

Personal Food Log APP

GROUP 5 Presentation as:

Ke Yang 300087074

Yi Pang 101100793

Yuhao Shen 300086198

Xu Zhang 500062651



Faculté de génie | Faculty of Engineering

uOttawa.ca



uOttawa

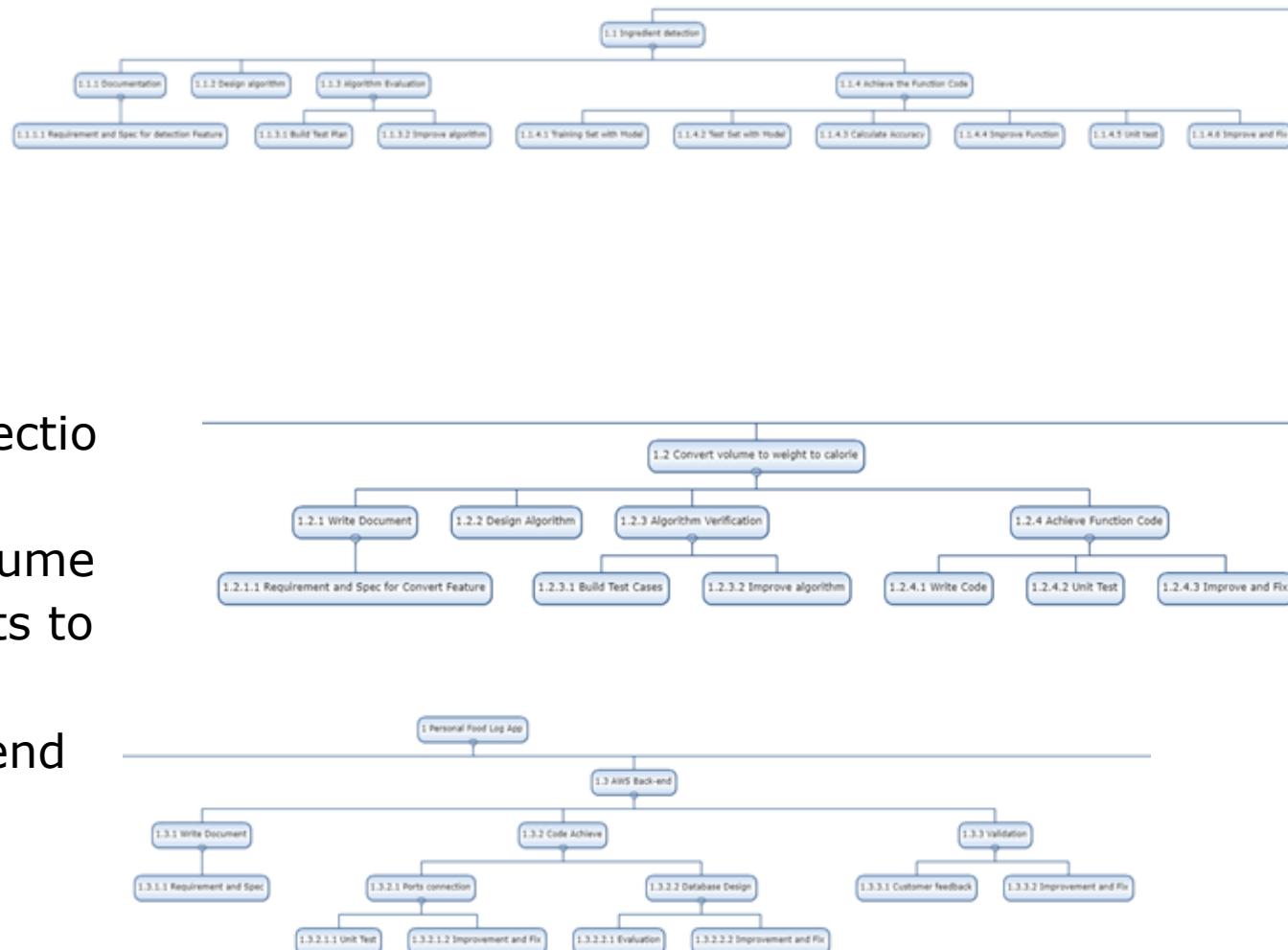
Requirements:

- **Purpose :**
 - Build a mobile app that can detect the food and calculate calories to estimate intakes for the user of DH inc.
- **Objectives :**
 - The project software can realize the function of ingredient detection
 - The project software can realize the function of calories calculation
 - The project if for a mobile App publication

WBS:

- **6 Features**

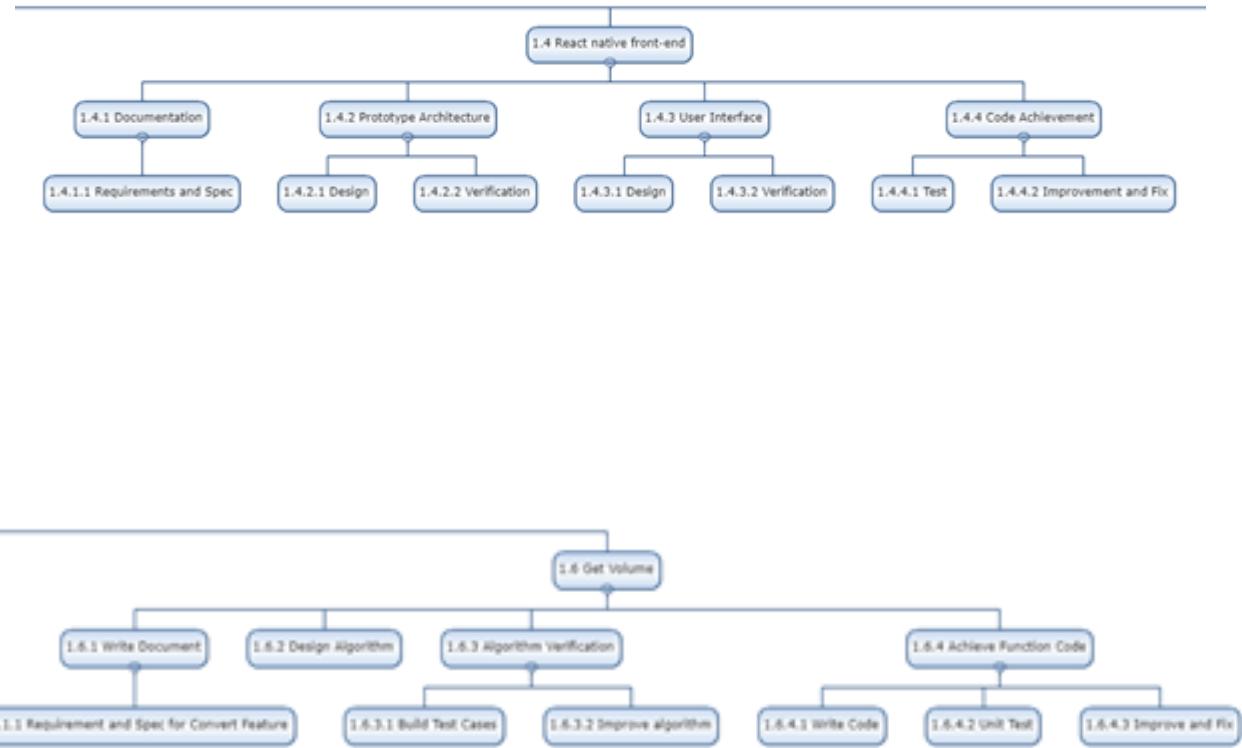
- Ingredient detection
- Convert volume from weights to calories
- AWS back-end



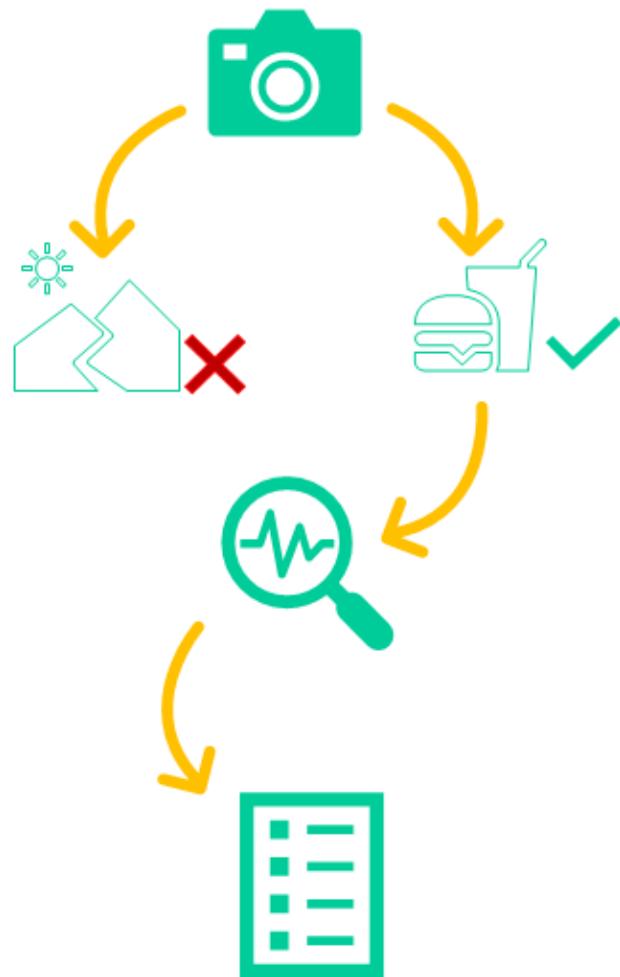
WBS:

- **6 Features**

- React native front-end
- Release test
- Get volume



Design:



- **How it works**

- Take a photo
- Confirm the photo
- Show ingredients
- Calculate calories

Schedule

- Total development duration is estimated to 34 weeks.
 - Pre-work 4 weeks (including recruiting, training)
 - Prototype design and test (1 week).
 - All features starts together, release every 2 weeks
- Within each release in each feature
 - Document confirm
 - Developing or design
 - Test and improvement
 - Release
- Release test

Schedule – one release

Schedule Control

- Schedule tracking
 - Report current progress according to schedule every 2 weeks
- Performance report
 - Every two weeks
- Progress variance solution
 - Re-estimate schedule and give new plan

Budget

- Our main budget comes from the salaries from developers, subcontract, project manager team, etc.
- Another cost comes from the activities and welfare, including work overtime fees.
- All services and contracts are assumed already equipped, for example: computers, software, and test mobile phone, etc.

Budget

	position	skill level	number of personnel	sub-total (CAD)
salaries	Project Manager	senior	1	63,333
	assistants	intermediate	3	112,000
	back-end developers	experienced	8	469,973
	back-end developers	less experienced	7	364,000
	UI designer	intermediate	6	292,000
	test staff	intermediate	6	324,000
	developer analyst	senior	2	114,666
	software experts	senior	1	90,000
activities	once a month, 200 per time			1,600
holiday gifts	3 times within the whole project			3,150
work overtime	maximum 5 hours per week for the whole project team			4080
total				1,838,802

Budget Control

- Cost Management
 - Record within every 2 weeks
 - Report the cost consumed
- Cost control
 - If the cost is over \$4,000 per release, report the problem and re-estimate budget is required.

Risks

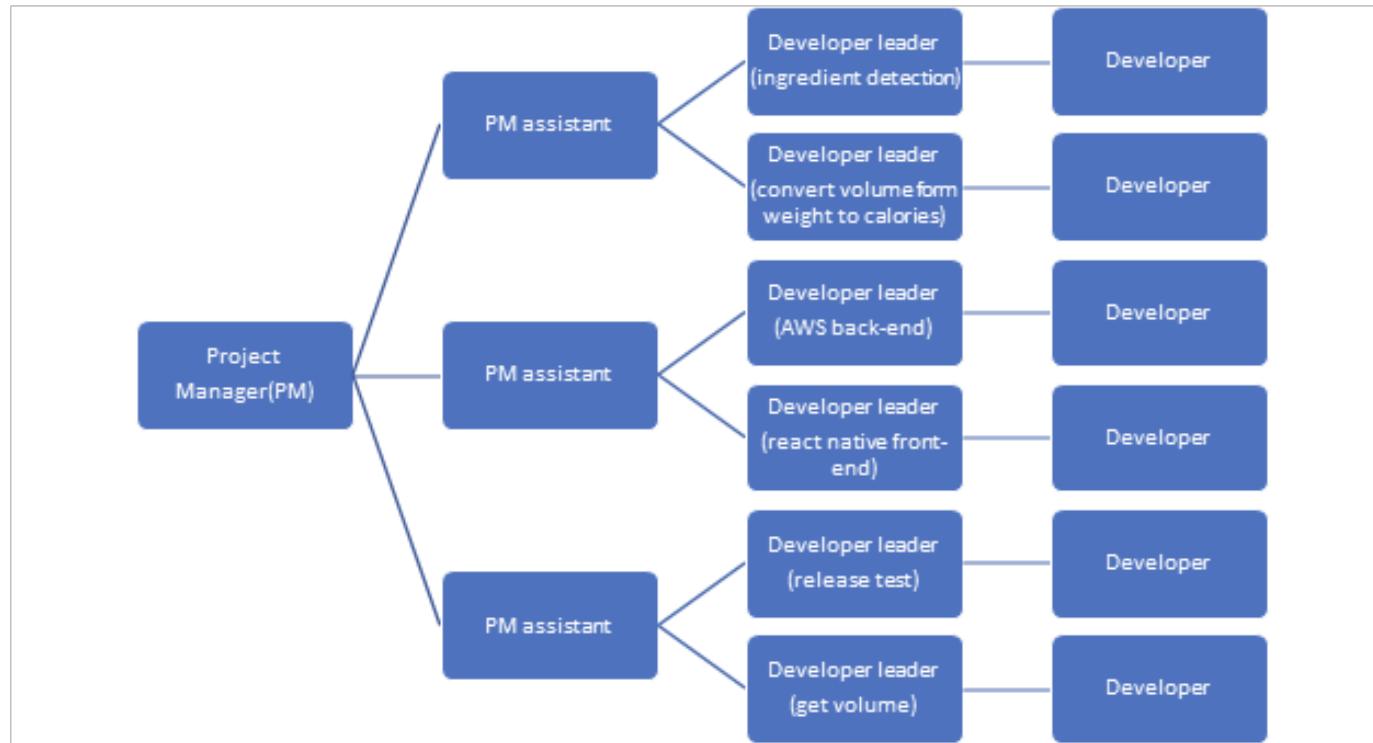
- **High risk:**
 - Subcontractors can not prove equal to work.
 - Solution: contact subcontractors ASAP to find a solution.
- **Medium risk:**
 - Developers quit halfway during development.
 - Solution: prepare extra developers for every features and requirements.
 - Computer break down during development.
 - Solution: using multiple storage ways to keep data safe.
- **Low risk**
 - Staff in team catch a sick.
 - Solution: find substitute staff ASAP

