Volgenau School of Engineering

Toward Practical Automation for Software Engineering

Kevin Moran, Ph.D.
Assistant Professor

CS Research Day

George Mason University

Friday, September 11th, 2020

GEORGE MASON UNIVERSITY

Software Engineering

Software Engineering

The methods and techniques by which developers design, create, test, and manage software

Software Engineering

The methods and techniques by which developers design, create, test, and manage software

My research goal: Design tailored ***automated approaches*** to help facilitate ***developer needs*** throughout the software development and maintenance lifecycle.

Practical Significance

Practical Significance

Blend *scientific discovery* with *practical significance*

Practical Significance

Blend *scientific discovery* with *practical significance*

How Can We Design Practical Automation?

Mining Software Repositories

GitHub

Bitbucket

Google Play

SOURCEFORGE

GitLab

Mining Software Repositories

Mining Software Repositories

Source Code Files

Software Documentation

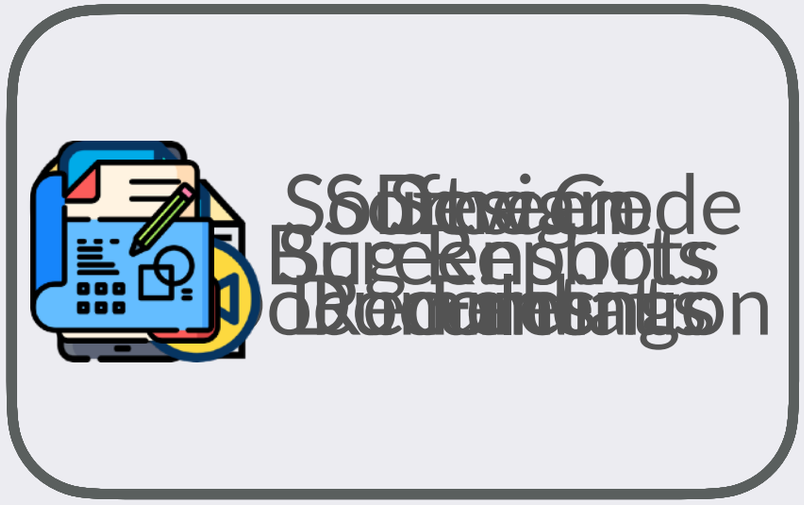
Screenshots

Screen Recordings

Bug Reports

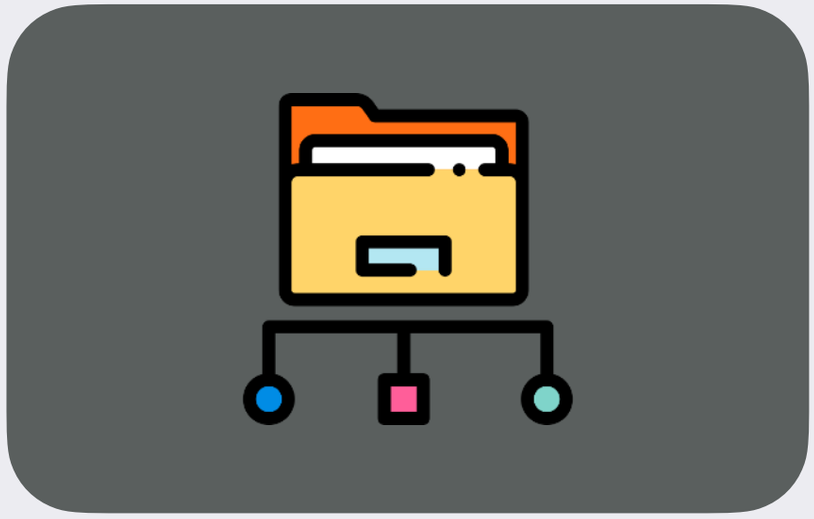
Design Documents

Learning Patterns from Software Data

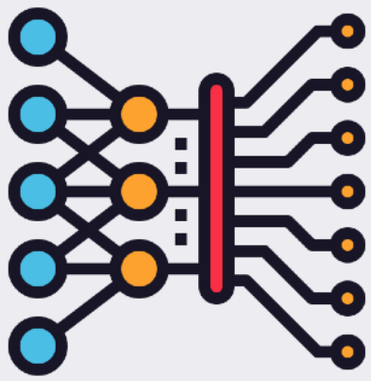


Learning Patterns from Software Data

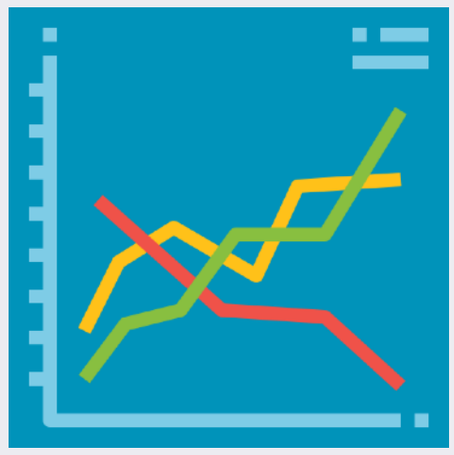
Software Repository Data



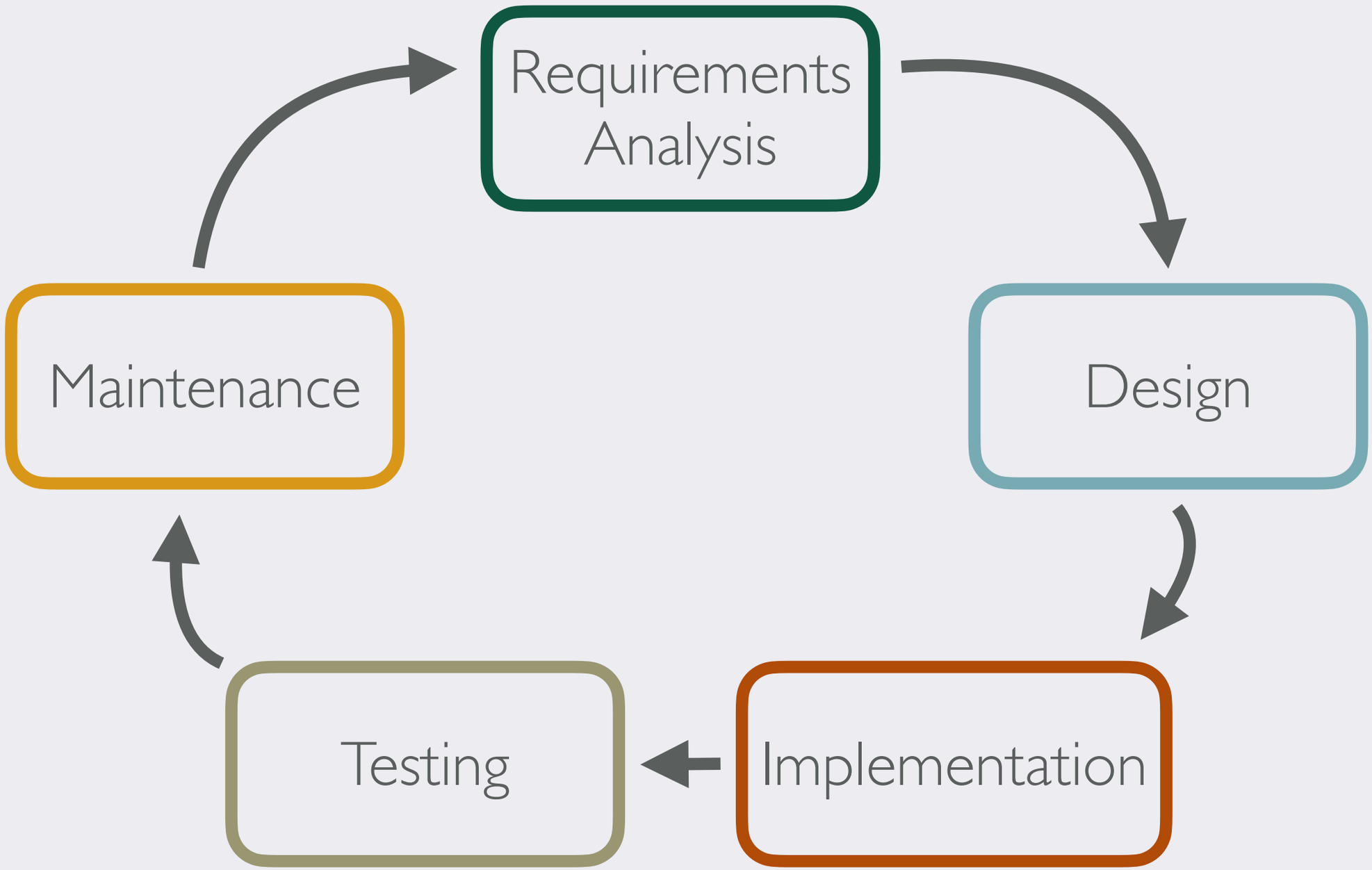
Machine Learning



Salient Patterns



Software Development Lifecycle



Overview

Overview

1. A Brief Look into my Research

Overview

1. A Brief Look into my Research

- Automatically Prototyping Mobile App GUIs

Overview

1. A Brief Look into my Research

- Automatically Prototyping Mobile App GUIs
- Translating Screen Recordings into Replayable Scenarios

Overview

1. A Brief Look into my Research

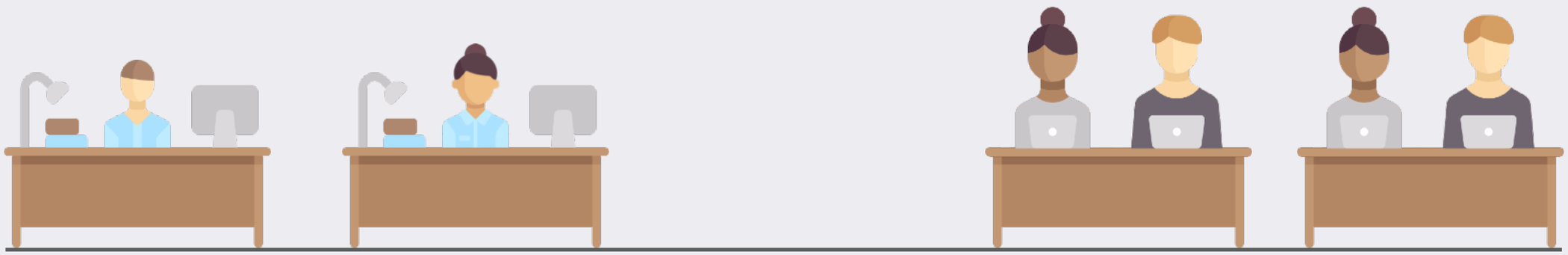
   - Automatically Prototyping Mobile App GUIs
   - Translating Screen Recordings into Replayable Scenarios

