

Tools for Advanced Prototyping for Mobile Apps

Joseph Kim
Happy Dojo

北京

伦敦

纽约

旧金山

圣保罗

上海

东京

QCon

全球软件开发大会

[上海站]

主办方 **Geekbang** 极客邦科技 **InfoQ**

信息安全

机器学习

人工智能

黑产

互联网金融 (FinTech)

团队管理

云计算

基础设施

软件性能

硅谷

微服务

互联网架构

2017年10月17-19日
上海·宝华万豪酒店

——> 扫描二维码
开启软件开发新思路





Geekbang> | EGO EXTRA GEEKS' ORGANIZATION NETWORKS
极客邦科技

EGO会员招募季

EGO旨在组建全球最具影响力的技术领导者社交网络，联结杰出的技术领导者学习和成长。

2017年6月30-7月10



扫码报名

SPEAKER INTRODUCE

**Joseph
Kim** Mobile Developer at
HappyDojo

Joseph Kim is a mobile developer at HappyDojo based in San Francisco. He focuses on working with startups build mobile apps for iOS and React Native. He loves building prototypes and is passionate about improving the process.

SPEAKER
ArchSummit 2017 ShenZhen

Summary

- Why prototype?
- How prototypes are built now and why they're important
- Introduce ideas and tools we're working on for creative developers to build prototypes in the future
- Explore technical considerations in building prototyping tools

Technical Designer - Designer that invests
in coding skills.

Creative Developer - Developer that invests
in UI/UX skills.

Why prototype?

Problem Scenario

- Your company just spent 6 months with
 - 4 developers
 - 1 designer
- Building an app that users can't understand and hate using

Problem Scenario

- Your company just spent 6 months with
 - 4 developers
 - 1 designer
- Building an app that users can't understand and hate using
- You didn't get the design (UX) right!

Problem Scenario

- Shipping a production app is a significant investment
- If the app flops it's a big loss

Problem Scenario

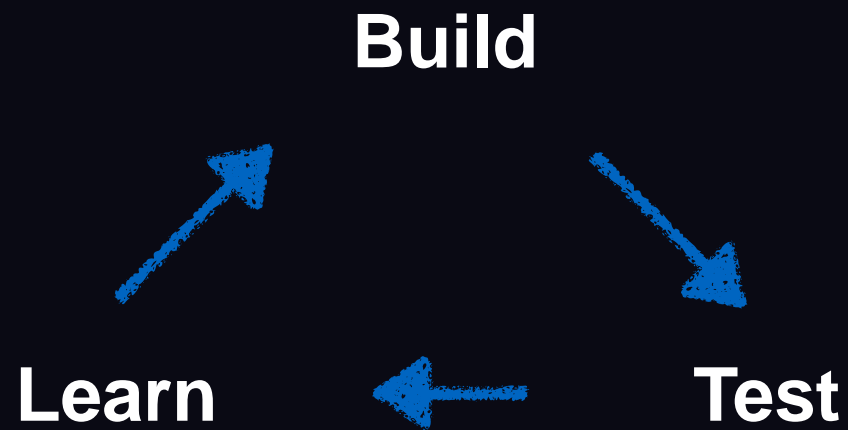
- Shipping a production app is a significant investment
- If the app flops it's a big loss
- What can we do we avoid this situation?

Test!

Test

- Test the app yourself until it feels right
- Test the app with new users from the target demographic and make sure it feels right for them

Build - Test - Learn



You just make the best product you can,
and you don't put it out until you feel it's
right.

- Steve Jobs

How do you know
when it feels right?

Getting good at cooking

- Try as many dishes as possible
- Try to copy them
- Experiment cooking as many dishes as possible
- This will expand your sense of taste
- Your sense of taste will tell you if something is right
- **WARNING** - Your taste will develop faster than your cooking skills.

Why prototype?

Prototypes

- Build much faster (5x, 10x+)
- Test more iterations before core engineering begins
- Get ideas for how to improve app
- Improvements on top of improvements increase the probability that you're building the right thing

Common Prototyping Scenarios

1. Creating a new app
2. Major redesign of existing app
3. Designing a new feature

Case Studies

1. Creating a new app: Work & Co

1. Creating a new app: Work & Co

- Digital Agency
- 90% repeat business rate
- Top customers like:
Apple, Facebook,
Google

“Prototypes already are and continue to become an even bigger and more a part of our process. It sounds insanely fast, but for all our projects it’s common to see working prototypes within the first week of designing,”

- Calvin Teoh, Work & Co

Framer

- Open source Javascript library especially good at prototyping animations and interactions
- Very popular with designers
- Paid for IDE and design tool




```
# Set default material design curve
Framer.Defaults.Animation =
  curve: "cubic-bezier(0.4, 0.8, 0.2, 1)"
  time: .5

# This imports all the layers for "superhero-material" into superheroMaterialLayers
superhero = Framer.Importer.load "imported/superhero-material"

# Enable accessing all layer groups by their name
Utils.globalLayers superhero

# Set statusBar and navbar superLayer to device so it's persistent
statusBar.superLayer = Framer.Device.screen
navbar.superLayer = Framer.Device.screen

# Hide existing floating action button (FAB)
pinkFab.visible = false

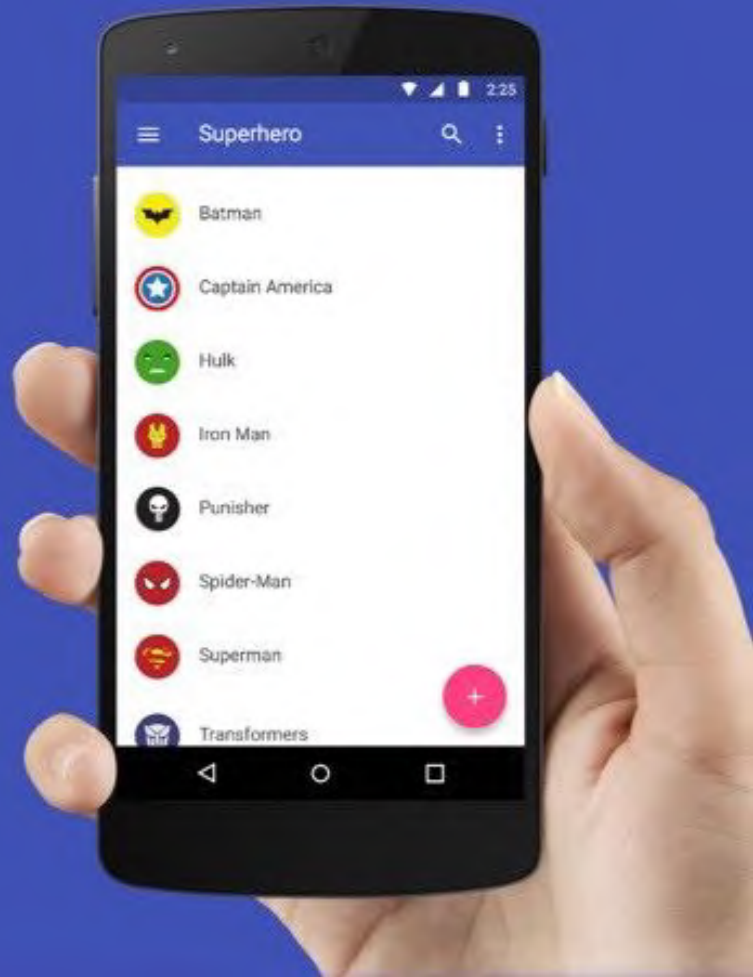
# Create new FAB
fab = new Layer
  width: 168
  height: 168
  borderRadius: "50%"
  backgroundColor: "#FF4081"
  borderColor: "rgb(255,255,255, .12)"
  borderWidth: .5
  x: Screen.width - 216
  y: Screen.height - 368
  shadowY: 18
  shadowBlur: 18
  shadowColor: "rgb(0,0,0, .24)"

# Add plus sign and center
add.superLayer = fab
add.center()

### Upload Screen ###
upload.visible = true

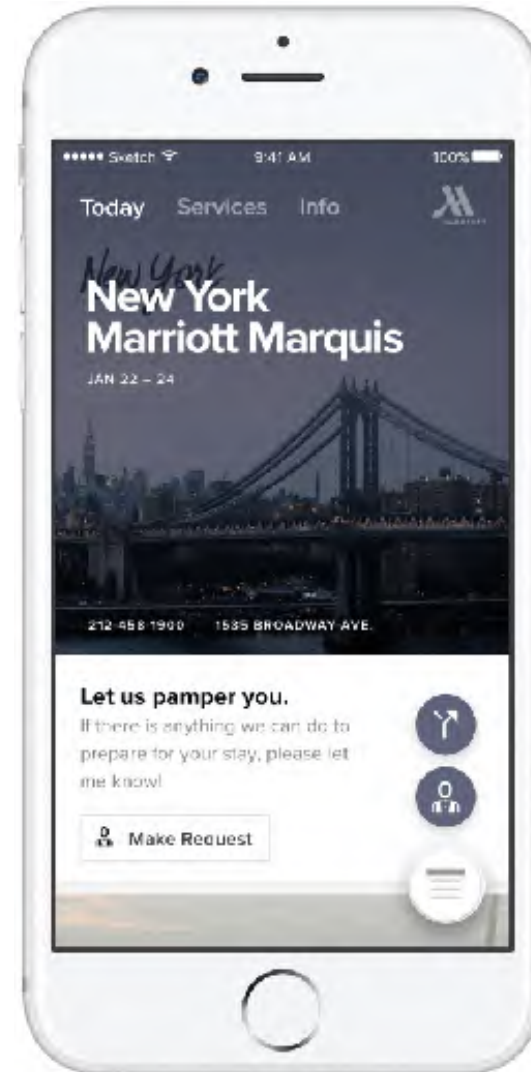
# Put invite screen before button
upload.placeBefore fab

# Create hit target for back arrow
backButton = new Layer
  width: 128
  height: 128
  x: 24
```



