System requirements specification:

**System Overview:**

The project shall produce the following information item content in accordance with the project’s policies with respect to the system requirements specification from the IEEE.

**System purpose:**

The purpose of the system is to produce an automatic way for the machine to learn to detect calories. The system will produce a way of detecting the volume of the food, correctly detect different items in the food and count the total calories. The system will provide the software with necessary data and method to detect and count the total calories. The system will produce an “Android” and “iOS” software, the hardware requirement is low as the system will do necessary resource intensive tasks in the cloud. The system will be learning using Supervised machine learning technique. The system will have different components to detect volume, detect items, count calorie which will interact with each other to produce the total calorie of the food. The data will be saved in the cloud. The system will securely save the data and use it to improve the learning of the machine. The system will produce a fast software with little hardware requirements for user’s device.

**System scope:**

The system to be produced is “Calorie counter”. The system will have different components such as- item detection, volume detection of each item, counting the total calorie, storing data safely and conducting resource intensive tasks in the cloud. These different components of the system will interact with each other and produce the finished software for the users. The customers need an application that will accurately count the total calories of a meal. The current software in the market does not accurately predict the calorie count and needs help of a thumb or a coin to detect the volume of the bowl. The system uses gradient descent technique to detect the size of the bowl and items without any help from the user. The system will use the same density on the surface to calculate the total calorie. Size of each items will get multiplied by the calorie value from nutritional table to calculate the total calories. The system will produce different subscription models. The paid subscription will produce detailed nutritional table. The system is expected to predict food calorie with 94.11% accuracy, which is more than any other calorie detection application in the market.

**System overview:**

The system overview describes the system with major element and their interaction with the user interfaces.

**System context:**

Major components:

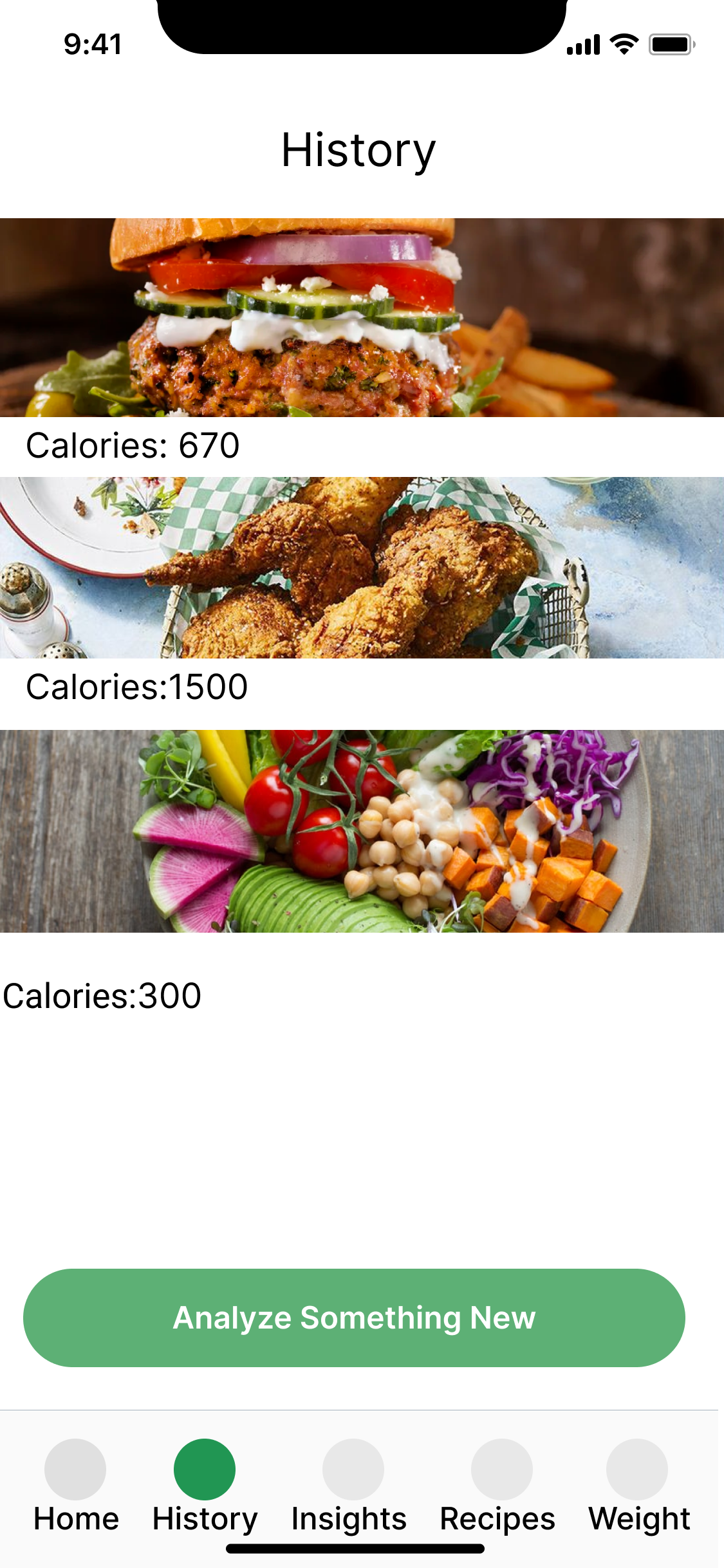
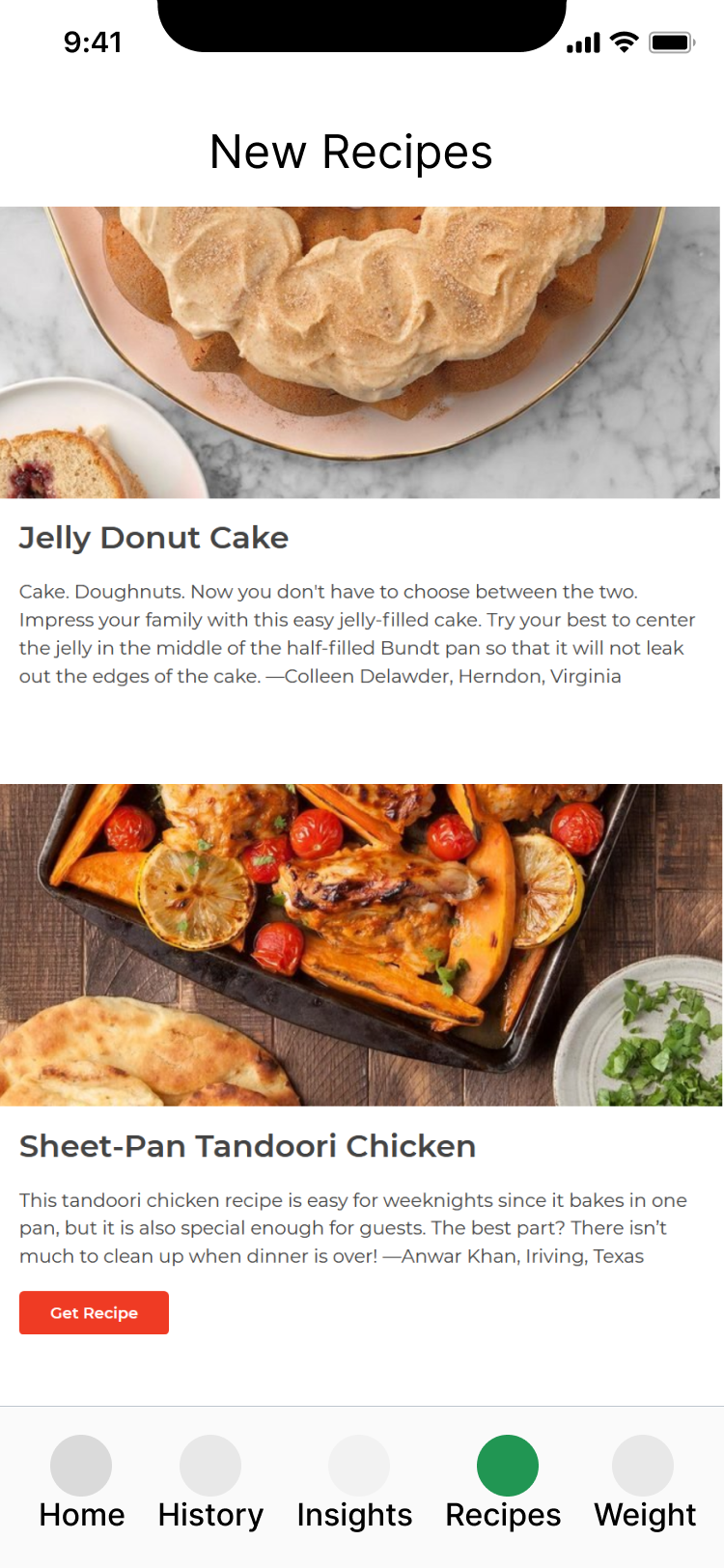
1. Detection of density of each item.