Organization:

In this project, it's a better choice to use **projectized structure**.

- All resource concentrate on one goal.
- Manager has stronger control of all activities.

Organization: internal structure

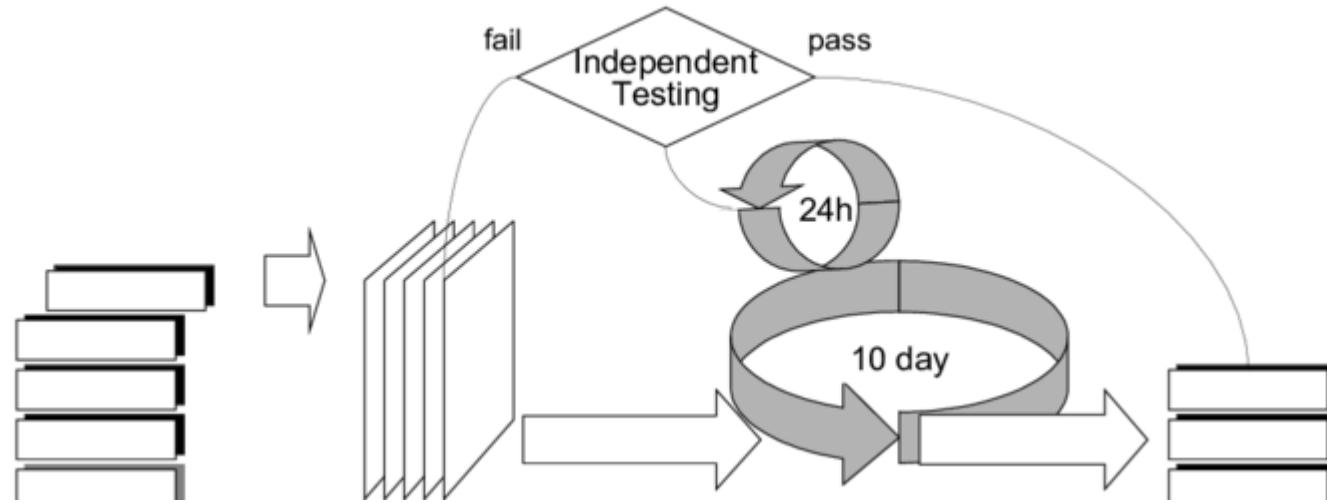


Development methodology:

In this project, we use **agile methodology** to develop our application.

- Fast react to every change.
- Fast development.

