MICHIGAN STATE

UNIVERSITY

Project Plan

Image Recognition Annotation and Validation Mobile Application

The Capstone Experience

Team Whirlpool

Shruti Avutapalli

Jessica Clappison

Jackie Li

Savanna Pinkoski

Jack Turak

Department of Computer Science and Engineering

Michigan State University

Fall 2018

From Students…
…to Professionals

Functional Specifications

- **Image Recognition Annotation**
  - Image capture through a mobile devices
  - TensorFlow Lite object detection
    - Annotate bounded food items via Yummly API
- **Provide Valid Training Data (Validation)**
  - Gamification Mechanism
    - Queue-in annotation submissions at random to be verified
      - ❖ Validation threshold
        - » Above – passed on to Yummly API
        - » Below – manual admin review needed
- **Leaderboard**
  - Promote internal (Whirlpool) competition
    - Increase user annotation submissions
    - Increase user verifications

*Overall objective is to provide Yummly with a vast range of data for a reliable training data set, so their system can learn and suggest meal recipes based off of food items found in users' home. *
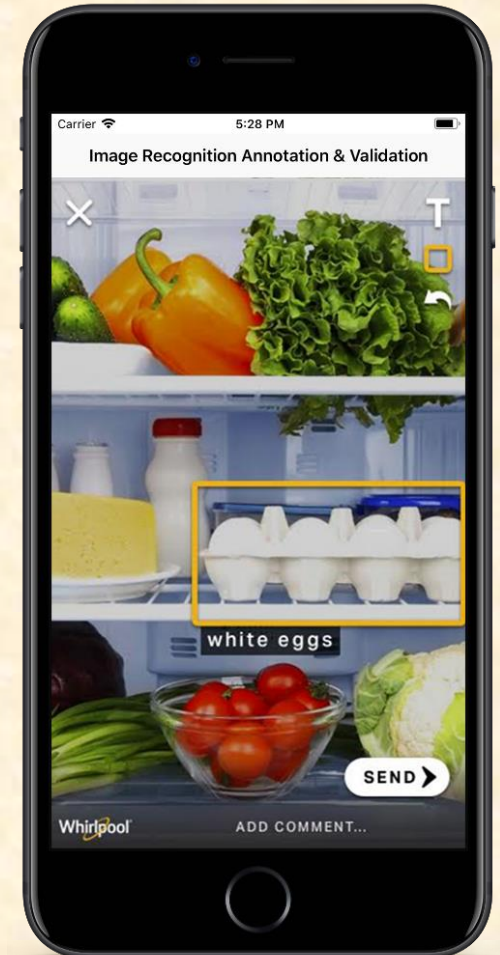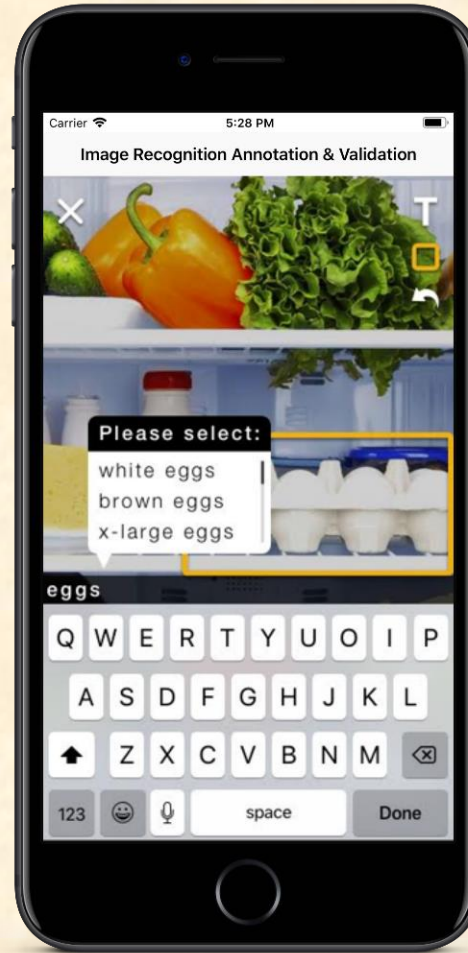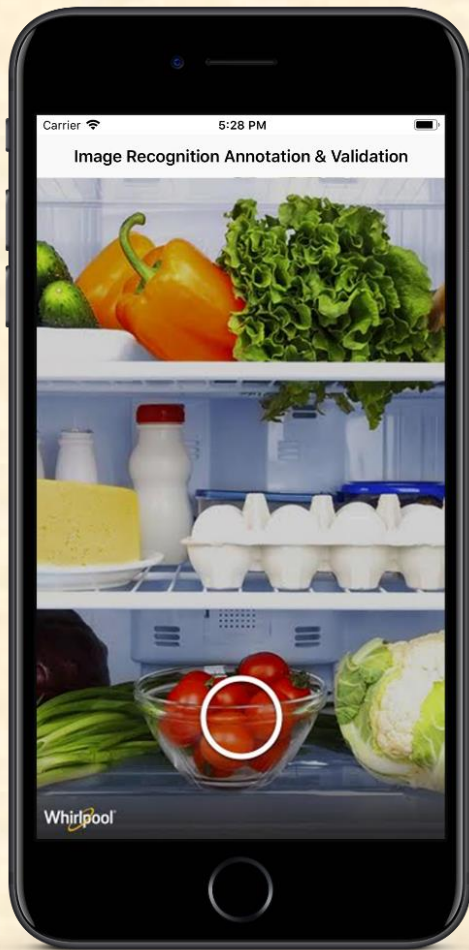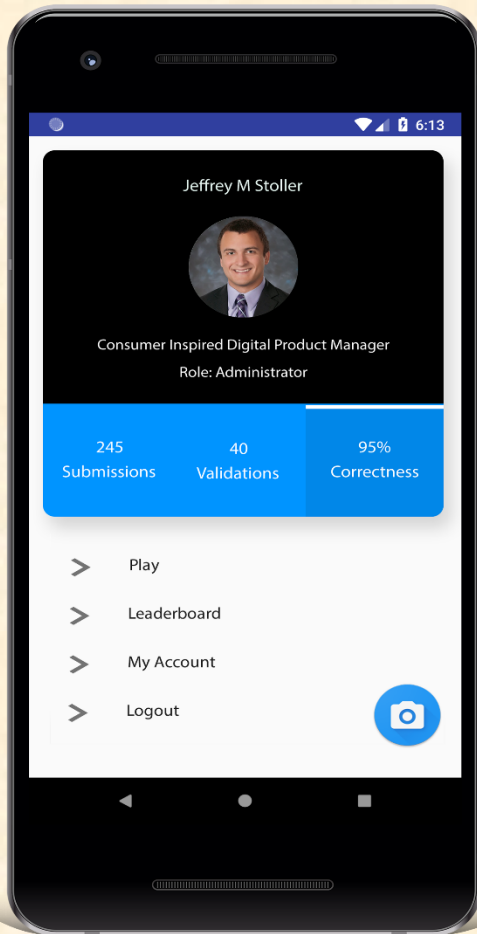
Design Specifications

