**AN IMAGE BASED FOOD CLASSIFIER FOR DIETARY SUPERVISION**

**A RESEARCH PORJECT SUBMITTED TO**

**THE DEPARTMENT OF COMPUTER SCIENCE,**

**SCHOOL OF COMPUTING AND ENGINEERING SCIENCES,**

**BABCOCK UNIVERSITY**

**IN PARTIAL FULIFILMENT OF THE REQUIREMENT**

**FOR THE AWARD OF BACHELOR OF SCIENCE (HONORS)**

**DEGREE IN COMPUTER SCIENCE**

**BY GROUP 44 (Computer Science)**

**OTUKOYA OMOBOSOLA CHIAMAKA 16/1909**

**AYENI BOLAJI HEPHZIBAH 16/1336**

**UGOKWE COSMOS CHINEMEREM 16/0988**

**MARCH 2019**

# **DECLARATIONS**

We, declare that this project, AN IMAGE BASED FOOD CLASSIFIER FOR DIETARY SUPERVISION was carried out by the following people:

**OTUKOYA OMOBOSOLA CHIAMAKA**

**…………………… ………………..**

**AYENI BOLAJI HEPHZIBAH**

**…………………... ………………..**

**UGOKWE COSMO CHINEMEREM**

**…………………… ………………..**

# **ABSTRACT**

Deficiency in Dietary Practices in the world is currently one of the major concerns for the world health organization (WHO) as stated in the fight against conference in 2018. Recent data has shown that Unhealthy **diet** contributes to approximately 678,000 deaths each year in the U.S., due to **nutrition**- and obesity-related diseases, such as heart disease, cancer, and type 2 diabetes. In the last 30 years, obesity rates have doubled in adults, tripled in children, and quadrupled in adolescents. This study aims to bridge the gap between poor eating habits and the health industry SMART EATS is an image based food classifier application that informs the consumer on the type of indigenous delicacies he/she is interested in then predicts its nutritional benefits based on the constituent in the food then proceeds to give a feedback to the consumer based on his/ her dietary preference.

Based on the review of related works it has been recorded that the invention of these systems has helped individuals with nutritional deficiencies to improve on their eating habits and this work not only continues with this advantage it also contributes more by segregating the content of a product from its package hereby reducing the rate at which smuggled goods enter the country.

On this basis it is recommended that Government health organizations, hospitals and healthcare centers, Nutritionist and dieticians use this system as a key factor in the management of mortality rate influenced by eating habits

# **ACKNOWLEDGEMENT**

We are using this opportunity to express our profound gratitude to everyone who supported us throughout the course of this Undergraduate project. We are thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. We are sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

we express our warm thanks to our Parents Dr & Mrs Adedeji Otukoya, Mr & Mrs Ugokwe, Engr & Mrs Sunday Ayeni and our most amiable supervisor DR. Okoro Raymond and to all our friends within and outside the departments for their support and guidance.

# **CERTIFICATION**

I, certify that this project title an image based food classifier for dietary supervision was carried out under my supervision in the Department of Computing and Engineering sciences, Babcock University.

**OTUKOYA OMOBOSOLA C 16/1909**

**……………...**

**AYENI BOLAJI H 16/1336**

**………………**

**UGOKWE COSMOS C 16/0988**

**………………**

**DR. OKORO RAYMOND, U.R DATE**

**Project Supervisor**

**……………………….. ………………**

**EXTERNAL EXAMINER DATE**

**……………………….. ………………**

**PROF. ADEKULE, Y.A DATE**

**Head of Department**

**……………………….. ………………**

Table of Contents

[DECLARATIONS…………………………………………………………………………………………1](#_Toc35336079)

[ABSTRACT 2](#_Toc35336080)

[ACKNOWLEDGEMENT 3](#_Toc35336081)

[CERTIFICATION 4](#_Toc35336082)

[1.0CHAPTER ONE 6](#_Toc35336083)

[1.1INTRODUCTION 6](#_Toc35336083)

[1.2BACKGROUND OF THE STUDY 8](#_Toc35336084)

[1.2.1CLASSIFICATION OF IMAGES 8](#_Toc35336085)

[1.3STATEMENT OF THE PROBLEM 9](#_Toc35336086)

[1.4OBJECTIVES OF THE STUDY 10](#_Toc35336087)

[1.5METHODOLOGY OVERVIEW 10](#_Toc35336088)

[**1.5.1** DEVELOPMENT TOOLS 11](#_Toc35336089)

[**1.5.2** DESIGN TOOLS 11](#_Toc35336090)

[1.6SIGNIFICANCE OF THE STUDY……………………………………………………………….11](#_Toc35336091)

[SCOPE OF THE STUDY 12](#_Toc35336092)

[1.8ORGANIZATION OF SUBSEQUENT CHAPTERS 12](#_Toc35336093)

[1.8.1 CHAPTER TWO 12](#_Toc35336094)

[1.8.2 CHAPTER THREE 13](#_Toc35336095)

[1.8.3 CHAPTER FOUR 13](#_Toc35336096)

[1.8.4 CHAPTER FIVE 13](#_Toc35336097)

[1.9EXPECTED OUTCOME 13](#_Toc35336098)

[2.0 CHAPTER TWO……………………………………………………………………………………6](#_Toc35336083)

[2.1INTRODUCTION 6](#_Toc35336083)

[2.2THEORETICAL FRAMEWORK 8](#_Toc35336084)

[2.3LITERATURE REVIEW 9](#_Toc35336086)

[2.3.1WHAT IS DIETARY SUPERVISION 10](#_Toc35336087)

[2.3.2COMPUTER VISION TASKS 10](#_Toc35336088)

[2.4REVIEW OF RELATED WORKS 11](#_Toc35336091)

[2.4.1REVIEW ON INVERSE COOKING 12](#_Toc35336092)

[2.5GENERAL COMMENTS ON REVIEWED LITERATURE 12](#_Toc35336093)

3.0 CHAPTER THREE 1

[3.1 NATURE OF THE PROJECT 1](#_Toc35336801)

[3.2 SYSTEM ANALYSIS AND DESIGN 2](#_Toc35336802)

[3.3 DESIGN MODEL 3](#_Toc35336803)

[3.2.1 STAGES OF THE ACTIVITIES 5](#_Toc35336804)

[3.3 REQUIREMENT SPECIFICATION 7](#_Toc35336805)

[3.3.1 “FUNCTIONAL REQUIREMENT 8](#_Toc35336806)

[3.3.2 NON-FUNCTIONAL REQUIREMENT 8](#_Toc35336807)

[3.3.3 SYSTEM REQUIREMENTS 10](#_Toc35336808)

[3.3.4 USER REQUIREMENTS 10](#_Toc35336809)

[3.4 CHOICE OF PROGRAMMING LANGUAGE 11](#_Toc35336810)

3.5[DEVELOPMENT TOOLS 13](#_Toc35336811)

[3.5.1 DEVELOPMENT ANALYSIS 13](#_Toc35336812)

[3.6 SYSTEM DESIGN 14](#_Toc35336813)

[3.6.1 CODING 14](#_Toc35336814)

[3.7 CONCEPTUAL DESIGN 15](#_Toc35336815)

[3.7.1 USE CASE DIAGRAM 15](#_Toc35336816)

[3.7.2 ENTITY RELATIONSHIP DIAGRAM 17](#_Toc35336817)

[3.7.3 SYSTEM FLOWCHART 17](#_Toc35336818)

[3.1 MAINTENANCE 17](#_Toc35336819)

[4.3 SYSTEM TESTING 8](#_Toc35336942)

[4.11 SYSTEM TESTING 9](#_Toc35336943)

[4.11.1 BLACK BOX TESTING 9](#_Toc35336944)

[4.11.2 HAPPY-PATH TESTING 10](#_Toc35336945)

[4.11.2 BACK-END TEST 10](#_Toc35336946)

5.0 [CHAPTER FIVE 11](#_Toc35336947)

5.1 [SUMMARY, CONCLUSION AND RECOMMEDATION 11](#_Toc35336948)

[5.1.1 SUMMARY 11](#_Toc35336949)

[5.1.2 CONCLUSION 12](#_Toc35336951)

[5.2 RECOMMENDATION 13](#_Toc35336951)

**CHAPTER ONE**

## INTRODUCTION

Computer Vision (CV) which has been in existence for over 50 years since the summer of 1996 where the pioneers of artificial Intelligence Seymour Paapert and Marvin Minsky at MIT artificial intelligence group devised a project titled summer vision Project. The Aim of the project was to build a system that can analyze a sense and identify objects in the scene.

David Marr a neuroscientist at MIT took ideas from the studies of the cerebellum, hippocampus and cortex from human perception and these became the building blocks of the modern computer vison thereby earning him the title as the Father of Modern Computer Vision.

Computer Vision is a technology that gives the computer the ability to see and understand images. Understanding vision and building visual systems is really understanding intelligence and by see, I mean to understand and not just to record pixels (Fei-Fei Li 2014)

Originally Computer vision focused on replicating parts of the complexity of the human vision system and enabling computers to identify and process objects in images and videos in the same way that humans do but thanks to the advances in artificial intelligent and innovation in deep learning and neural networks CV now comprises of object classification, object identification, object verification, object detection, object segmentation and object recognition.

Computer vision is not just limited to detecting an image but also semantic image interpolation (Image to speech) which is the ability to relay the image in understandable Natural Language.

The most impressive thing I have seen recently is the ability of these deep learning systems to understand an image and produce a sentence that describes it in natural language (Yoshua Bengio, 2015)

One of the questions in both Neuroscience and Machine Learning has been on how the brains work, and how we can approximate that with its algorithms? The reality is that there are very few working and comprehensive theories of brain computation; so despite the fact that Neural Nets are supposed to “mimic the way the brain works,” Researches in the area of AI is still unsure about neural nets.