Development methodology: scrum

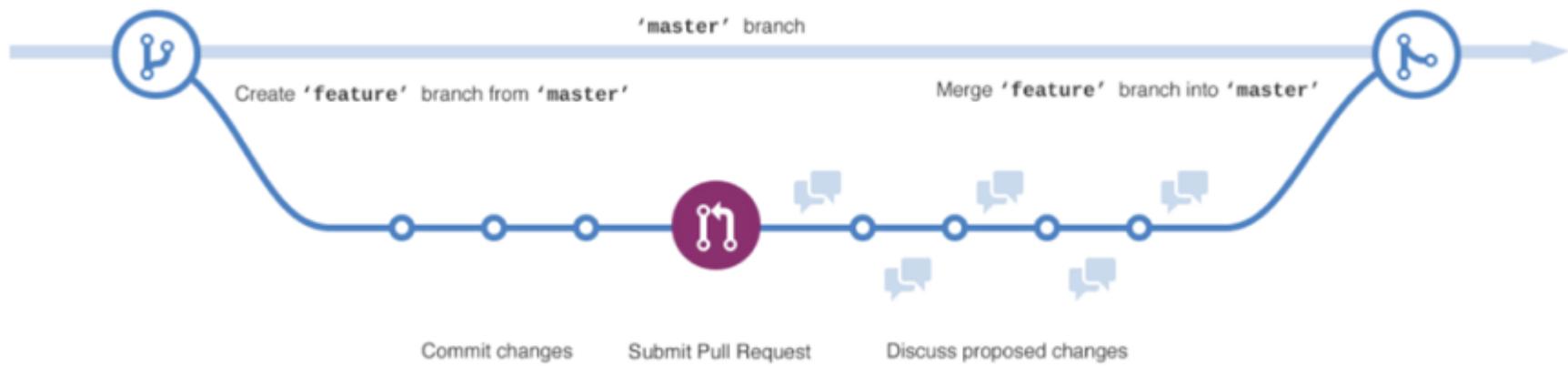


We broke each story into many technical tasks. Then we iteratively do release planning, implementation and testing based on these tasks. Finally we deliver these releases as our milestones

Configuration management:

For version control: **Git**

- Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.



Configuration management:

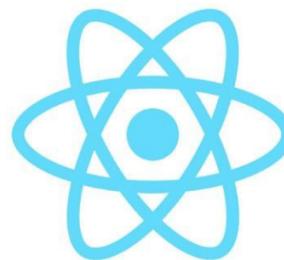
For requirement change control:

- 1.collect
- 2.review
- 3.schedule in the future scrum cycle

- Every change should be reported by feature tech lead, reviewed by pm assistant, agreed by manager and customers.