2. Overview of other Research Areas & Future Work

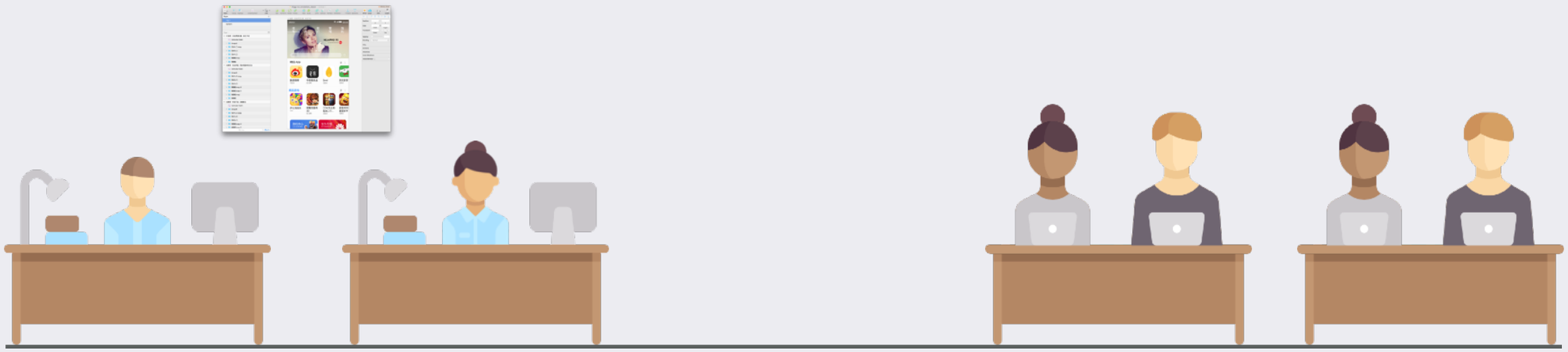
A Brief Look into my Research

Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps

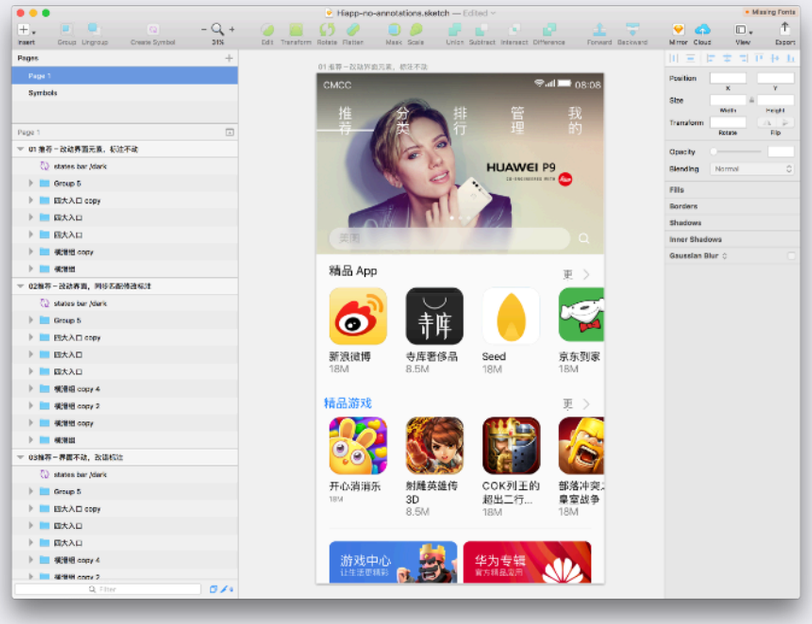
Research Problem

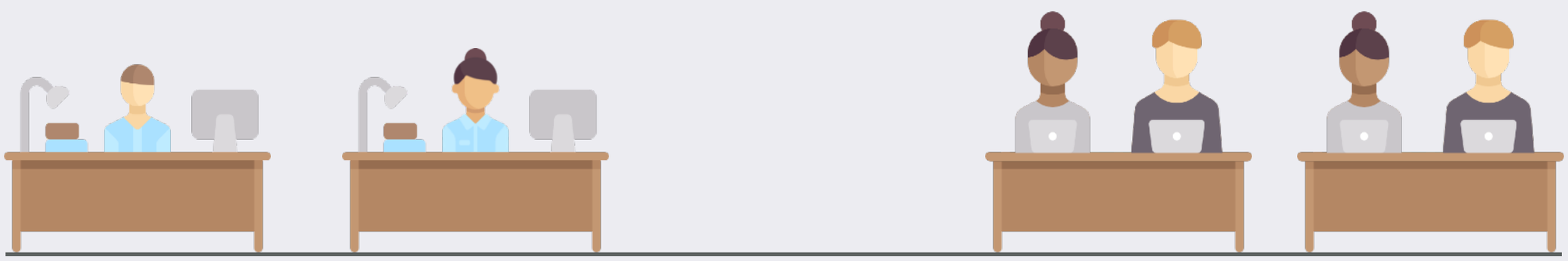


Research Problem



Research Problem

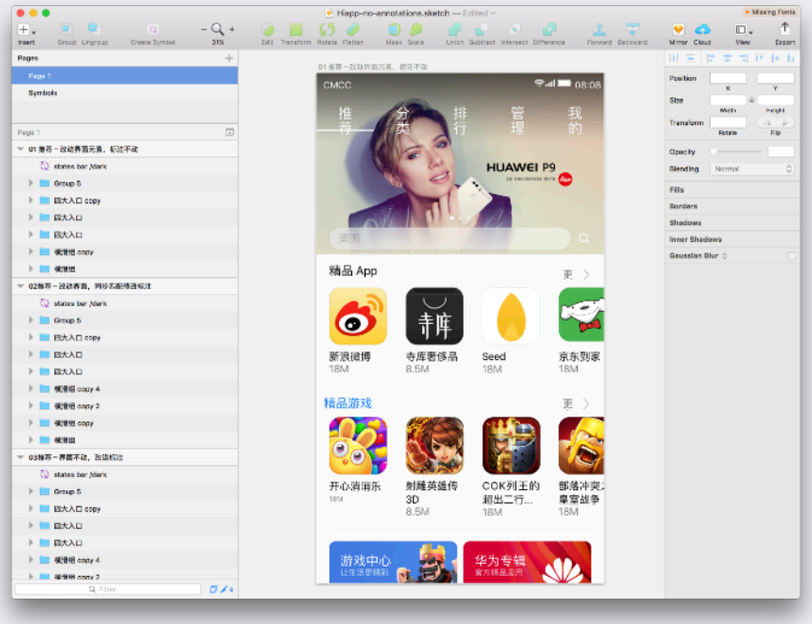


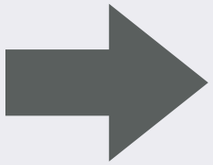


UI/UX Design Team

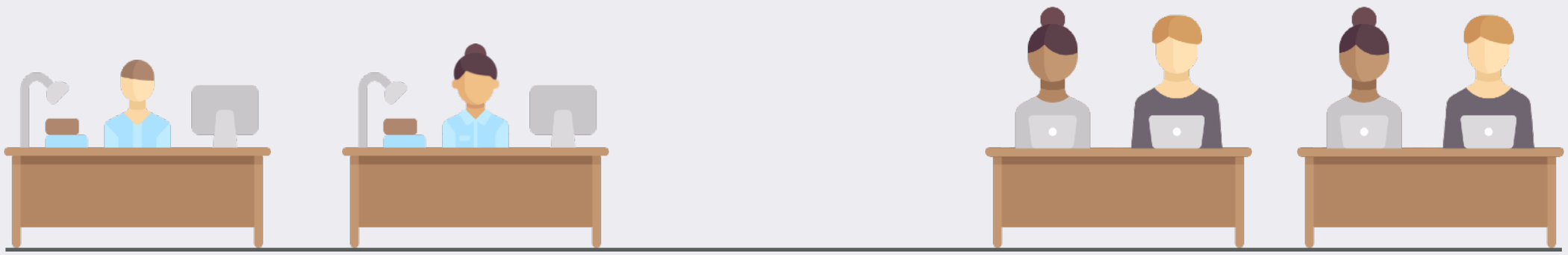
Development Team

Research Problem





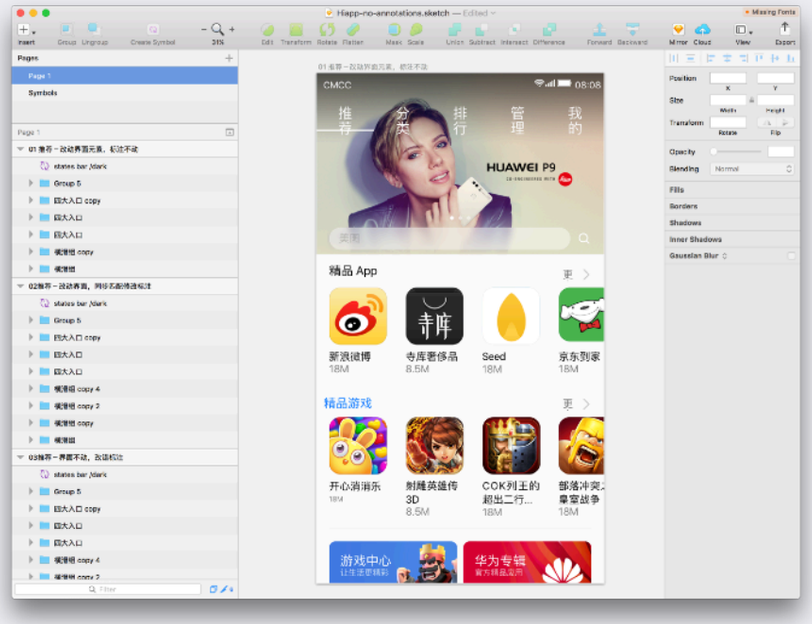
Prototype GUI Code

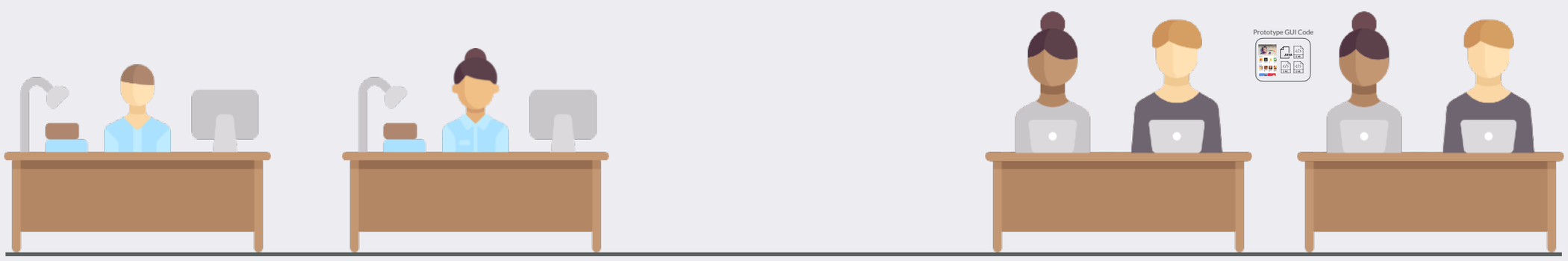


UI/UX Design Team

Development Team

Research Problem





UI/UX Design Team

Development Team

First Principles

How does a developer translate a GUI mock-up into code?

First Principles

How does a developer translate a GUI mock-up into code?

1. ***Detect*** or identify GUI-components that exist in a mock-up

First Principles

How does a developer translate a GUI mock-up into code?

1. ***Detect*** or identify GUI-components that exist in a mock-up
2. ***Classify*** these GUI-components into their constituent types

First Principles

How does a developer translate a GUI mock-up into code?

1. ***Detect*** or identify GUI-components that exist in a mock-up

2. ***Classify*** these GUI-components into their constituent types

3. ***Assemble*** the GUI-components into a hierarchy and stipulate styles

First Principles

First Principles

Detection

First Principles

Detection Classification

First Principles

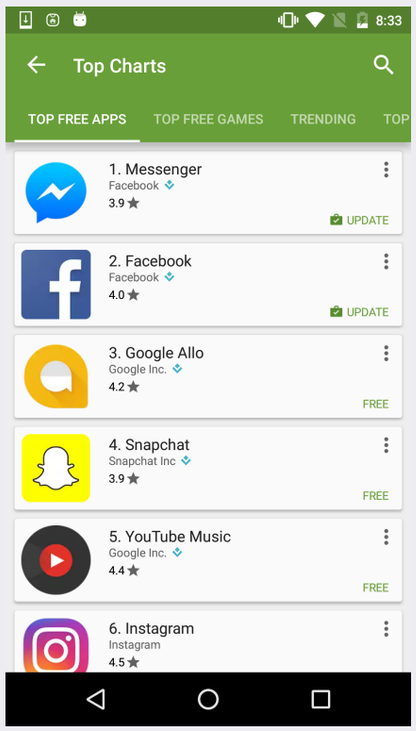
Detection Classification Assembly

First Principles

Detection — Classification — Assembly

Phase 1: GUI-Component Detection

Computer Vision-Based Detection*

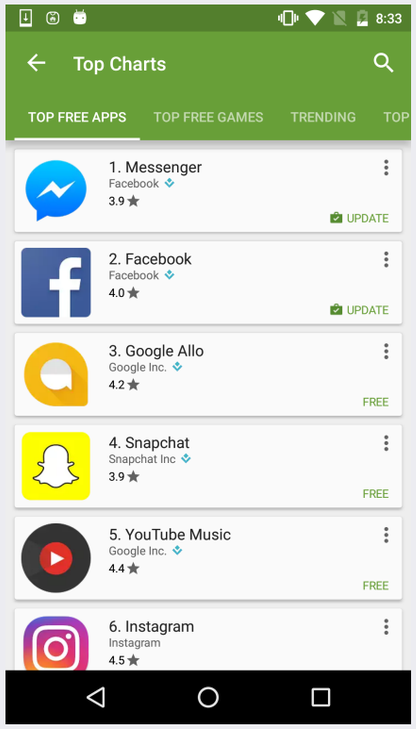


Input Image

*Tuan Anh Nguyen and Christoph Csallner. 2015. Reverse Engineering Mobile Application User Interfaces with REMAUI, In Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE '15)

