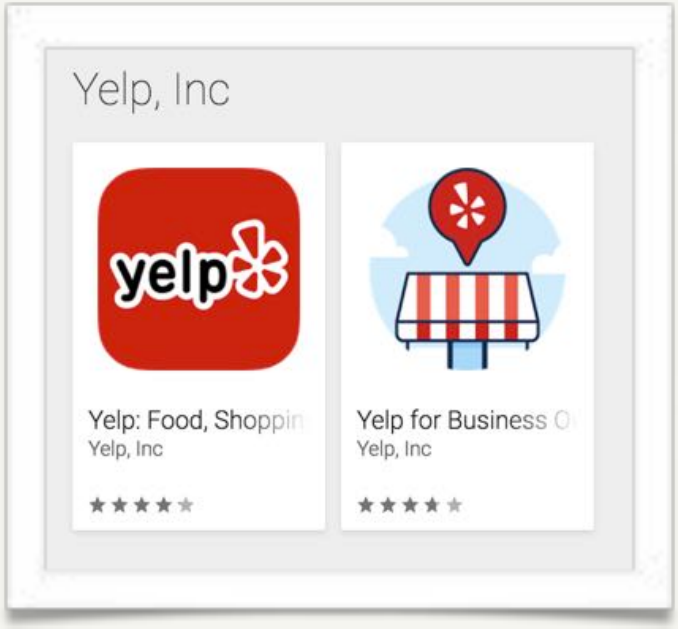
Tags

- Components: Contains coded patterns and examples
- Voice & Tone: Provides guidance on how language should be used
- Designers Kit: Includes a sketch/Photoshop/Figma kit file for designers
- Storybook: Provides components with relevant stories for React Storybook

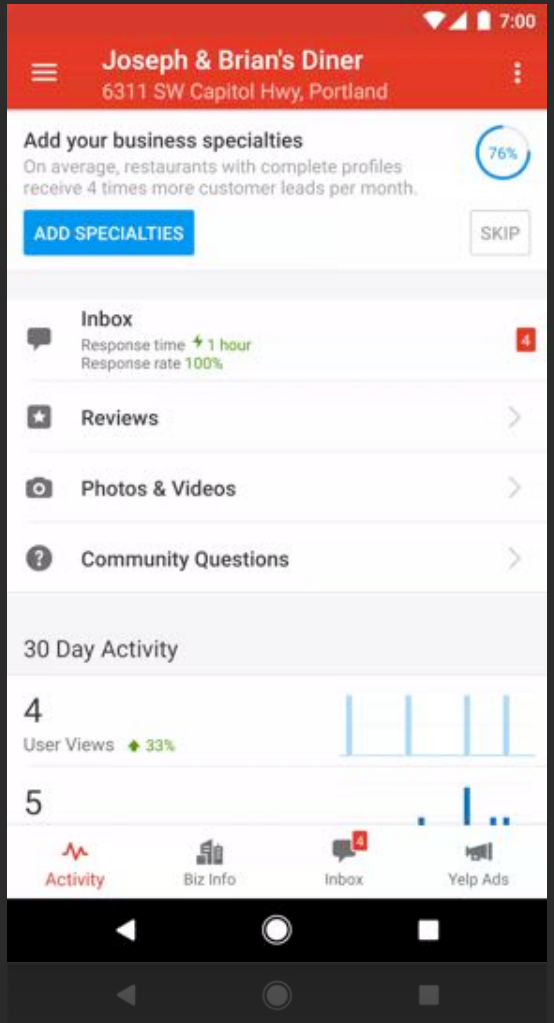
	Components	Voice & Tone	Designers Kit	Storybook
[illegible]	✓	✓	✓	✓
[illegible]			✓	
[illegible]	✓	✓	✓	✓
[illegible]	✓		✓	
[illegible]	✓	✓	✓	
[illegible]	✓		✓	
[illegible]	✓		✓	
[illegible]	✓			
[illegible]	✓	✓	✓	
[illegible]	✓			
[illegible]	✓			
[illegible]	✓	✓		
[illegible]	✓	✓		
[illegible]	✓			
[illegible]	✓			