The same paradox holds true for computer vision, since research is yet to decide on how the brain and the eye process images it is difficult to say how well the algorithms used approximate human internal mental process. On a certain level computer vision is about pattern recognition, so one of the ways scientists train computers to understand visual data is by feeding it a maximum amount of labeled data and subjecting those data to various software techniques or algorithms that allow the computer to recognize patterns in all elements relating to those labels.

## **BACKGROUND OF THE STUDY**

**COMPUTER VISION TASKS**

Integrating computer vision can span from the area of business to health and can virtually serve everybody including the virtually impaired which is why in 2015 mark Zuckerberg acknowledged the urgency of having such technology. In other to achieve the aforementioned there is a need to consider many standard task such as CLASSIFICATION.

### CLASSIFICATION OF IMAGES

Image Classification refers to a process in computer vision that can classify an image according to its visual content. The task of predicting what an image represents is called Image Classification.

The OCR is a multi-task convolutional neural network that incorporates image classification and was developed by Emmanuel Goldberg. Optical Character Recognition (OCR) is a technology that is used for the conversion of images of written text, scanned document, scene-photo (text on billboards) into text.

Image Classification can only tell the probability that an image represents one or more of the classes that the model was trained on. It cannot tell then position or identify the object within the image.

## **STATEMENT OF THE PROBLEM**

In the fast moving digital world imagine a seamless shopping experience where all the visual content is instantly shoppable and decision making for customer is made possible through an automated process of collecting and analyzing digital data. The goal in retail shopping should not be to only make the shopping experience in retail stores faster, more accurate and more enjoyable but also to alleviate the process of food intake and dietary assessment. The problem image classification of food and drink face is the large scale visual representation challenge which deals with defining the structure of the object which is particularly a difficult task as food and drink images differ in nature. Another issue is with drink recognition there is a limited amount of information that can be gained using only the images of drink an example of such information is the packaging style of the drink, the drink’s color. All these obstacles makes the classification of food and drink a challenging problem in computer vision.

Hence this study presents a framework that incorporates the use of deep convolutional neural networks to recognize food and drink items, detect the content of a product regardless of the appearance and also recommend dietary assessments based on the products nutritional value.

## **OBJECTIVES OF THE STUDY**

The general objective of this study is to build an Image based Food identifier for dietary supervision using computer vision technology

However the specified objectives are to:

1. Identify and classify challenges in the existing system
2. Design an image classification framework which incorporates NLP
3. Evaluate the image classification API
4. Implement the developers model

## **METHODOLOGY OVERVIEW**

The software development life cycle (SDLC) model that was used in the course of the project is the Agile SDLC model which is a combination of the iterative and incremental process models. However in order to carry out the project the following activities were completed

1. Group Interviews was conducted and Literature reviews in areas such as AI, Data Mining, and Ophthalmology.
2. Create a Theoretical Framework
3. Design a Unified Modelling Language (UML) as a modelling tool
4. Amazon Machine Learning tool was used to implement the classification model

### **DEVELOPMENT TOOLS**

1. Arduino Uno – This is a microcontroller board developed by Adruino.cc, it is equipped with a set of both digital and analog IO pins that passes both power and data to the components.
2. Raspberry Pi – This is a small single-board computer that runs on Linux which has General-purpose IO pins.
3. Arduino software (IDE)- This IDE makes it easy to write and upload instructions to the Arduino board

### **DESIGN TOOLS**

Design tools to be used include: JavaScript, HTML was used for web development MySQL Workbench was used for database development

## **SIGNIFICANCE OF THE STUDY**

The effect of this work will help

1. Reduce nutritional risk for customers with certain health issues
2. National Health Agency detect Fake and illegal products
3. Identify product content accurately regardless of the appearance for instance kerosene inside a Babcock table water bottle thereby reducing smuggling of contrabands
4. Generate an estimation of nutritional values which can help healthy customers improve their dietary planning and assessment.

## **SCOPE OF THE STUDY**

The scope of study is to design an image classification technology for retailers and buyers which addresses the importance of dietary and health as relating to drinks and local food The Application will enable customers detect the product and classify the content of the product regardless of its appearance. It would also be able to recommend health tips based on the user’s health history by reading the nutritional value of the product. It will also be able to project the product’s price along with the expiration date either by text or speech in order to aid easy and fast shopping experience. This application will also have the ability to make instant nutrition and calorie estimates from the chosen product and also take into account the cashless payment method that will enable customers make payment by scanning the barcode of the product as well as alert a security body when a product has been stolen.

## **ORGANIZATION OF SUBSEQUENT CHAPTERS**

### CHAPTER TWO

Chapter 2 of this project discusses the background research involved to gain a better understanding of what the system is about. It goes into detail design patterns and several other topics that are important in computer vision.

### CHAPTER THREE

The Specification and Design of Chapter 3 gathers tools and methods for achieving the task; the requirement specification is attained, and it goes ahead to design the food classification model which is the blue print of what the system would carry out

### CHAPTER FOUR

Chapter 4 discusses the implementation of smart eats, subjecting it to system testing strategies, software maintenance and evaluation.

### CHAPTER FIVE

The final Chapter concludes this project and “gives a summary of the overall project progress and results. Additionally, it also suggests further work that can improve what has already been accomplished”.

## EXPECTED OUTCOME

## The Expected outcome of this study is an application that classifies drinks and local food and recommends dietary assessments to the user by reading products nutritional value and comparing with users provided health history.

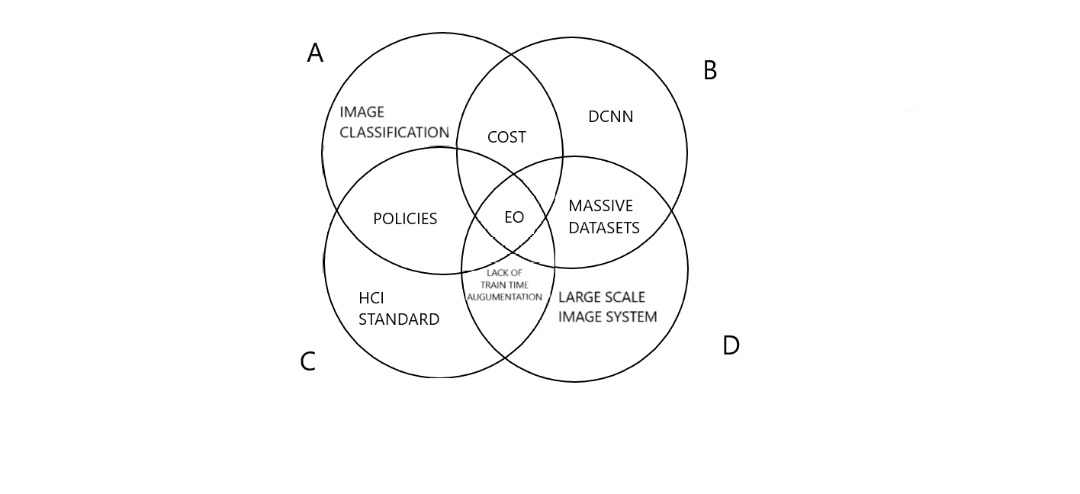
**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 INTRODUCTION**

It is arguably true that not many of today’s restaurants, gyms, hospitals, shopping malls are familiar with the new and effective approach the implementation of computer vision in food and drink applications will present. Although in recent years researchers in this field have done a good job in implementing these applications one of many being the FOOD Ai “Caloriemama” which allows you to take a picture of your meal and view the nutritional value. Applications from this field dramatically reduce the rate of harmful consumption of certain products however those applications are limited in the aspect of user interaction which is one of the gaps our project attends to by incorporating NLP in order to aid human-computer interaction.

**2.2 THEORETICAL FRAMEWORK OF THE IMAGE CLASSIFIER MODEL**



*Figure 1 Theoretical Framework of the image classifier model*

(Cooperative prefix Theoretical framework by Okoro et. al 2014)



*Figure 2 Example of an unprecedented death due to intake of allergic substance in food.*

2.3 LITERATURE REVIEW

2.3.1 WHAT IS DIETARY SUPERVISION

Dietary Supervision, also known as “foodservice management”, is the practice of providing nutritional options for individuals and groups with [diet](https://en.wikipedia.org/wiki/Diet_(nutrition)) concerns through supervision of [foodservices](https://en.wikipedia.org/wiki/Foodservice).

2.3.2 COMPUTER VISION TASKS

The area of Computer Vision basically deals with anything that humans see and perceive.

There are so many tasks that we humans do almost subconsciously that we hardly think are even worth mentioning. However, for a computer to learn to perform or even try to mimic such things are very difficult hence the use of deep learning.

The use of deep learning algorithms are achieving state-of-the-art results in challenging problems as relating to computer vision. It is not just the performance of deep learning models on benchmark problems that is most interesting; it is the fact that a single model can learn meaning from images and perform vision tasks

The many standard tasks in computer vision that require special considerations include:

1. IMAGE CLASSIFICATION

Image Classification refers to the process in computer vision that can classify an image according to its visual content the task of predicting what an image represents is called image classification.

Image classification is the process of taking an **input** (like a picture) and outputting a **class** (like “cat”) or a **probability** that the input is a particular class (“there’s a 90% probability that this input is a cat”).

1. OBJECT IDENTIFICATION AND RECOGNITION

Object identification and recognition is an area in AI that focuses on recognizing different objects in images and videos. This involves drawing a boundary box around one or more objects in an image. Object recognition is a key technology behind driverless cars, enabling them to recognize a stop sign or distinguish a pedestrian from a lamppost.

1. IMAGE SEGMENTATION

Image Segmentation is a technique in digital image processing to partition an image into multiple parts or regions. It is the process of taking a digital image and segmenting it into multiple segments of pixels, this could involve separating foreground from foreground or clustering regions of pixels based on similarities in colour or shape.