Phase 1: GUI-Component Detection

Computer Vision-Based Detection*

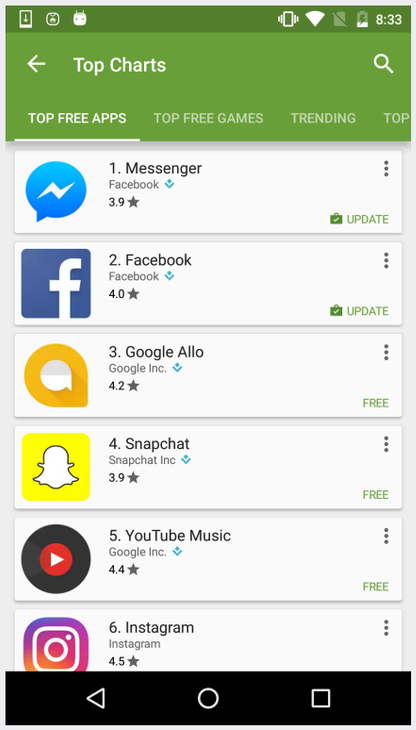


Input Image

*Tuan Anh Nguyen and Christoph Csallner. 2015. Reverse Engineering Mobile Application User Interfaces with REMAUI, In Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE '15)

Phase 1: GUI-Component Detection

Computer Vision-Based Detection*



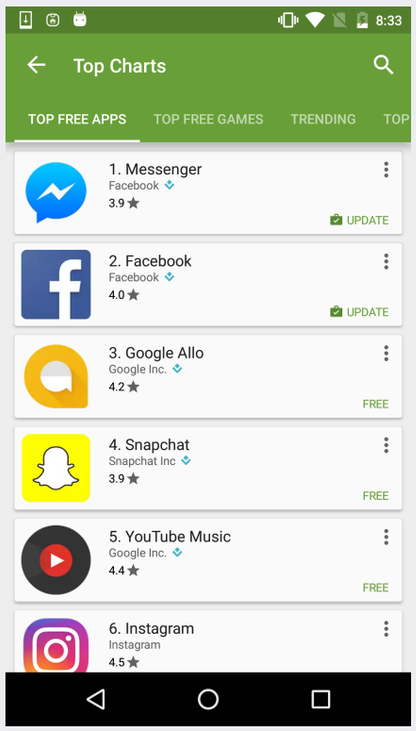
Input Image

Canny's Edge Detection

*Tuan Anh Nguyen and Christoph Csallner. 2015. Reverse Engineering Mobile Application User Interfaces with REMAUI, In Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE '15)

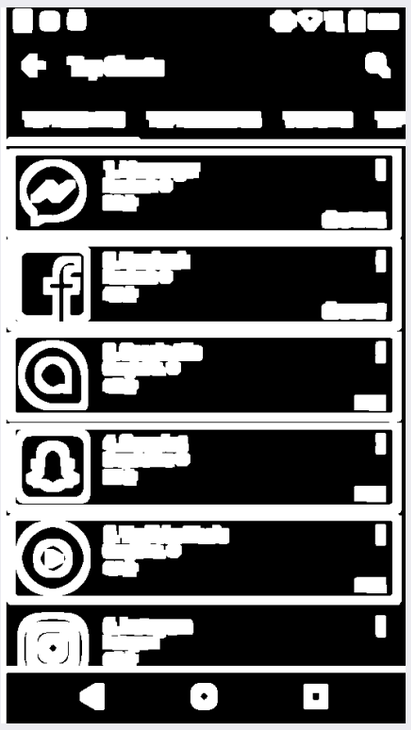
Phase 1: GUI-Component Detection

Computer Vision-Based Detection*



Input Image

Canny's Edge Detection

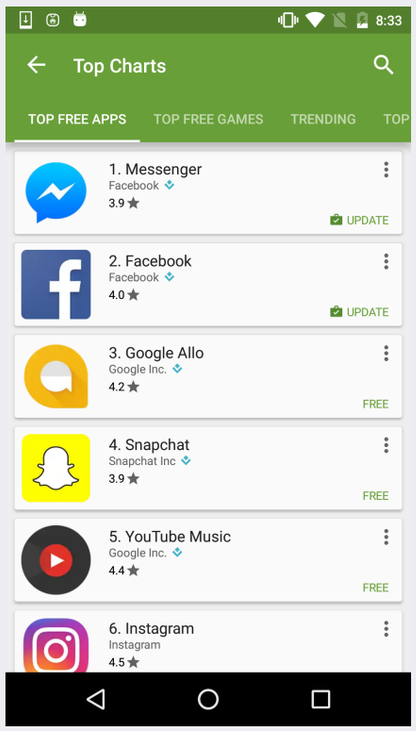


Edge Dilation

*Tuan Anh Nguyen and Christoph Csallner. 2015. Reverse Engineering Mobile Application User Interfaces with REMAUI, In Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE '15)

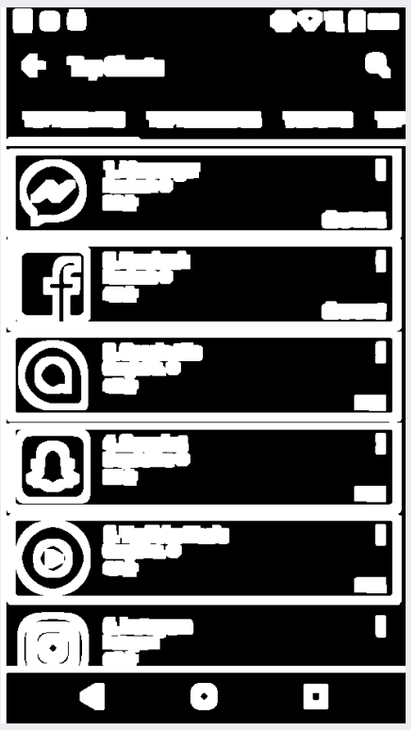
Phase 1: GUI-Component Detection

Computer Vision-Based Detection*

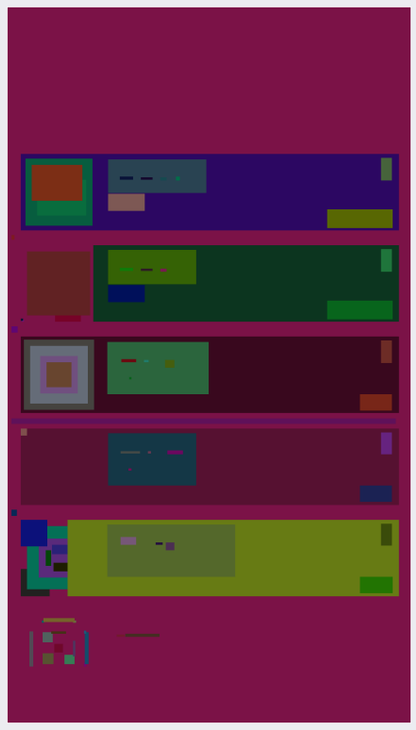


Input Image

Canny's Edge Detection



Edge Dilation

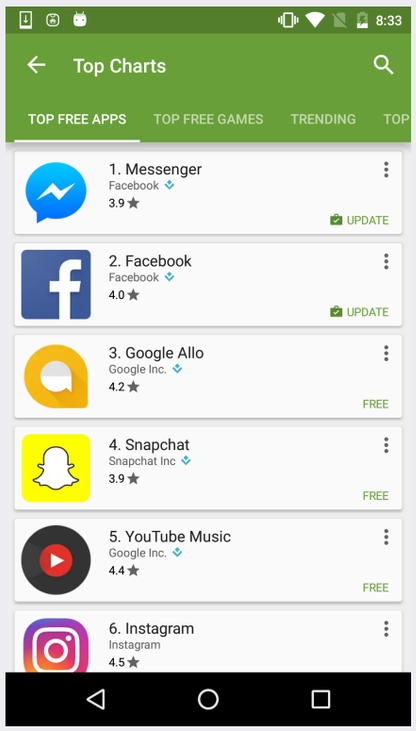


Contour Bounding Boxes

*Tuan Anh Nguyen and Christoph Csallner. 2015. Reverse Engineering Mobile Application User Interfaces with REMAUI, In Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE '15)