- **Home Page**
  - Navigation between dashboards
- **Image Capture and Annotation Interface**
  - Camera view & capture
  - Text annotation w/ Yummly API
- **Validation/Game Interface**
  - Provide reliable training data
- **Tutorial**
  - Application walk-through
- **User Stats**
  - User role, scores, & submission history
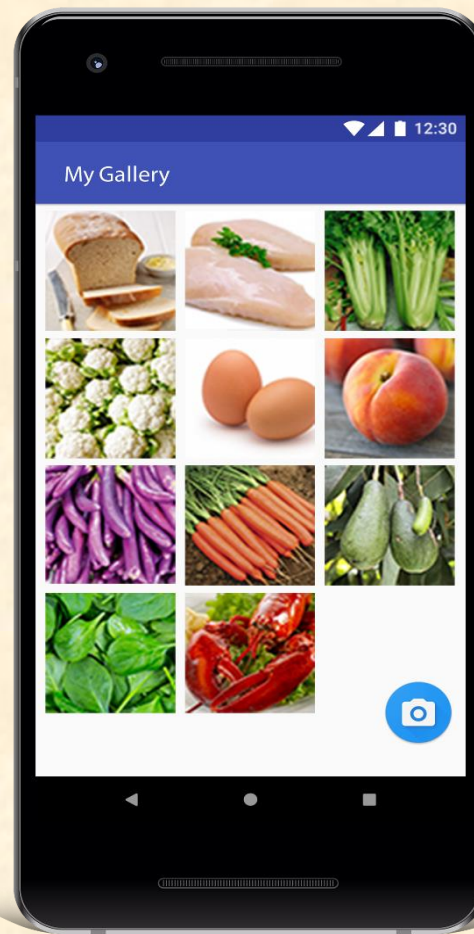- **Leaderboard**
  - Current standings

Screen Mockup: Image Recognition & Annotation Interface (iOS)

Screen Mockup: Android Application

Home Screen Overview

User Submission Gallery

Validation Game

Technical Specifications

- **External Data**
  - Yummly API
    - Text fields to identify detected items
- **Front End –** Native UI
  - iOS – Swift, iOS 11+, CocoaPods
    - Table View, Collection View, Navigation, Tab Bar, Page View, GLKit View Controllers
  - Android – Java, API 21+ (Lollipop)
- **Back End**
  - Firebase
    - Firestore – Database
    - Storage – Image Storage
    - Authentication – Whirlpool Domain, whitelist
  - TensorFlow Lite
    - Object detection
      - Use of bounding boxes