1. OBJECT VERIFICATION

Object Verification is a computer vision task that deals with checking if an image matches its identity

1. OBJECT DETECTION

Object detection is the craft of detecting instances of a certain class like animals, humans and many more in an image or video. Object Detection is the task of simultaneously classifying (what) and localizing (where) object instances in an images, it is a subset of object recognition. An Object detection model can identify which of the known set of objects might be present and provide information about their positions within the image. Detection is done by sliding boxes multiple sizes over an image and testing each box for the presence of the target object

2.4 REVIEW OF RELATED WORKS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S/N | AUTHOR/YEAR | TITLE | OBJECTIVE | METHODOLOGY | OUTCOME | GAPS |
| 1 | Amaia Salvador et al.  2019 | Inverse Cooking: Recipe Generation from Food Images | An inverse cooking system that recreates cooking recipes given food images | The use of a novel architecture, modeling their dependencies without imposing any order, and then generates cooking instructions by attending to both image and its inferred ingredients simultaneously. It extensively evaluates the whole system on the large-scale Recipe database | The model predicted sets of ingredients from food images, showing that modeling dependencies matters. It also considered instruction generation conditioned based on images and inferred ingredients, highlighting the importance of reasoning about both modalities at the same time | 1. The dataset used in the nutrient project was only specialized to food and drinks consumed in Central Europe. |
| 2 | Simon Mezgec et al. 2019 | Mixed deep learning and natural language processing method for fake-food image recognition and standardization to help automated dietary assessment. | To establish a combination of an established and a validated food-choice research method with a new food matching technology to automate data collection and analysis. | The methodology combines fake-food image recognition using deep learning and food matching and standardization based on natural language processing | The ﬁnal accuracy of the deep learning model trained on fake-food images acquired by 124 study participants and providing ﬁfty-ﬁve food classes was 92·18%, while the food matching was performed with a classiﬁcation accuracy of 93%. | This model was developed with the aim to detect fake food only, so it is very inapplicable in other aspects of food computing and food classification for dietary supervision. |
| 3 | WEIQING MIN et al.  2019 | A Survey on Food Computing | A study on different methods from computer science for food related studies including the analysis of food data with different modalities | A survey that targets the study of computing technology for the food area and offers a collection of research studies and technologies to benefit researchers and practitioners working in different food-related fields. | The survey provides an extensive review of the most notable works to date on the datasets, tasks and applications of food computing, from food-oriented data acquisition and analysis, perception, recognition, retrieval, recommendation, prediction and monitoring to its various applications and services. | While the survey gives a broad overview of food computing, it is still a qualitative view of this aspect of computing, hence it does not have any quantitative results on food computing. |
| 4 | Lili Pan et al.  2019 | Image Augmentation-Based Food Recognition with Convolutional Neural Networks | The objective of this study was to develop a novel approach to image retrieval for food ingredients which would allow the functionality of small sized food datasets. | In the development of the new approach, image augmentation was used using image transformation techniques to enlarge the datasets. | The model achieved an accuracy of 88.4% which was more than 5% accurate than other models with larger datasets. | While this model helps solve the problem of using deep convolutional neural networks with small datasets, the solution is more of a workaround rather than a direct solution as it uses image augmentation to achieve a dataset large enough for the neural network. |
| 5 | Wenyan Jia et al.  2019 | Automatic food detection in egocentric images using artificial intelligence technology | To develop an artificial intelligence (AI)-based algorithm which can automatically detect food items from images acquired by an egocentric wearable camera for dietary assessment. | To study the human diet and lifestyle, large sets of egocentric images were acquired using a wearable device, called eButton, from free-living individuals. The images of foods taken from these devices were used to create to separate datasets named: eButton1 and eButton2 | A cross data-set test was conducted on eButton data set 1. The overall accuracy of food detection was 91·5 and 86·4 %, respectively, when one-half of data set 1 was used for training and the other half for testing. For eButton data set 2, 74·0 % sensitivity and 87·0 % specificity were obtained if both ‘food’ and ‘drink’ were considered as food images. Alternatively, if only ‘food’ items were considered, the sensitivity and specificity reached 85·0 and 85·8 %, respectively | The only limitation of this work was the fact that the Neural network developed only recognized food and did not perform image classification or ingredient data retrieval. |

1. 2.4.1 REVIEW ON INVERSE COOKING: RECIPE GENERATION FROM FOOD IMAGES
2. During the course of this research we came across this article that aimed at recreating food recipes given food images and it peeked an interest to us as it has certain similarities to our proposed work. As our system, this system predicts ingredients as sets by means of novel architecture and then generates cooking instructions by attending to both the image and its inferred ingredients simultaneously. After extensive evaluation of both system on the scale of 10,000 – 1,000,000 datasets we can say that both systems are able to achieve high accuracy in making predictions regarding food.
   1. GENERAL COMMENT ON REVIEWED LITERATURE

As identified above the limitations regarding the related works will be addressed in our project as it will be available without the use of internet, every product will not only be recognized but classified based on its content and not appearance, the cost of getting the application will be of little or no expense.

**CHAPTER THREE**

**SYSTEM METHODOLOGY AND DESIGN**

3.0 INTRODUCTION

After much review of related works and project, we proceeded to document the process that we went through in the development of the project. This chapter contains the system design, model and specification. It explains the model that has been adopted during this research to implement the design of the software in prospect. In the course of this chapter we will consider the software development methodology adopted for this project and why it was adopted. We will also conduct analysis on the system as a whole and the requirements for which the system must meet in order to satisfy the development goal.

Further down this project we will explain the system and its design which will lay emphasis on

1. The System Structure
2. Database Design
3. System Design
4. System Security Implementation

The methodology used after a systematic review of the related literatures for the design of image identifier application is the Waterfall Software Development Process Model.

3.1 NATURE OF THE PROJECT

This project is a publicly available web based API that can be installed by anybody in possession of an Android device. It will also be available as a web based application that will be accessible through web browser and will be hosted on web browser. It will implement a database system for the storage and management of data.

3.2 SYSTEM ANALYSIS AND DESIGN

System Analysis and Design (SAD) explains the requirements and materials that are put together to facilitate the development of a software or system or a solution to an identified problem. System Analysis and Design is used to study, design and implement enhancements in the support of users and the functioning of the tasks that can be accomplished using computerized systems.

System analysis can therefore be explained simply as the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements to the system. It is also where requirements identified for the proposed system takes place. A system analyst usually carries out this task.

System design is a process of planning a new business software or system or one to replace or complement an existing system. It is the process of defining the architecture, components, modules, interface and data for system to satisfy specified requirements”.

3.3 DESIGN MODEL

“The software process model may be well-defined as an abstract representation of a software process. It is a set of related activities that leads to the production of the software.

Software process models integrate software engineering approaches and practices and are the fundamentals for managing large-scale software and IT projects. “The variations of the model accommodate the generic framework activities which defines a workflow that invokes each activity in different manner this is why Ion Summerville is 2013 acknowledged the fact that in order to satisfy engineering principles and practices software development methodologies

Must be adopted.

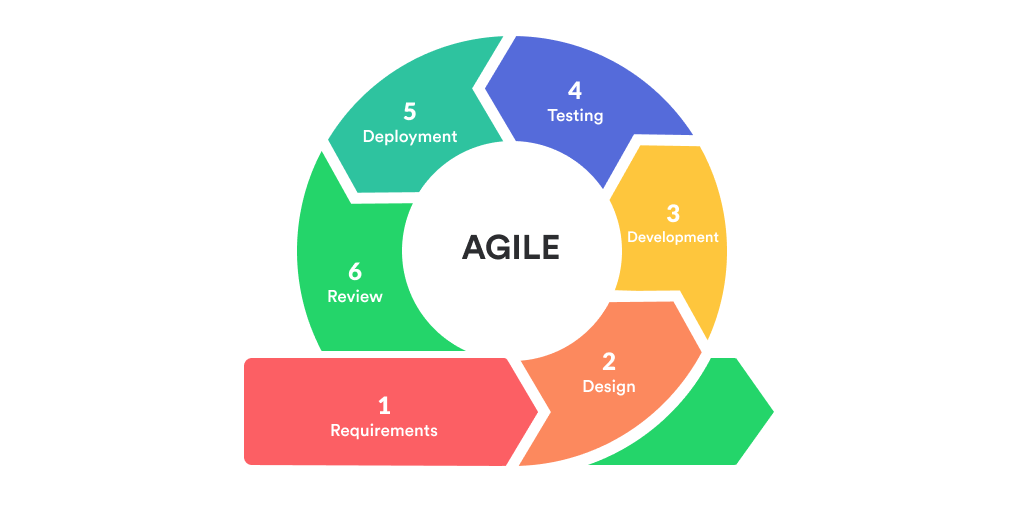
The creation of the model for proper understanding of the requirements and how best to achieve them is entailed in the analysis and design. Considering the nature of the system as stated in the section above and other factors like the requirements and the structure of the system that will still be explained along the line, we have adopted the Agile Development system design model because the agile software development emphasizes on four core values:

1. Iterative Development:- Because of the nature of the software and the need for a substantial dataset , the software will have to evolve and grow based on its daily input from users hence the need for a process model that support iterative development.
2. Intensive Team interactions: - As this is a hands on group project, a process model that supports team work will be more suitable than one that does not. Each Agile iteration usually takes several weeks and aims to delivers a complete working software version which can be more feasible if more than one person is working on the software. This fosters quick development but makes its maintenance more complicated as more time is spent to find the problem when there's no detailed software description.
3. Early customer feedback:- Agile is about working in close collaboration both across the team and with the customers and because the software will be available as a web based application customer feedback is essential. As earlier said in the significance of this study, the software will be able to make predictions to help the customers dietary needs based on health preference, therefore the need of a process model which supports customer involvements and early feedback in order to improve the quality of the software.
4. Working Software:- The models of this group put more focus on delivering a functioning part of the application quickly. They pay less attention to detailed software documentation (detailed requirement specification, detailed architecture description), and more to software testing activities which is essential to this study as there is a timeline and the milestones have to be attained.
5. Quick response to change :-They also allow for continuous software improvement with easy fixes and changes, quick updates, and feature addition, and help to deliver applications that satisfy users’ needs better.