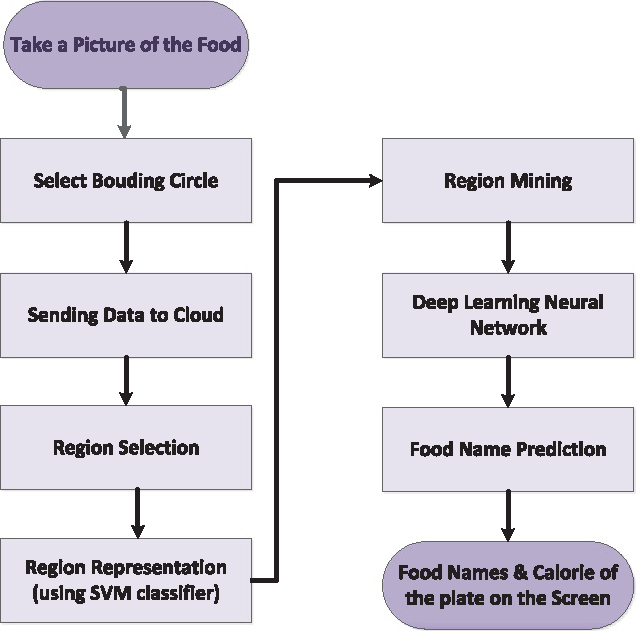
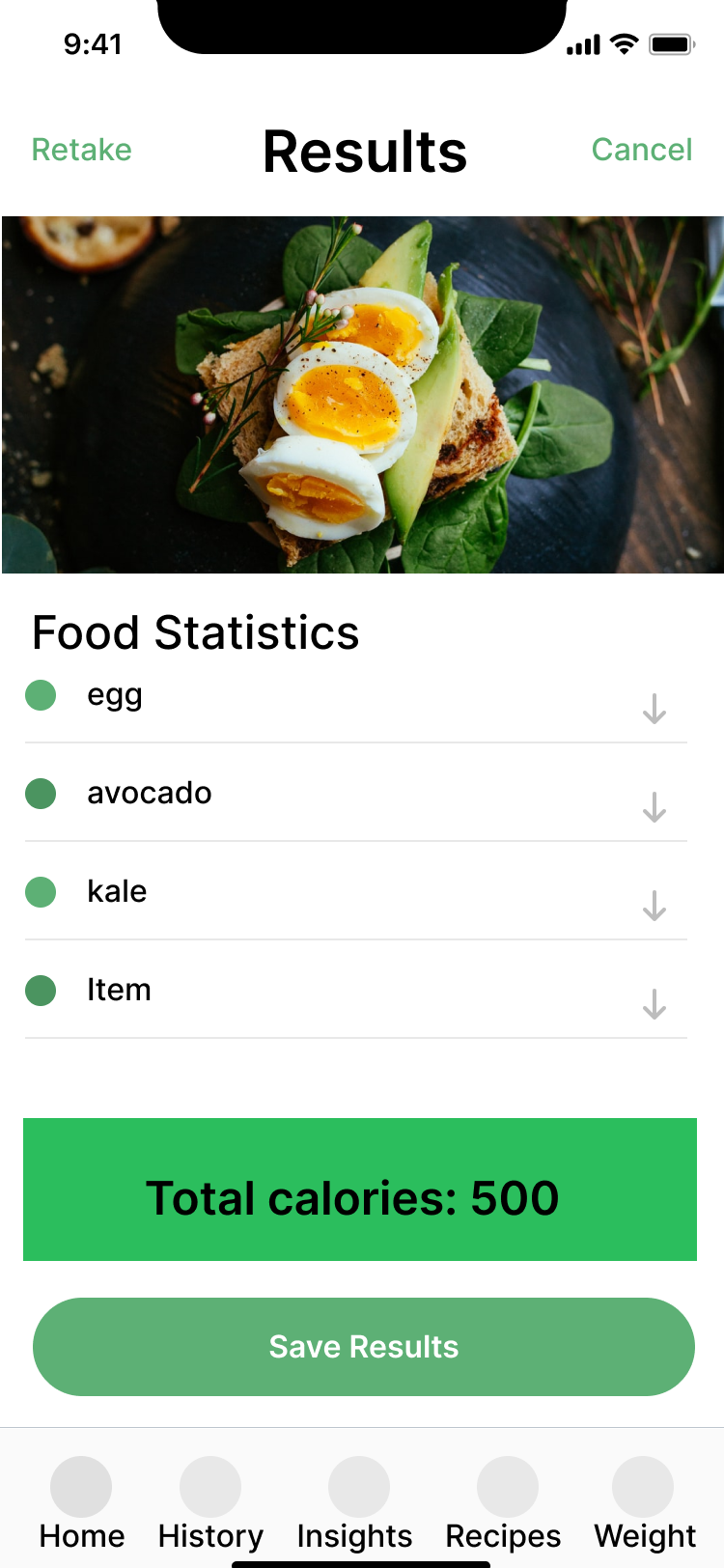
2.Detection of the size of bowl.

