Dietise

UCI Master of Computer Science

MEET THE TEAM

Eric Cai
eacai@uci.edu

Backend AI, Database Developer
Project Manager

Created two noSQL databases in MongoDB and Firebase Firestore. Used Python to create scripts that cleaned and managed the data for the databases. Used Python's Keras API to make a Food Image Classification Model transfer learned on MobileNet V2.

Matt Bilello
mlbilell@uci.edu

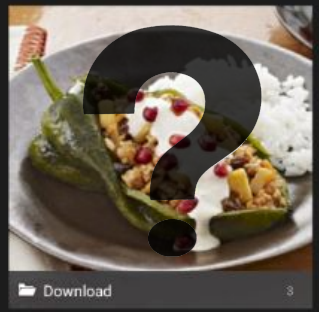
Front-end iOS React Developer.

Built cross-platform application with focus on the iOS native application. Learned Javascript along with React/React Native frameworks in order to implement the front-end application. Features include user sign-up, user sign-in, API calls to Firebase for recipe/user data in the database, camera use for taking pictures, displaying information, navigating between screens smoothly, and hook variables to save state locally and globally.

Zhenyuan (Kevin) Gao
zhenyug1@uci.edu

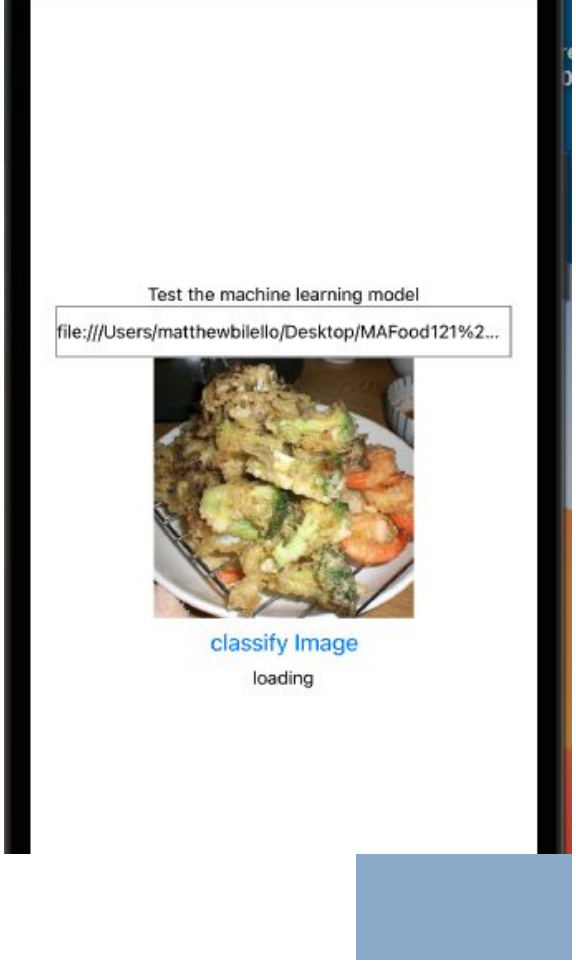
Front-end Android React Developer

Utilized React Native FIrebase API to fetch data, add data and delete data if necessary.
Created the UI for android.
Integrated multiple functionalities including camera, ML model integration, Firebase recipe display with the help of useContext, useState, hook variables and other libraries.



Problem Statement

A positive side effect of the current global pandemic, is that an increasing number of people are finding joy in cooking meals. With the increased interest in cooking, people are looking for better ways to find recipes. **However, how can we find a recipe for a delicious-looking dish when we don't know what the dish is?**



Solution

Our product uses a **novel AI food image classification model to identify dishes** from a picture taken by the user and recommends appropriate recipes.

Product Preview

Product Overview

Our product is a recipe recommendation application.