Accordingly, in the course of this research, existing systems and related works have been seen to adopt the Agile process models and have produced working and successful applications in the field of artificial intelligence this is why Andrej Karpathy, a Director of AI at Tesla agrees that Agile processes can make this cycle go faster, since engineers will choose a smaller feature set to focus on for 2-4 week sprints rather than attempt to build an entire piece of software in one go.

Although Agile process model is suitable for this project it consists of some limitations like due to its the lack of detailed planning and openness to changes it makes it difficult to accurately estimate budget, time and people required for the project

Below is a graphical illustration of the various steps involved in the agile development methodology”.”



*Figure 3 Diagrammatic representation of the Agile Development Design Model.*

3.2.1 STAGES OF THE ACTIVITIES

**Planning and Requirements Gathering:** The Analysis Phase is where the project lifecycle begins. The Analysis Phase is where you break down the deliverables into the more detailed business requirements. The Analysis Phase is also the part of the project where you identify the overall direction that the project will take through the creation of the project

Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This formal consists of four basic steps.

1. **Elicitation** – I ask questions, you talk, I listen
2. **Validation** – I analyze, I ask follow-up questions
3. **Specification**– I document, I ask follow-up questions
4. **Verification** – We all agree
5. **Design :** At the second phase of the software development life cycle, the developers are actually designing the architecture. All the different technical questions that may appear on this stage are discussed by all the stakeholders, including the customer. Also, here are defined the technologies used in the project, team load, limitations, time frames, and budget. The most appropriate project decisions are made according to the defined requirements..
6. **Iteration/construction (coding):** After the requirements approved, the process goes to the next stage — actual development. Programmers start here with the source code writing while keeping in mind previously defined requirements. The system administrators adjust the software environment, front-end programmers develop the user interface of the program and the logics for its interaction with the server.
7. **Testing and Debugging:** The testing phase includes the debugging process. All the code flaws missed during the development are detected here, documented, and passed back to the developers to fix. The testing process repeats until all the critical issues are removed and software workflow is stable. In the course of this project we will be undergoing the four types of testing namely
8. **Unit Testing**
9. **Integration Testing**
10. **System Testing**
11. **Acceptance Testing**
12. **Release:** In this phase you’re nearly ready to release your product into the world. The finished product after the testing will undergo about 10-13 iterations to confirm the finality of the software,
13. **Maintenance and User Feedback:** When the program is finalized and has no critical issues it is time to launch it for the end users. After the new program version release, the tech support team joins. This department provides user feedback; consult and support users during the time of exploitation. Moreover, the update of selected components is included in this phase, to make sure, that the software is up-to-date and is invulnerable to a security breach.

3.3 REQUIREMENT SPECIFICATION

For any App to be developed, its requirements must be examined by a system analyst, then the problem the system wants to solve must be defined, the user must specify what he wants and expect from the App, that is user requirement. The App is developed based on the user requirement.

The requirement specification is the description of what the users of the App will be able to do with the App, what the system will do on its own, and the functionalities an App should have and how it should react in certain situation. Requirements are descriptions of the functionalities or services provided by a system and its operational constraints that reflect the need of customers for a system that help solves some problems. Two basic types of requirement specification include

1. Functional requirements
2. Non-functional requirements”

3.3.1 “FUNCTIONAL REQUIREMENT

“These are statements of services that system will provide, how the system will react to particular input and how the system will behave in a particular situation. It describes the system functions services and operational constraints in details and documentation will precisely define what will be implemented. The client expects the system to maintain information he is interested in but not to process them, therefore the functionality requirement is related to the information content”.

3.3.2 NON-FUNCTIONAL REQUIREMENT

These are otherwise known as “Quality Attributes” as they “show the constraints on the functionality of the system, constraints such as time and speed. You can write a specific line of code to implement them, rather they are “emergent” properties that arise from the entire solution. Some of these requirements are listed below

1. **Efficiency**

“Efficiency can be explained at the extent to which the software system handles capacity, response time and throughput. Efficiency requirement address the user concern for how far the system functions, how efficiently the system takes input and processes outputs, and how much can be processed at a time. When eliciting efficiency requirement consider needs regarding response time, throughput, storage capacity and process capacity”.

1. **“Reliability**

Reliability is the extent to which software system consistently performs the specified functions without the failure. The software should not crash on software bugs. It should be able to detect error and correct the errors for example if a wrong username or password is used, the software should throw an error and stop at any attempt to login”.

1. **Integrity**

Integrity is the “degree to which the data maintained by the software are accurate, authentic and without corruption”. Integrity requirement address the user concern for the accuracy and authenticity of the data. The database feature of the software should help ensure identification, authentication, authorizing and accountability of user’s login process is monitored.

1. **Modifiability**

Modifiability is the “degree to which changes to a software system can be developed and deployed efficiently and cost effectively. The” software should be created with the intention that upgrades would be done. Therefore, a provision for adjustment should be made.

1. **Security**

The software would have provisions for their sessions to be secured. “This would be done by implementing a criterion for registration and login procedure””

3.3.3 SYSTEM REQUIREMENTS

“System requirements give descriptions to the systems services, that is, what is expected from the system as a whole. The system requirements contain and describes the user requirements from the system”

1. Grant access to anybody that uses the system
2. Grant access to high-set functions when users they log in
3. Grant ability to view Products
4. Grant ability to give dietary recommendations based on eating preference (Registered Users)
5. Grant ability to select products and put into the cart
6. Grant ability to purchase products
7. Grant ability to view purchase history

3.3.4 USER REQUIREMENTS

“They are used to describe the necessities/requirements/prospects of the end users of the system from the system. They describe the activities that users must be able to perform”

1. User must be able to access the mobile application
2. User must be able to login
3. User must be able to view and select products
4. User must be able to Purchase products
5. User must be able to make reviews on products
6. User must be able to view product history
7. User must be able to access health advice on various products (Registered Users)

Viii User must be able to log out

* 1. **ANALYSIS OF THE PROPOSED SYSTEM**

The main objective of the project is to build an Image based Food identifier for dietary supervision using computer vision technology other functions the system will provide includes:

1. **Secure access:** at the launch of the application, the user is prompted for an authenticated username and password without which access would be denied. Users are allowed to create accounts and change password after verification through email.
2. **Proficient storage facility:** every user data will be automatically stored in the system’s memory
3. **Management:** as each user would be assigned a tutor/lecturer to facilitate and monitor the learning process of each student
4. **User-Friendly Interface:** the software has a very user friendly interface. Thus the users will feel very easy to work on it.

**3.5 CHOICE OF PROGRAMMING LANGUAGE**

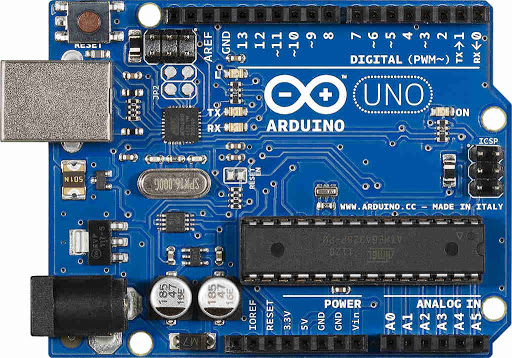
Selecting the correct programming language is a decision that needs to be made very early in the process. We treat this decision-making process with a very high level of emphasis, and it’s a collaborative effort amongst our internal team as well as our clients. Before choosing a programming, language there are so many dependent factors. Some of which are; technological fit, architectural environment, overall client fits. The application built is a web based application programming interface, an android based app with java programming language. “An Android application is a mobile application developed using the Android SDK and targeted toward devices running the Android operating system or runtime. Native apps are the most common applications that you can find in app stores (application marketplaces) today. Native applications are usually developed using higher level programming languages, such as Java for Android, Objective-C for iOS, or C# for Windows Phone. The native APIs are provided to the developer as part of the platform SDK.

The coding/programming language used were explained and include the following

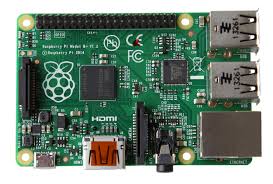
1. “Java: The Android platform allows developers to write managed code using Java to manage and control the Android device. Android applications can be developed by using the Java programming language and the Android SDK.
2. PHP: stands for Hypertext Pre-processor. It's an open source, server-side, scripting language  for the development of mobile and web applications. By scripting language, we mean a program that is script-based (lines of code) written for the automation of tasks.
3. HTML: is used quite a lot in data transfer because it's an intermediate, platform independent **format**. XML Schema can be used to enforce a particular structure and **form** in an XML document, and thus can be used to create custom mark-up languages”
4. MySQL WORKBENCH: This is a unified visual tool for database architects, developers and DBAs. It provides data modelling, SQL development and comprehensive administration tools for server configuration, user administration, backup and much more.
5. JAVASCRIPT: This is a programming language of the web used to build interactive websites. It can be used to add dynamic behaviour, store information and handle requests and response.

DEVELOPMENT TOOLS

1. Arduino Uno – This is a microcontroller board developed by Adruino.cc, it is equipped with a set of both digital and analog IO pins that passes both power and data to the components.



1. *Figure 4 An Arduino board*
2. Raspberry Pi – This is a small single-board computer that runs on Linux which has General-purpose IO pins.