3. Detection of each item.

4.Calculate the result in the cloud.

5. New recipes based on user’s history.

****     

Cloud storage

|  |  |  |  |
| --- | --- | --- | --- |
| Food Name | Measure | Weight(grams) | Energy |
| Apple, with skin | 1 | 140 | 80 |
| Orange | 1 | 135 | 116 |

detailed results for paid customers

Cloud computation

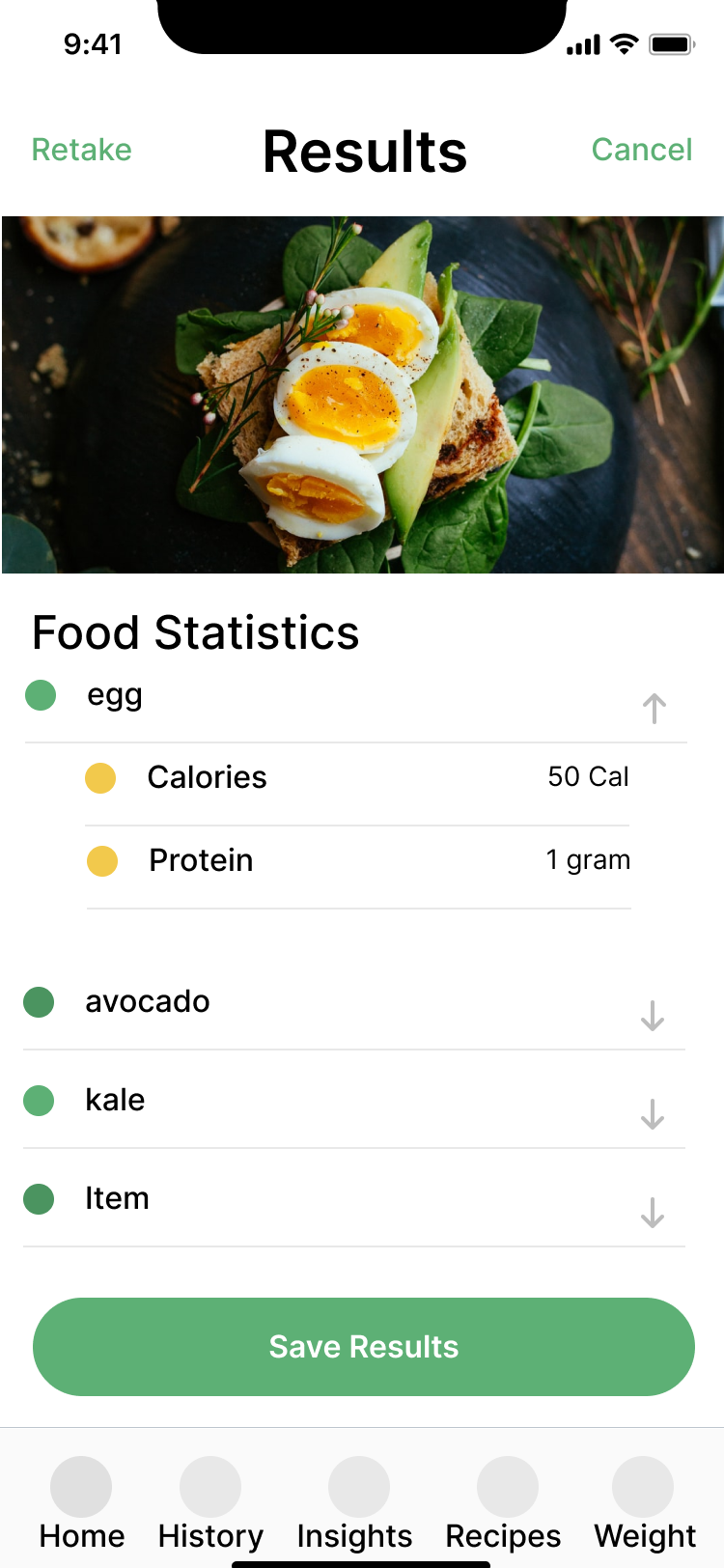


Fig: Major components and User interface interactions.