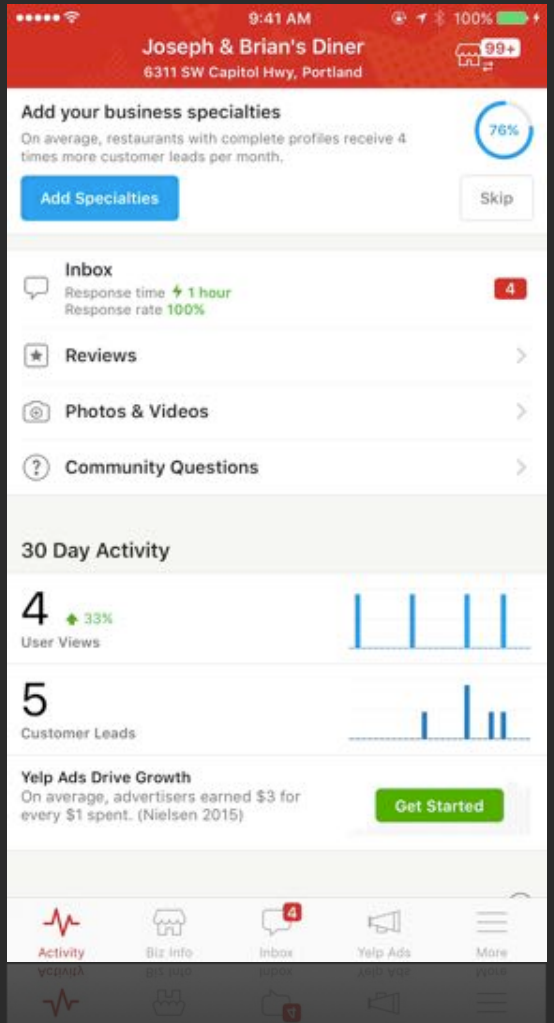
Consistency @Yelp

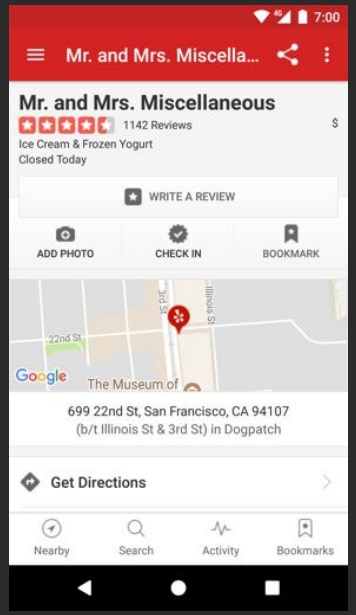
Mobile Apps

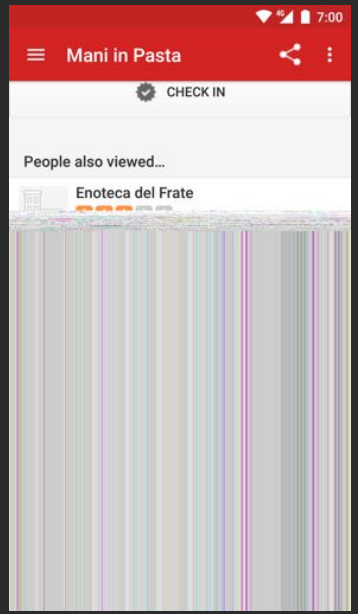

Yelp, Inc
Yelp: Food, Shoppin
Yelp, Inc
Yelp for Business O
Yelp, Inc

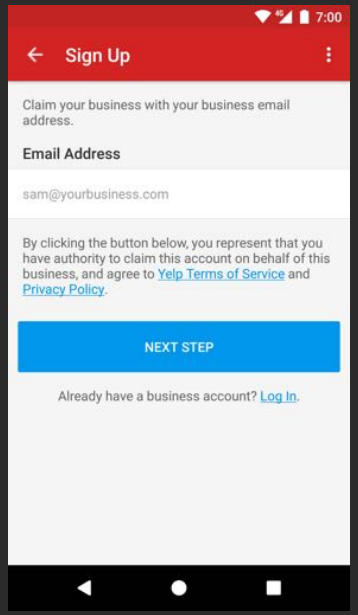
iPhone Apps
1.
Yelp – Nearby Restaurant...
View in iTunes
2.
Yelp for Business Owners
View in iTunes

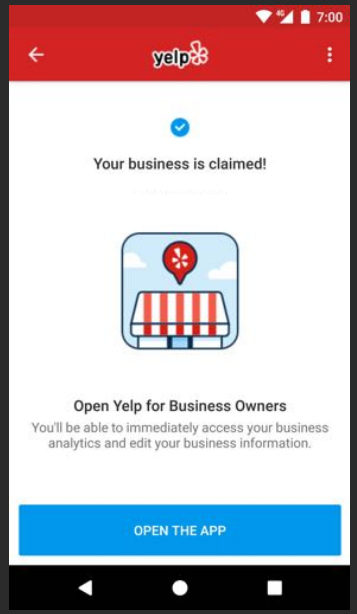

7:00
Joseph & Brian's Diner
6311 SW Capitol Hwy, Portland
Add your business specialties
On average, restaurants with complete profiles receive 4 times more customer leads per month.
76%
ADD SPECIALTIES
SKIP
Inbox
Response time 1 hour
Response rate 100%
4
Reviews
Photos & Videos
Community Questions
30 Day Activity
4
User Views 33%
5
Activity
Biz Info
Inbox
Yelp Ads

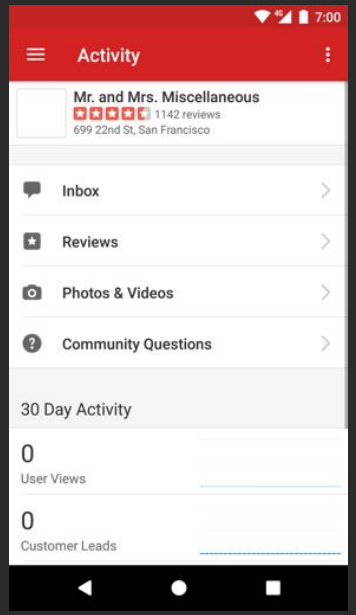

9:41 AM
100%
Joseph & Brian's Diner
6311 SW Capitol Hwy, Portland
99+
Add your business specialties
On average, restaurants with complete profiles receive 4 times more customer leads per month.
76%
Add Specialties
Skip
Inbox
Response time 1 hour
Response rate 100%
4
Reviews
Photos & Videos
Community Questions
30 Day Activity
4
33%
User Views
5
Customer Leads
Yelp Ads Drive Growth
On average, advertisers earned $3 for every $1 spent. (Nielsen 2015)
Get Started
Activity
Biz Info
Inbox
Yelp Ads
More


7:00
Mr. and Mrs. Miscella...
Mr. and Mrs. Miscellaneous
1142 Reviews
$
Ice Cream & Frozen Yogurt
Closed Today
WRITE A REVIEW
ADD PHOTO
CHECK IN
BOOKMARK
3rd St
Illinois St
22nd St
Google
The Museum of
699 22nd St, San Francisco, CA 94107
(b/t Illinois St & 3rd St) in Dogpatch
Get Directions
Nearby
Search
Activity
Bookmarks