1. *Figure 5 A Raspberry Pi board*
2. Arduino software (IDE)- This IDE makes it easy to write and upload instructions to the Arduino board

3.5 DEVELOPMENT ANALYSIS

“The analysis of a system depends on its model and it’s a very important process that precedes system implementation. It involves a number of steps that ought to be followed”. These steps include:

1. **“Requirements Identification:** this entails ensuring that the necessary tools needed and an outline of all other requirement are established are acquired from the beginning.
2. **The Design:** after the application requirement and tools are in place, the project team must know all entities and how the app will look and how it will function.
3. **The Implementation:** After the design analysis, the project team can now implement the design of the system.
4. **Validation:** after designing the app, the team must test the designed app for bugs or errors and ensure that the app meets all requirement and functionalities needed by the end users.
5. **Maintenance:**  the App must be maintained from time to time. For instance, adding new content, removing old content”

3.6 SYSTEM DESIGN

­­ The system design phase comes up just immediately after the system analysis, because the system to be developed would have been critically studied with all its requirements clearly stated. The System design phase is a description of how the system will fulfil or meet its stated objectives (Henry et al. 2014). The system design gives a logical framework or a conceptual model of the systems to be developed so that the software developer can know how the system is expected to look and behave in different given scenarios. The system design involves the plan of how the system to be developed will behave and how every component of the system will function”. The system was designed with Graphical User Interface such that users of the system will launch it with web browser thereby login in through the provided login interface After login, the user interface will present the tasks or menu of actions to be taken to the user on the user panel. The system was designed using UI/UX design tools like Adobe Experience Design and Framer JS because of their interactive and flexible characteristics.

”

3.6.1 CODING

The system was designed using technologies that support the building of native android mobile software. The system will be designed to be backward compatible with older version of the android OS. The languages to be used for the coding includes HTML, JAVA, JAVASCRIPT and MySQL.

3.7 CONCEPTUAL DESIGN

“Conceptual Design was the earliest phase of design in which drawings or solid modes were the dominant tools and products. The conceptual design phase provided a description of the proposed system in terms of a set of integrated ideas and concepts about what it was to do, behave and look like, that was understandable by the users in the manner intended” (Henry et al. 2014). According to Fourer (2003), the conceptual design includes the system architecture such as UML Diagrams, Flow charts, the Use Case Diagrams as well as the Activity Diagram as indicated below.

3.7.1 USE CASE DIAGRAM

“A Use Case Diagram in its simplest form is a representation of a user’s interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. Use-case diagram can help provide higher-level view of the system. It has been said before that “Use case Diagrams are the blueprints for your system” (Wiki 2017). They provide the simplified and graphical representation of what the system must actually do. The Use-Cases are divided into two (2) categories; the detailed roles of each user and the general outlook of the entire system”” (Whitten 2004).

The various actors identified are:

**Primary Actors**

* The Customer
* The Administrator

**Secondary Actor**

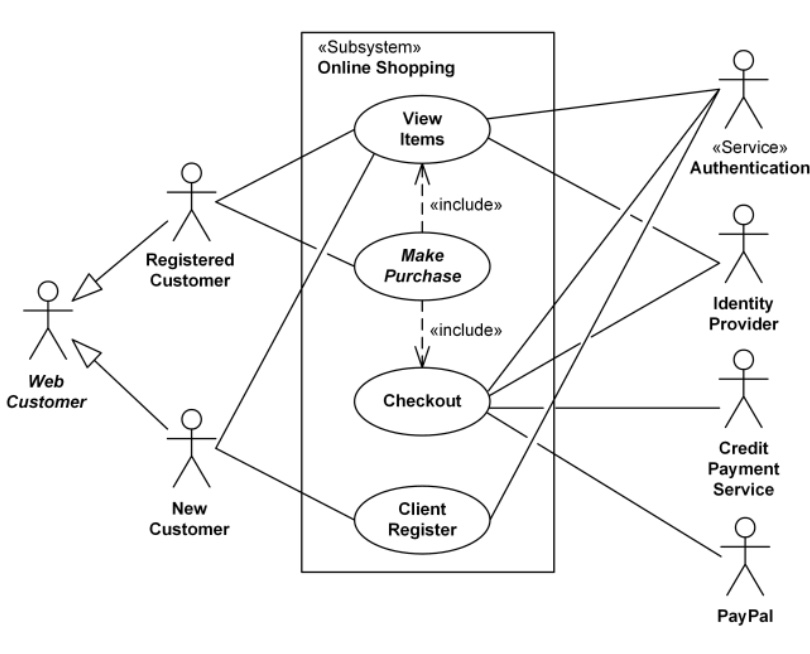
* The Developers
* The System

**Actor: The Customer**

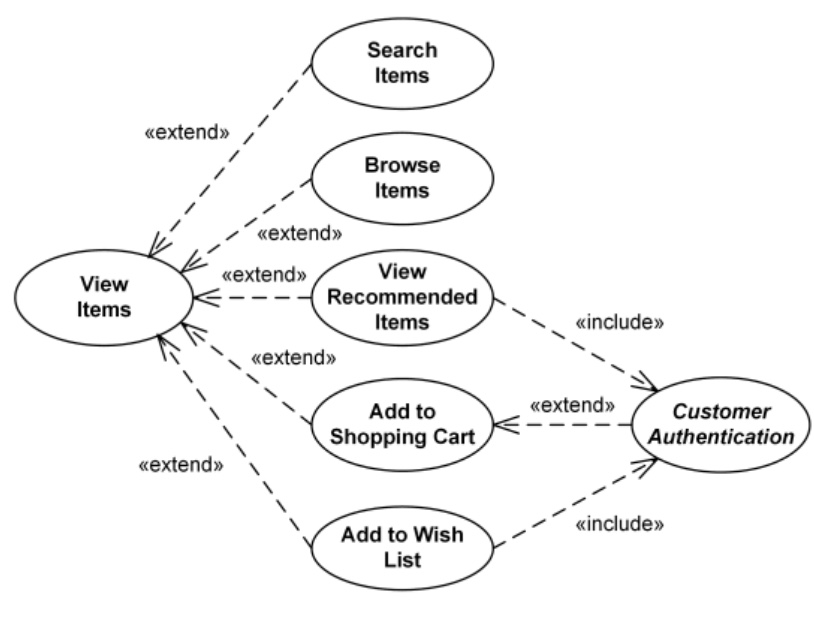
This user can “perform the following functions on the system”

1. User can login to the application
2. User can view the Products
3. User can view nutritional facts about the product
4. User can add product to cart
5. User can purchase products
6. User can make reviews
7. User can log out of the application

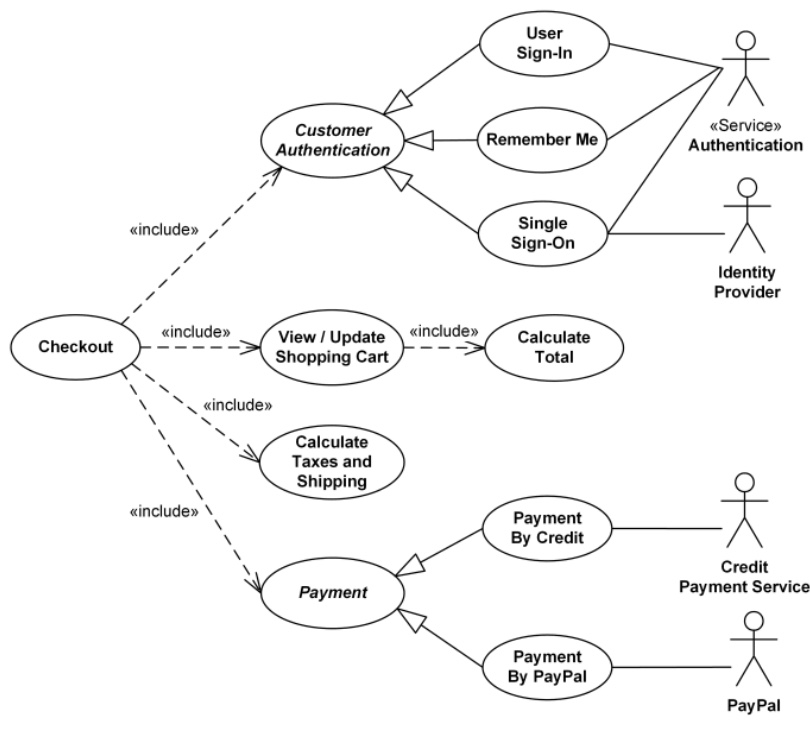
The following Use Case diagrams is to aid better understanding and give a virtual representation of the system.



*Figure 6 An Online Customer Use Case*



*Figure 7 A View Items Use Case*

 *Figure 8 A Make Payment Use Case*

3.7.2 ENTITY RELATIONSHIP DIAGRAM

“An entity-relationship model (ER model) describes the inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which groups the things of interest) and specifies relationships that can exist between instances of those entity types. In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database”. According to the entity diagram below the system database has only four (4) tables each table consist of set of concurrent activities that help immensely in ensuring the smoothness of the system usage.

3.7.3 SYSTEM FLOWCHART

“System flowchart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting these with arrows. This diagrammatic representation can give a step-by-step solution to a given problem. Process operations are represented in these boxes, and arrows connecting them represent flow of control. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields. Different symbols are used in the flowchart to represent input, output, decision, connectors and process” (Nakayenga, 2011).

3.10 MAINTENANCE

“This phase of the waterfall model is virtually never-ending phase. Generally, problems with the system developed (which are not found during the development life cycle) come up after its practical use starts, so the issues related to the system are solved after deployment of the system. Not all the problems come in picture directly but they arise time to time and needs to be solved; hence this process is referred to as maintenance”.