Basic Features Overview

User Login/Sign-up
Search Recipes: (Limited Daily)
- Text Input Search
- Model Prediction Search

Save Favorite Recipes

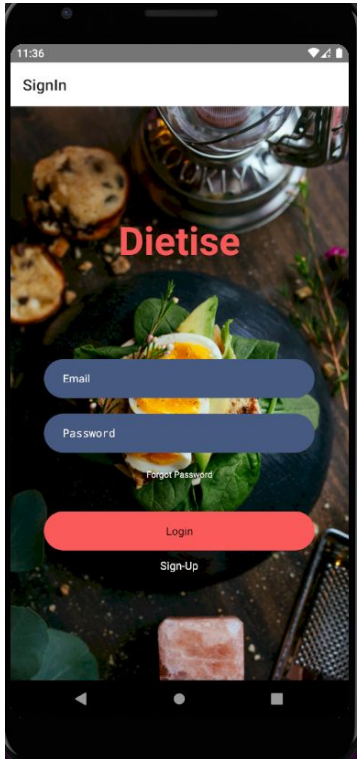
Premium/Paid Features Overview

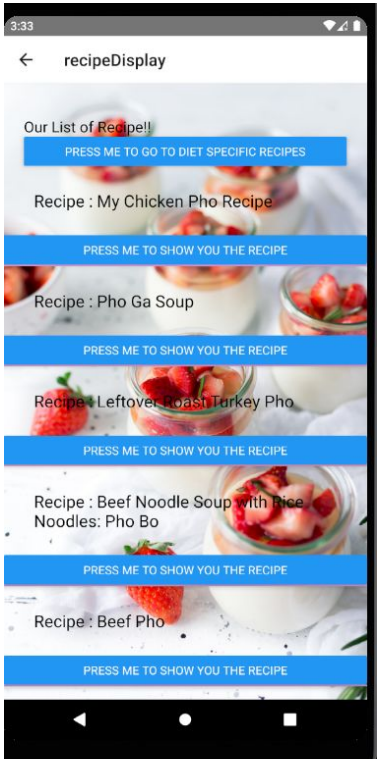
Unlimited Recipe Searches
Save User Diet Information
Filter Recipes Based on Diet Restrictions

Model and Database Overview

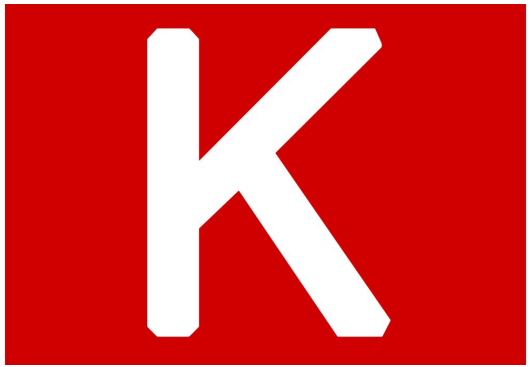
Model:
- Identify 121 unique dishes

Database:
- >100,000 Recipes available





Tech Stack



Python Keras CNN Model

Used Jupyter Notebook and Tensorflow Keras' Python API to create and train a Food Image Classification CNN model. This model was transfer learned using MobileNet V2 as the base model.

Firebase FirestoreDB

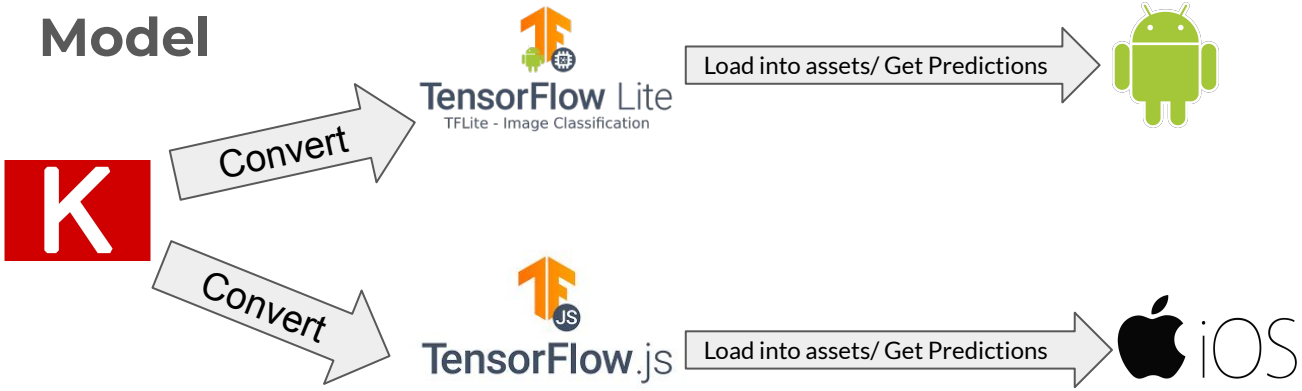
Google Firebase noSQL Database Solution FirestoreDB. We used this to host our database for our recipes, available diet options, and user data. Firebase has packages and API's that are easily connected to React Native