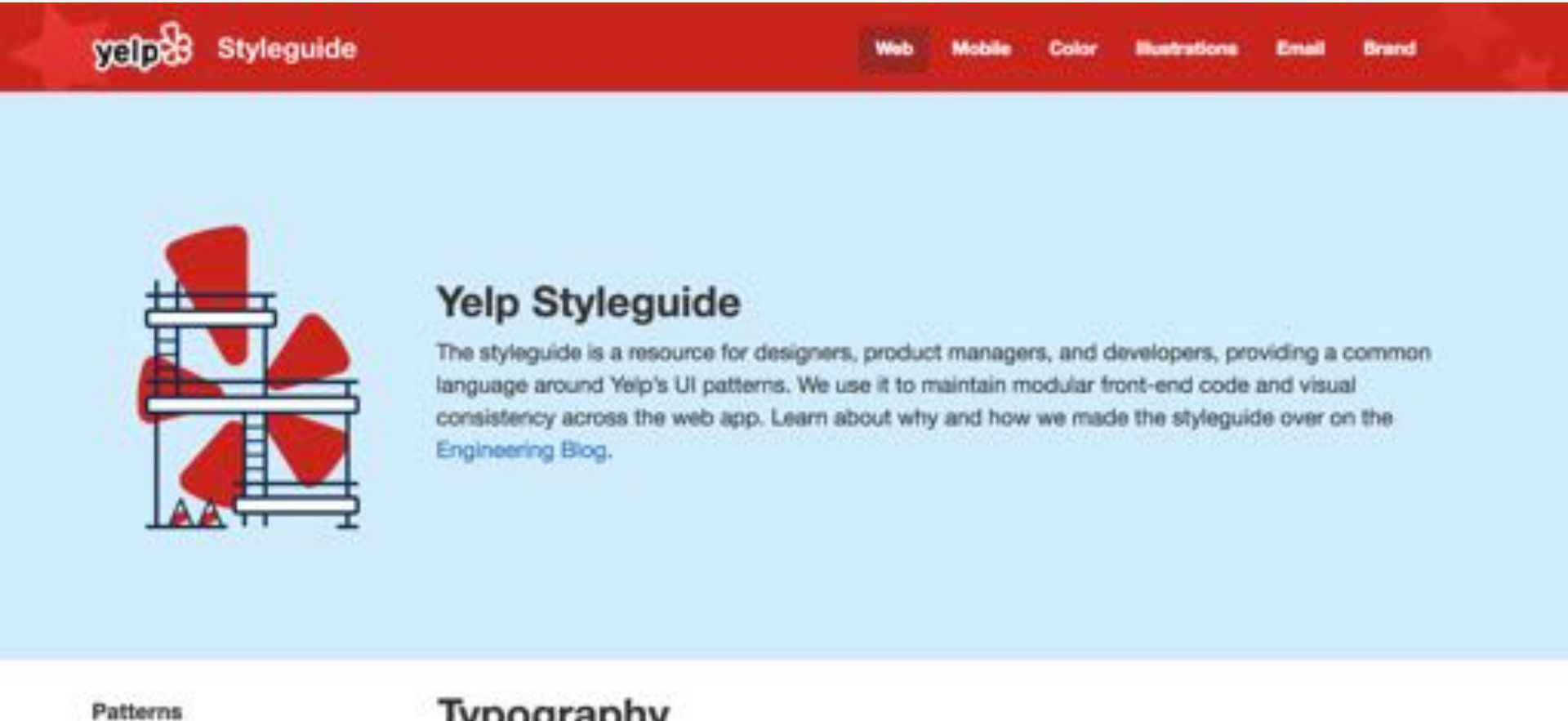

7:00
Mani in Pasta
CHECK IN
People also viewed...
Enoteca del Frate


7:00
Sign Up
Claim your business with your business email address.
Email Address
sam@yourbusiness.com
By clicking the button below, you represent that you have authority to claim this account on behalf of this business, and agree to Yelp Terms of Service and Privacy Policy.
NEXT STEP
Already have a business account? Log In.

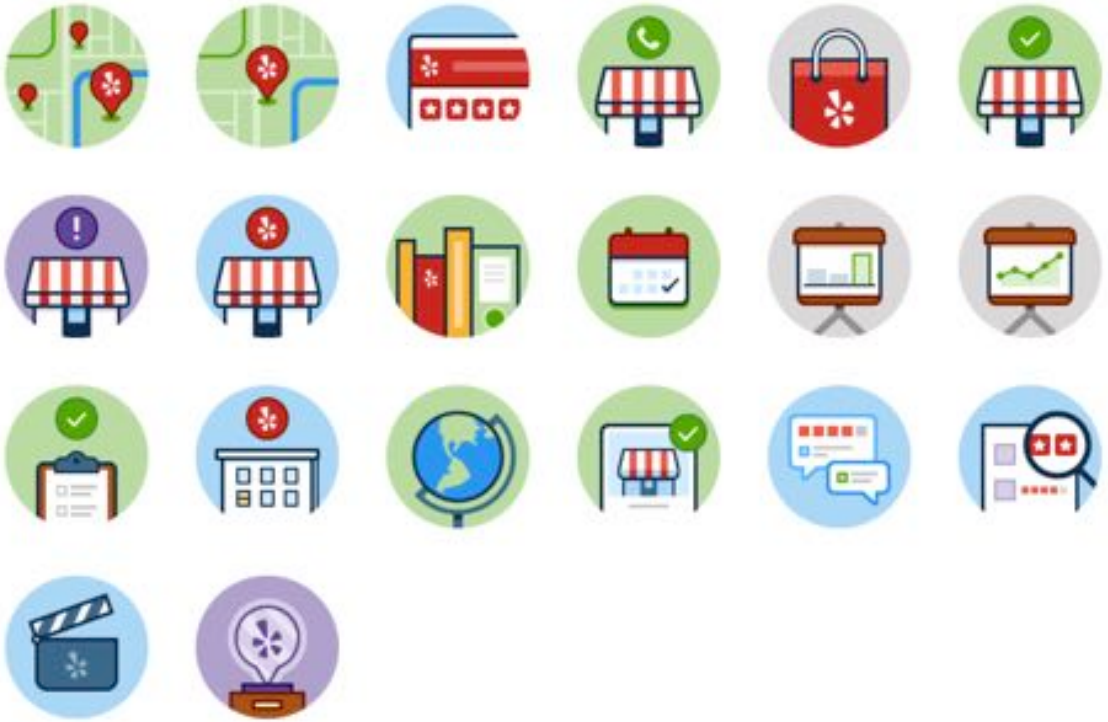

7:00
yelp
Your business is claimed!
Open Yelp for Business Owners
You'll be able to immediately access your business analytics and edit your business information.
OPEN THE APP


7:00
Activity
Mr. and Mrs. Miscellaneous
1142 reviews
699 22nd St, San Francisco
Inbox
Reviews
Photos & Videos
Community Questions
30 Day Activity
0
User Views
0
Customer Leads