1. Creating a new app: Work & Co

- Example prototype made in framer for Marriott
- <https://blog.framer.com/powering-design-agency-workflows-63f3dd8faf0c>



Benefits of this approach

- Work & Co can rapidly iterate with their clients
- Prototypes can be validated through user tests
- No more long cycles where design firms spend a whole month just doing research before jumping in and experimenting
- With rapid cycle time a large space of app design can be explored

2. Major redesign: Uber

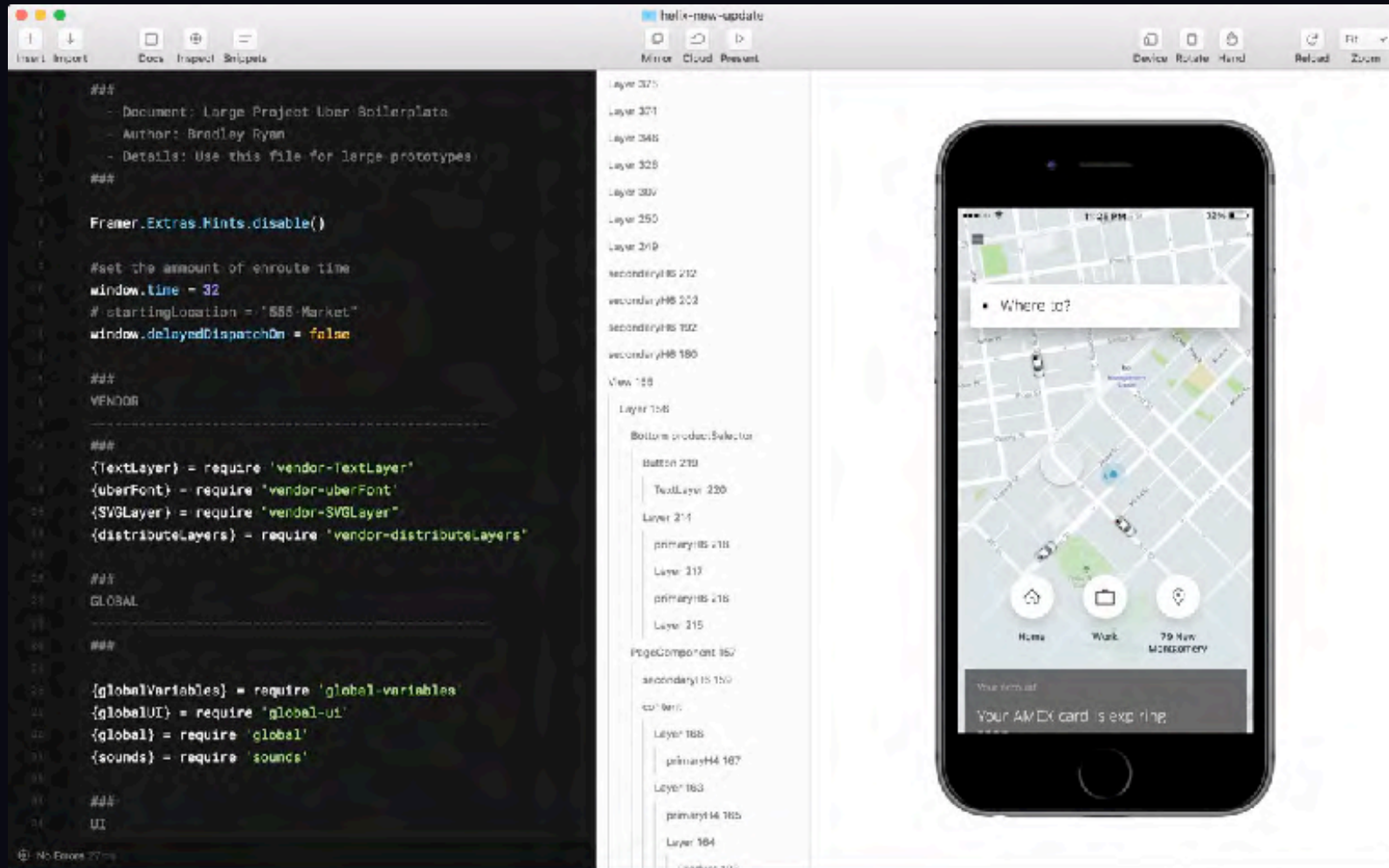
2. Major redesign: Uber

- In 2016, Uber began complete redesign of app
- Team prototyped and user tested multiple iterations before major development began

“Framer is integral to our workflow, allowing us to perfect animations and motions and then validate and test flows even before handoff to the engineer.”

- Bradley Ryan, Uber

2. Major redesign: Uber



- https://www.youtube.com/watch?v=X8axMiUi_bE

Benefits of this approach

- A major new feature motivating prototype was UberPOOL
- Redesigns can have massive scope
- Hard to find design flaws until you start filling out the details of the system
- Uber can get the user experience right the first time before handing off the design to engineering
- Engineers can use the prototype as a reference and spec when building the production version

3. Designing a new feature: Facebook

3. Designing a new feature: Facebook

- Vast majority of designers at Facebook are building prototypes
- Investing in prototyping tools such as Origami Studio

“Prototyping has fundamentally changed the way we design products at Facebook”

- Matej Hrescak, Facebook

Origami Studio

- Visual programming language
- Based on Origami which was built on top of Quartz Composer

The Origami Studio Finally Launch Best Prototyping Tool From Facebook - UX hacker



Press `esc` to exit full screen

Interaction

- Layer Photo
- Enable
- Down
- Tap
- Position
- Force

Switch

- Flip
- On / Off
- Turn On
- Turn Off

Pop Animation

- Number 1
- Progress
- Bounciness 5
- Speed 10

Transition

- Progress 1
- Start 0.38
- End 1

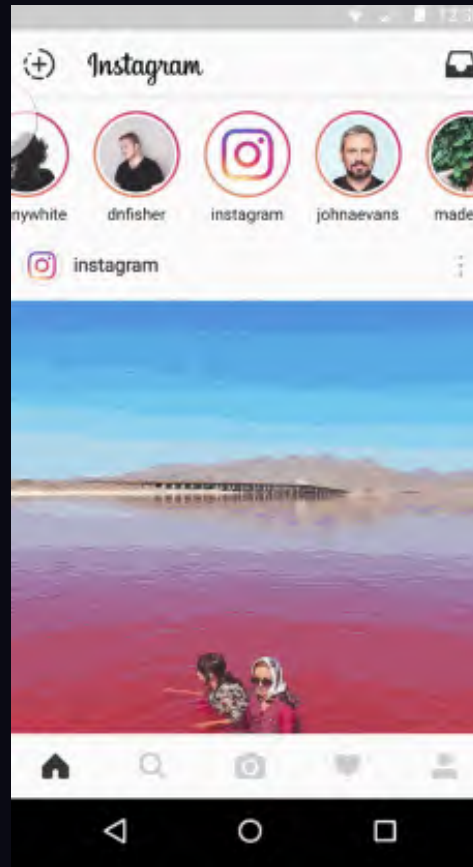
Photo

- Scale 1

7:06 / 11:15



3. Designing a new feature: Facebook



Benefits of this approach

- Engineers can focus on implementation
- Engineers are building something the design team has confidence in
- Designers can specify the animation and interactions in a highly precise way through the prototype

Common themes

Effective prototyping
makes engineering time
more productive

Prototypes are part of the new spec

- With apps involving animation and gestures, it's unrealistic to write out the spec in words
- Prototypes as part of the spec save engineering time

Spending core engineering time
on something users won't
understand or like is a huge
waste

Prototyping is an
industry best practice

Companies that don't
prototype are at a huge
disadvantage to those that do

Common Prototyping Scenarios

1. Creating a new app
2. Major redesign of existing app
3. Designing a new feature

1. Scenario - Creating a new app



I want to build the next great home sharing app like Airbnb, but I want it to be more social like Instagram.

What do we do?
How should we build the prototype?

Many tools for creating prototypes

Many tools for creating prototypes

- Framer
- Principle
- Marvel
- Origami Studio
- proto.io
- Craft Prototype
- Many more...

