



NUTRITICS
NUTRITION ANALYSIS SOFTWARE

Software Engineering Group Project: Nutritics

Software Design Specification Document

Client: Ciaran O'Neill

Group 15

Rory Flynn, Cormac O'Higgins, Leong Kai Ler, Owen Duffy, Luan Williams, James Healy

Table of Contents

1. Introduction

- 1.1 Overview - Purpose of System
- 1.2 Scope
- 1.3 Objectives and Success Criteria
- 1.4 Definitions, Abbreviations
- 1.5 References 2

2. System Design

- 2.1 Design Overview
 - 2.1.1 High-level overview of how the system is implemented, what tools, frameworks and languages are used etc.
- 2.2 System Design Models
 - 2.2.1 System Context
 - 2.2.2 Use cases (from Requirements Document)
 - 2.2.3 System Architecture
 - 2.2.4 Class Diagrams
 - 2.2.5 Sequence Diagrams
 - 2.2.6 State Diagrams

1. Introduction

1.1 Overview - Purpose of System

The purpose of building this API system with Optical Character Recognition (OCR) feature is to interpret nutrition labels and organise them into trustworthy and accurate information stored in Nutritics database. Nutritics provide softwares that manage recipes, analyse diets and activities, thus creating meal plans to their clients, such as dietician, sport nutritionist and food manufacturer sectors.

1.2 Scope

This project comprises of a new feature offered by Nutritics as an approach to address the problem of disjunction between software developers and nutritional professionals. This new feature is required to turn photos of food products' nutrition panels into text form and thus, adding newly gathered information into Nutritics database.

1.3 Definitions, Abbreviations

- **OCR: Optical character recognition** or **optical character reader** is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo or from subtitle text superimposed on an image

1.4 References

Nutritics. *About Nutritics*. <https://www.nutritics.com/p/about> (accessed February 2, 2018).

2. System Design

2.1 Design Overview

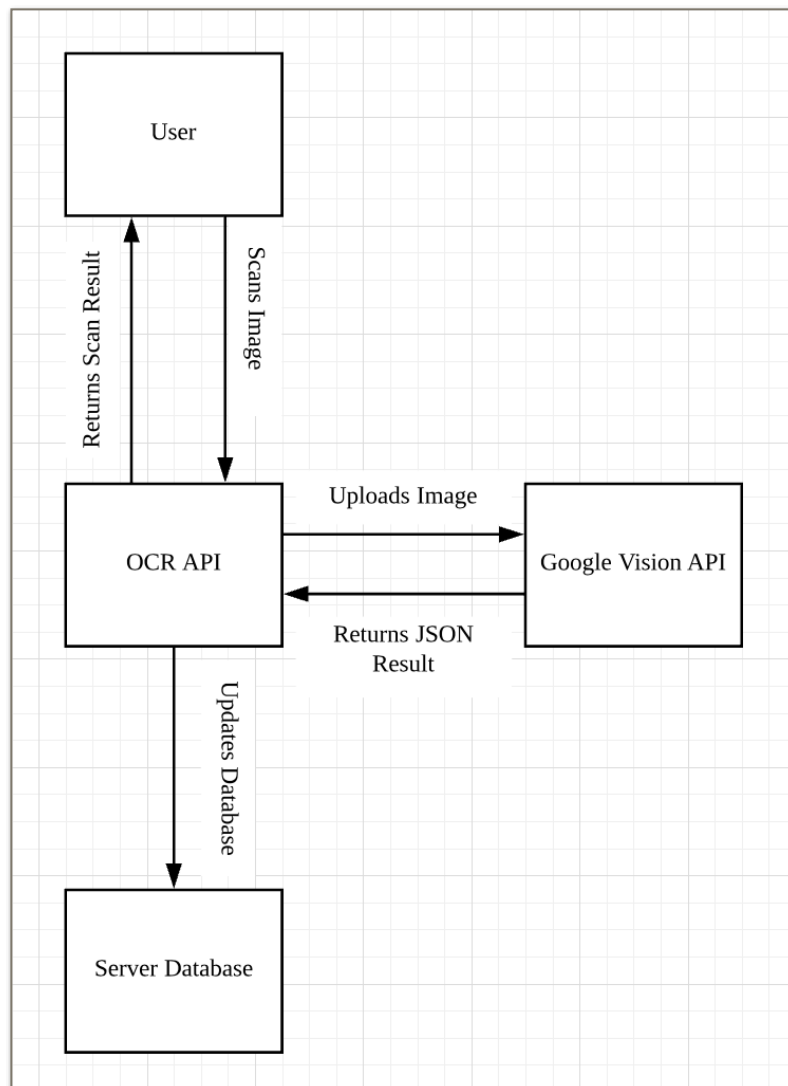
- 2.1.1 High-level overview of how the system is implemented, what tools, frameworks and languages are used etc.