Environment

- Tools
- Software packages
- Infrastructure



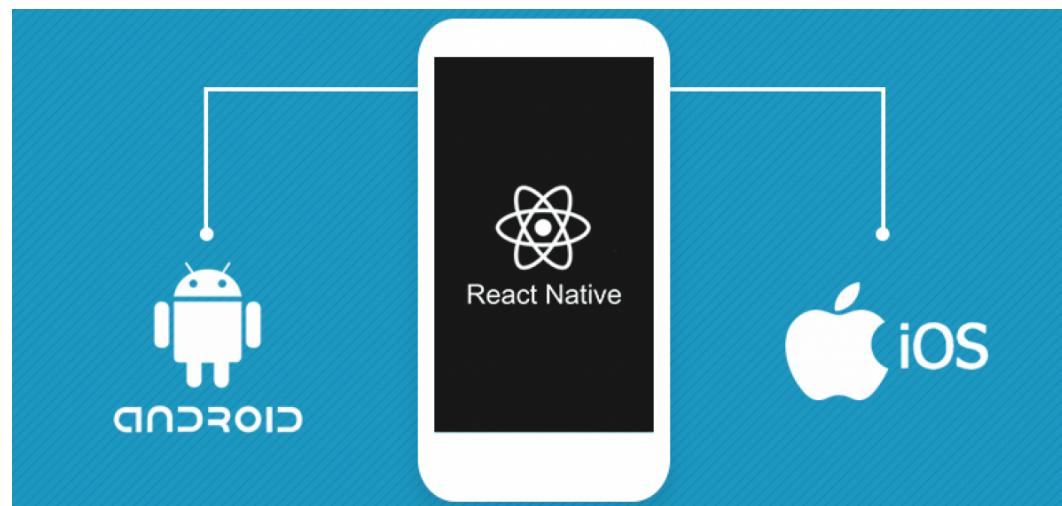
React Native



AWS Amplify

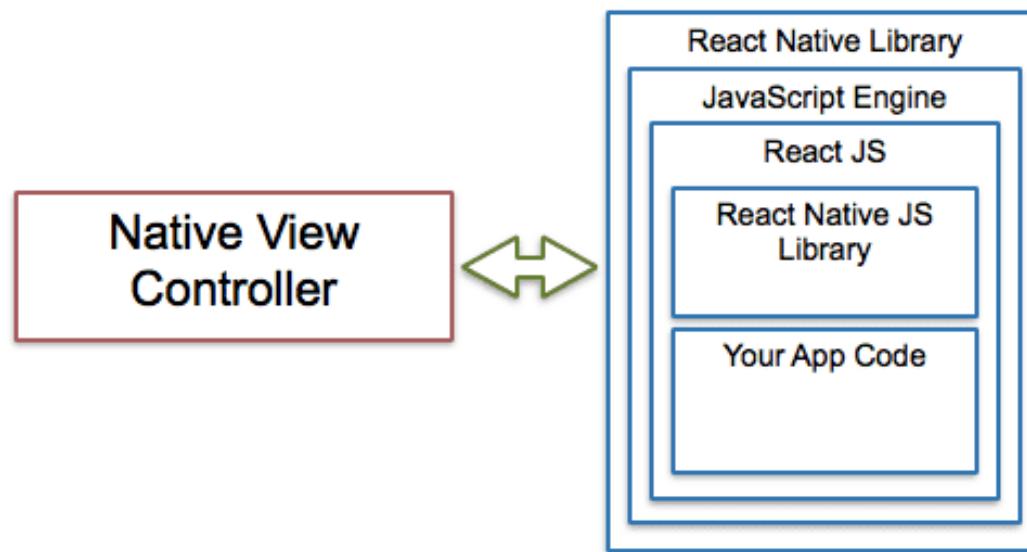
Front End - React Native

React Native is an open-source mobile application framework created by Facebook. It is used to develop applications for **Android**, **iOS**, Web and UWP by enabling developers to use React along with native platform capabilities. In a nutshell, React Native allows you to build mobile applications that look, feel and perform much more like native applications.



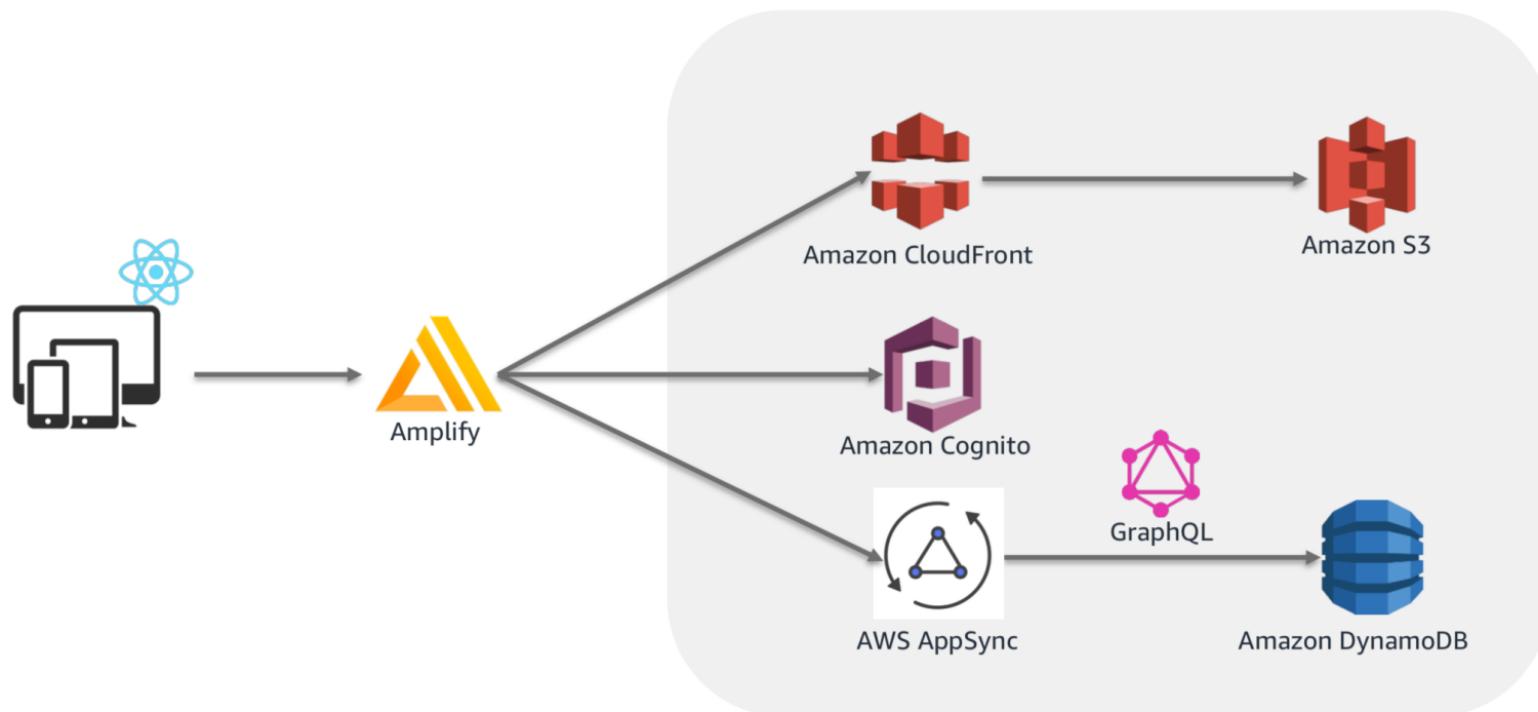
Front End - React Native

React Native runs in a background process directly on the end-device and communicates with the native platform via a serialisation, asynchronous and batched Bridge. React components wrap existing native code and interact with native APIs via React's declarative UI paradigm and JavaScript.