**CHAPTER FOUR**

**IMPLEMENTATION AND TESTING**

In this chapter, implementation from the above-mentioned design from user case discussed in previous chapter is shown as Figure 4.1, short narrative of the screen shot is also included. Algorithm and models used within the conduit platform has been instantiated on an Android device for the first phase implementation feasibility index of this project product called” SMART EATS

**4.1 INTRODUCTION**

The implementation and testing of the software would be detailed in this chapter.

Implementation entails the construction of the proposed system using the programming languages and software tools indicated in the methodology. Several high-fidelity designs in form of screenshots was used to illustrate the graphical layout of the SMART EATS and the various modules that exist in the web based software. The function of each module will also be detailed. Unit testing and integration testing will also be included.

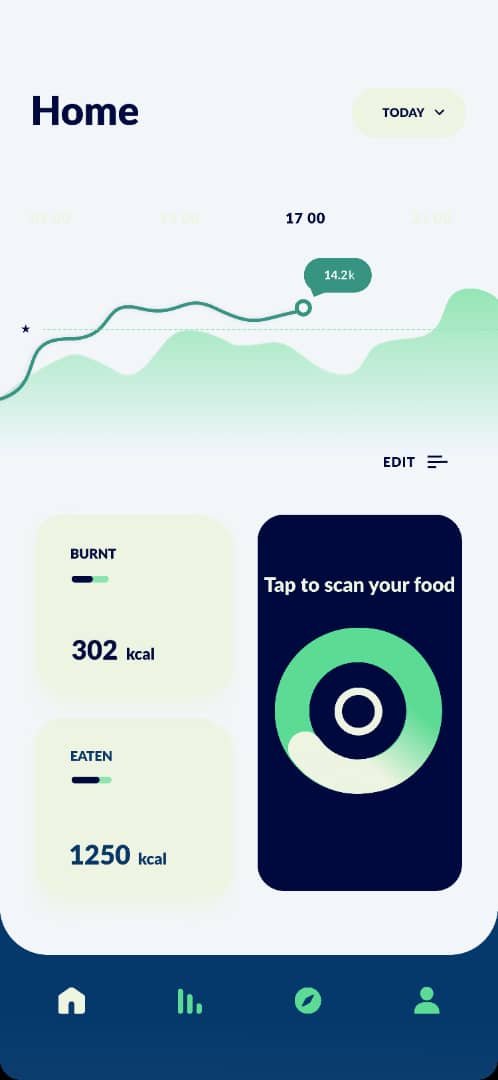
**4.2 SYSTEM REVIEW AND IMPLEMENTATION**

**4.2.1 SPLASH PAGE**



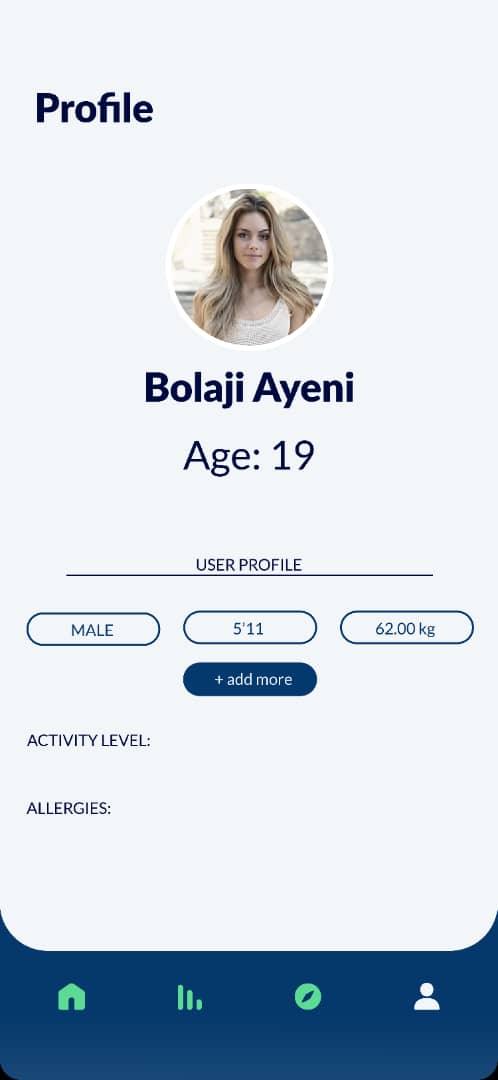
1. *Figure 6 SMART EATS Splash Page*

**4.2.2 HOME PAGE**



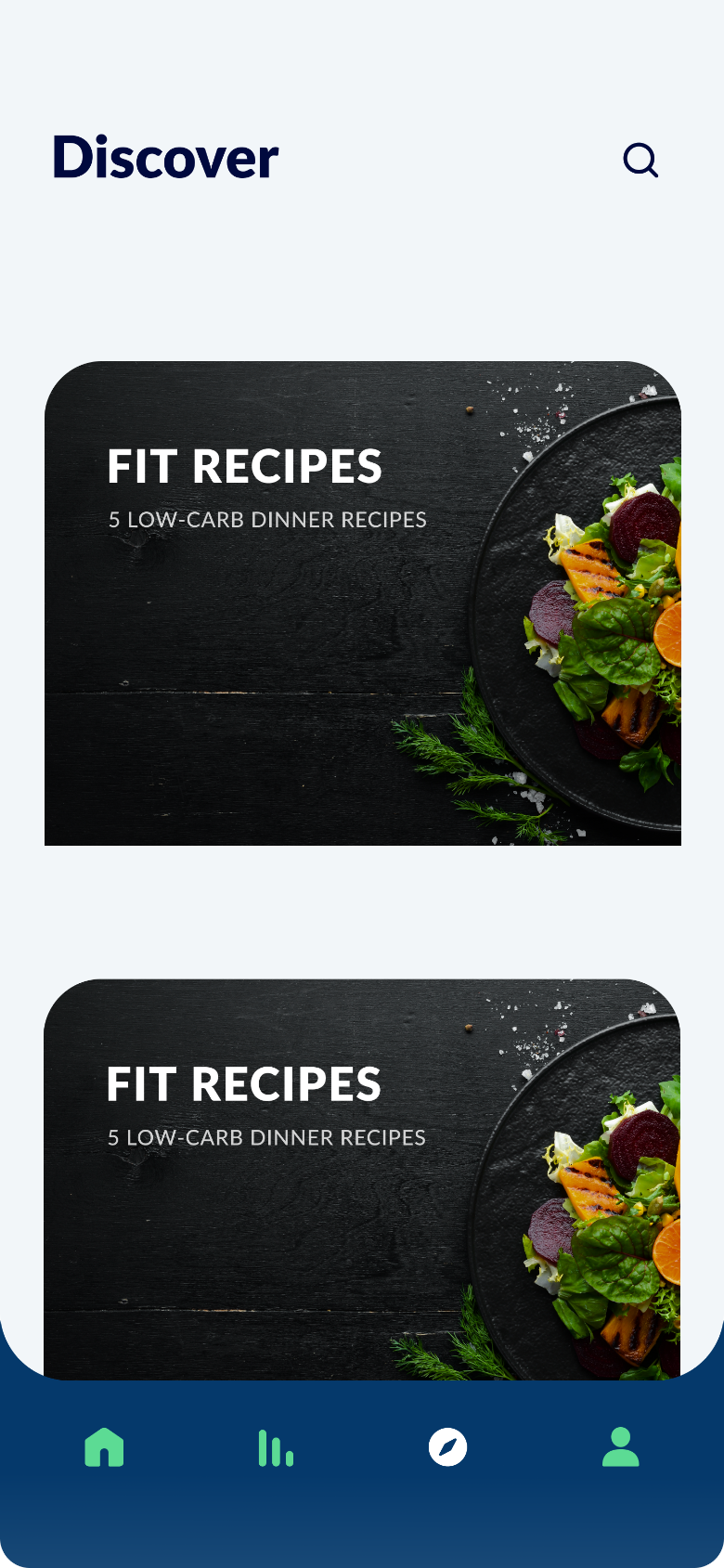
1. *Figure 7 SMART EATS Home Page*

**4.2.3 PROFILE PAGE**



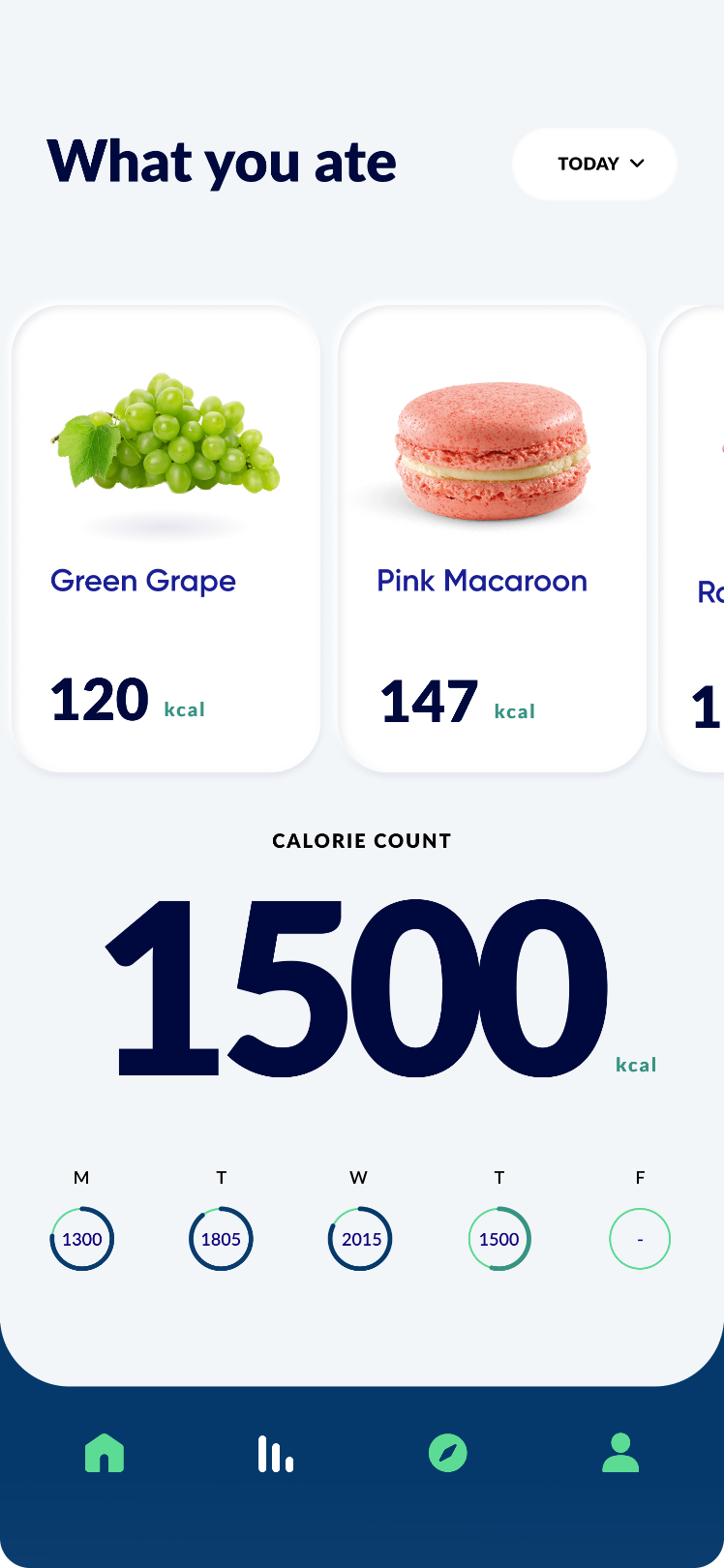
1. *Figure 8 SMART EATS User Profile Page*