Tool / Framework	Usage
Google CLOUD VISION API	Google Cloud Vision API enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy to use REST API. It quickly classifies images into thousands of categories, detects individual objects and faces within images, and finds and reads printed words contained within images. Users can build metadata on their image catalog, moderate offensive content, or enable new marketing scenarios through image sentiment analysis. They can also analyze images uploaded in the request or integrate with their image storage on Google Cloud Storage.
XAMPP	XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends. It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy as well.
PhpStorm	PhpStorm provides the best code completion, refactorings, on-the-fly error prevention, and more for PHP coding. It incorporates all the features of WebStorm and offers full-fledged support for PHP and Databases/SQL support

PHP is used to develop an API that helps the client scan and uploads image to be processed by the Google Cloud Vision API and store the return JSON formatted result in the client database, which is also programmed in PHP.

2.2 System Design Models

2.2.1 System Context

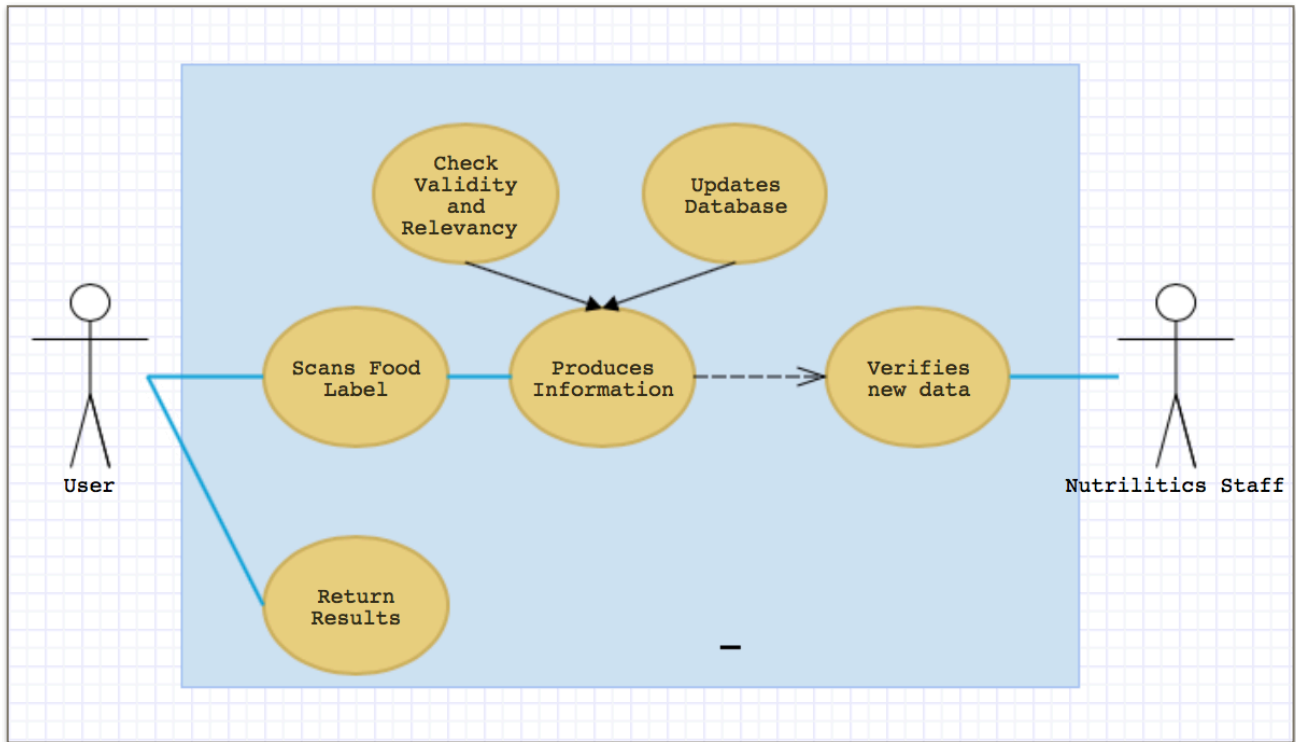


The System Context Design defines the boundary between the system and its environment, showing the entities that interact with it.

The System consists of the user scanning an image of food label using the OCR API by Nutritics. The API then sends the image to the Google Vision API which will then returns its interpretation in the format of JSON.