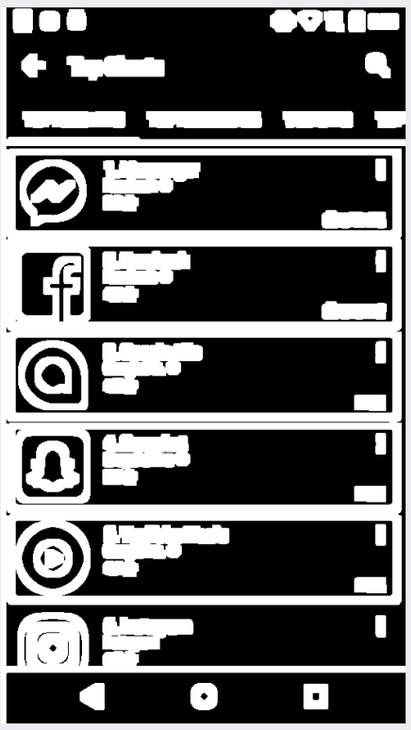
Phase 1: GUI-Component Detection

Computer Vision-Based Detection*

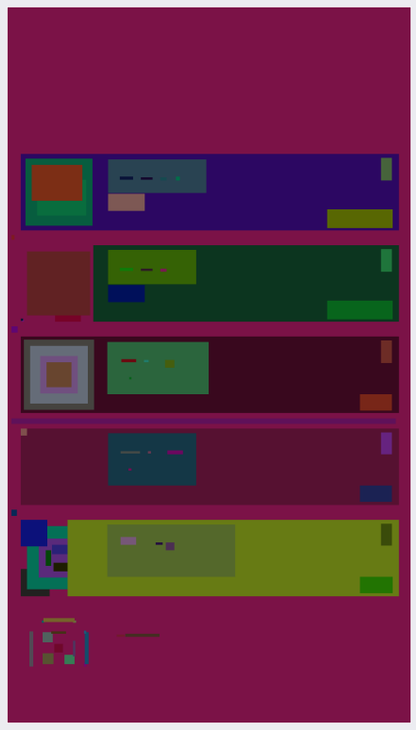


Input Image

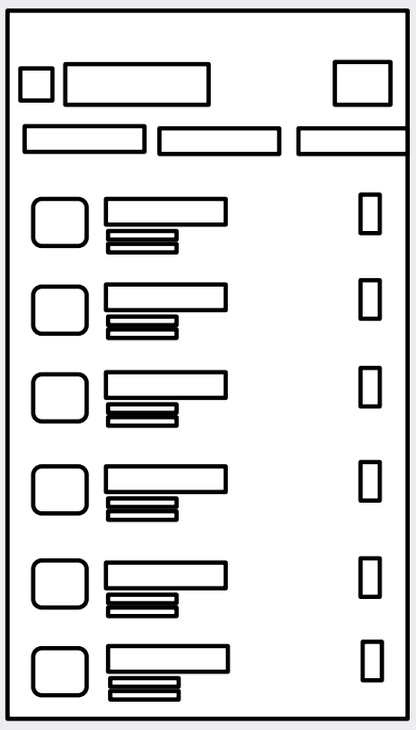
Canny's Edge Detection



Edge Dilation



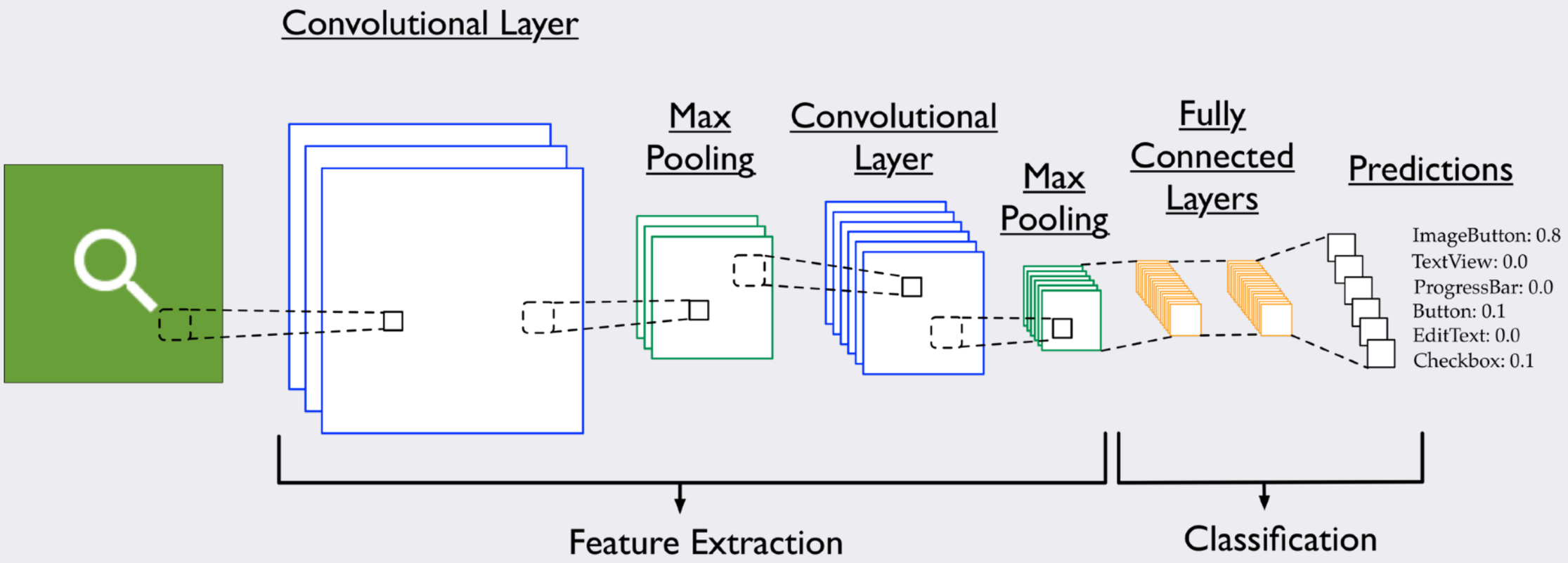
Contour Bounding Boxes



GUI-Component Bounding Boxes

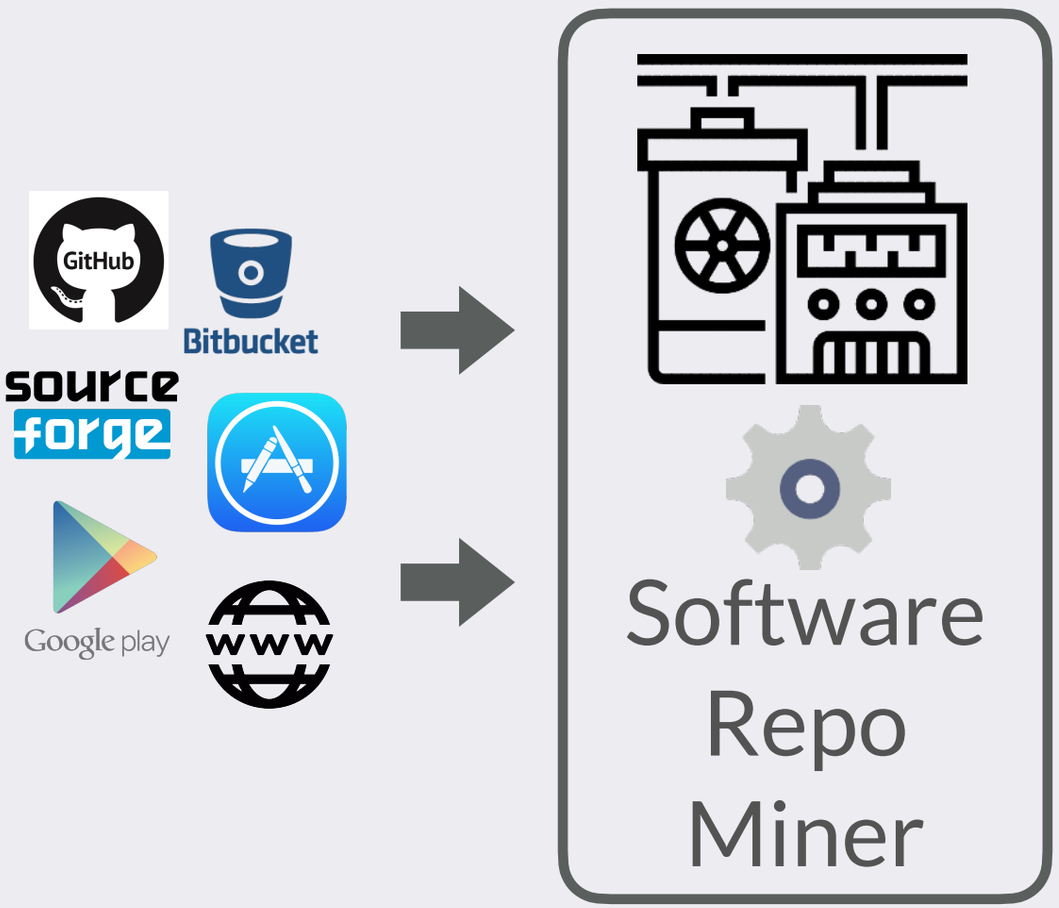
*Tuan Anh Nguyen and Christoph Csallner. 2015. Reverse Engineering Mobile Application User Interfaces with REMAUI, In Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE '15)