**4.2.4 DISCOVER PAGE**



1. *Figure 9 SMART EATS Discover Page*

**4.2.5 USER CONSUMPTION PAGE**



1. *Figure 11 SMART EATS User Consumption Page*

**4.3 SYSTEM TESTING**

System Testing is the testing of a complete and fully integrated software product. Usually, software is only one element of a larger computer-based system. Ultimately, software is interfaced with other software. The aim of testing is to examine fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole. As the software is created and added to the developing system, testing is performed to ensure that it is working correctly and efficiently. Testing is generally focused on two areas:

1. **Internal efficiency (Alpha testing)**
2. **External efficiency (Beta testing**)

They are being regarded to as the “Customer Validation methodologies (Acceptance Testing types) that help in building confidence to launch the product, and thereby results in the success of the product in the market. Even though they both rely on real users and different team’s feedback, they are driven by distinct processes, strategies, and goals. These two types of testing together increase the success and lifespan of a product in the market. Testing can be a labour-intensive process, due to its iterative nature. Other forms of testing may include:

4.4.1 BLACK BOX TESTING

The interface testing was conducted using the black box testing technique. “It is a Software Testing method that analyses the functionality of an application without knowing much about the internal structure of the item that is being tested and compares the input value with the output value”. The black box testing is aimed at finding some bugs such as “Incorrect or missing functions, Interface errors, Errors in data structures or external database access, Behaviour or performance errors, Initialization and termination errors. Black box testing” technique has been used in testing the user interface and database to ensure user friendliness and enabling new users to become acquainted

with your system without learning too much which will aid easy navigation around the system. . This method of testing was adopted to eliminate any chances of user dissatisfaction, if the users are not able to navigate the application or carry out use cases then the application is essentially not serving its purpose.

4.4.2 HAPPY-PATH TESTING

“The objective of Happy Path Testing is to test an application successfully on a positive flow. It does not look for negative or error conditions, but focuses only on the valid and positive inputs through which application generates the expected output”. “The happy path user will understand clearly he domain and functionality of an application, familiarity with the UI, operates exactly the way according to the software, remembers everything, Understands the prompts and responses made by the product, doesn’t make any mistake, does not get distracted”. Happy path testing enables the workflow of the application to be tested by supplying valid data,

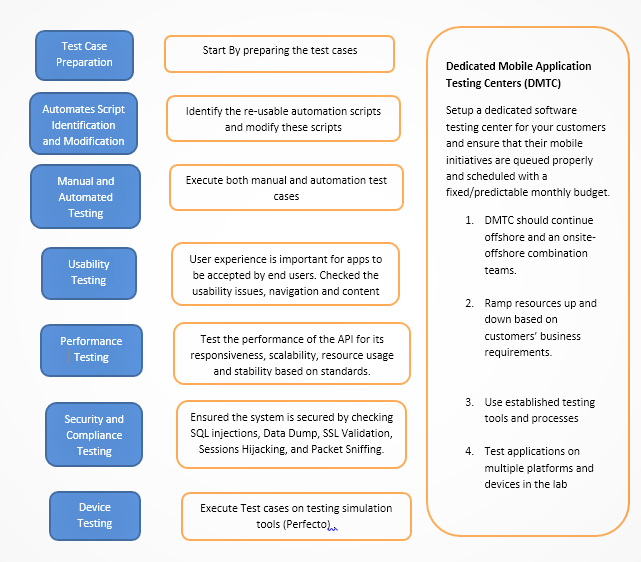
4.4.3 BACK-END TEST

The back end is used for clarifying the integrity, accuracy and consistency of data in the database. The database was designed using MySQL and was tested using the apache Wamp server. Data stored in the database is done using the JSON (Javascript Object Notation) format which makes use of key value pairs resulting In a tree like structure representing the totality of data that is stored instead of schemas and tables that are present in conventional SQL databases, this means that the constraints which places restrictions on the type of data that is stored is provided in the application code and not in the back-end infrastructure.

The testing was done in this project by rendering the coded programs into web browser to view how and where all the elements appear in the web app. These are some of the test that were employed

1. **Identified Type of Testing**: SMART EATS application was made to conform to the Andriod app so that the end consumers can get familiar with diverse devices. To ensure that he app worked on all android platforms we selected a combination of manual testing, automation testing and testing in cloud simulator.
2. **Manual and Automated Testing**: The development process followed for tgis project was SCRUM ( part of agile testing approach ) and each sprint was two weeks ago. Every two weeks development team delivered a logically completed product to the quality assurance Team and Quality Assurance(QA) did run test cases on the build. Regression was a challenge because in each build QA team had to ensure that the previously completed items were still working. To accomplish this. QA team had automated basic set functionalities using Experitest and ran the automation scripts on each build which saved good amount of time. We used JIRA to file and track defects.
3. **Usability Testing** : User Acceptance testing was conducted for this QA/ST build. This was achieved by asking the beta users ( a set of hundred online shoppers) to browse every single page within the mobile app. This ensured that every bug was fixed before the app is tagged ready for release and that the app captured all necessary features and functionality.
4. **Performance Testing** : The functionality, performance (Stability, responsiveness, resource, usage, stability parameters as per standards) and user interface for the delivered build was thoroughly tested.
5. **Device Testing**: The app was tested across multiple OS versions/devices using the device anywhere simulation tool and also physically tested in our dedicated mobile application testing labs. Within the time frame given, Andriod OS such as 4.0 to 6.0 were covered.
6. **Mainatined Test Case Sheets**: During the whole process the QA team maintained the test case sheet (with all functionalities implemented) at last certified to build release. The entire end-to-end mobile application testing process involved collective effort between onsite and off-siteteams.
7. **Generated Test Summary Report**: This I sa test summary report which provides details of any important information uncovered by the test conducted, including assessments of the quality of the testing effort, the quality of the software system under test and statistics derived from incident reports. This helps to improve any future test planning and extension from SMART EATS upgrades in the future.

The figure below shows the different testing phases SMART EATS has passed through.

*Figure 11- SMART EATS Testing Process Model*

**CHAPTER FIV****E**

**5.1 SUMMARY, CONCLUSION AND RECOMMEDATION**

**5.1.1 SUMMARY**

**SMART EATS** is an image-based food classifier that Identifies and classifies images of food and outputs not only the name of the food but its nutritional contents and then makes predictions to the consumer according to his/her dietary needs. SMART EATS is powered using object recognition

It was design using an object recognition algorithm known as DCNN which stands for Deep convolutional neural networks because of its ability to make high accurate predictions of its outcome and also DCNN makes the image processing computationally manageable through filtering the connections by proximity. This simply means it identifies the relationship between objects based on how similar the pixels are which eliminates background clustering thereby reducing the risk of wrong predictions. This system was built from studying existing systems and acquisition of large amount of datasets as said earlier, data of food images and data from user inputs. The system has been tested fully and discovered to be fully functional and very optimal that is it has eliminated all the items stated in the problem statement. This system if properly utilized and managed will improve healthy living and fulfil every item discussed in the significance of the study. It will also be a benchmark for any application relating to itself that wants to focus on local food and how it can aid eating habits

This project is could be an asset to the Babcock investment group particularly Babcock Shoprite (Babrite) in which simplicity is introduced and complexity forgone for the sake of easier system. If this system is put to use, it will be a very good addition to the school and will reduce the health deficiencies and well as give a more precis inventory as the application will be available online hence digitalized records will be kept.

**5.1.2 CONCLUSION**

As earlier said by Bethenny Frankel a diet is a bank account, good food choices are good investments, in the a world where technology is fast approaching to be the way of life of every individual in every aspect of life the intervention of a technology in a country where eating disorders like obesity, diabetics and anorexia are at its peak is likely to be very much appreciated.