The API then updates the data onto Nutritics database to be stored and returns the result to the user to verify the scan successful.

2.2.2 Use cases (from Requirements Document)



UML Use Case Scenario Descriptions

1.

Name:	Scans Food Label
Participating Actor:	User
Entry Condition:	User scans a label on a food product
Exit Condition:	Scanning was successful.
Normal Scenario:	User scans the image of the label by accessing the camera of the mobile device with the app.
Error Scenario:	Scanning failed.

2.

Name:	Verifies new data
Participating Actor:	Nutritics staff
Entry Condition:	A food product (can also not be a food) with a food label or nutritional values per serving unrecognised or unseen by the OCR API
Exit Condition:	The new food product is verified by Nutritics staff.
Normal Scenario:	Nutritics staff checks if the label scanned is actually a food. Nutritics calculates the nutritional values per serving of the food listed on the label to determine if the values are true, valid and relevant.
Error Scenario:	Nutritics staff determined errors or fallacy on the label and thus verification of the new data fails.

3.

Name:	Return results
Participating Actor:	User
Entry Condition:	Verification of a new data scanned a user is done.
Exit Condition:	The user receives a JSON text format of the contents verified and its results.
Normal Scenario:	The OCR API creates a JSON text format of the food label scanned. The OCR API sends the text and results to the user.
Error Scenario:	The user does not receive the relevant or correct results and text.

4.

Name:	Produces new Information
Participating Actor:	-(managed by the OCR API)
Entry Condition:	The OCR API receives a new image scanned by the user.
Exit Condition:	The new image is interpreted successfully and relevant information on it is extracted to form a new piece of nutritional information of a food product.
Normal Scenario:	The OCR API attempts to read and interpret relevant information on the food label image it receives. The API checks the validity and relevancy of the information and updates it on the database.
Error Scenario:	The newly scanned image cannot be recognised by the API to create a new information and is sent to the Nutritics staff to be verified.

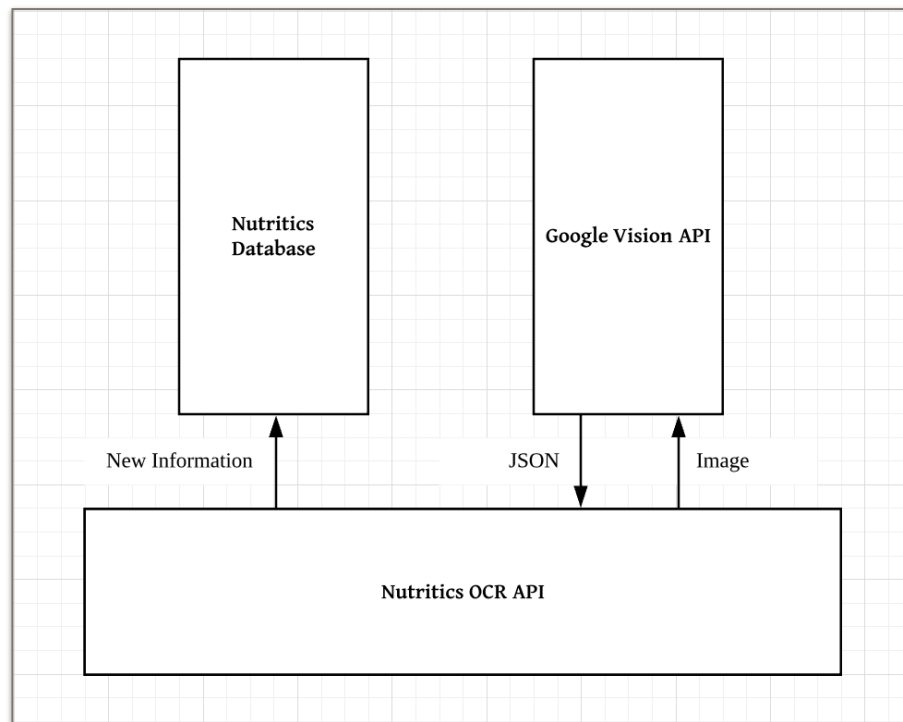
5.

Name:	Checks Validity and Relevancy
Participating Actor:	-(managed by the OCR API)
Entry Condition:	A new information is processed by the API and needs to be examined.
Exit Condition:	The API approves the new information.
Normal Scenario:	API search for related words on the image structure. API works out if each nutritional values corresponds to their types logically.
Error Scenario:	The API is unable to determine the validity and relevancy of the information and it is sent to the Nutritics staff to handle it.

6.