PHASE 2: GUI-COMPONENT CLASSIFICATION

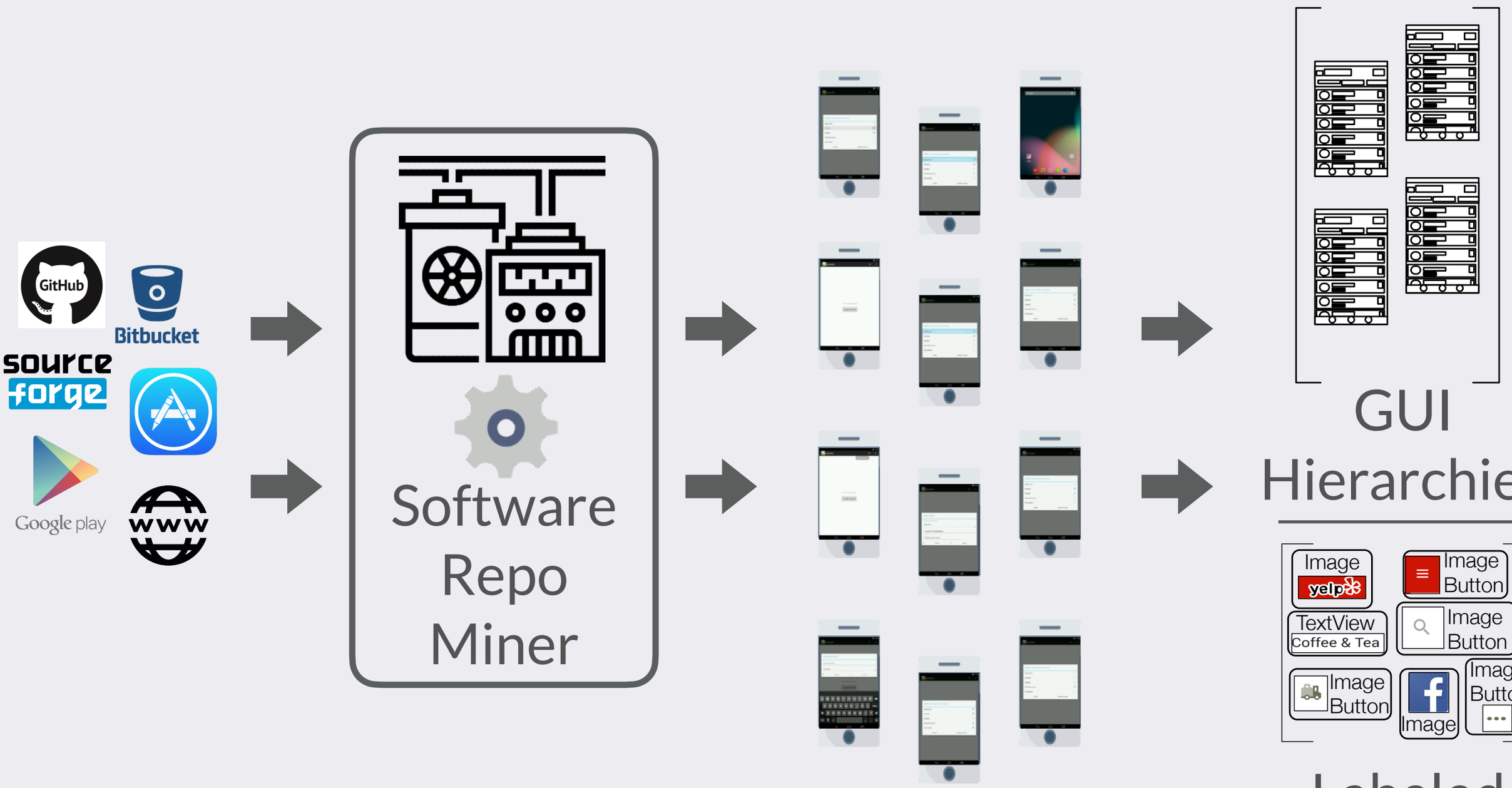


Phase 2: GUI-Component Classification

PHASE 2: GUI-COMPONENT CLASSIFICATION



Phase 2: GUI-Component Classification



Applications Generated By ReDraw

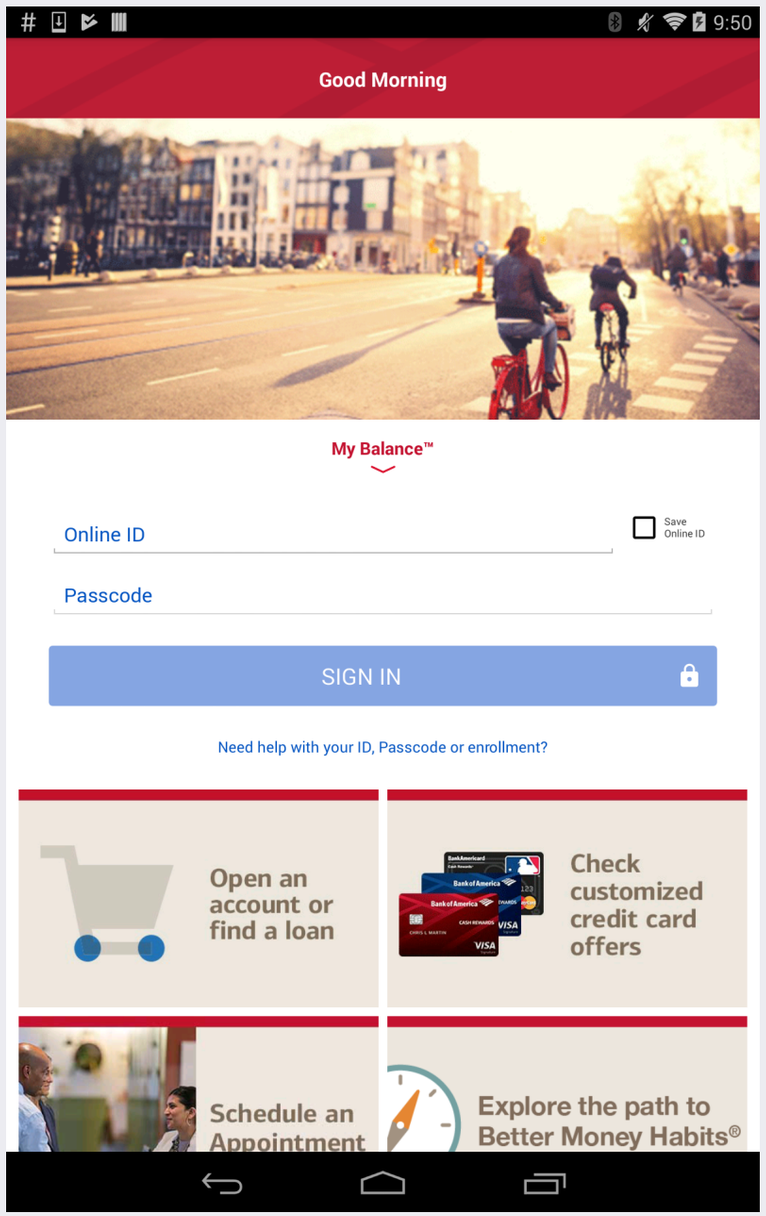
Textgram

A) Original Application

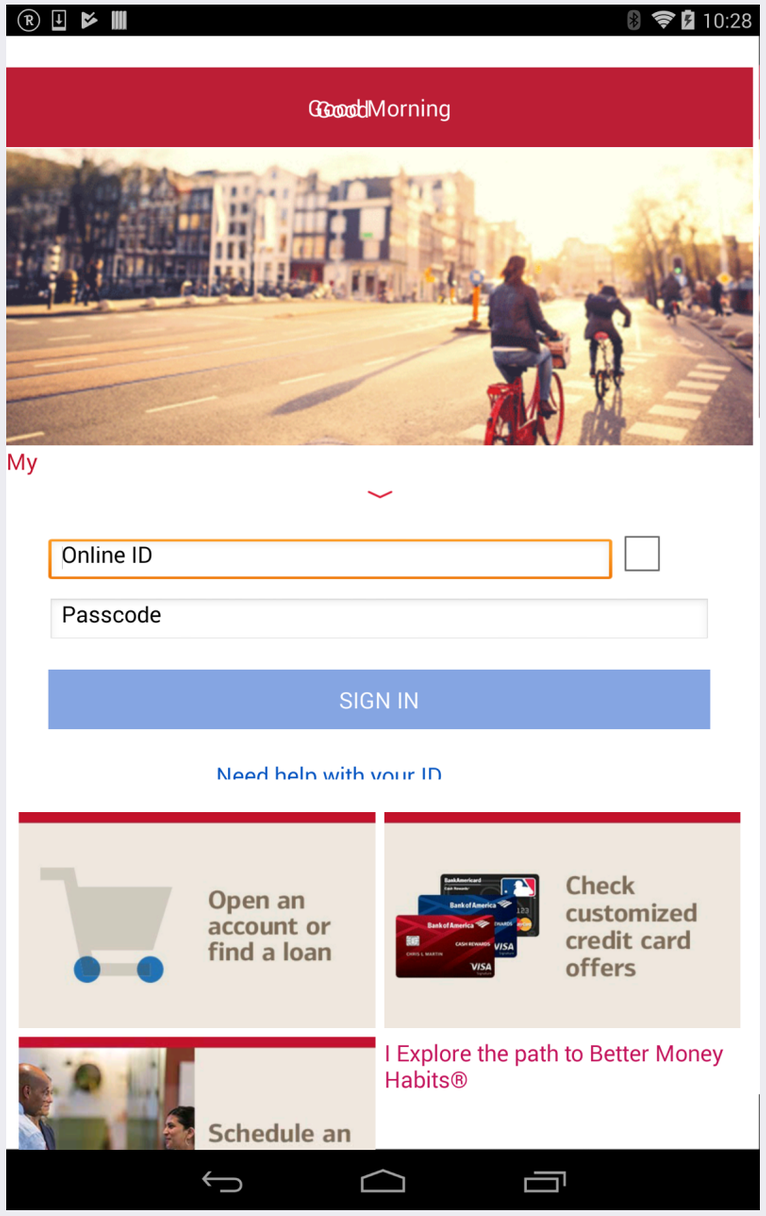
B) ReDraw App (MockUp)

Applications Generated By ReDraw

Bank of America



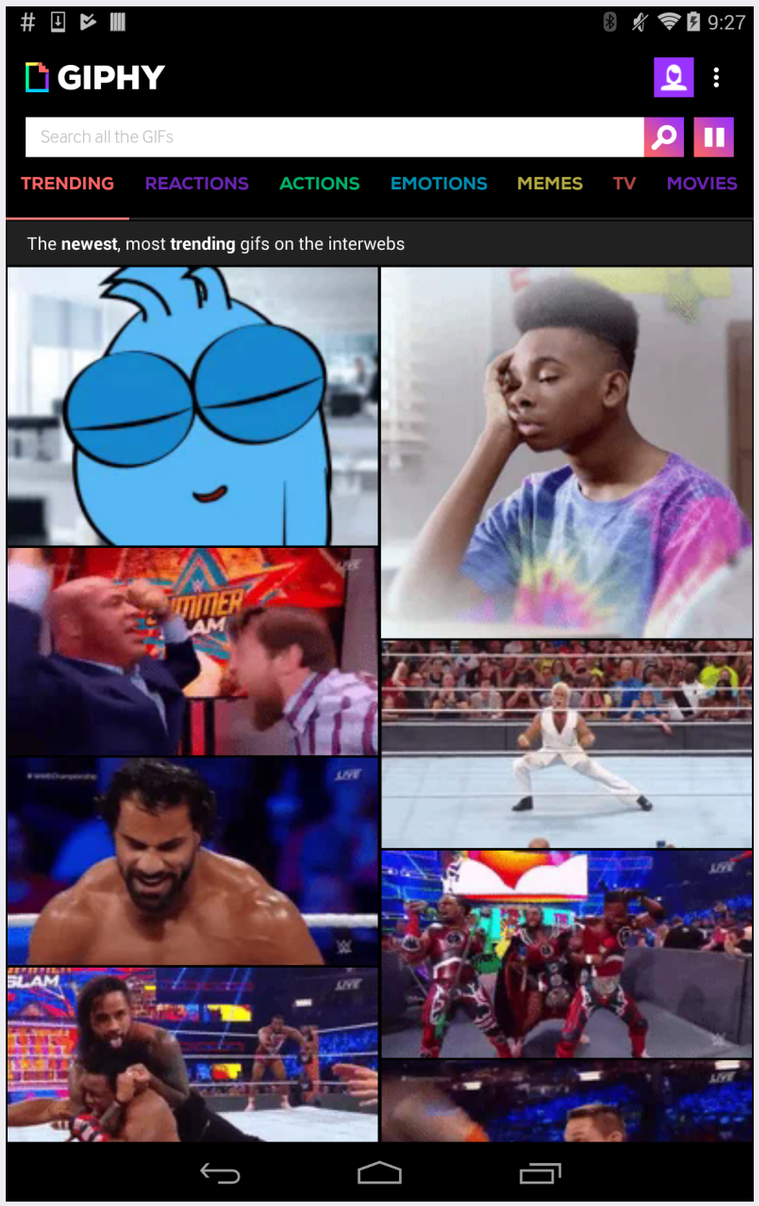
A) Original Application



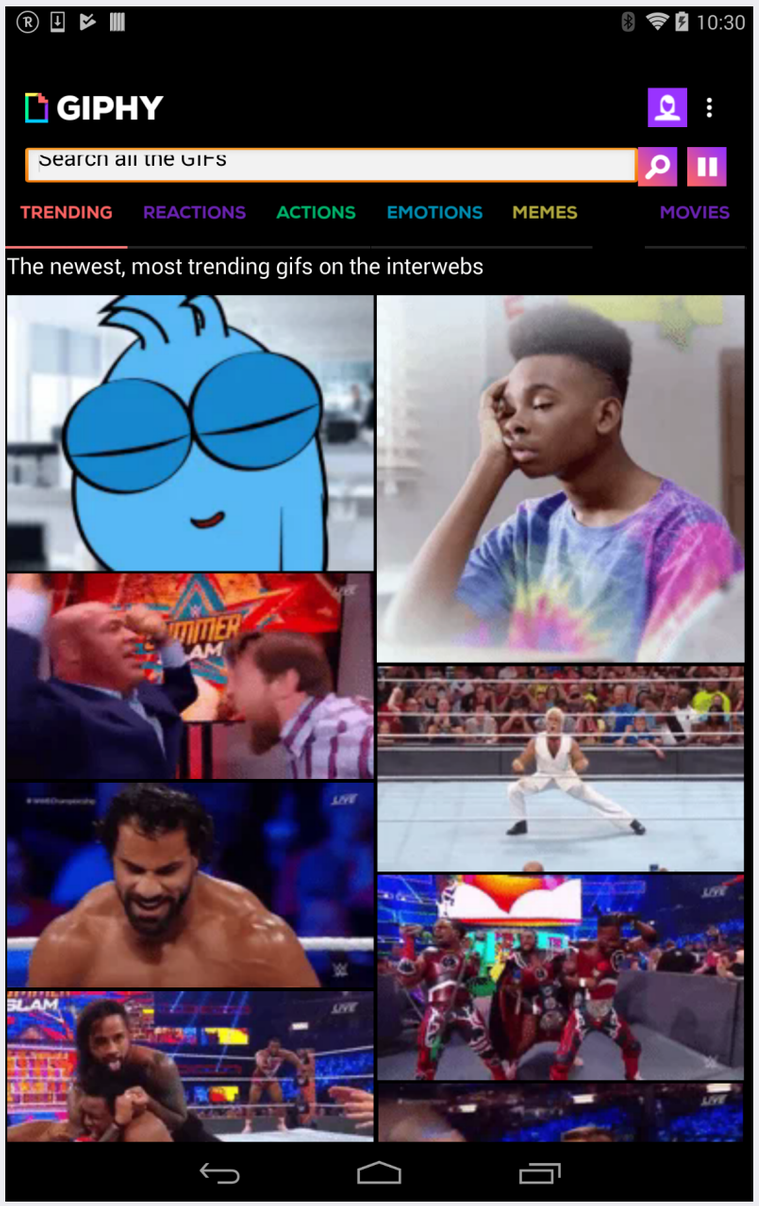
B) ReDraw App (MockUp)

Applications Generated By ReDraw

Giphy



A) Original Application

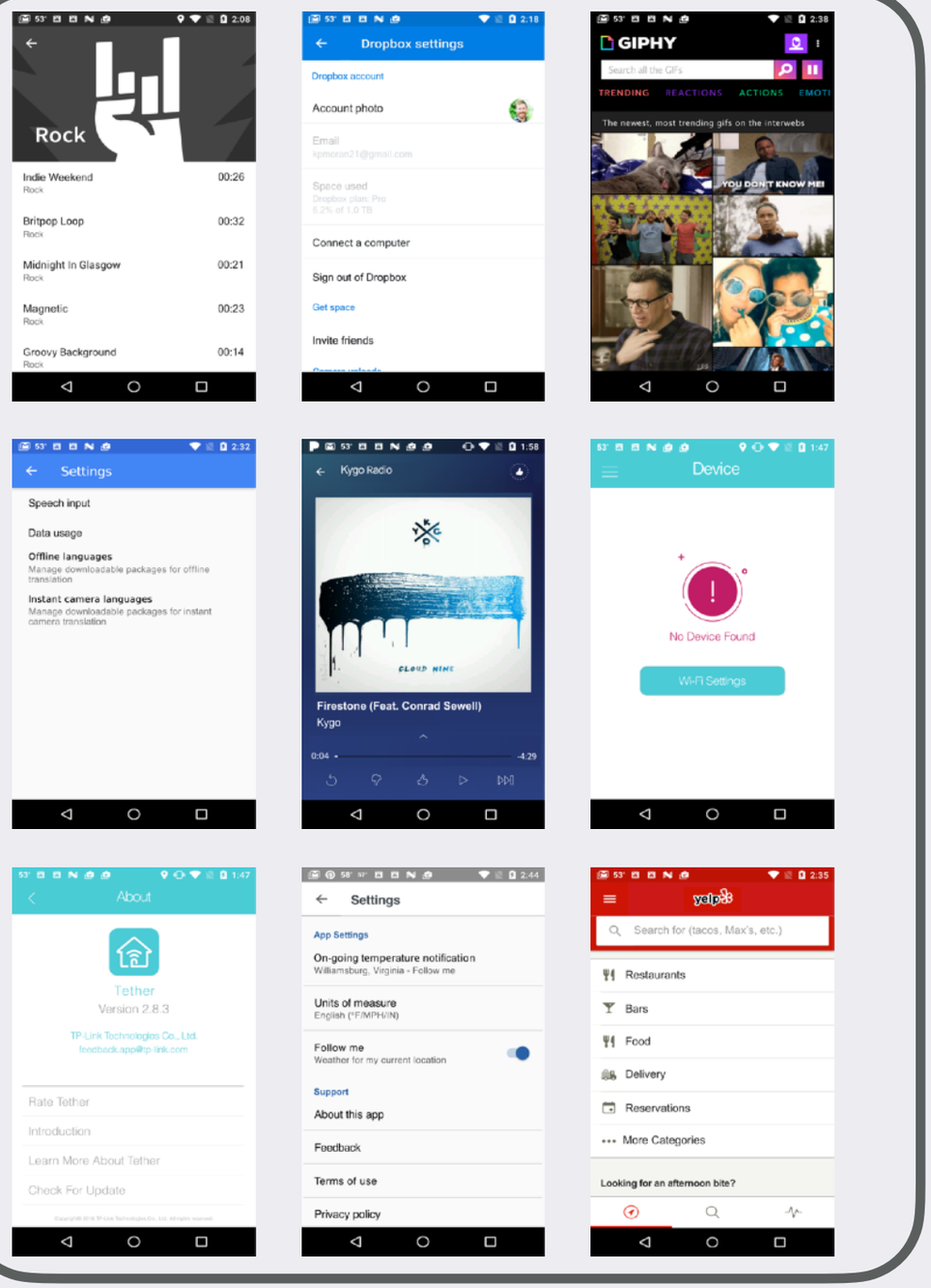


B) ReDraw App (CV)