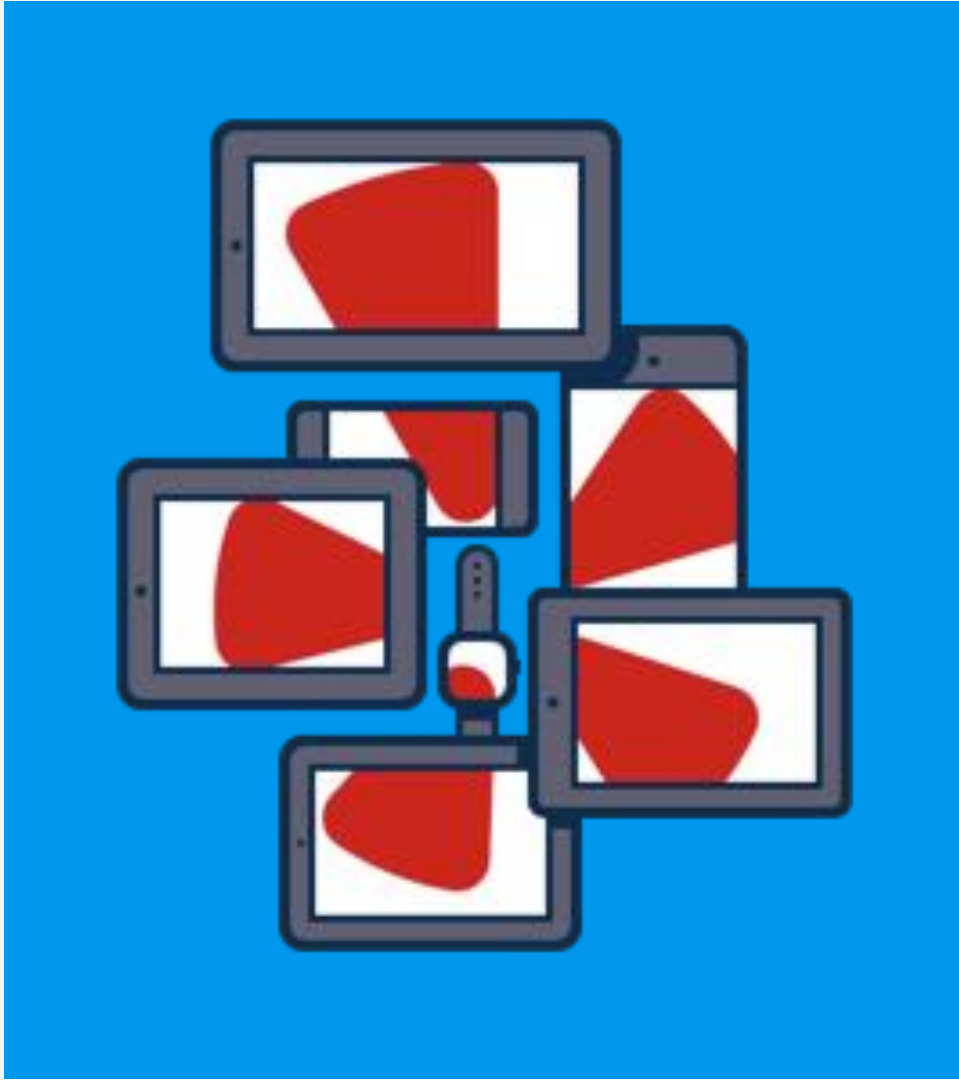
yelp.com/styleguide



Red-Dark
D32323
Red-Regular
F15C4F
Red-Light
F8ADA8
Red-Extra-Light
FCD6D3
Black-Dark
050505
Black-Regular
333333
Black-Light
4D4D4D
Black-Extra-Light
666666
Gray-Dark
999999
Gray-Regular
CCCCCC
Gray-Light
E6E6E6
Gray-Extra-Light
F5F5F5
Green-Dark
308941
Green-Regular
41A700
Green-Light
B6DBA6
Green-Extra-Light
DAECD2
Yellow-Dark
FEC011
Yellow-Regular
FEDD30
Yellow-Light
FFEE98
Yellow-Extra-Light
FFF7CC
Orange-Dark
F15C00
Orange-Regular
FF9242
Orange-Light
FFD8A0
Orange-Extra-Light
FFEBCF
Mocha-Dark
791900
Mocha-Regular
B15118
Mocha-Light
F1BD79
Mocha-Extra-Light
F8E3C7
Blue-Dark
0073BB
Blue-Regular
0097EC
Blue-Light
A5D8F5
Blue-Extra-Light
D0ECFB
Slate-Dark
092D4B
Slate-Regular
336A8B
Slate-Light
9AB5C5
Slate-Extra-Light
CDDAE2
Purple-Dark
281F58
Purple-Regular
6C3EB2
Purple-Light
B2A2CB
Purple-Extra-Light
DAD1E4



The Android Styleguide Library



Design

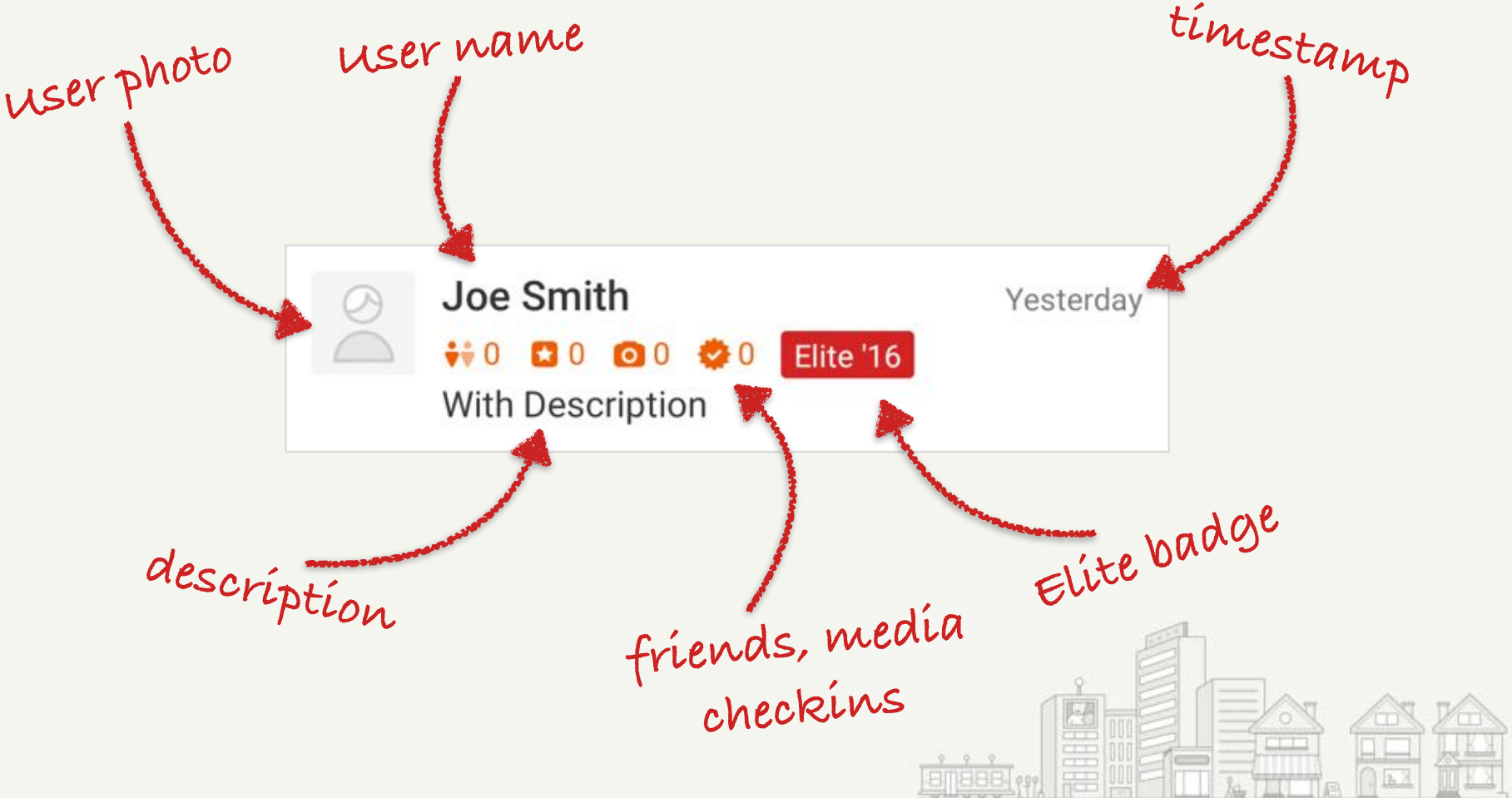
Build

Share

Design

Does it fit?