Framer

- Interaction are written in code, but the library is only used for prototypes and is not a general purpose framework
- Since the implementation is in Javascript it feels slow which hurts its realism

Origami Studio

- Interactions are fast and can be used to generate code, but the more complexity you add the slower it gets
- You have to use a visual programming language to program interactions

Not necessarily the best prototyping tools for creative developers

- Biased towards designers
- Trade-off expressiveness for simplicity
- Hard to incorporate real data
- As a developer, I am skeptical about learning a framework or new software that's only good for prototyping

Creative developer priorities for prototyping tools

- Fast dev-to-test loop (~ 1s)
- Minimal time to get started
- Realistic
- Want to use a general purpose framework
- Supporting tools
- Code reuse, or at least easily portable

How should we build our prototype?

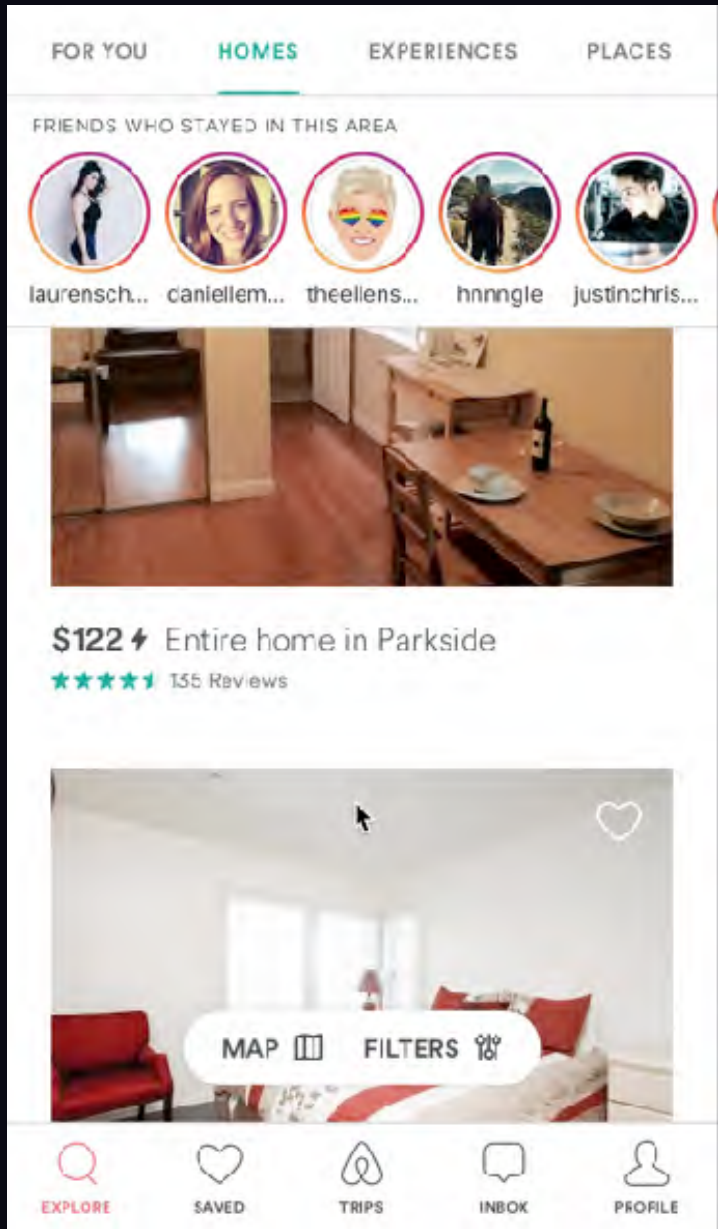


I want to build the next great home sharing app like Airbnb, but I want it to be more social like Instagram.

Proposal - Use React Native to build the prototype

React Native

- Cross-platform mobile development framework
- Supports iOS and Android
- Very performant animations and gestures
- Write code in React and Javascript
- Can build hybrid apps and easily bridge to native components



Fast dev-to-test loop (~ 1s)

- React Native bundler can automatically detect changes and reload the app
- Works on simulator or on device
- Faster than normal native development

Minimal time to get started

- Just exploring a new idea
- Don't want to recreate the UI, yet

Basil - Experimental tool to convert native iOS UI to React Native

- Export UIView hierarchy to JSON and images
- Convert to React Native
 - Currently just converting everything to a hierarchy of images
- Minimize the time to get started prototyping

```

_MarqueelessVerticalCollectionView-1.json
_(GC19D).SComponentLibrary35VerticalCollectionVi
Airbnb-Ison.json
capture-5207302682.json
capture-5207303606.json
capture-5207304454.json
capture-5207306511.json
capture-5207307142.json
capture-5207308980.json
capture-5207310524.json
capture-5207321121.json
capture-5207329326.json
capture-5207329848.json
capture-5207335363.json
capture-5207336906.json
ExploreCollapsibleTabCarouselComponent-1.json
GradientComponent-4.json
IGRootView-1.json
IGTabBar-1-520816.json
IGTabBar-1-5208166043.json
IGTabBar-1.json
IGToolBarz.json
MessageThreadComponent-1.json
RCTRootView-1-5208166523.json
RCTRootView-1.json
UITransitionView-1-5208073290.json
UITransitionView-1.json
UIView-1-5207309430.json
UIView-1-5207310790.json
UIView-1-5207337259.json
UIView-1-5208166147-5210673991.json
UIView-1-5208166147.json

IGStoryTrayCell-2
  UIView-19
    IGRadioButtonOwnerPictureImageView-3
      IGCircularButton-5
        UIImageView-21
          UILabel-14
          UILabel-15
      IGStoryGradientRingView-5
    ICLiveCuratorContainerView-3
      IGStoryGradientRingView-6
      IGLiveRippleEffectView-3
    IGProfilePictureImageView-3
      IGCircularButton-6
        UIImageView-22
      UIView-20
    IGLiveIndicatorView-3
      IGInsetLabel-3
      UIImageView-23
    UIImageView-20
  IGStoryBadgeView-3
    UIImageView-24
  IGStoryPeekPreviewContainerView-3
  IGStoryTrayCell-3
    UIView-21
      IGRadioButtonOwnerPictureImageView-4
        IGCircularButton-7
          UIImageView-26
          UILabel-16
          UILabel-17
      IGStoryGradientRingView-7
    IGLiveCuratorContainerView-4
      IGStoryGradientRingView-8
      IGLiveRippleEffectView-4
    IGProfilePictureImageView-4
      IGCircularButton-8
        UIImageView-27
      UIView-22
    IGLiveIndicatorView-4
      IGInsetLabel-4
      UIImageView-28
    UIImageView-25
  IGStoryBadgeView-4
    UIImageView-29
  IGStoryPeekPreviewContainerView-4

```



Key	Value
alpha	1
className	IGStoryPeekPr
frame	
origin	
x	8
y	4
size	
height	56
width	56
hidden	0
image2x	
imageRefId	20d83c562b6
isEditable	1
isView	1
layoutMargins	
bottom	8
left	8
right	8
top	8
objectId	IGStoryPeekPr
superview	
refId	IGStoryTrayCe
tag	0
UIColor	
a	1
b	1
className	UIColor
g	0.4784313725
r	0

The image shows a screenshot of the Xcode IDE. On the left is a component library with a list of files. A blue arrow points to the file `UIView-1-5208166147.json`. In the center is a tree view of the app's UI hierarchy, with `IGStoryTrayCell-3` selected. On the right is a preview of an Instagram-like app interface, showing a post by `wiedzmapani` with a photo of food. To the right of the preview is a property inspector table.

Key	Value
alpha	1
className	IGStoryPeekPr
frame	
origin	
x	8
y	4
size	
height	56
width	56
hidden	0
image2x	
imageRefId	20d83c562b6
isEditable	1
isView	1
layoutMargins	
bottom	8
left	8
right	8
top	8
objectId	IGStoryPeekPr
superview	
refId	IGStoryTrayCe
tag	0
UIColor	
a	1
b	1
g	0.4784313725
r	0

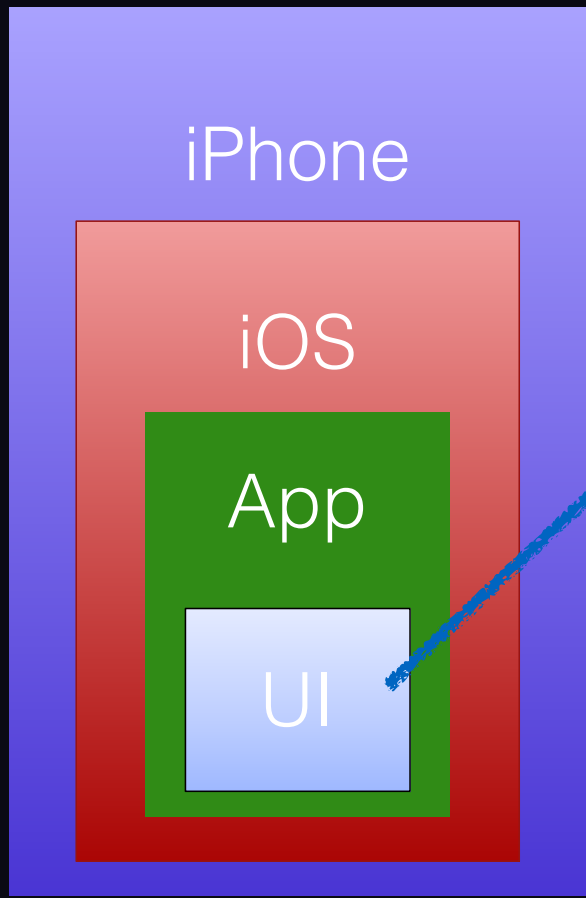
Component library

View Hierarchy

IGStoryPeekPreviewContainerView-3

The screenshot displays the Xcode View Hierarchy tool. On the left, a list of views is shown, with 'IGStoryPeekPreviewContainerView-3' highlighted. A blue arrow points from this node to the Instagram story preview in the center. Another blue arrow points from the same node to the 'Properties' pane on the right, which shows the view's attributes:

Key	Value
alpha	1
className	IGStoryPeekPr
frame	
origin	
x	8
y	4
size	
height	56
width	56
hidden	0
image2x	
imageRefId	20d83c562b6
isEditable	1
isView	1
layoutMargins	
bottom	8
left	8
right	8
top	8
objectId	IGStoryPeekPr
superview	
refId	IGStoryTrayCe
tag	0
UIColor	
a	1
b	1
g	0.4784913725
r	0



JSON
Images

React
Native


```
IGRootView-1.js — DesignCanvasViewer
JS IGRootView-1.js x
9356   --return this.defaultRender();
9357   }
9358 }
9359
9360 export class IGTabBarButton1 extends React.Component {
9361   defaultRender() {
9362     --return (
9363     --<<View
9364     --<<<style={{
9365     --<<<<position: 'absolute',
9366     --<<<<left: 0,
9367     --<<<<top: 0,
9368     --<<<<height: 45,
9369     --<<<<width: 75,
9370     --<<<<opacity: 1,
9371     --<<<<overflow: 'visible',
9372     --<<<<}}>
9373     --<<<<Image
9374     --<<<<<resizeMode={'contain'}
9375     --<<<<<source={require('./images/ee9af61d9c6edf211cbad0c66d665484.png')}>
9376     --<<<<<style={{
9377     --<<<<<position: 'absolute',
9378     --<<<<<left: 0,
9379     --<<<<<top: 0,
9380     --<<<<<height: 45,
9381     --<<<<<width: 75,
9382     --<<<<<}}>
9383     --<<<</>
9384     --<<<<<UIImageView61 />
9385     --<<<<<UIImageView62 />
9386     --<<<</View>
9387     --<<<</>
9388   )
9389 }
9390
9391 render() {
9392   --return this.defaultRender();
9393 }
9394
9395 export class UIImageView61 extends React.Component {
9396   defaultRender() {
9397     --return (
9398     --<<<View
9399     --<<<<style={{
```

Ln 9375, Col 25 Spaces: 2 UTF-8 CRLF JavaScript

Process using Basil

- Capture multiple screens from Airbnb and Instagram
- Isolate pieces of the UI as reusable components
- Generate the reusable components as React Native
- Restitch React Native components with new functionality

_MarqueelessVerticalCollectionView-1.json
 _T(GC19D).SComponentLibrary35VerticalCollectionVi
 Airbnb-Ison.json
 capture-5207302682.json
 capture-5207303605.json
 capture-5207304454.json
 capture-5207306511.json
 capture-5207307142.json
 capture-5207308980.json
 capture-5207310524.json
 capture-5207321121.json
 capture-5207329326.json
 capture-5207329848.json
 capture-5207335363.json
 capture-5207336966.json
 ExploreCollapsibleTabCarouselComponent-1.json
 GradientComponent-4.json
 IGRRootView-1.json
 IGStoryPeekPreviewContainerView-3.json
 IGTabBar-1-520816.json
 IGTabBar-1-5208186643.json
 IGTabBar-1.json
 IGToolBar2.json
 Instagram-Stories.json
 MessageThreadComponent-1.json
 RCTRootView-1-5208165523.json
 RCTRootView-1.json
 UICollectionView-1.json
 UITransitionView-1-5208073280.json
 UITransitionView-1.json
 UIView-1-5207309430.json
 UIView-1-5207310780.json
 UIView-1-5207337259.json
 UIView-1-5208106147-5210673991.json
 UIView-1-5208186147.json

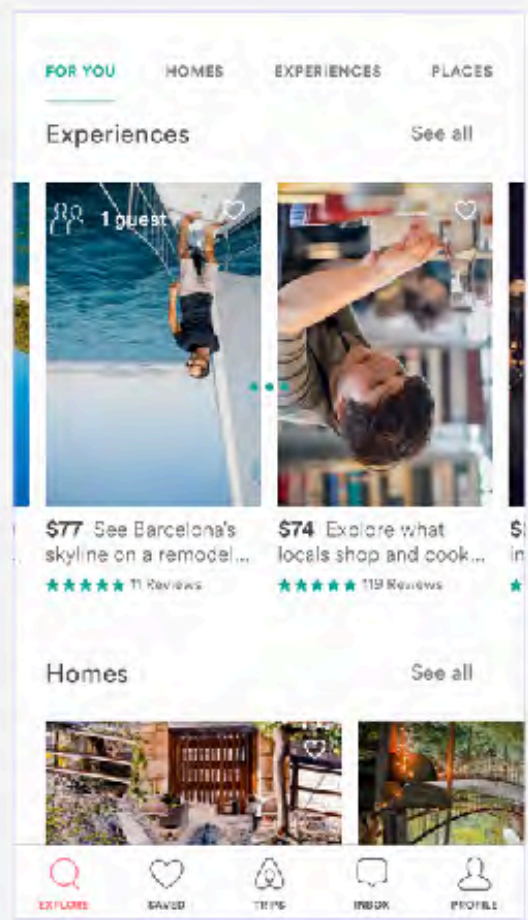
IGStoryTrayCollectionViewCall-1
 UIView-13
 UICollectionView-2
 IGStoryTrayColl-1
 UIView-15
 IGRae_OwnerPictureImageView-1
 IGCircularButton-1
 UIImageView-11
 UILabel-10
 UILabel-11
 IGStoryGradientRingView-1
 ICLiveCuratorContainerView-1
 IGStoryGradientRingView-2
 IGLiveRippleEffectView-1
 IGProfilePictureImageView-1
 IGCircularButton-2
 UIImageView-12
 UIView-16
 IGLiveIndicatorView-1
 IInsetLabel-1
 UIImageView-13
 UIImageView-10
 UIImageView-9
 IGStoryBadgeView-1
 UIImageView-14
 IGStoryPeekPreviewContainerView-1
 IGStoryTrayNuxCell-1
 UIView-17
 IGRae_OwnerPictureImageView-2
 IGCircularButton-3
 UIImageView-16
 UILabel-12
 UILabel-13
 IGStoryGradientRingView-3
 IGLiveCuratorContainerView-2
 IGStoryGradientRingView-4
 IGLiveRippleEffectView-2
 IGProfilePictureImageView-2
 IGCircularButton-4
 UIImageView-17
 UIView-18
 IGLiveIndicatorView-2
 IInsetLabel-2
 UIImageView-18

Key	Value
alpha	1
className	IGStoryTrayCo
frame	
origin	
x	0
y	33
size	
height	91
width	375
hidden	0
image2x	
imageRefId	8f2e8821967c
isEditable	1
isView	1
layoutMargins	
bottom	8
left	8
right	8
top	8
objectId	IGStoryTrayCo
subviews	
0	
refId	UIView-13
1	
refId	UIView-14
tag	0
tintColor	
a	1
b	1
className	UIColor
g	0.4784313725
r	0