Study: Developer Utility

Context

9 Screens from 8 Popular Apps



Mobile Front-End Developer

Google

Mobile Designer

HUAWEI

Mobile Researcher

facebook

Study: Results

Study: Results

"It's a good starting point... From a development standpoint, the thing I would appreciate most is getting a lot of the boilerplate code done [automatically]"

Study: Results

"It's a good starting point... From a development standpoint, the thing I would appreciate most is getting a lot of the boilerplate code done [automatically]"

"There are going to be edge cases for different layouts, but these are easily fixed after the fact"

STUDY: RESULTS

"It's a good starting point... From a development standpoint, the thing I would appreciate most is getting a lot of the boilerplate code done [automatically]"

"There are going to be edge cases for different layouts, but these are easily fixed after the fact"

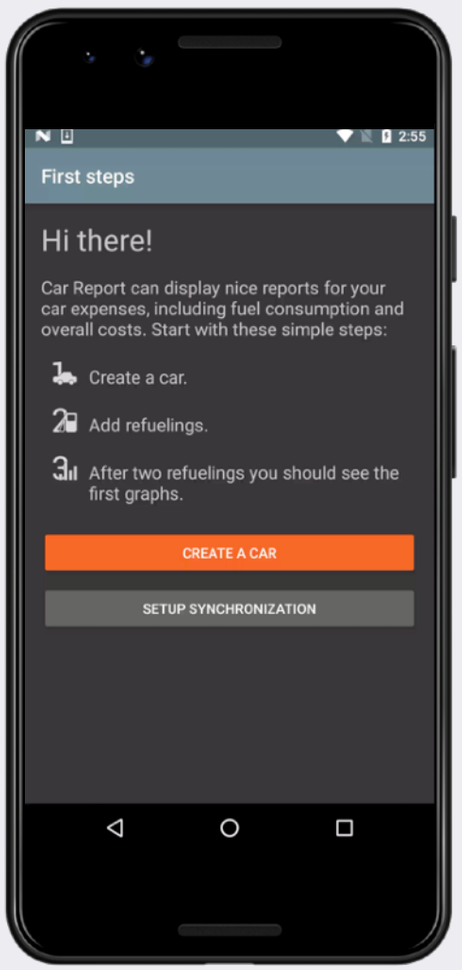
"The key thing is fast iteration. A developer could generate the initial view [using ReDraw], clean up the layouts, and have a working app. If a designer could upload a screenshot, and without any other intervention [ReDraw] could update the [existing] xml this would be ideal."

Translating Video Recordings of Mobile Apps into Replayable Scenarios

Video-Based Bug Reporting

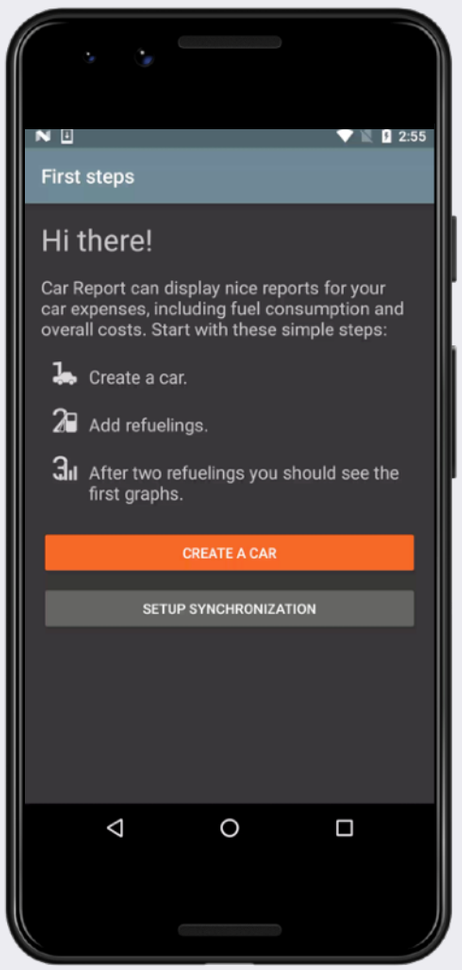
VIDEO-BASED BUG REPORTING

Our Solution: Video2Scenario (V2S)



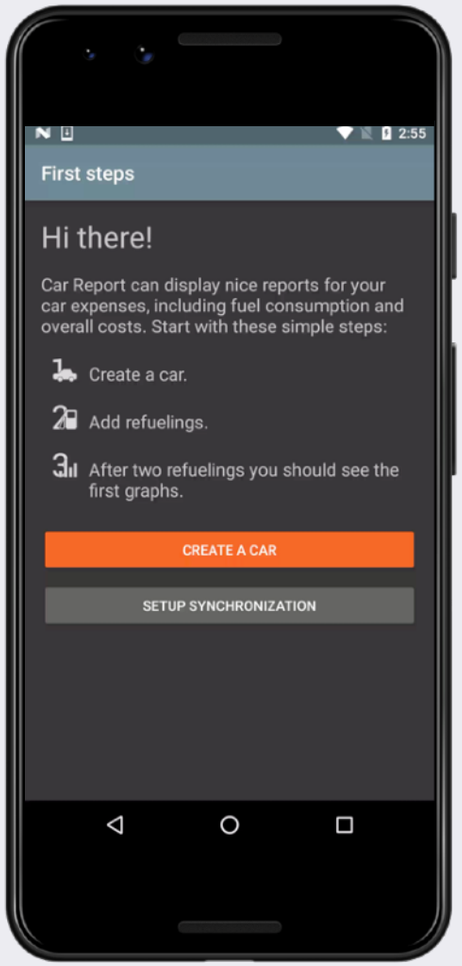
Input Screen Recording

Our Solution: Video2Scenario (V2S)



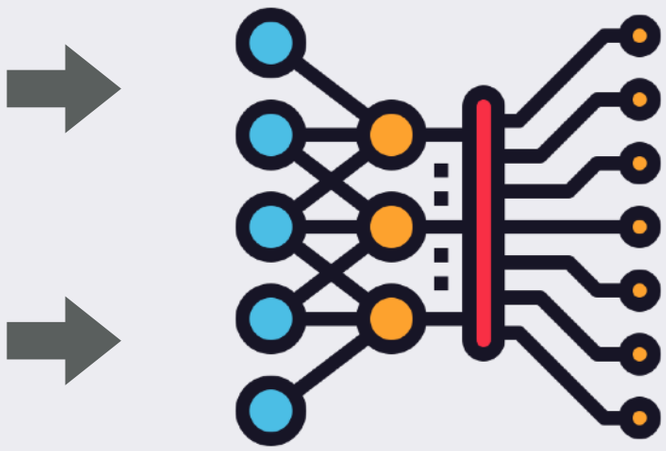
Input Screen Recording

Our Solution: Video2Scenario (V2S)

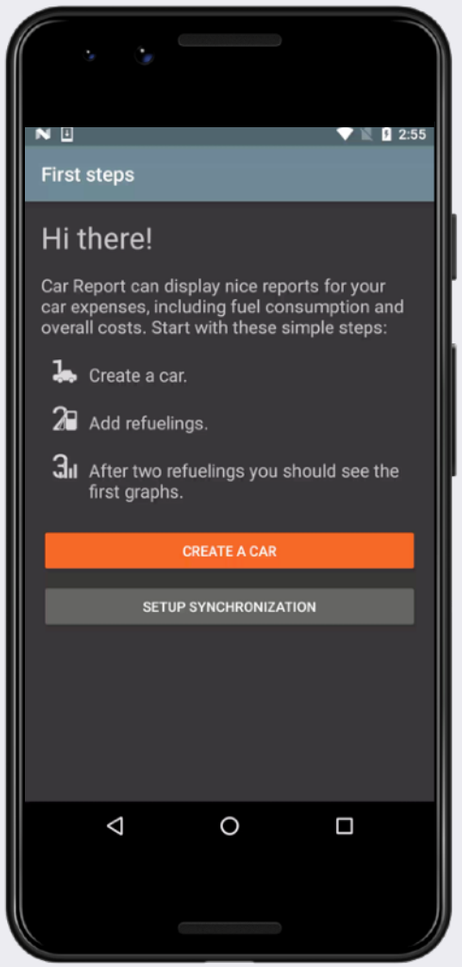


Input Screen Recording

Neural Object Detection & Classification

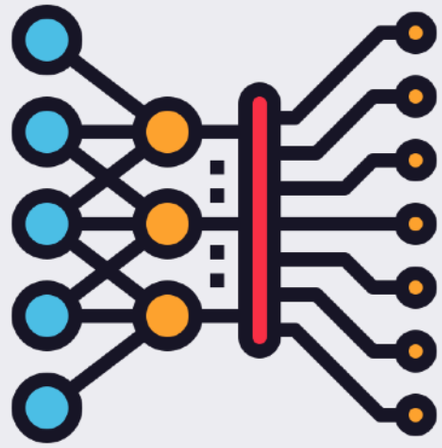


Our Solution: Video2Scenario (V2S)



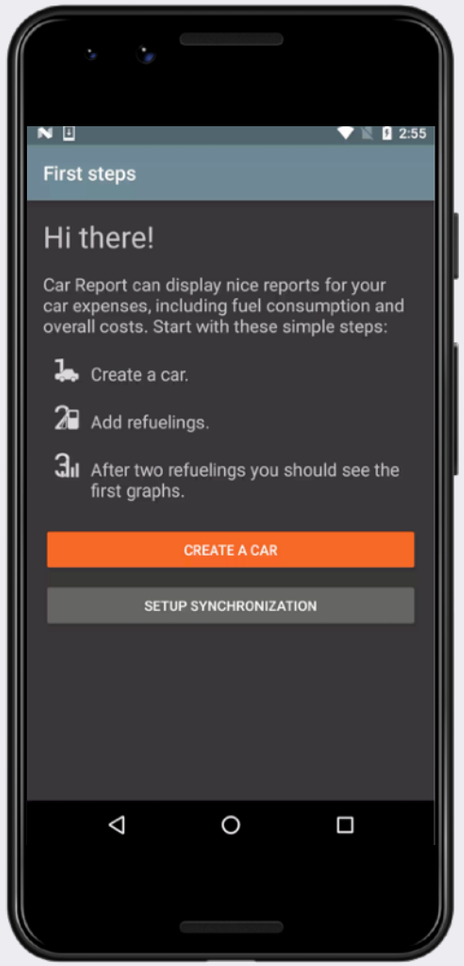
Input Screen Recording

Neural Object Detection & Classification



Replay Script

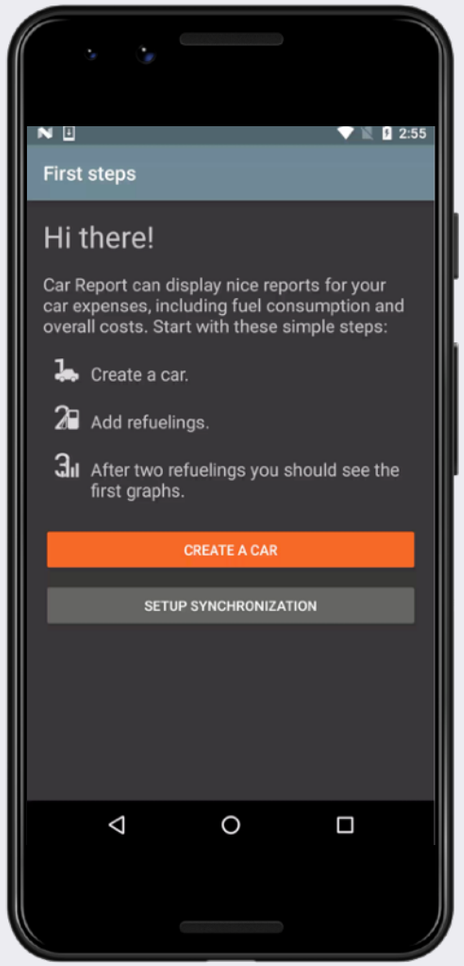
Our Solution: Video2Scenario (V2S)