- Can be **reused**?
- Is it **visible to the user**?
- Development **plan**?


user photo
user name
timestamp
Joe Smith
Yesterday
0 0 0 0
Elite '16
With Description
description
friends, media
checkins
Elite badge

Attributes

```
<resources>
</resources>
```

Attributes

```
<resources>
    <declare-styleable name="UserPassport">
    </declare-styleable>
</resources>
```

Attributes

```
<resources>
    <declare-styleable name="UserPassport">
        <!-- Determines user's name -->
        <attr name="userPassportName" format="string"/>
    </declare-styleable>
</resources>
```

Attributes

```
<resources>
    <declare-styleable name="UserPassport">
        <!-- Determines user's name -->
        <attr name="userPassportName" format="string"/>

        <!-- Determines user's description/role -->
        <attr name="userPassportDescription" format="string"/>
    </declare-styleable>
</resources>
```

Attributes

```
<resources>
    <declare-styleable name="UserPassport">
        <!-- Determines user's name -->
        <attr name="userPassportName" format="string"/>

        <!-- Determines user's description/role -->
        <attr name="userPassportDescription" format="string"/>
    </declare-styleable>
    <attr name="userPassportStyle" format="reference"/>
</resources>
```

Theme

Theme

```
<resources>
    <!-- This theme is the parent of all themes of Yelp's android apps. -->
    <style name="YelpStyleguideTheme"/>
</resources>
```

Theme

```
<resources>
    <!-- This theme is the parent of all themes of Yelp's android apps. -->
    <style name="YelpStyleguideTheme" parent="Theme.AppCompat.Light.DarkActionBar"/>
</resources>
```

Theme

```
<resources>
    <!-- This theme is the parent of all themes of Yelp's android apps. -->
    <style name="YelpStyleguideTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <item name="userPassportStyle">@style/UserPassport</item>
    </style>
</resources>
```

Styles

Styles

```
<style name=“UserPassport"/>
```

Styles

```
<style name=“UserPassport">
    <item name="userPassportName">Joe Smith</item>
</style>
```

Styles

```
<style name=“UserPassport">
    <item name="userPassportName">Joe Smith</item>
    <item name="userPassportDescription">Owner of Sample Business</item>
</style>
```

UserPassport

```
public class UserPassport extends RelativeLayout {
```

```
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;
```

```
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;

    public void setName(String name) {
        mUserName.setText(name);
    }
```

```
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;

    public void setName(String name) {
        mUserName.setText(name);
    }

    public void setDescription(String description) {
        if (TextUtils.isEmpty(description)) {
            mDescription.setVisibility(GONE);
        } else {
            mDescription.setVisibility(VISIBLE);
            mDescription.setText(description);
        }
    }
```

```
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;
```

```java
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;

    public UserPassport(final Context context) {
        super(context);
        init(context, null, 0);
    }

    public UserPassport(final Context context, final AttributeSet attrs) {
        super(context, attrs);
        init(context, attrs, R.attr.userPassportStyle);
    }

    public UserPassport(final Context context, final AttributeSet attrs,
final int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        init(context, attrs, defStyleAttr);
    }
```

```java
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;

    public UserPassport(final Context context) {
        super(context);
        init(context, null, 0);
    }

    public UserPassport(final Context context, final AttributeSet attrs) {
        super(context, attrs);
        init(context, attrs, R.attr.userPassportStyle);
    }

    public UserPassport(final Context context, final AttributeSet attrs,
final int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        init(context, attrs, defStyleAttr);
    }
```

```
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;

    private void init(
           final Context context, final AttributeSet attrs, final int defStyleAttr) {

    }
```

```
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;

    private void init(
           final Context context, final AttributeSet attrs, final int defStyleAttr) {

       LayoutInflater.from(context).inflate(R.layout.user_passport, this, true);

    }
```

```
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;

    private void init(
           final Context context, final AttributeSet attrs, final int defStyleAttr) {

       LayoutInflater.from(context).inflate(R.layout.user_passport, this, true);

       mUserName = (TextView) findViewById(R.id.user_name);
       mDescription = (TextView) findViewById(R.id.description);

    }
```

```
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;

    private void init(
           final Context context, final AttributeSet attrs, final int defStyleAttr) {

       LayoutInflater.from(context).inflate(R.layout.user_passport, this, true);

       mUserName = (TextView) findViewById(R.id.user_name);
       mDescription = (TextView) findViewById(R.id.description);

       final TypedArray styles =  context.obtainStyledAttributes(attrs,
                    R.styleable.UserPassport, defStyleAttr, 0);

    }
```

```java
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;

    private void init(
           final Context context, final AttributeSet attrs, final int defStyleAttr) {

       LayoutInflater.from(context).inflate(R.layout.user_passport, this, true);

       mUserName = (TextView) findViewById(R.id.user_name);
       mDescription = (TextView) findViewById(R.id.description);

       final TypedArray styles =  context.obtainStyledAttributes(attrs,
                    R.styleable.UserPassport, defStyleAttr, 0);

       setName(styles.getString(R.styleable.UserPassport_userPassportName));
       setDescription(styles.getString(
                               R.styleable.UserPassport_userPassportDescription));

    }
```

```java
public class UserPassport extends RelativeLayout {

    private TextView mUserName;
    private TextView mDescription;

    private void init(
           final Context context, final AttributeSet attrs, final int defStyleAttr) {

       LayoutInflater.from(context).inflate(R.layout.user_passport, this, true);

       mUserName = (TextView) findViewById(R.id.user_name);
       mDescription = (TextView) findViewById(R.id.description);

       final TypedArray styles =  context.obtainStyledAttributes(attrs,
                    R.styleable.UserPassport, defStyleAttr, 0);

       setName(styles.getString(R.styleable.UserPassport_userPassportName));
       setDescription(styles.getString(
                               R.styleable.UserPassport_userPassportDescription));

       styles.recycle();
    }
```

```
<style name="UserPassport">
    <item name="userPassportName">Joe Smith</item>
    <item name="userPassportDescription">@null</item>
    <item name="userPassportTint">@color/orange_dark_interface</item>
    <item name="userPassportNameColor">@color/black_regular_interface</item>
    <item name="userPassportSize">Regular</item>
    <item name="userPassportShowName">true</item>
    <item name="userPassportShowIcons">true</item>
    <item name="userPassportEliteYear">-1</item>
    <item name="userPassportFriends">0</item>
    <item name="userPassportReviews">0</item>
    <item name="userPassportPhotos">0</item>
    <item name="userPassportCheckIns">0</item>
    <item name="userPassportShowCheckIn">false</item>
</style>
```

```
<style name="UserPassport.White">
    <item name="userPassportName">Joe Smith</item>
    <item name="userPassportDescription">@null</item>
    <item name="userPassportTint">@color/orange_dark_interface</item>
    <item name="userPassportNameColor">@color/black_regular_interface</item>
    <item name="userPassportSize">Regular</item>
    <item name="userPassportShowName">true</item>
    <item name="userPassportShowIcons">true</item>
    <item name="userPassportEliteYear">-1</item>
    <item name="userPassportFriends">0</item>
    <item name="userPassportReviews">0</item>
    <item name="userPassportPhotos">0</item>
    <item name="userPassportCheckIns">0</item>
    <item name="userPassportShowCheckIn">false</item>
</style>
```

```
<style name="UserPassport.White">
    <item name="userPassportName">Joe Smith</item>
    <item name="userPassportDescription">@null</item>
    <item name="userPassportTint">@color/white_interface</item>
    <item name="userPassportNameColor">@color/white_interface</item>
    <item name="userPassportSize">Regular</item>
    <item name="userPassportShowName">true</item>
    <item name="userPassportShowIcons">true</item>
    <item name="userPassportEliteYear">-1</item>
    <item name="userPassportFriends">0</item>
    <item name="userPassportReviews">0</item>
    <item name="userPassportPhotos">0</item>
    <item name="userPassportCheckIns">0</item>
    <item name="userPassportShowCheckIn">false</item>
</style>
```

```
<style name="UserPassport.White">
    <item name="userPassportTint">@color/white_interface</item>
    <item name="userPassportNameColor">@color/white_interface</item>
</style>
```