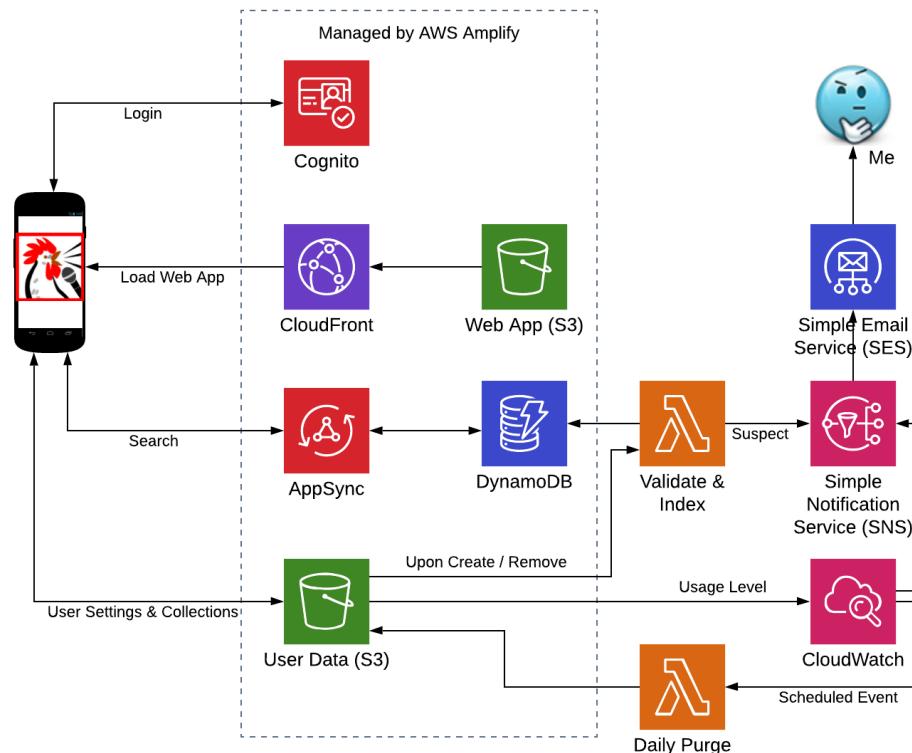
Back End - AWS Amplify

AWS Amplify provisions and manages backends for mobile applications and provides a simple framework to easily integrate backend with React Native frontends.



Back End - AWS Amplify

Mobile applications require cloud services for actions that can't be done directly on the device, such as offline data synchronization, storage, or data sharing across multiple users.



Food Recognition Model

We build and train our model using **jupyter notebook**, which is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

The framework we use in our project is **TensorFlow Object Detection API**, which is an open source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models.



Dataset - Food 101

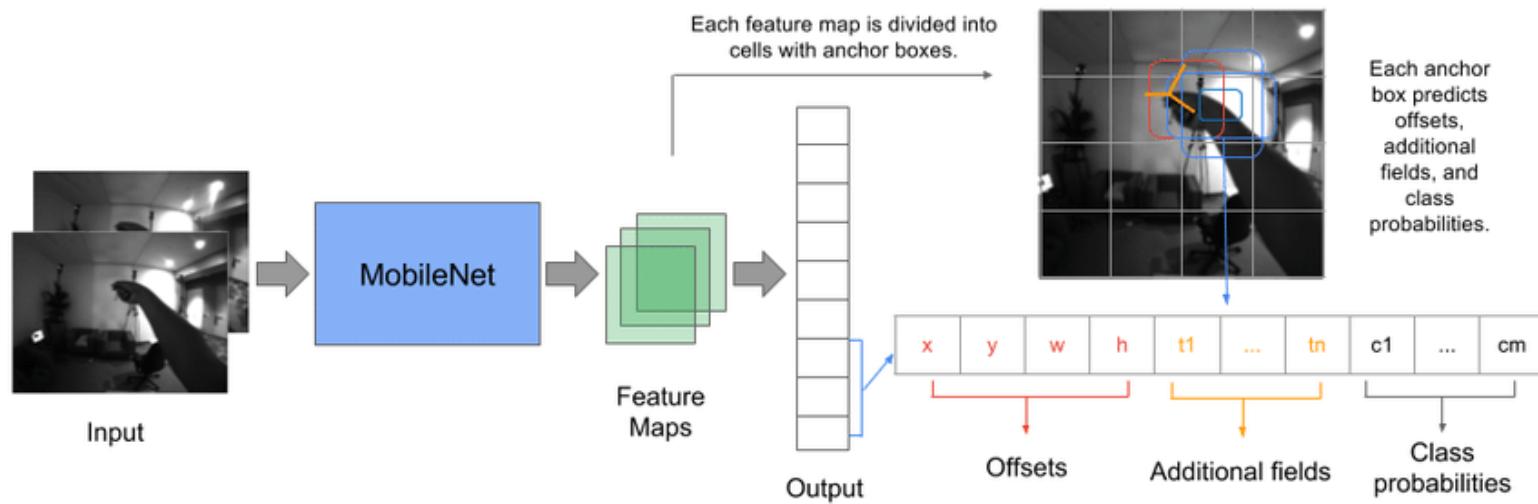
Because we want to achieve high accuracy, we need a large dataset of food images. And **Food 101** is a good collection of food images which we can use for this project.