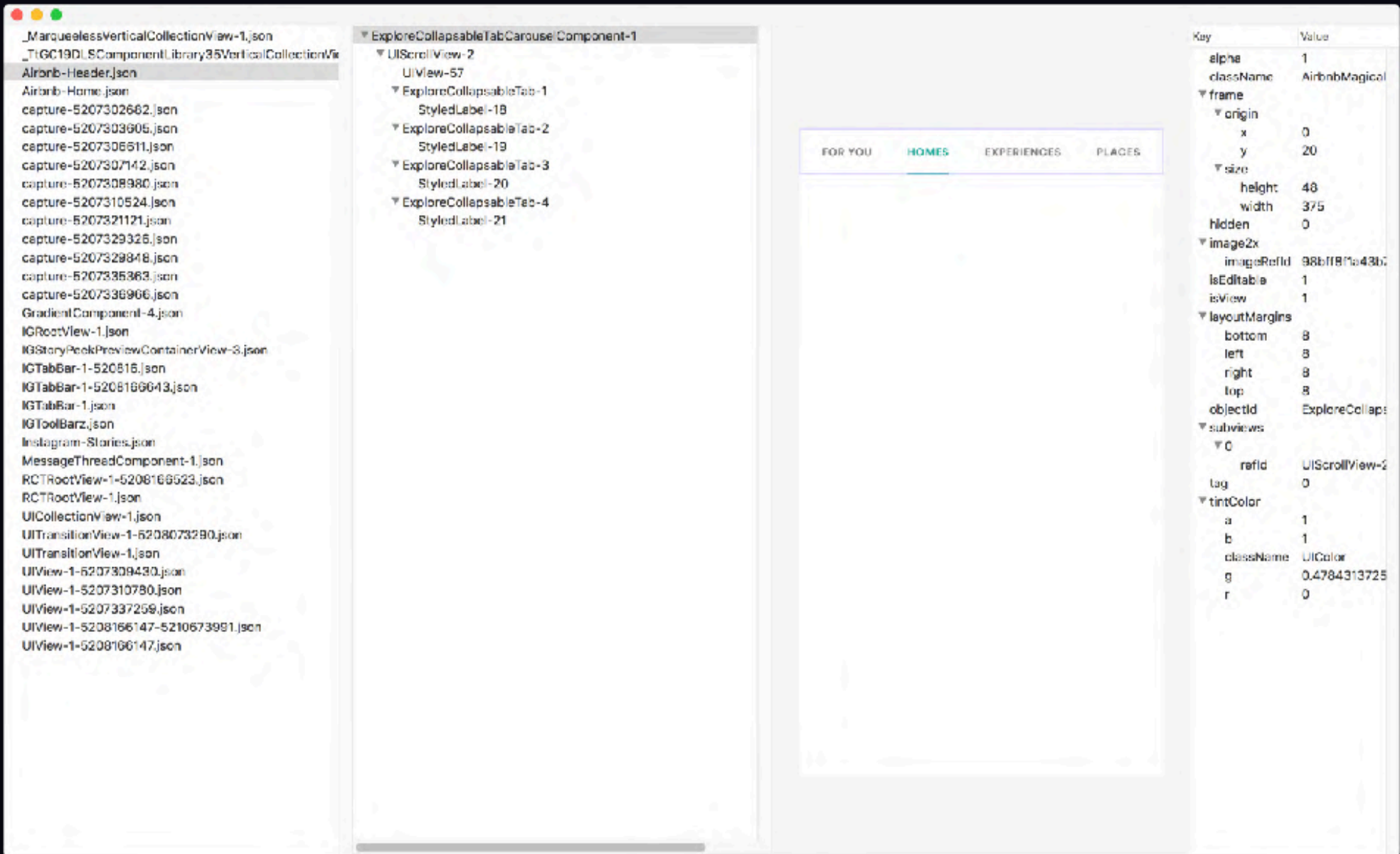
```

_MarqueelessVerticalCollectionView-1.json
_TTGC19DLSComponentLibrary35VerticalCollectionVie
Airbnb-Header.json
Airbnb-Home.json
capture-5207302662.json
capture-5207303605.json
capture-5207306511.json
capture-5207307142.json
capture-5207308980.json
capture-5207310524.json
capture-5207321121.json
capture-5207329326.json
capture-5207329848.json
capture-5207335363.json
capture-5207336906.json
GradientComponent-4.json
IGRootView-1.json
IGStoryPeekPreviewContainerView-3.json
IGTabBar-1-520816.json
IGTabBar-1-5208166043.json
IGTabBar-1.json
IGToolBar.json
Instagram-Stories.json
MessageThreadComponent-1.json
RCTRootView-1-5208166523.json
RCTRootView-1.json
UICollectionView-1.json
UITransitionView-1-5208073290.json
UITransitionView-1.json
UIView-1-5207309430.json
UIView-1-5207310780.json
UIView-1-5207337259.json
UIView-1-5208166147-5210673991.json
UIView-1-5208106147.json

UIWindow-1
  UIView-1
    UIView-2
      UIView-5
        UILayoutContainerView-1
          UITransitionView-1
            UINavigationControllerWrapperView-1
              UILayoutContainerView-2
                UINavigationTransitionView-1
                  UINavigationControllerWrapperView-2
                    ExploreSearchResultsComponent-1
                      ExploreSearchResultsEmptyStateCol
                        UIScrollView-1
                          UIView-9
                            UStackView-1
                              StyledLabel-2
                                ExploreEmptyStateButton
                                  ExploreEmptyStateButt
                                    UIButtonLabel-1
                                      StyledLabel-1
                                        UIView-10
                                          _TTGC19DLSComponentLibr
                                            UIView-11
                                              DestinationCardCompo
                                                ImageComponent-1
                                                  ImageView-1
                                                    UIView-12
                                                      GradientCompon
                                                        UIView-13
                                                          StyledLabel-3
                                                            UICollectionView-1
                                                              UIView-14
                                                                UIImageView-2
                                                                  LoaderComponent-1
                                                                    _TTGC19DLSComponentLi
                                                                      _TTGC19DLSComponentLi
                                                                      _TTGC19DLSComponentLi
                                                                UIImageView-1
                                                                  UIView-7
                                                                    ExploreCollectionContainerView-1
                                                                      _MarqueelessVerticalCollectionVie
                                                                        _TTGC19DLSComponentLibrary
                                                                          UIView-15
                                                                            _TTGC19DLSComponentLi
  
```

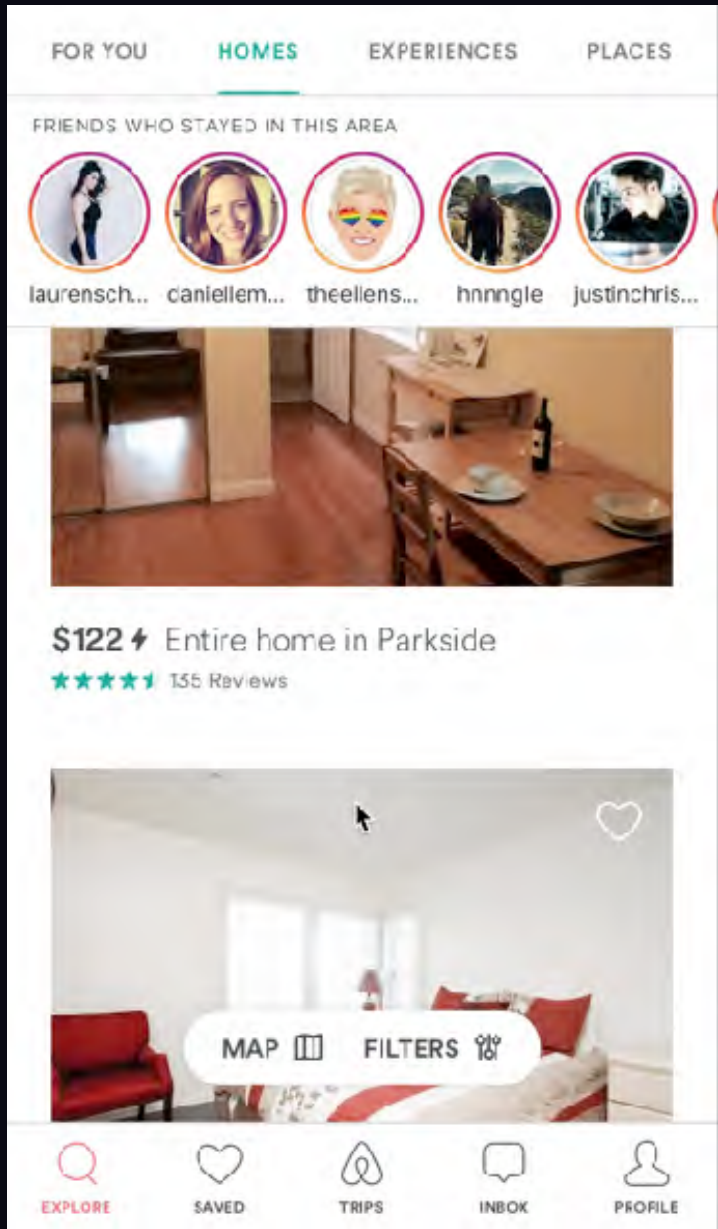


Key	Value
alpha	1
background...	
a	1
b	0
className	UIColor
g	0
r	0
className	UIWindow
frame	
origin	
x	0
y	0
size	
height	867
width	375
hidden	0
image2x	
imageRefId	Ofc3c086a98e
isEriable	1
isView	1
layoutMargins	
bottom	8
left	8
right	8
top	8
objectId	UIWindow-1
subviews	
0	
refid	UIView-1
tag	0



Restitching it back together

```
return (  
  <View>  
    { default:RootRender }  
    <View>  
      style={{  
        top: 48 + 20,  
        backgroundColor: 'white',  
        paddingVertical: 5,  
        borderTopWidth: 1,  
        borderBottomWidth: 1,  
        borderColor: '#d6d5c6',  
      }}>  
      <Text>  
        style={{  
          paddingLeft: 14,  
          marginVertical: 4,  
          fontSize: 18,  
          textAlign: 'left',  
          fontWeight: '300',  
          color: '#1e1e1e',  
          letterSpacing: 0.5,  
        }}>  
        { 'FRIENDS WHO STAYED IN THIS AREA' }  
      </Text>  
      { /* InstagramHeadWrapper is our new scrollable Instagram bar */ }  
      <InstagramHeadWrapper  
        onButtonClick={{buttonIndex: number, buttonPositionX: number}} => {  
        this.setState({  
          activeButton: {  
            index: buttonIndex,  
            x: buttonPositionX,  
          }  
        }  
      );  
        horizontalPadding={s}  
      />  
    </View>  
    { /* Scrollable feed view */ }  
    <ScrollableContentView />  
    <FloatingSegmentedButtonContainerComponent1 />  
    { /* Modal when selected Instagram head */ }  
    { modal }  
  </View>  
);  
}
```



Realistic

- Users can test on device
- Native performance
- Gestures
- Animation
- Real data

Supporting tools













- React Native has an excellent ecosystem of tools available and being built
- React Native packager for rapid reload while developing
- Chrome debugger for understanding and fixing bugs
- Typescript for type safety
- Visual Studio Code to edit JS/TS
- XCode for debugging
- Buddybuild for reshipping prototypes to stakeholders

General purpose framework

- React Native is worth learning for mobile developers even if it is not being used for production

Who's using React Native?