The user will take a picture with the software and the system will send it to cloud. The cloud will conduct the necessary computation and provide the user with results. The user will not have access to the computation part. The user will be able to access only his scanned items. Cloud will store these data for machine learning purposes. After, the computation the cloud will show the results based on the subscription of the customer. The software part will be responsible for user interactions and not the computational parts of the system.

**System functions:**

Our approach is to learn about a detector for each food category from a set of training images with only binary labels indicating whether the image contains this food category or not. Recognizing multiple food items and categorizing them is a very challenging task. We used Selective Search that has been broadly used as a detection method by many object detectors. The purpose of this method is to group super-pixels to generate a hierarchy of small to large regions. The auto-calibration part of the application must calculate the quantity of the item accurately without requiring user’s coin or thumb. The results must be calculated in the cloud. There must be away to take the user review after each scan and add to the data, for the machine to learn. The application must save data securely and restrict users to access other user’s data. The nutritional insights must give insights based on each user’s food habits and health condition. The new recipes component of the system should suggest new ideas based on that user’s personal likings. There should a list of new recipes for users. The application must follow Amazon cloud computing rules to use cloud computation. The computational parts must be completed in the cloud so that user device’s resources do not get used. The food detection technique must accurately detect items in difficult conditions such as bad lighting, uncommon shape of the bowl, state of the food etc. The quantity detection technique must use auto-calibration. The health organization can only use this application with paid subscription. Data should be saved securely, so that no user can see or use other user’s information. The cloud provided security measures must be taken to ensure safety. Databases will save the information after grouping together similar information to ensure speed of the application. The reviews must be incorporated to the wrong/unsatisfied results to improve our system.

**User characteristics:**

Types of user:

There are 3 types of users- unpaid, paid, Health companies. Health company must have a paid subscription to use the application. The system only shows the total calorie count to the unpaid customers. Whereas paid customers and health companies gets detailed result with number of vitamins, and amount of energy, carbohydrate, minerals, protein, fat, fibre etc. All paid users will get nutritional insights based on their weight and food habit.

Types of device:

The application will be out in “Android” and “iOS” operating systems. Both types of application will have similar User interface and functionalities.

**Functional requirements:**

Measuring the volume of food:

The system will have a component that measures the size of the bowl from user photo without asking for any help from the user. It should correctly measure the size without a thumb or a coin. The system must finish the measurement within 1 second of user taking the photo.

Accuracy of detecting food:

The system must use “map Reduce” to detect regions, mining those regions and use convolutional neural network to accurately detect each item in those regions. The accuracy of counting total calorie must be more than 94.11%. The system must count total calorie in 2 seconds of time.

Total calorie count:

The system will have individual items and their weight in the meal. The system will calculate the values using nutritional table and show the result to the user.

Cloud:

The whole computation will take place in the cloud. The cloud will save the user images, table values and programs that compute those values. The user data will be saved in the cloud. The cloud will process those data in batch and make it ready for machine learning. All the above steps must happen in the cloud within 4 second of time.

Data:

User data shall be securely stored in the cloud. Any sort of commercial usage of these data is prohibited. Any user shall only have access to his/her own history of scans.

**Usability requirements:**

1. Efficiency:

The system must compute the total calorie with 94.11% accuracy. The system will use cloud resources for resource intensive tasks. User’s device will only use the resources necessary to show interfaces of the application. The system will complete all the calculation within 4 seconds of time which is fast considering the complexity of the task.

1. Easy to use:

The user interfaces of the system must be intuitive and easy to understand. When a user first registers and starts using the application, it should not take him more than 3 seconds to start scanning the first meal. There must be a way to swipe and switch between the interfaces.

1. Error tolerance:

The system must not crash under any circumstances. There are two inputs that, system accepts from the user. The weight log and manual entry. The system will accept only particular type of values and the values have limits so that it does not compile with an error.

1. Engaging:

The system provides nutritional insights based on each customer’s history of scans and body weight. The application also gives new recipes the customers like the history.

1. Effective:

It guarantees accuracy of almost 95% in food detection. Measures the volumes of food without any user input and correctly estimates food density and total calorie.