It contains images of food, organized by type of food. It was used in the Paper "Food-101 – Mining Discriminative Components with Random Forests" by Lukas Bossard, Matthieu Guillaumin and Luc Van Gool.



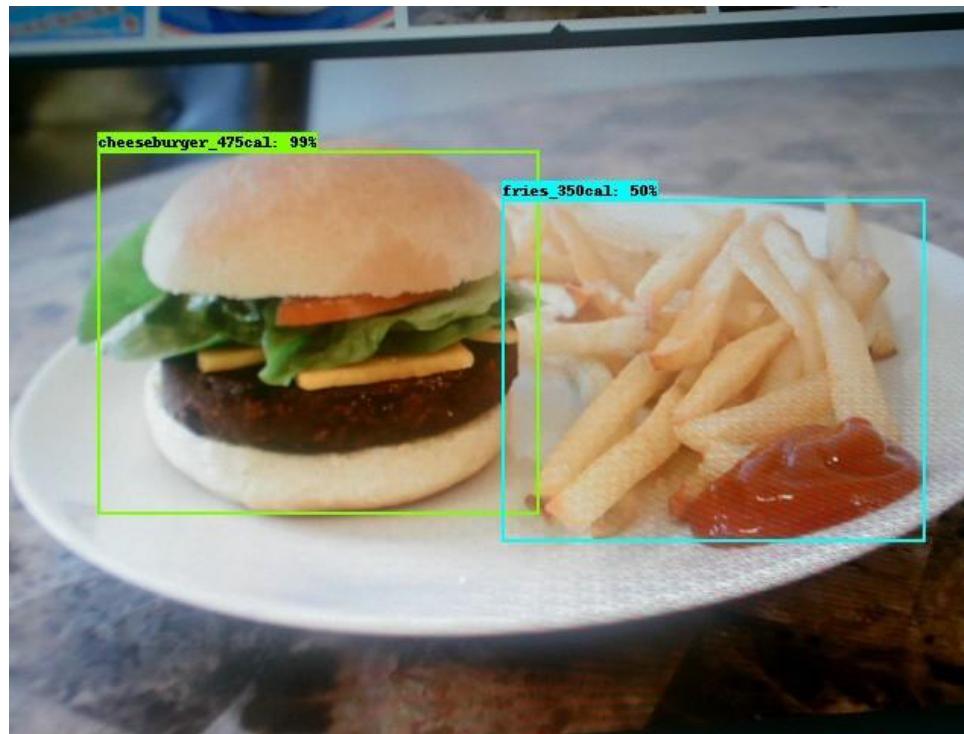
Pre-trained Model - Mobileset SSD

Training a model from beginning to end with our dataset can take a long time (weeks) and a high-end GPU. Instead, we use a pre-trained model and retrain it with our dataset to replace the classes in that model with our classes. For this project, we chose **Mobileset SSD** as the base model, which offer high speed and are ideal for detection on video feeds.



Our Results

The food detection result of our model is shown below:



References

- 1: React Native. [Online] <https://facebook.github.io/react-native/>
- 2: AWS Amplify. [Online] <https://aws.amazon.com/amplify/>
- 3: R. Aaron, et a. “Treatment of Nonunions With Electric and Electromagnetic Fields,” Clin Orthop, vol. 419, 2004
- 4: Parisa Pouladzadeh ; Shervin Shirmohammadi. You are what you eat: So measure what you eat!. IEEE.
- 5: Parisa Pouladzadeh ; Shervin Shirmohammadi. Mobile Multi-Food Recognition Using Deep Learning. IEEE.