isInEditMode()

isInEditMode()

```
public void setMediaCount(int photo, int video, int media) {

}
```

isInEditMode()

```
public void setMediaCount(int photo, int video, int media) {
    mMediaCount.setText(String.valueOf(media));
}
```

isInEditMode()

```
public void setMediaCount(int photo, int video, int media) {
    mMediaCount.setText(String.valueOf(media));
    Resources resources = getResources();
    String contentDesc;
}
```

isInEditMode()

```
public void setMediaCount(int photo, int video, int media) {
    mMediaCount.setText(String.valueOf(media));
    Resources resources = getResources();
    String contentDesc;
    if (video == 0) {
        contentDesc = resources.getString(R.string.photo_count, photo);
    } else if (photo == 0 && video > 0) {
        contentDesc = resources.getString(R.string.video_count, video);
    } else {
        contentDesc = resources.getString(R.string.photo_and_video_count, media);
    }
}
```

isInEditMode()

```
public void setMediaCount(int photo, int video, int media) {
    mMediaCount.setText(String.valueOf(media));
    Resources resources = getResources();
    String contentDesc;
    if (video == 0) {
        contentDesc = resources.getString(R.string.photo_count, photo);
    } else if (photo == 0 && video > 0) {
        contentDesc = resources.getString(R.string.video_count, video);
    } else {
        contentDesc = resources.getString(R.string.photo_and_video_count, media);
    }
    mMediaCount.setContentDescription(contentDesc);
}
```

isInEditMode()

```
public void setMediaCount(int photo, int video, int media) {
    mMediaCount.setText(String.valueOf(media));
    if (!isInEditMode()) {
        Resources resources = getResources();
        String contentDesc;
        if (video == 0) {
            contentDesc = resources.getString(R.string.photo_count, photo);
        } else if (photo == 0 && video > 0) {
            contentDesc = resources.getString(R.string.video_count, video);
        } else {
            contentDesc = resources.getString(R.string.photo_and_video_count, media);
        }
        mMediaCount.setContentDescription(contentDesc);
    }
}
```

.Swift
.XML
.CSS

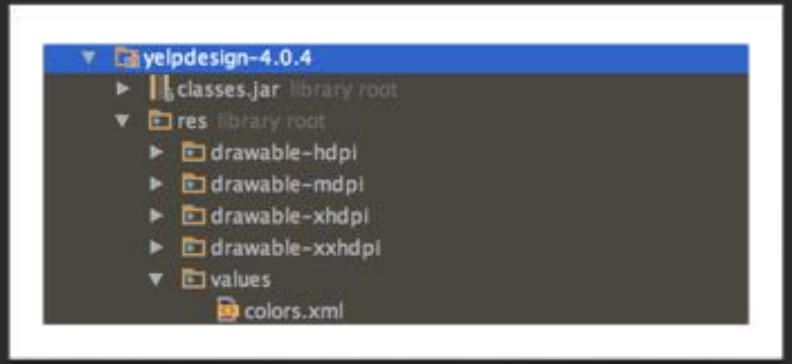
.SVG

Assets

```
dependencies {

    // Yelp asset libs
    compile 'com.yelp:yelpicons:135.0.0'
    compile ‘com.yelp:yelpdesign:4.0.4’

}
```

Assets

```
dependencies {

    // Yelp asset libs
    compile 'com.yelp:yelpicons:135.0.0'
    compile 'com.yelp:yelpdesign:4.0.4'

}
```



Color

Color

```
<color name="black_extra_light_interface">#666666</color>
<color name="black_regular_interface">#333333</color>
<color name="blue_dark_interface">#0073bb</color>
<color name="blue_extra_light_interface">#d0ecfb</color>
<color name="blue_regular_interface">#0097ec</color>
<color name="gray_dark_interface">#999999</color>
<color name="gray_extra_light_interface">#f5f5f5</color>
<color name="gray_light_interface">#e6e6e6</color>
<color name="gray_regular_interface">#cccccc</color>
<color name="green_extra_light_interface">#daecd2</color>
<color name="green_regular_interface">#41a700</color>
<color name="mocha_extra_light_interface">#f8e3c7</color>
<color name="mocha_light_interface">#f1bd79</color>
<color name="orange_dark_interface">#f15c00</color>
<color name="orange_extra_light_interface">#ffebcf</color>
<color name="purple_extra_light_interface">#dad1e4</color>
<color name="red_dark_interface">#d32323</color>
<color name="red_extra_light_interface">#fcd6d3</color>
<color name="slate_extra_light_interface">#cddae2</color>
<color name="white_interface">#ffffff</color>
<color name="yellow_dark_interface">#fec011</color>
<color name="yellow_extra_light_interface">#fff7cc</color>
```

Color

```
<color name="black_extra_light_interface">#666666</color>
<color name="black_regular_interface">#333333</color>
<color name="blue_dark_interface">#0073bb</color>
<color name="blue_extra_light_interface">#d0ecfb</color>
<color name="blue_regular_interface">#0097ec</color>
<color name="gray_dark_interface">#999999</color>
<color name="gray_extra_light_interface">#f5f5f5</color>
<color name="gray_light_interface">#e6e6e6</color>
<color name="gray_regular_interface">#cccccc</color>
<color name="green_extra_light_interface">#daecd2</color>
<color name="green_regular_interface">#41a700</color>
<color name="mocha_extra_light_interface">#f8e3c7</color>
<color name="mocha_light_interface">#f1bd79</color>
<color name="orange_dark_interface">#f15c00</color>
<color name="orange_extra_light_interface">#ffebcf</color>
<color name="purple_extra_light_interface">#dad1e4</color>
<color name="red_dark_interface">#d32323</color>
<color name="red_extra_light_interface">#fcd6d3</color>
<color name="slate_extra_light_interface">#cddae2</color>
<color name="white_interface">#ffffff</color>
<color name="yellow_dark_interface">#fec011</color>
<color name="yellow_extra_light_interface">#fff7cc</color>
```