System Architecture

User Creation/Sign-In

Authentication

Data Storage

Firestore

Image Storage

Storage

Android

iOS

TensorFlow Lite
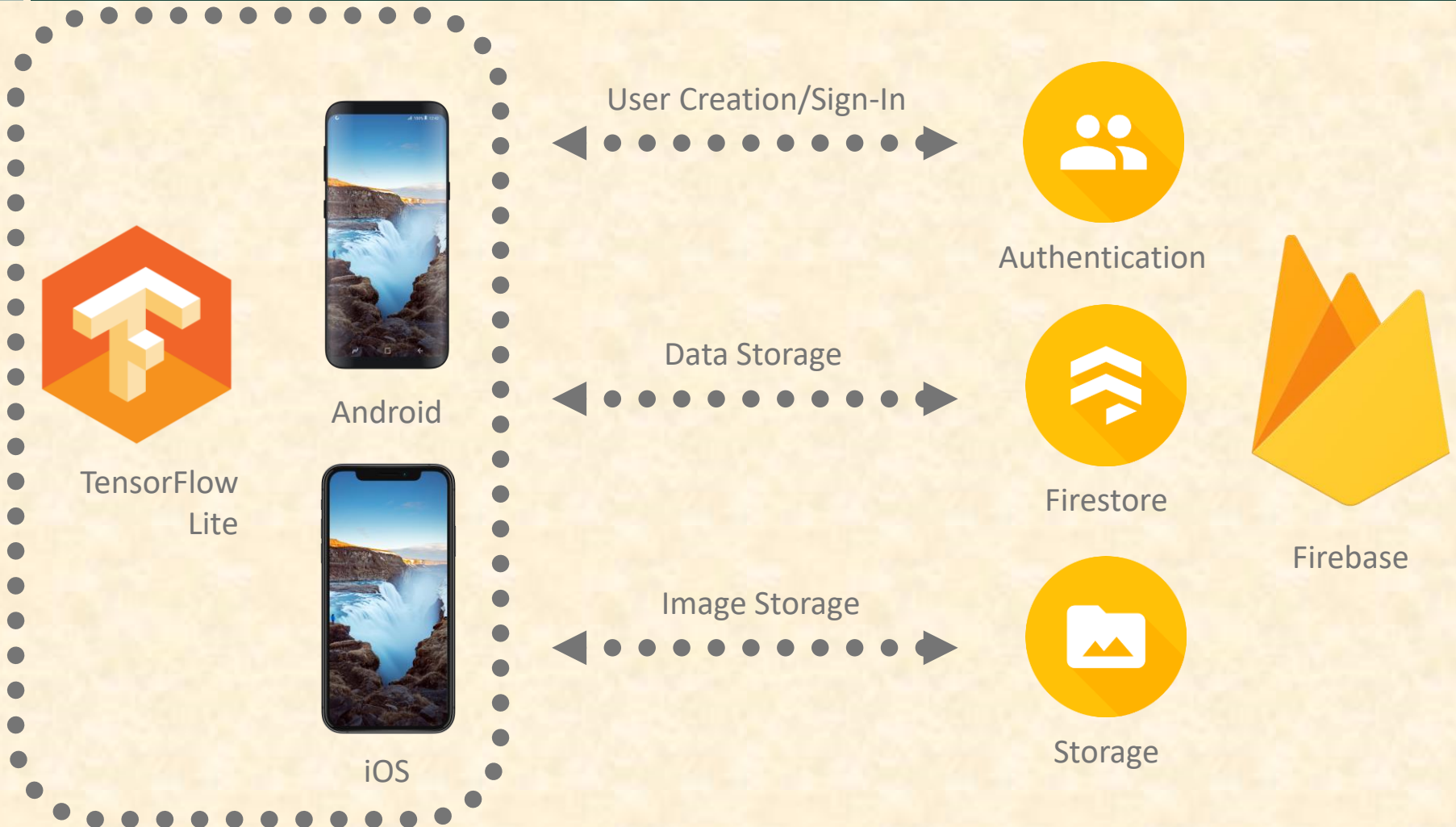
Firebase

System Components

- **Hardware Platforms**
  - Mobile Devices
    - Android
    - Apple
- **Software Platforms / Technologies**
  - Android Studio – Version 3.1.4
  - Xcode - Version 9.4.1
    - Swift – using storyboards
  - Firebase
    - Firestore
    - Storage
    - Authentication
  - TensorFlow Lite

Risks

- ## Non-Uniform Cross Platform UI Design – Medium
  - Description : There is no simple way to ensure that the apps have extremely similar UI appearances, which could cause user confusion and make our final product appear unprofessional.
  - Mitigation : Maintaining consistent collaboration between both IOS and Android team as we produce our app. The overall structure needs to function the same even if the back-end of each app functions differently.

- ## Object Detection in Image – High
  - Description : Currently there is no definitive plan for how we will detect the ingredients in any given image.
  - Mitigation : We have to research and implement TensorFlow Lite to help with object detection. Furthermore we can reach out to other peers who are familiar with TensorFlow Lite.

- ## Suboptimal System Architecture – Low
  - Description : We can access the firebase API and can store and put data on it but we have yet to fully implement firebase in the production of our app. We don't know how reliable, secure, or if there are any hidden restrictions that might limit our progress.
  - Mitigation : Get advice/approval through client. Check that all of its capabilities match all of the all the expectations we have that it will accomplish for us.

- ## Substandard UI – Medium
  - Description : We don't want any confusion when it comes to using our app. Our UI design needs to be intuitive and user friendly.
  - Mitigation : Have people outside our team test our app as we finalize it to ensure a user friendly UI.

Questions?