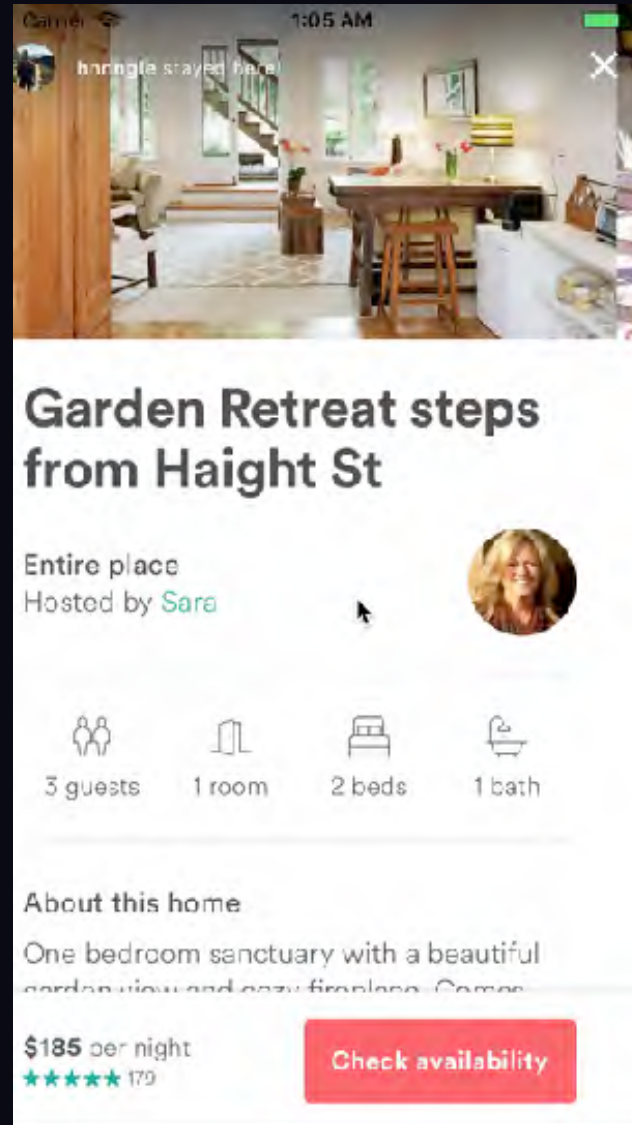
Thousands of apps are using React Native, from established Fortune 500 companies to hot new startups. If you're curious to see what can be accomplished with React Native, check out these apps!

 Facebook iOS - Android Using React Native in the Facebook App	 Facebook Ads Manager iOS - Android How We Built the First Cross-Platform React Native App	 Instagram iOS - Android React Native at Instagram	 FB iOS - Android Tutorial: Building the FB conference app
 Airbnb iOS - Android Hybrid React Native Apps at Airbnb	 Walmart iOS - Android React Native at Walmart Labs	 Tencent iOS - Android	 Tencent QQ Android QQ is China's largest messaging platform, with over 800 million active accounts
 Baidu Mobile (手机百度) iOS - Android Baidu Mobile is a search engine used by over 600 million people in China	 Bloomberg iOS - Android How Bloomberg Used React Native to Develop Its new Consumer App	 Vogue iOS	 List Android Building List for Android with React Native

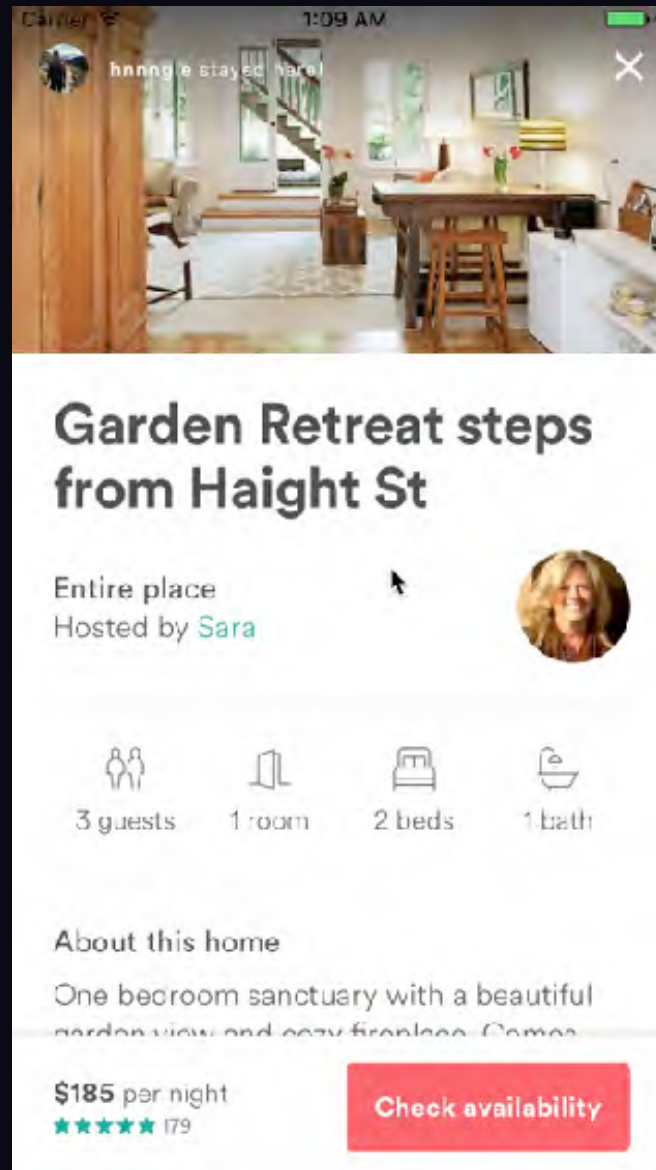
Code Reuse

- Prototypes in React Native can be used as a reference to re-implement natively.

Original swiping stories



Swiping stories with overlay



2. Scenario - Major redesign of existing app



Let's redo our
entire flagship
app in the
Google Material
way.

2. Major Redesign

- Create new components from Sketch and existing app
- Re-stitch together and prototype new material-style animations in React Native