Color

```
<color name="black_extra_light_interface">#666666</color>
<color name="black_regular_interface">#333333</color>
<color name="blue_dark_interface">#0073bb</color>
<color name="blue_extra_light_interface">#d0ecfb</color>
<color name="blue_regular_interface">#0097ec</color>
<color name="gray_dark_interface">#999999</color>
<color name="gray_extra_light_interface">#f5f5f5</color>
<color name="gray_light_interface">#e6e6e6</color>
<color name="gray_regular_interface">#cccccc</color>
<color name="green_extra_light_interface">#daecd2</color>
<color name="green_regular_interface">#41a700</color>
<color name="mocha_extra_light_interface">#f8e3c7</color>
<color name="mocha_light_interface">#f1bd79</color>
<color name="orange_dark_interface">#f15c00</color>
<color name="orange_extra_light_interface">#ffebcf</color>
<color name="purple_extra_light_interface">#dad1e4</color>
<color name="red_dark_interface">#d32323</color>
<color name="red_extra_light_interface">#fcd6d3</color>
<color name="slate_extra_light_interface">#cddae2</color>
<color name="white_interface">#ffffff</color>
<color name="yellow_dark_interface">#fec011</color>
<color name="yellow_extra_light_interface">#fff7cc</color>
```

Build

Summary: **[android-styleguide] Adding a new Ripple background to FlatButtons**

Review Request #124540 - Created and updated 11 minutes ago

Description

```
Adding a fancy ripple background
```

Testing Done

```
After any change, please run `docs/create_doc.sh` and check that the documentation in
docs/build/html is updated and functional.

Adding a module:
  Did you include screenshots of your module within docs/WidgetName/?
  Did you create any tests?
  Did you add the component into styleguidetestapp?
  Did you make sure it was previewable?

Modifying a module:
  Did you update the screenshots in docs/WidgetName?
  Did you run any tests?
  Did you update the component into styleguidetestapp?
  BizApp Buildbot Run:
  ConsumerApp Buildbot Run:
```

VCS & CI

- git **submodule**
- Run the **Build** for
  - **submodule**
  - **consumer app**
  - **business app**

Custom Lint Checks

Custom Lint Checks

```
Button b = new Button(context);

SwitchCompat switchCompat = new SwitchCompat(context);

Snackbar.make(getRootView(), “Test”, LENGTH_SHORT).show();
```

Custom Lint Checks

```
@SuppressLint("")
Button b = new Button(context);

@SuppressLint("")
SwitchCompat switchCompat = new SwitchCompat(context);

@SuppressLint("")
Snackbar.make(getRootView(), “Test”, LENGTH_SHORT).show();
```

Custom Lint Checks

```
@SuppressLint("NonStyleguideButtonInstance")
Button b = new Button(context);

@SuppressLint("")
SwitchCompat switchCompat = new SwitchCompat(context);

@SuppressLint("")
Snackbar.make(getRootView(), “Test”, LENGTH_SHORT).show();
```

Custom Lint Checks

```
@SuppressLint("NonStyleguideButtonInstance")
Button b = new Button(context);

@SuppressLint("NonStyleguideToggleInstance")
SwitchCompat switchCompat = new SwitchCompat(context);

@SuppressLint("")
Snackbar.make(getRootView(), “Test”, LENGTH_SHORT).show();
```

Custom Lint Checks

```
@SuppressLint("NonStyleguideButtonInstance")
Button b = new Button(context);

@SuppressLint("NonStyleguideToggleInstance")
SwitchCompat switchCompat = new SwitchCompat(context);

@SuppressLint("NonStyleguideSnackbarInstance")
Snackbar.make(getRootView(), “Test”, LENGTH_SHORT).show();
```

Custom Lint Checks

```
// Using stock Button because […] + Ticket number.
@SuppressLint("NonStyleguideButtonInstance")
Button b = new Button(context);

@SuppressLint("NonStyleguideToggleInstance")
SwitchCompat switchCompat = new SwitchCompat(context);

@SuppressLint("NonStyleguideSnackbarInstance")
Snackbar.make(getRootView(), “Test”, LENGTH_SHORT).show();
```

Custom Lint Checks

Custom Lint Checks

```
<Button
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

<Switch
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

Custom Lint Checks

```
<Button
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:ignore="NonStyleguideButtonTag" />

<Switch
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:ignore="NonStyleguideToggleTag" />
```

build.gradle

build.gradle

```
android {
    lintOptions {

    }
}
```

build.gradle

```
android {
    lintOptions {
        abortOnError true
        warningsAsErrors true

    }
}
```

build.gradle

```
android {
    lintOptions {
        abortOnError true
        warningsAsErrors true

        lintConfig file("lint.xml")
    }
}
```

build.gradle

```
android {
    lintOptions {
        abortOnError true
        warningsAsErrors true