**Performance requirements:**

Machine learning requires training models to learn. Our objective is to get total calorie count within 4 seconds. This can be accomplished by performing all operations at the same time. GPU have large number of cores compared to CPU’s resulting in 2 to 3 times faster processing speed. Nvidia GTX 1080 (8 GB) VRAM can train an average of 14kexamples/second. It can also perform operations on a batch of images of 128 or 256 images at once in just a few milliseconds. A CPU such as core i7-7500U can train an average of 115 examples/second. Nvidia GTX 1080 is suitable to perform in our expectations. 32 GB Ram is required to support all necessary functionalities.

Most of the work will done in the cloud following cloud computational rules. Amazon cloud supports serverless applications, meaning it can by itself predict and provide service in high demanding situations.

**System Interface requirements:**

CLOUD COMPUTATION Cloud Storage

& ML

Result

User Android or

iOS device

All these computation and storage must follow rules of Amazon cloud under all circumstances. The computation is same for Android or iOS operating systems. Data is processed properly before sending it to next stage.

**System operations:**

Human system integration requirements:

Most important part of the program is “item detection” in the meal. A small error on this part causes the result of total calories to be significantly off. There are some applications in the market that does the same thing. The system is supposed to detect items with 95% accuracy, which is the main selling point of this application. Any error on this part can impact the whole system in a negative way.

Maintainability requirements:

Time:

Mean downtime should not be more than 30 minutes of time. Maximum downtime should not be more than 1 hour of time. Turnaround time should not be more than 4 seconds. Mean time to repair should not be more than 30 minutes and maximum time to repair should not be more than 1 hour time.

Rate:

Item detection- 5 staff hours/month.

Size estimation- 3 staff hours/month.

Maintenance action indices:

Each staff takes 20 dollars per hours.

Reliability requirements:

1. The system must detect food item with accuracy of 95% or more, so that people.

Can rely on the results.

1. The system should accurately measure the volume of the bowl and density of each item so that the total calorie count is accurate.
2. For paid customers, system must display detailed list of items and their quantity accurately.
3. User login information must be hashed and stored into the database for security.
4. The system must not use any user data for any other than machine learning purposes.
5. The system must calculate the total calorie within 4 seconds of user taking the photo.
6. User’s scanned items must be saved in the cloud and should be limited only to that user.
7. The system should not be unavailable more than 1 percent of time.
8. The mean time between failures should be more than 20 days.
9. Time to recovery for the system should not be more than 2 hours.
10. The system should sanitize all user value in a limit to reduce error of the system.
11. The system should produce same results if same meal gets scanned multiple times.
12. The system must give nutritional insights based on the past history, which should not be random.
13. The system must not crash for any wrong user input.

Other quality requirements:

The software will support “Android” and “iOS” operating systems. The software will only display results, takes input from user. Heavy computations will get done in the cloud. Those input and output techniques in those operating systems are basic operations. The portability is moderate to high. The system will be compatible to most of the versions of those operating systems.

**Physical characteristics:**

Adaptability requirements:

The system will predict total calorie with 95% accuracy. Some results might deviate from accuracy for adverse conditions such as- lighting, photo angle etc. User review will help the system to learn so that the system gets it accurate over time. System must be built in a modular way, to ensure expansion. The advantage of modular development is high availability. Updates or new development can be done in isolation and added to the system. The system might need extra cloud resources to keep up with expansion. The “Amazon” serverless deployment can allocate those resources under short notice. The serverless deployment also ensures the allocation of the hardware resources.

Environmental conditions:

The system will work in cloud environment. The system must abide by those rules. The system will be deployed in serverless conditions. The system codes and data communication rules must follow cloud rules. The system components must be modular for easy deployment. Amazon supports Dynamo DB. The system must use Dynamo DB for scalability.

System security:

Facility:

The system will be deployed in “Amazon” cloud. “Amazon” provides physical security for such system. The data will be saved in the data warehouse which are also very secure. It is impossible to physically access those data without verification.

System:

The system will be built in a modularized way. The data of one module will be limited to that module. If one module needs data from another module, it will be passed securely, so that the access is limited to only that data. The data

Software:

The software is the mean to take input and show results. Software itself will have access to no data. There is no way to manipulate cloud data from user’s software.

Privacy:

Each user can only view his/her own history of scans. User will not have access to someone else’s data. User must login to access his data. These stored data will strictly be used for machine learning purposes. These data will not be used commercially under any circumstances.