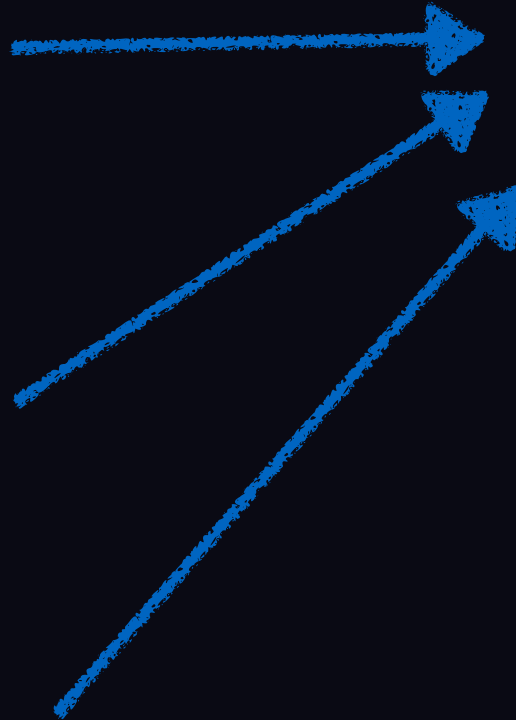
Sketch

Android UI

iOS UI

JSON
Images

React
Native



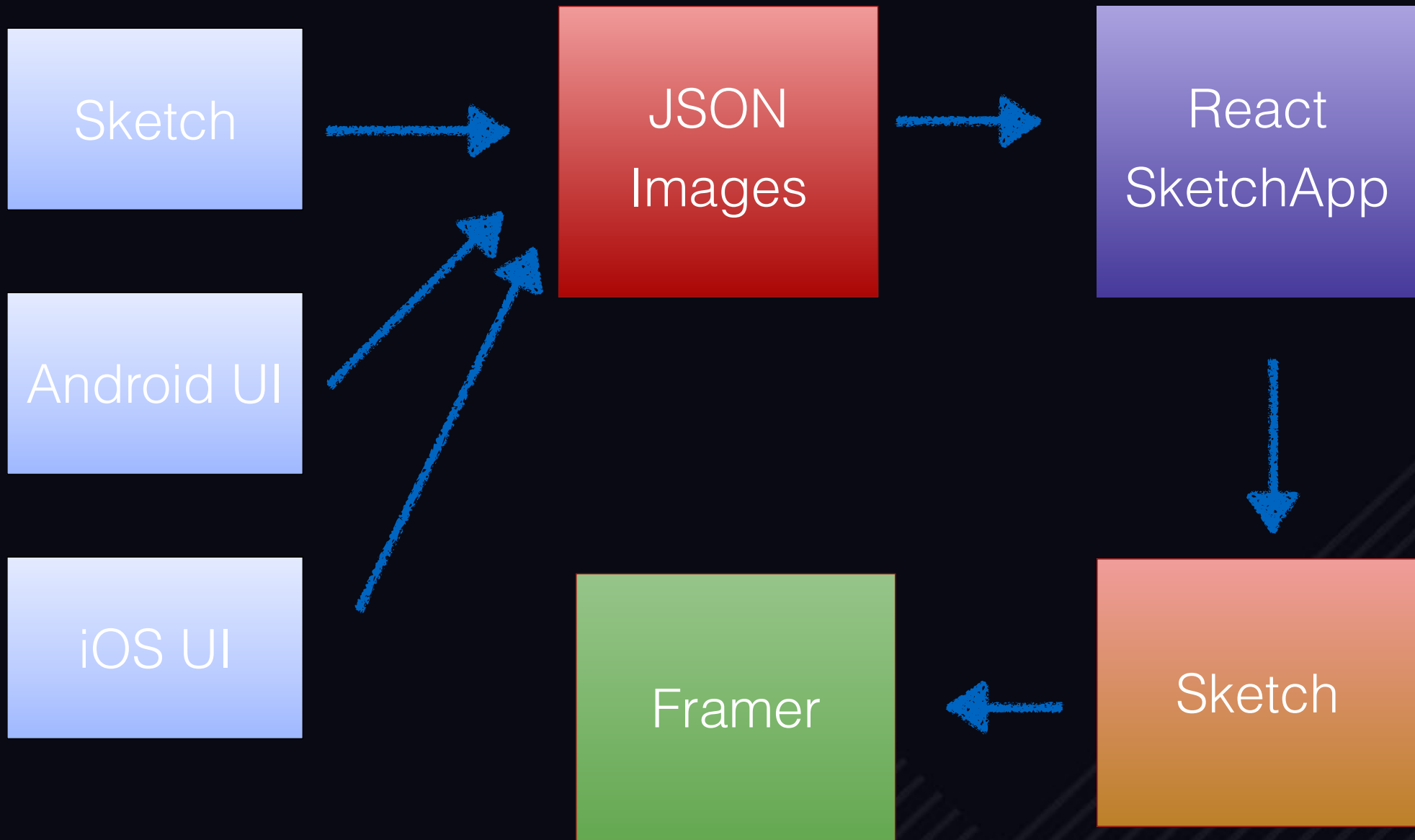
3. Scenario - Designing a new feature



We need to
make the
sharing flow
more delightful.
Our designer is a
wiz with Framer.
Have them send
me some comps.

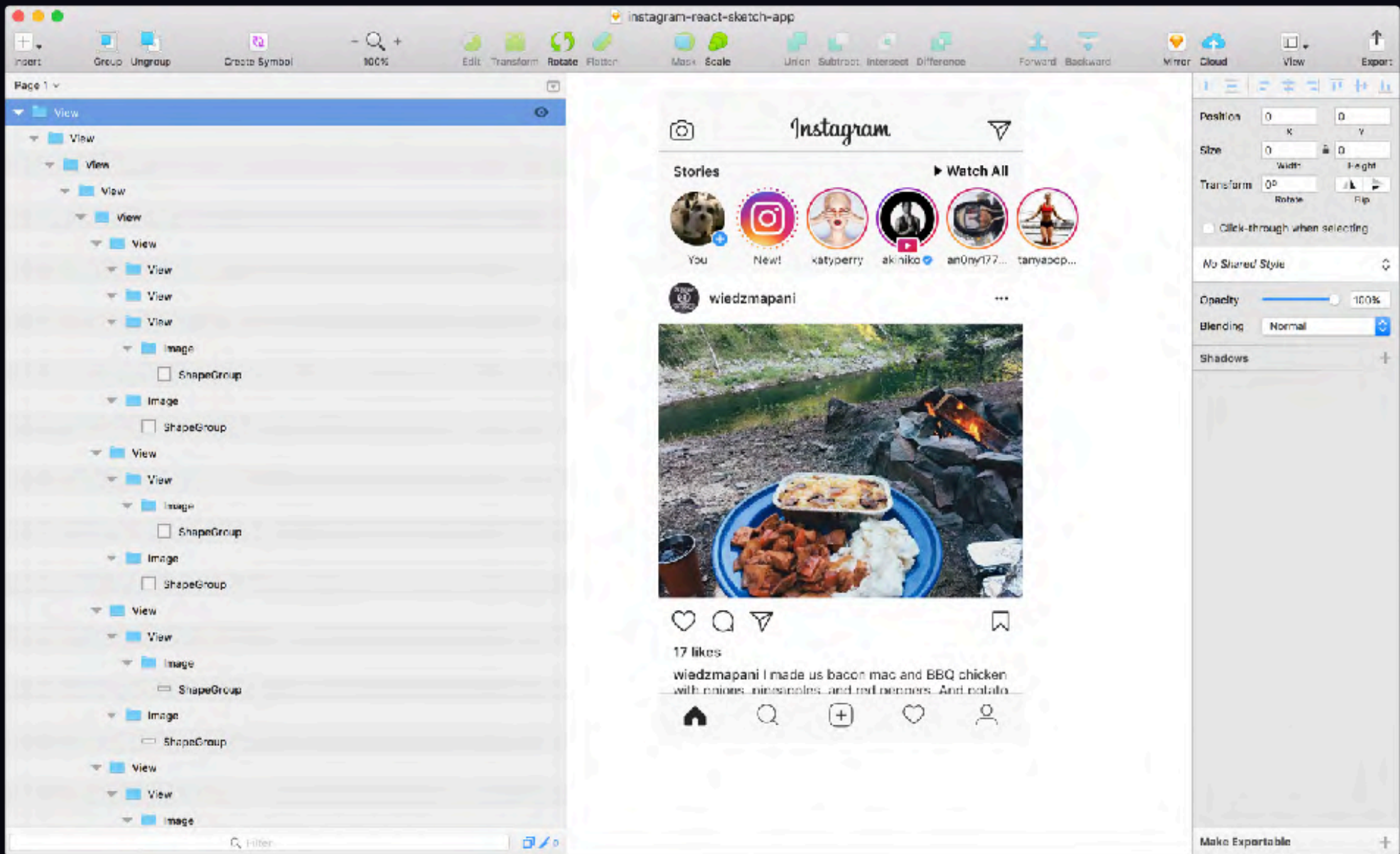
3. Designing a new feature

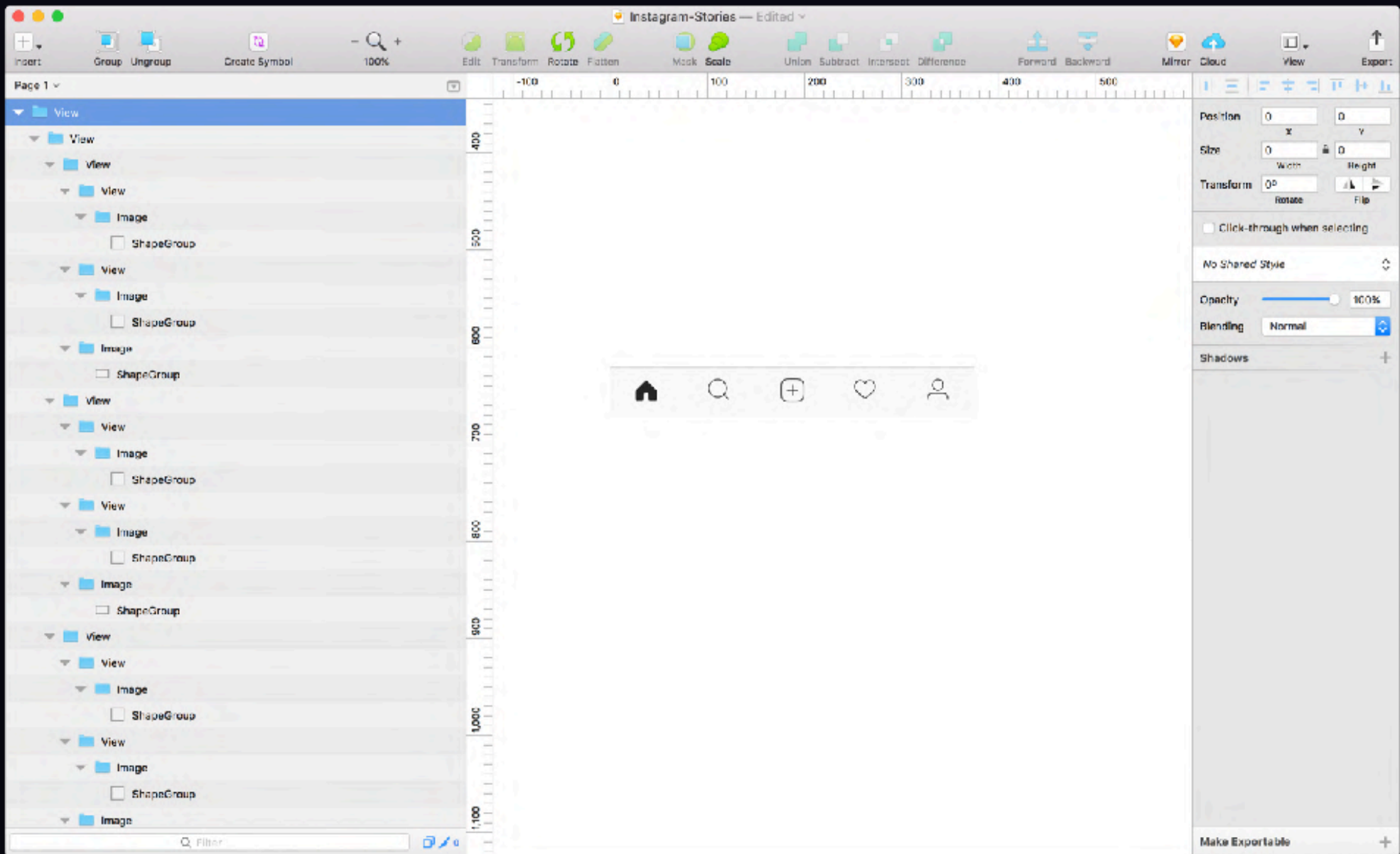
- In this scenario we want a designer to build the prototype using Framer
- So instead of generating React Native we generate React SketchApp which is used to make a Sketch file.
- Then we will import Sketch into Framer