Accurate
Scenario Replay

Replay Script

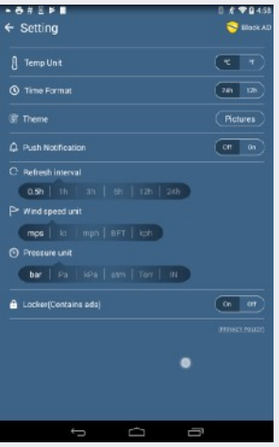
Our Solution: Video2Scenario (V2S)



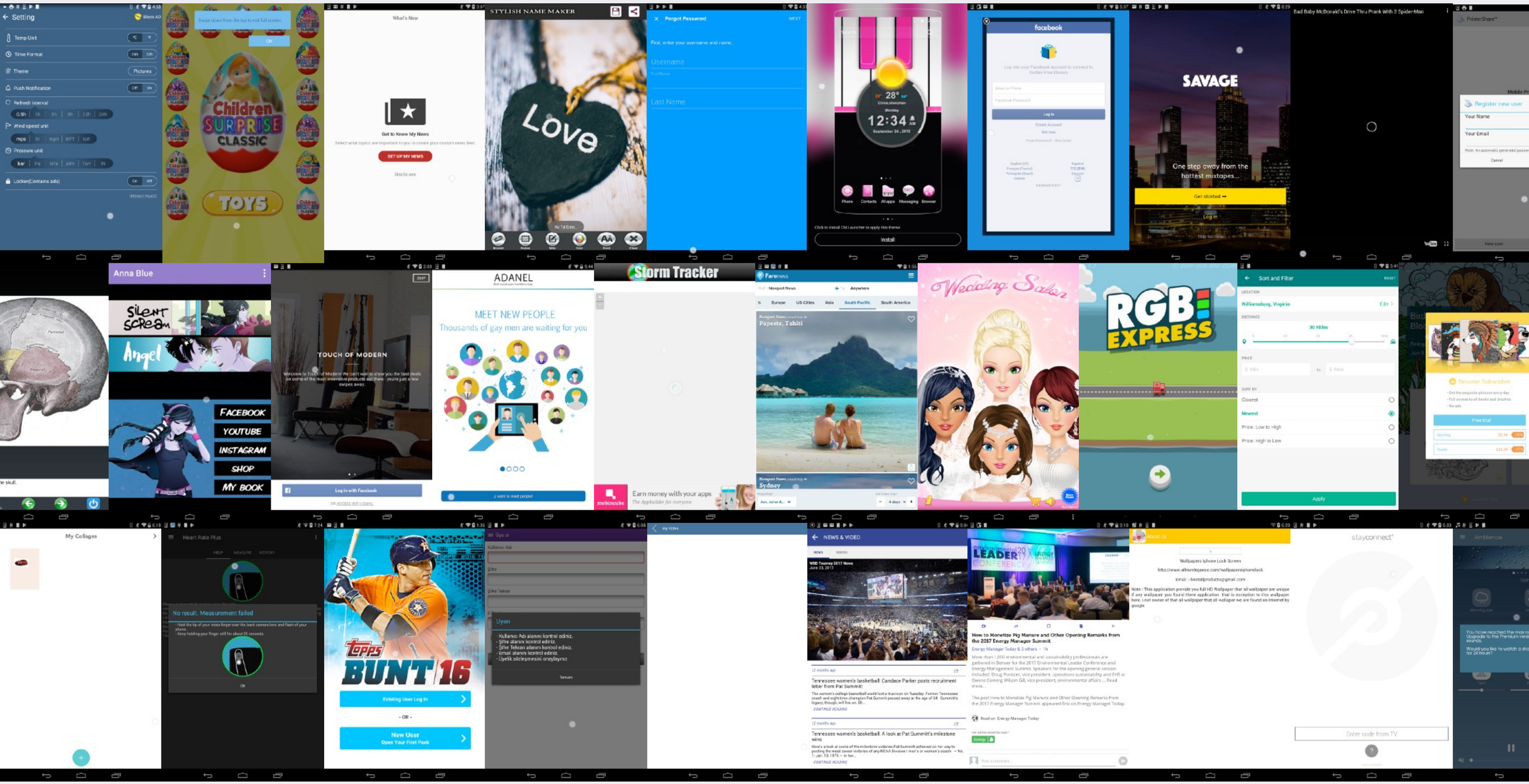
Accurate
Scenario Replay

Replay Script

Phase 1: Touch Detection



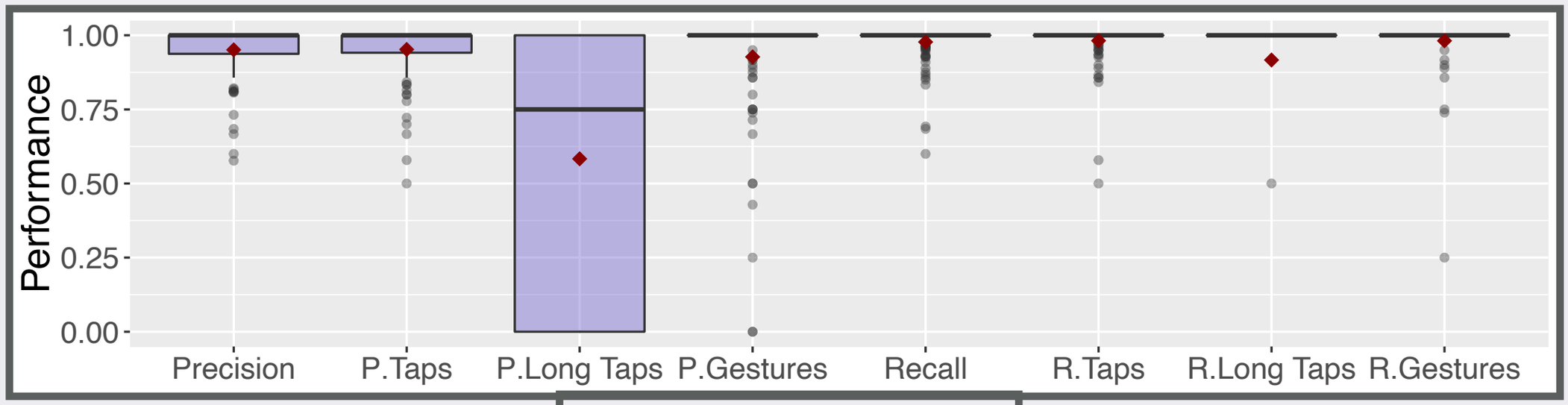
Phase 1: Touch Detection



Accuracy of Replay

Scenario Replay Precision & Recall

Controlled Study



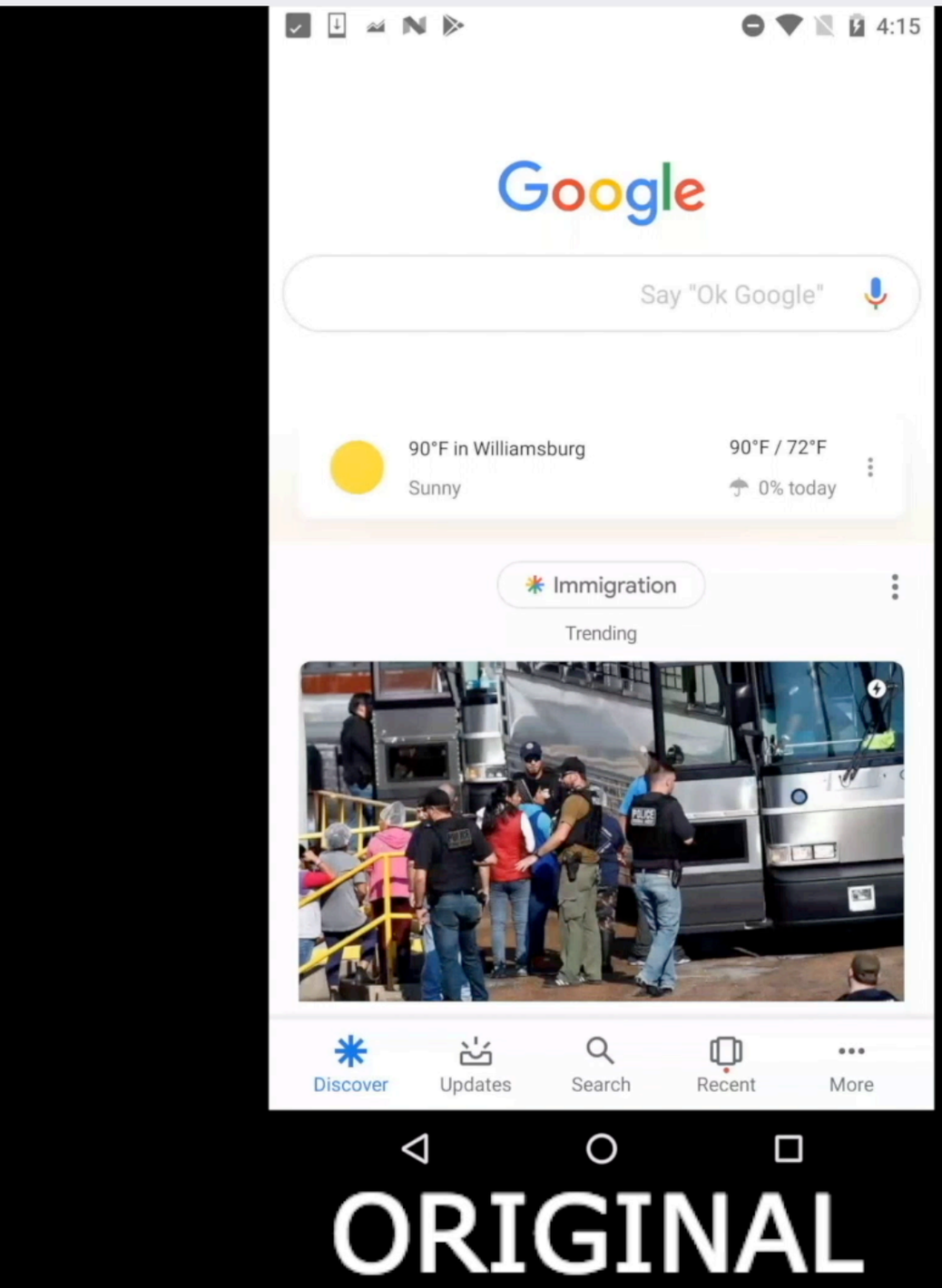
Popular Apps

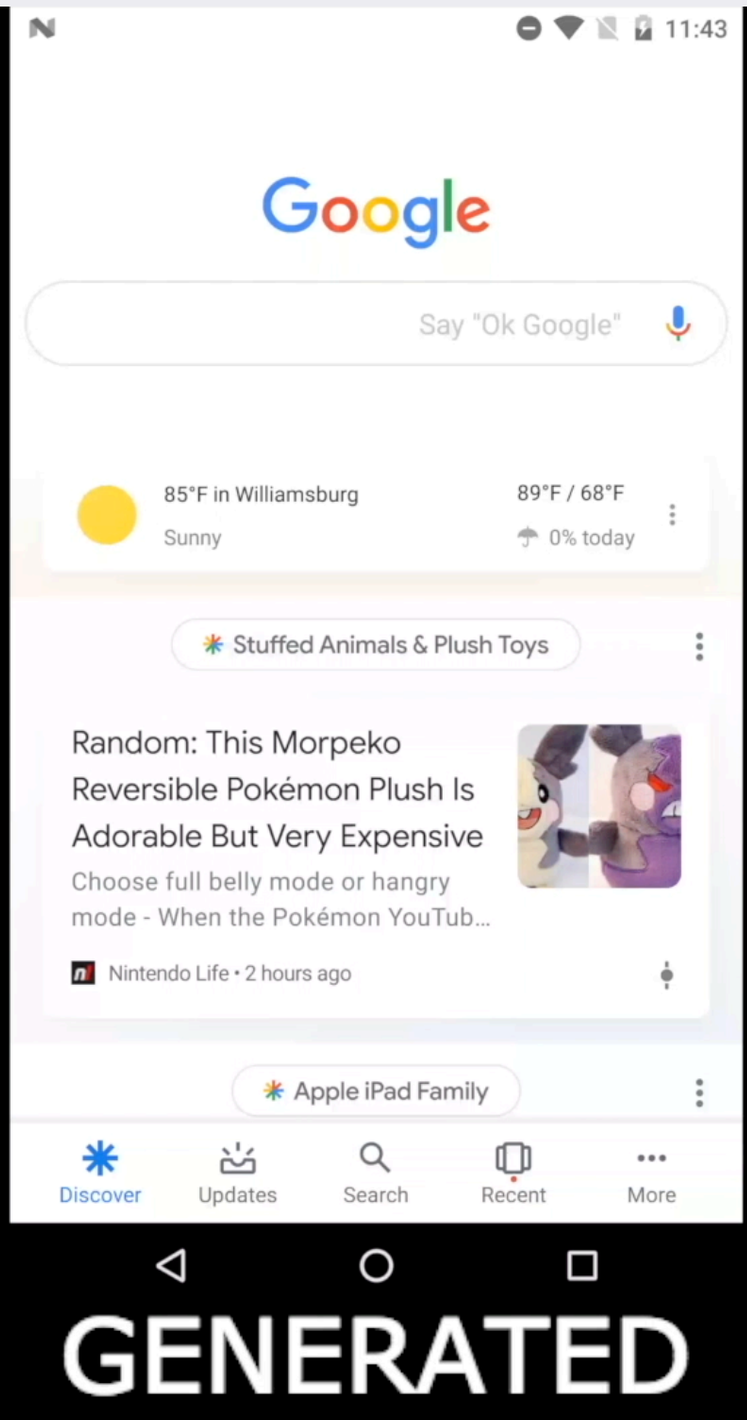
Accuracy of Replay

Scenario Replay Precision & Recall

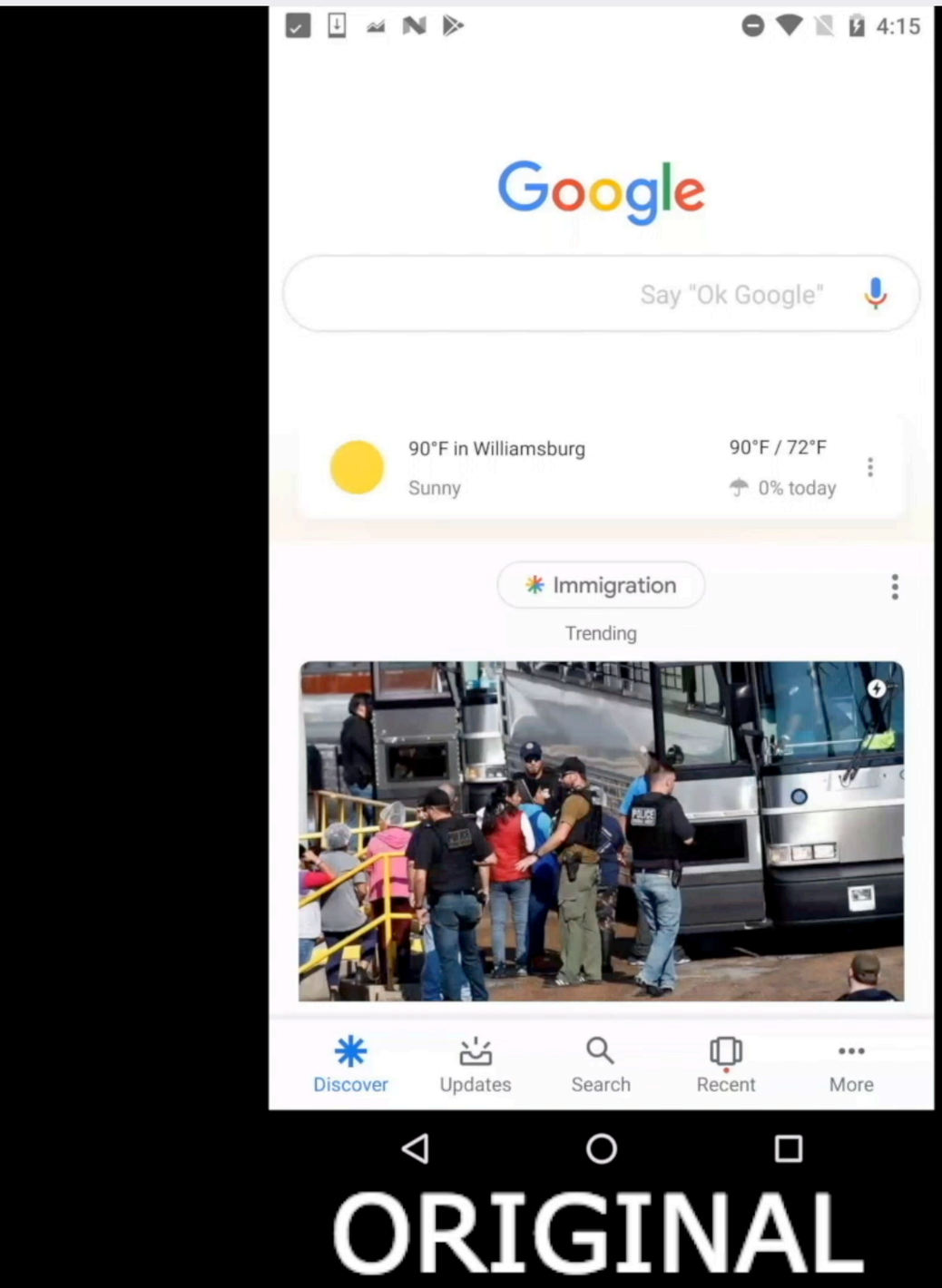
Overall, precision and recall are ***~95%*** and ***~98%*** respectively.

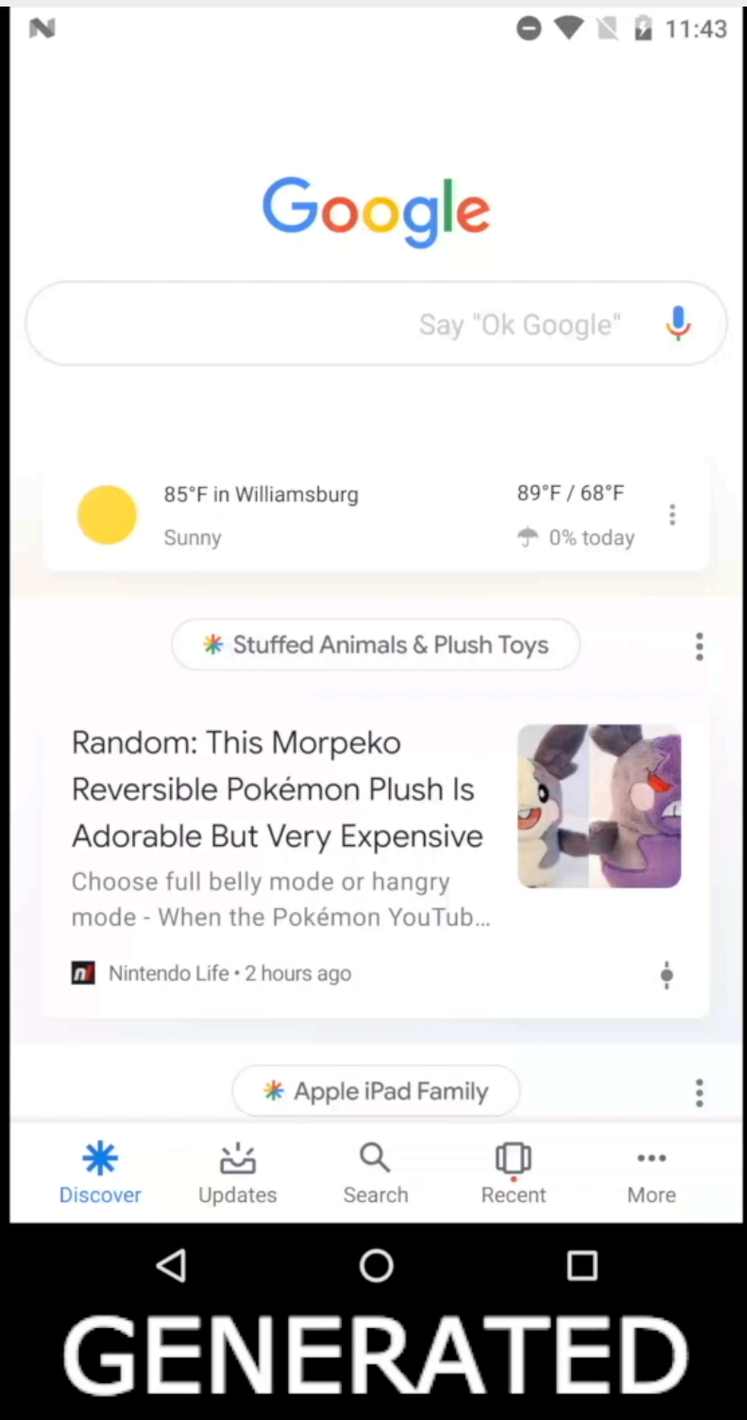
RQ3: Accuracy of Replay





RQ3: Accuracy of Replay





Overview of Other Research Areas & Future Work

Research Areas & Future Work

Research Areas & Future Work

Mobile Software Development

Research Areas & Future Work

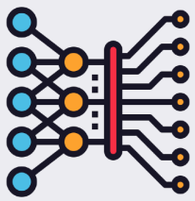
Mobile Software Development

Deep Learning & Software Engineering

Research Areas & Future Work

Mobile Software Development

Deep Learning & Software Engineering



Transforming Bug Reporting

Research Areas & Future Work

Mobile Software Development

Deep Learning & Software Engineering

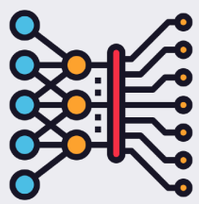
Transforming Bug Reporting

Improving Software Traceability

Research Areas & Future Work

Mobile Software Development

Deep Learning & Software Engineering



Transforming Bug Reporting

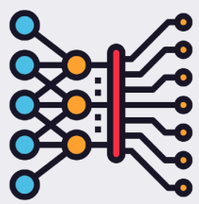
Improving Software Traceability

Developer-Centric Software Security

Research Areas & Future Work