React Native JS

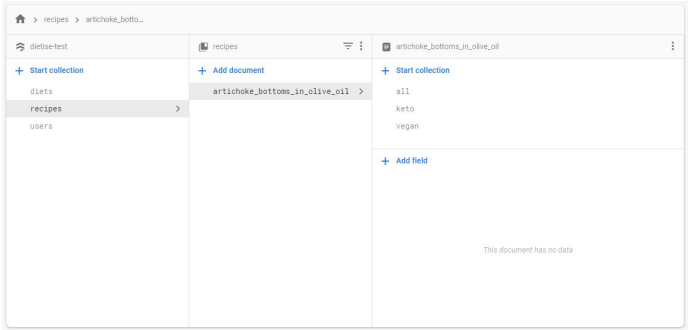
Used React Native JS to build our App. We choose React as it would allow for Cross-Platform Development. We developed both an Android and iOS version of our Application.

Product Architecture

Model

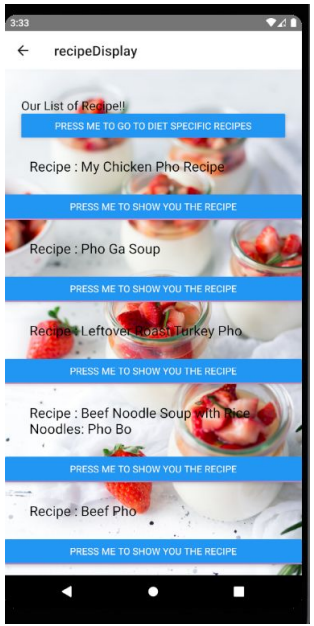


FirestoreDB

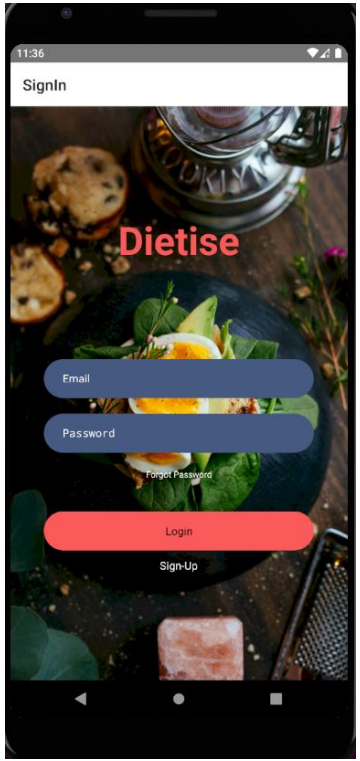


Fetch and Receive Data

Application



Product RoadMap



SPRINT ONE (3/29 - 4/11)

Week 1: Brainstorming Product Ideas, Researching Database Options and Started AI Model Design, Choose React Native for Cross-Platform Development
Week 2: Built Skeleton App using React, Made noSQL MongoDB database for Recipes, and Finished Creating AI Model

SPRINT TWO (4/12 - 4/23)

Week 3: Trained Our Model to 60% Training/Validation Accuracy and Sample Recipe Data into MongoDB Database
Week 4: Made Skeletons for APP Features: Login Page, Home Page, Search Recipe , and Use Camera

SPRINT THREE (4/24 - 5/6)

Week 5: Trained Our Model to 70% Accuracy, Switched to Hosting Database using Firebase FirestoreDB (noSQL) and Integrated Database with App
Week 6: Made Skeletons for APP Features: Display Picture, Display Recipes, Display Profile

SPRINT FOUR (5/7 - 5/18)

Week 7/ Week 8: Finished Implementation for APP Features: Login/Sign-up, Display Profile, Display Recipes, My Recipes, Search Recipe by Name
Integrated Model with App, Added Diet-based Data to FirestoreDB

SPRINT FIVE (5/19 - 6/1)

Week 9: Completed Implementation of Model Id and Search and Pick Diet, Filter Recipes by Diet Features
Week 10: Final Testing, Fix Bugs, Final UI Adjustments