This system underwent a complex process which required determination of a suitable classification system, selection of training samples, image preprocessing, feature extraction, selection of suitable classification approaches, post‐classification processing, and accuracy assessment. The user's need, scale of the study area, economic condition, and analyst's skills are important factors influencing the selection of remotely sensed data, the design of the classification procedure, and the quality of the classification results. With these, the objectives of the system design is achieved. In other to allow future expansion of this system, it has been designed in such a way that it gives a room for modification should any need arises. This system was designed with a well-designed database that has been normalize and does not give room for redundancies. Although the system has been trained using neural networks of about 10,000 local foods it is not undermined that new foods will be produced as the meal innovations are growing exponentially as the days go by therefore any need to continue this work will have to keep in mind that new datasets will be needed to be trained so as to maintain the relevance of the system.

**5.2 RECOMMENDATION**

SMART EATS will serve many beneficiaries including: medical practitioners, culinarians or even the consumers, it should be adapted by the National Agency for Food Drug Administration Control (NAFDAC) for its ability to give information on food based on constituents.

REFERENCES

Abel, T., & Satterfield, D. (2020). HCI and User Experience Design Education: Principles for Ethical and Responsible Practices in Inclusive Research and Application. In *UXD and UCD Approaches for Accessible Education* (pp. 80-95). IGI Global.

Aizawa, K. (2020). Image Recognition-Based Tool for Food Recording and Analysis:FoodLog. In *Connected Health in Smart Cities* (pp. 1-9). Springer, Cham.

Blechert, J., Lender, A., Polk, S., Busch, N., & Ohla, K. (2019). Food-pics\_extended—an image database for experimental research on eating and appetite: additional images, normative ratings and an updated review. *Frontiers in psychology*, *10*, 307.

Jia, W., Li, Y., Qu, R., Baranowski, T., Burke, L. E., Zhang, H., ... & Sun, M. (2019). Automatic food detection in egocentric images using artificial intelligence technology. *Public health nutrition*, *22*(7), 1168-1179.

Liang, D. I. A. O., & Gu, H. (2019). *U.S. Patent Application No. 16/230,815*.

Mezgec, S., Eftimov, T., Bucher, T., & Seljak, B. K. (2019). Mixed deep learning and natural language processing method for fake-food image recognition and standardization to help automated dietary assessment. *Public health nutrition*, *22*(7), 1193-1202.

Min, W., Jiang, S., Liu, L., Rui, Y., & Jain, R. (2019). A survey on food computing. *ACM Computing Surveys (CSUR)*, *52*(5), 1-36.

Pan, L., Qin, J., Chen, H., Xiang, X., Li, C., & Chen, R. (2019). Image augmentation-based food recognition with convolutional neural networks. *Comput. Mater. Continua*, *59*(1), 297-313.

Pan, L., Qin, J., Chen, H., Xiang, X., Li, C., & Chen, R. (2019). Image augmentation-based food recognition with convolutional neural networks. *Comput. Mater. Continua*, *59*(1), 297-313.

Qiu, B. (2019). *Food recognition and nutrition analysis using deep CNNs* (Doctoral dissertation, McGill University).

Recipe generation from food images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 10453-10462).

Salvador, A., Drozdzal, M., Giro-i-Nieto, X., & Romero, A. (2019). Inverse cooking:

Song, W., Jiang, N., Wang, H., & Vincent, J. (2020). Use of smartphone videos and pattern recognition for food authentication. *Sensors and Actuators B: Chemical*, *304*, 127247.

Sun, Y., Xue, B., Zhang, M., & Yen, G. G. (2019). Evolving deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation*.

Zhou, D. X. (2020). Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, *124*, 319-327.

Zhu, X., Li, Z., Zhang, X. Y., Li, P., Xue, Z., & Wang, L. (2019). Deep convolutional representations and kernel extreme learning machines for image classification. *Multimedia Tools and Applications*, *78*(20), 29271-29290.

**APPENDIX**

*import* {AppConfig} *from* *"./config"*;

*const* host *=* AppConfig.host;

*import* axios *from* *'axios'*;

*const* axiosClient *=* axios.create({

    baseURL: host,

    timeout: *50000*, *//50 seconds*

});

*class* ApiService {

*static* ping *=* *async* () *=>* {

*const* results *=* *await* axiosClient.get(*'/ping'*);

        console.log(results.data);

*return* results.data

    };

*static* loadClasses *=* *async* () *=>* {

*const* results *=* *await* axiosClient.get(*'/api/classes'*);

*const* data *=* results.data;

        console.log(data);

*return* data

    };

*static* predict *=* *async* (*uri*) *=>* {

*let* uriParts *=* uri.split(*'.'*);

*let* fileType *=* uriParts[uriParts.length *-* *1*];

*let* data *=* *new* FormData();

        data.append(*'file'*, {

            uri: uri,

            name: *`photo.${fileType}`*,

            type: *`image/${fileType}`*,

        });

        console.log(data);

*const* response *=* *await* axiosClient.post(*'/api/classify'*, data, {

            headers: {

*'Content-Type'*: *'multipart/form-data'*,

*'Accept'*: *'application/json'*

            }

        });

        console.log(response.data);

*return* response.data

    }

}

*export* {ApiService};

*import* React *from* *'react'*;

*import* { Platform, StatusBar, StyleSheet, View } *from* *'react-native'*;

*import* { AppLoading, Asset, Font, Icon } *from* *'expo'*;

*import* AppNavigator *from* *'./navigation/AppNavigator'*;

*export* *default* *class* App *extends* React.Component {

  state *=* {

    isLoadingComplete: *false*,

  };

  render() {

*if* (*!***this**.state.isLoadingComplete *&&* *!***this**.props.skipLoadingScreen) {

*return* (

        <AppLoading

*startAsync=*{**this**.\_loadResourcesAsync}

*onError=*{**this**.\_handleLoadingError}

*onFinish=*{**this**.\_handleFinishLoading}

        />

      );

    } *else* {

*return* (

        <View *style=*{styles.container}>

          {Platform.OS *===* *'ios'* *&&* <StatusBar *barStyle="default"* />}

          <AppNavigator />

        </View>

      );

    }

  }

  \_loadResourcesAsync *=* *async* () *=>* {

*return* Promise.all([

      Asset.loadAsync([

        require(*'./assets/images/robot-dev.png'*),

        require(*'./assets/images/robot-prod.png'*),

      ]),

      Font.loadAsync({

*// This is the font that we are using for our tab bar*

*...*Icon.Ionicons.font,

*// We include SpaceMono because we use it in HomeScreen.js. Feel free*

*// to remove this if you are not using it in your app*

*'space-mono'*: require(*'./assets/fonts/SpaceMono-Regular.ttf'*),

      }),

    ]);

  };

  \_handleLoadingError *=* *error* *=>* {

*// In this case, you might want to report the error to your error*

*// reporting service, for example Sentry*

    console.warn(error);

  };

  \_handleFinishLoading *=* () *=>* {

**this**.setState({ isLoadingComplete: *true* });

  };

}

*const* styles *=* StyleSheet.create({

  container: {

    flex: *1*,

    backgroundColor: *'#fff'*,

  },

});

*const* AppConfig *=*

{

    title : *"What Food Is It?"*

    ,host : *"https://food-img-classifier.herokuapp.com"*

};

*const* description *=* *`*

*This app lets you submit a photo of food and returns the predicted food category. 🍕*

*The model was developed using the [food-101 dataset](https://www.vision.ee.ethz.ch/datasets\_extra/food-101/) and the [fastai](https://github.com/fastai/fastai) deep learning library, which is built on PyTorch.*

*`*;

*export* {AppConfig, description}

import pandas as pd

import requests

import glob

from io import BytesIO

import numpy as np

import os

import shutil

import pprint

import json

learn = create\_cnn(data, models.resnet34, metrics=error\_rate,pretrained=True)

learn.lr\_find()

learn.recorder.plot()

lr = 1e-2

learn.fit\_one\_cycle(8 , lr)

model\_name="resnet34"

learn.save(f'{model\_name}-stage-1')

learn.load(f'{model\_name}-stage-1')

learn.unfreeze()

learn.lr\_find(start\_lr=1e-09, end\_lr=1e-3)

learn.recorder.plot(skip\_end=10)

learn.fit\_one\_cycle(5, max\_lr=slice(1e-8,1e-4))

learn.save(f'{model\_name}-stage-2')

learn.load(f'{model\_name}-stage-2');

learn.load(f'{model\_name}-stage-2');

interp = ClassificationInterpretation.from\_learner(learn)

interp.plot\_top\_losses(9, figsize=(15,11))

interp.plot\_confusion\_matrix(figsize=(50,50), dpi=30)

interp.most\_confused(min\_val=2)

final\_model\_name = f'{model\_name}-final'

learn.save(final\_model\_name)

learn.load(final\_model\_name);

learn.data.classes

data2 = ImageDataBunch.single\_from\_classes(path, data.classes

, tfms=get\_transforms()

, size=224).normalize(imagenet\_stats)

learn = create\_cnn(data2, models.resnet34)

learn.load(final\_model\_name)

data2.classes, data2.c

def fetch\_image(url):

response = requests.get(url)

img = open\_image(BytesIO(response.content))

return img

img = fetch\_image(bibimbap\_url)

pred\_class,pred\_idx,outputs = learn.predict(img)

pred\_class , pred\_idx, outputs

def predict(url):

img = fetch\_image(url)

pred\_class,pred\_idx,outputs = learn.predict(img)

res = zip (learn.data.classes, outputs.tolist())

predictions = sorted(res, key=lambda x:x[1], reverse=True)

top\_predictions = predictions[0:5]

pprint.pprint( top\_predictions)

return img.resize(500)

shutil.rmtree("../models",ignore\_errors=True)

final\_model\_directory = os.getcwd()+ "/../models"

final\_model\_name='model.pkl'

learn.export(final\_model\_directory+f"/{final\_model\_name}")

with open('../models/classes.txt', 'w') as f:

json.dump(learn.data.classes,f)

learn3= load\_learner(final\_model\_directory,final\_model\_name)