Mobile Software Development

Deep Learning & Software Engineering



Transforming Bug Reporting

Improving Software Traceability

Developer-Centric Software Security

Automating Development of Accessibility Features

Research Areas & Future Work

Applied Techniques:

Machine Learning
Natural Language Processing
Computer Vision
Program Analysis

Join the Research Group!

SWE Research Group

Dr. Kevin Moran
Assistant Professor

Join the Research Group!

SWE Research Group

Dr. Kevin Moran
Assistant Professor

You? { Next Generation Bug Reporting Systems

You? { NLP for Code Generation

You? { Computer Vision for Software Engineering

You? { Software Engineering for Machine Learning

Join the Research Group!

SWE Research Group

Dr. Kevin Moran
Assistant Professor

You? { Next Generation Bug Reporting Systems

You? { NLP for Code Generation

You? { Computer Vision for Software Engineering

You? { Software Engineering for Machine Learning

Academic Collaborators

University of Minnesota

UT Dallas — The University of Texas at Dallas

William & Mary — Chartered 1693

Georgia Tech

Università degli Studi del Sannio

USI — Università della Svizzera italiana

Join the Research Group!

SWE Research Group

Dr. Kevin Moran
Assistant Professor

You? { Next Generation Bug Reporting Systems

You? { NLP for Code Generation

You? { Computer Vision for Software Engineering

You? { Software Engineering for Machine Learning

Academic Collaborators

Industrial Collaborators

Thank you!

Kevin Moran
Assistant Professor
kpmoran@gmu.edu
https://www.kpmoran.com

Feel free to come chat (virtually)!

Additional Slides