Name:	Updates databases
Participating Actor:	-(managed by the OCR API)
Entry Condition:	A newly produced information is to be stored in the database for future reference.
Exit Condition:	The information is successfully stored.
Normal Scenario:	The API stores the information in the database by organising the correct content values of each nutrient at the right category.
Error Scenario:	The database does not accept the new information, thus update on database fails.

2.2.3 System Architecture

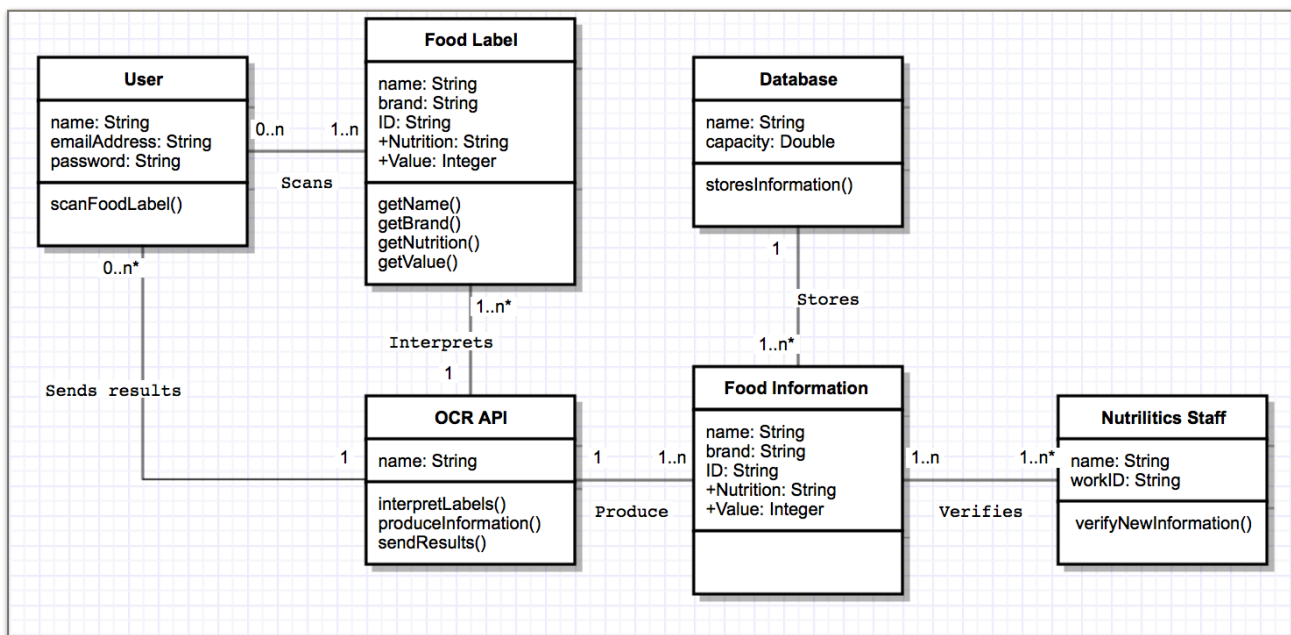


Overview

The architecture concept employed in the system incorporates the use of Google Vision API as it can easily **detect broad sets of objects** in images commonly found within images. Moreover, it offers an excellent Optical Character Recognition (OCR) to **detect text** within images, along with **automatic language identification**. Vision API supports a broad set of languages.

In general, the Nutritics OCR API is only responsible in sending the uploaded image to Google Vision API to extract required information and updates them in the Nutritics Database. The OCR API should be able to accept various image formats such as PNG, JPG, PDF and etc. After receiving the JSON result from Google Vision API, it should be able to store the data accordingly in the server database.