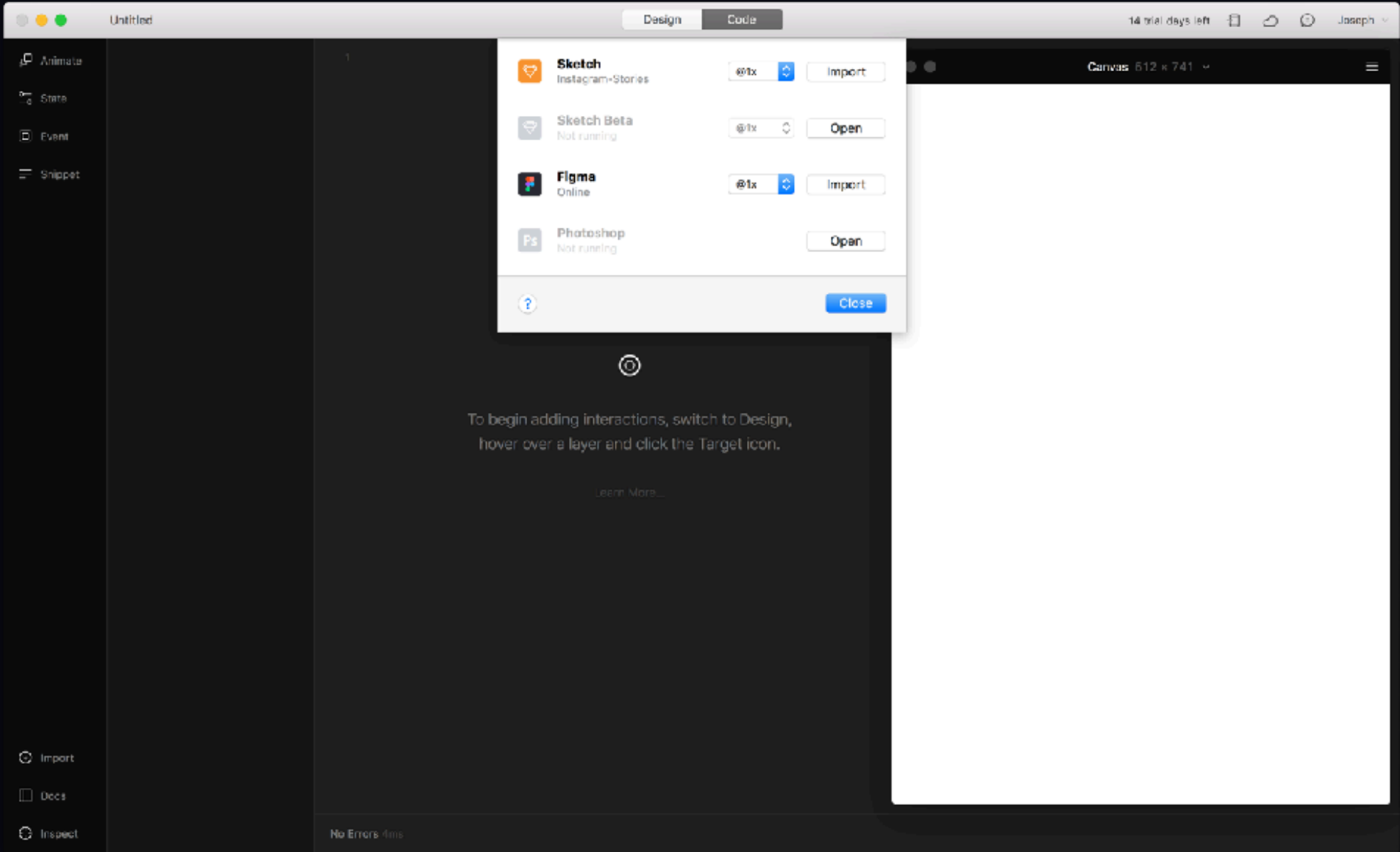


React SketchApp

- Built by Airbnb
- Renders React Native to Sketch
- <https://github.com/airbnb/react-sketchapp>







ICTabBar.framer Design Code 14 trial days left Joseph

View

- View1
 - View2
 - View6
 - Image2
 - View9
 - Image3
 - Image1
 - View3
 - View10
 - Image5
 - View11
 - Image6
 - Image4
 - View4
 - View12
 - Image8
 - View13
 - Image9
 - Image7
 - View5
 - View14
 - Image11
 - View15
 - Image12
 - Image10
 - View8
 - View16
 - Image14
 - View17

```

1 # Import file 'Instagram-Stories'
2 sketch = Framer.Importer.load("imported/Instagram-Stories#1x", scale: 1)
3
4 oldY = sketch.View.y
5 sketch.View.y += 100
6 sketch.View.opacity = 0
7
8 sketch.View.animate
9   - y: oldY
10  - opacity: 1.00
11  - options:
12    - time: 1
13    - curve: Spring(damping: 0.25)
14    - delay: 0.3
15

```

No Errors 14ms

iPhone 7 375 x 667 (85%)

Technical considerations building prototyping tools

Basil Embed

- Static library that exposes view hierarchy via embedded http server
- Can be injected into own apps
- Can be injected into 3rd party apps if iOS phone is jail broken

Electron

- We originally tried building the app as an electron app
- Lot of abstraction layers
- API is not as direct as macOS
- Performance and interactivity such as drag-and-drop a challenge

Mac App

- Familiar as an iOS developer
- Relatively performant on first pass
- Code seems to be more maintainable

Model driven architecture

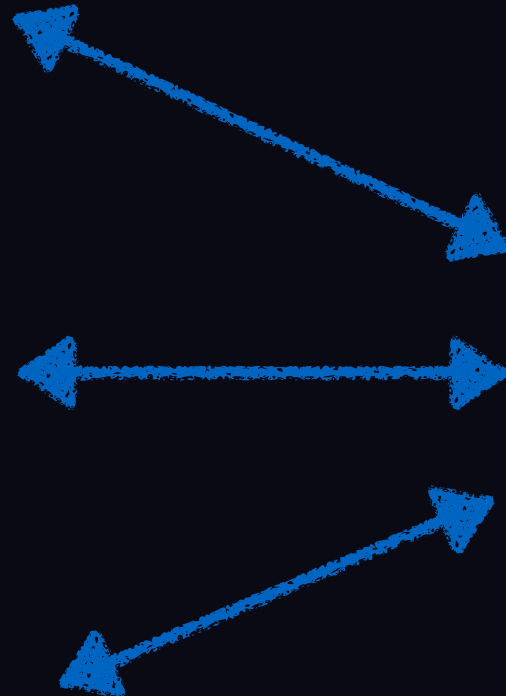
- Images and position, size
- Platform specific meta-data
- Convert from any platform to common model
- Convert from common model back to any platform

Sketch

Android UI

iOS UI

JSON
Images



How will the process of
prototyping continue to
evolve?

Still a large gap between
tools and processes between
design and development

With tools like Basil that can convert from one format to another we can reduce the overhead from switching tools

Design is not just what it looks like and
feels like.
Design is how it works.

- Steve Jobs

Design is not just what it looks like and feels like.

Design is how it works.



- Steve Jobs

Engineers need to be a part of design process

Prototyping is a creative process

By investing in creative skills
and tools engineers can become
more effective at prototyping

THANKS!

让创新技术推动社会进步

HELP TO BUILD A BETTER SOCIETY WITH
INNOVATIVE TECHNOLOGIES

Geekbang >

极客邦科技

InfoQ_{ueue}

专注中高端技术人员的技术媒体



EGO EXTRA GEEKS' ORGANIZATION
NETWORKS

高端技术人员学习型社交平台



StuQ_{ueue}
斯达克学院

实践驱动的 IT 教育平台