        lintConfig file("lint.xml")
        baseline file("lint-baseline.xml")
    }
}
```

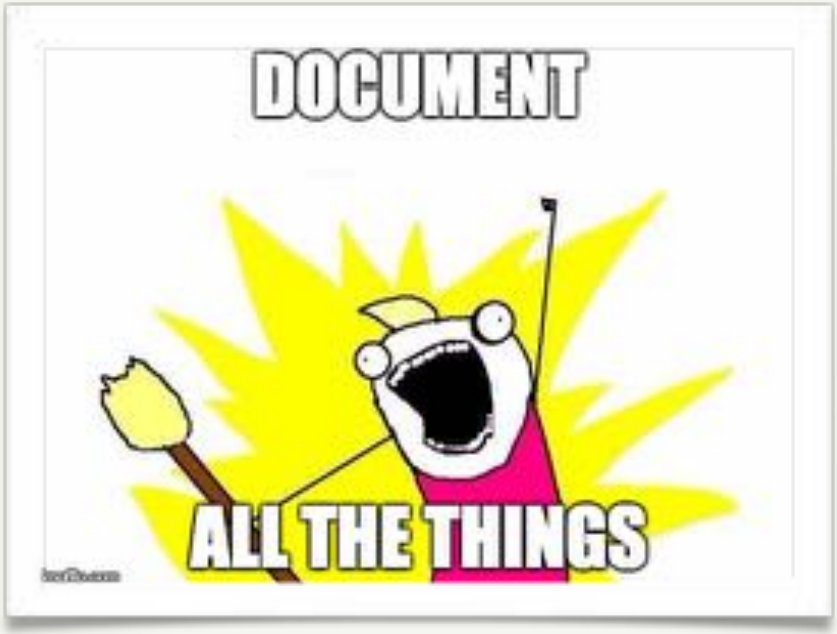
Test your component

- Your component ❤️ **Espresso**?
- Do you handle **state changes**?
- `contentDescription ?`

Share

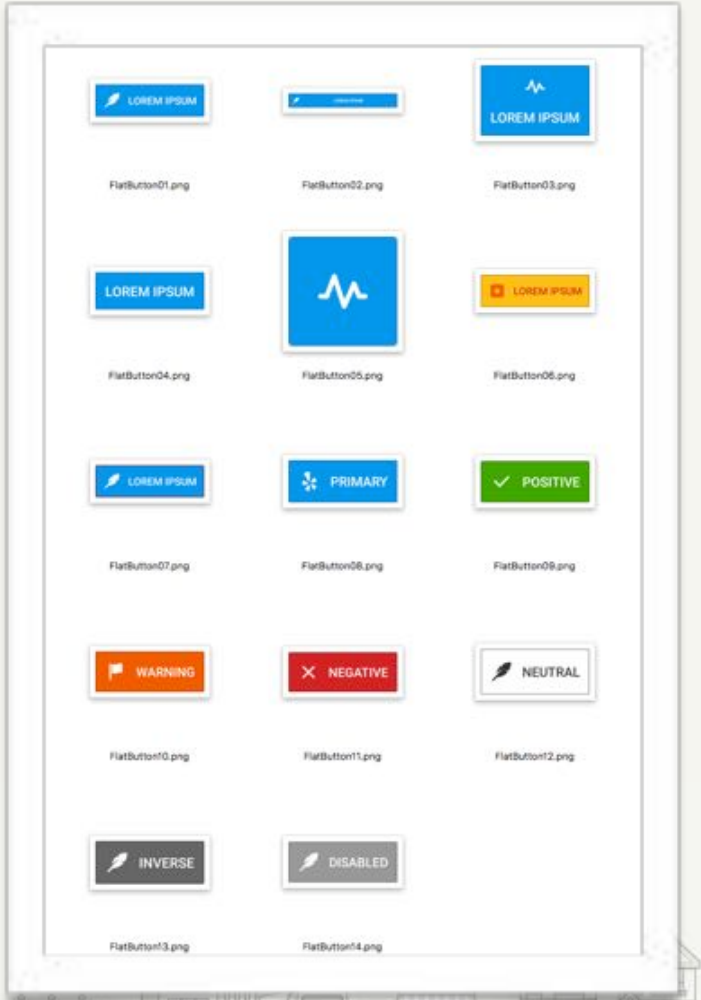
Documentation

- Provide **Javadoc**
- Add **Screenshots**
- Document **Attributes**
- Document **Styles**



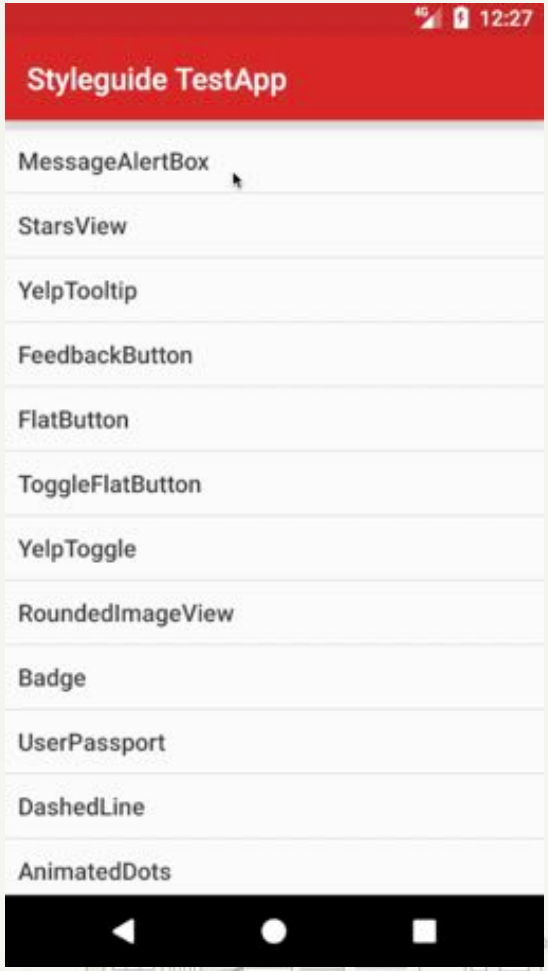
Screenshots capture

- v0.1: **Manual** Screenshots
- v0.2: **Automated** locally
- v0.3: **Automated** with CI 💫



StyleguideTestApp

- Components **Showcase**
- For **Designer** 🎨
- For **Developer** 🔧



Taking Screenshots with Espresso

```
public class ScreenshotViewActions {

}
```

Taking Screenshots with Espresso

```
public class ScreenshotViewActions {

    public static ViewAction screenshot(final String folderName, final String fileName) {
        return new ViewAction() {

        };
    }
}
```

Taking Screenshots with Espresso

```
public class ScreenshotViewActions {

    public static ViewAction screenshot(final String folderName, final String fileName) {
        return new ViewAction() {
            // Other methods omitted.

            @Override
            public void perform(UiController uiController, View view) {
                ScreenshotsUtil.takeScreenshot(folderName, fileName, view);
            }
        };
    }
}
```

Sample Espresso Test

Sample Espresso Test

```
public class StarsViewActivityTests {

}
```

Sample Espresso Test

```
public class StarsViewActivityTests {

    @Test
    public void takeScreenshot() throws InterruptedException {
        onView(withId(R.id.stars_view_4)).perform(setStarsNumber(4));

    }
}
```

Sample Espresso Test

```
public class StarsViewActivityTests {

    @Test
    public void takeScreenshot() throws InterruptedException {
        onView(withId(R.id.stars_view_4)).perform(setStarsNumber(4));
        onView(withId(R.id.stars_view_5)).perform(setStarsNumber(5),
                screenshot(FOLDER_NAME, "stars_with_text"));

    }
}
```

Sample Espresso Test

```
public class StarsViewActivityTests {

    @Test
    public void takeScreenshot() throws InterruptedException {
        onView(withId(R.id.stars_view_4)).perform(setStarsNumber(4));
        onView(withId(R.id.stars_view_5)).perform(setStarsNumber(5),
                screenshot(FOLDER_NAME, "stars_with_text"));

        ScreenshotUtil.fullScreenshot(FOLDER_NAME, "stars_fullscreen");
    }
}
```

Bend it, don’t **break** it!

Source: GIPHY

We are hiring!
www.yelp.com/careers/

@YelpEngineering

github.com/yelp

yelp.com/careers

engineeringblog.yelp.com

Nicola Corti
@cortinico
nco@yelp.com