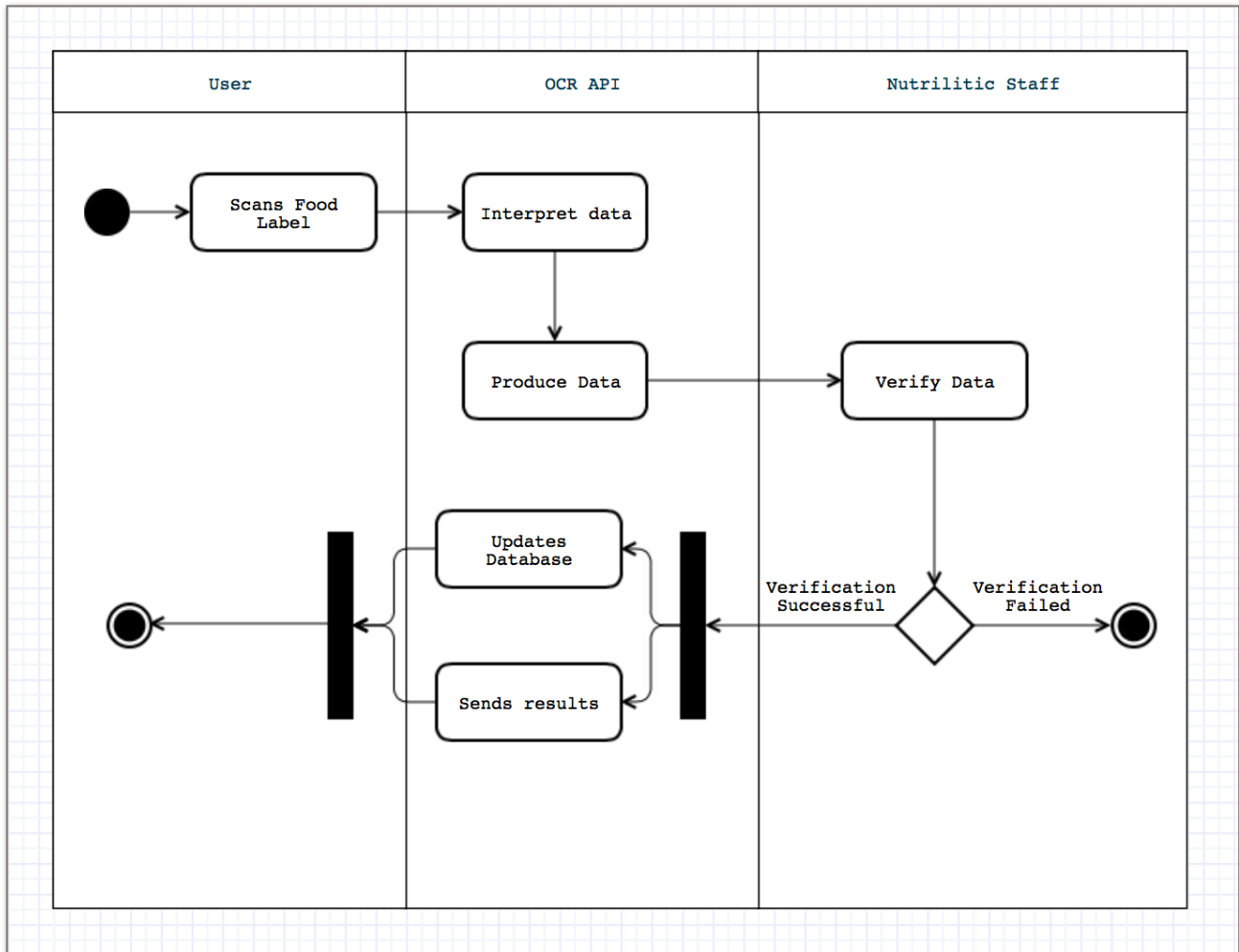
2.2.4 Class Diagrams



The class diagram provides a representation the interaction and relationships between all elements in the operation of Nutritics new API feature.

Firstly, users are to scan image of food labels using services provided by Nutritics. The image will then be uploaded to the Google OCR vision API to scan for relevant information. Then, the Google API will return a result of requisite data in the format of JSON which will then be stored in the database. At the same time, the results have to verified by operating Nutritics staff to make sure the information are valid and accurate. Lastly, the JSON formatted result will be send back to the user by the OCR API in text format as verification of a successful scan.

2.2.5 Sequence Diagrams



The UML sequence diagram displays the typical user activity of users using the OCR API by Nutritics. Firstly, the user have to scan a food label and forward it to the OCR API for interpretation. Then, it will extract information using Google Vision API and produce data returned in JSON format. The data is then verified by Nutritics staff. If the new data is somehow recognised as invalid, inaccurate or fake, the whole operation is will automatically failed and the data will not be stored in the database. On the other hand upon successful verification, the data will be updated in the Nutritics database and the results will also be sent by the OCR API back to the user for viewing.

2.2.6 State Diagrams

State Description:

- The state is initiated when a client of Nutritics scans a food label using Nutritics app.
- Then, the client uploads the image to Nutritics, where the OCR API will upload the image to Google Vision API to extract relevant information from the label.
- Afterwards, the result is returned in a JSON format to Nutritics.
- This result is verified by Nutritics staffs. If the verification fails, this new data will not be stored into the database and thus, the scan is deemed as failed.
- However, if verification is successful, the user will receive the result as a response as verification successful. The data will also be updated into Nutritics database .

