DEPARTMENT OF ROBOTICS & ARTIFICIAL INTELLIGENCE

Food Vision using EfficientNet & ResNet

Maria Ali (22108160)
Ali Haider (22108108)

Supervised by: Khawaja Tahir

Department of Robotics & Artificial Intelligence
Faculty of Computing and Engineering Sciences
SZABIST University
Islamabad, Pakistan

Fall 2024

Table of Contents

SRS Report

1. Introduction

1.1 Purpose

The Food Vision system is a state-of-the-art application designed to classify food items from images and estimate their nutritional content. By utilizing deep learning models like EfficientNet and ResNet, it offers accurate predictions and a seamless user experience. The software aims to bridge the gap between advanced AI technology and real-life dietary management needs. Its purpose can be outlined as:

- **Health Monitoring**: Assists users in tracking their daily nutritional intake, enabling better dietary decisions.
- **Convenience**: Simplifies the process of identifying food and its nutritional value without manual effort or prior expertise.
- **Educational Use**: Serves as a tool for dietitians and health professionals to educate clients on nutrition and calorie management.
- **Real-Time Insights**: Provides immediate results, making it suitable for quick assessments in fast-paced environments like restaurants or fitness centers.
- **Scalability**: Designed to expand into mobile applications, making it accessible anytime, anywhere.

Real-Life Applications:

- **For Health Enthusiasts**: Helps users manage their calorie intake and monitor nutrition, aiding in weight management and fitness goals.
- **For Dietitians**: Offers a reliable tool for creating personalized diet plans based on identified food items.
- **In Restaurants**: Assists chefs and diners by providing detailed nutritional insights into dishes.
- **For Researchers**: Facilitates research in AI-driven food recognition and its impact on health sciences.
- **For Everyday Users**: Provides a simple way to explore and learn about the nutritional content of their meals.

This integration of AI with practical use cases makes Food Vision a valuable asset in promoting healthier and more informed lifestyles.

1.2 Scope

The Food Vision system is a computer vision-based solution that leverages pre-trained deep learning models like EfficientNet and ResNet. The key objectives include:

- Classifying food items into 101 predefined categories.
- Estimating calorie values for the identified food items.
- Providing a user-friendly interface for image uploads and displaying results.

The project will be deployed as a desktop or web-based application and can be extended to include mobile app support in the future.

1.3 Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
CNN	Convolutional Neural Network, used for image recognition.
EfficientNet	A state-of-the-art CNN for efficient image classification.
ResNet	A deep learning architecture optimized for image classification.

1.4 References

1. Food-101 Dataset: https://data.vision.ee.ethz.ch/cvl/food-101.tar.gz
2. TensorFlow Documentation: https://www.tensorflow.org

1.5 Overview

This document provides a detailed specification of the Food Vision system, covering functional and non-functional requirements, system features, and external interface requirements. It will serve as a guide for both development and testing phases.

2. Overall Description

2.1 Product Perspective

The Food Vision system is a standalone application that integrates with TensorFlow and Keras libraries for food classification. It uses the Food-101 dataset as its primary training data and employs transfer learning techniques for high accuracy in classification and calorie estimation.

2.2 Product Functions

- **Food Classification**: Predicts the category of the food item from an uploaded image.
- **Calorie Estimation**: Estimates calorie values based on the classified food item.
- **Image Upload Interface**: Provides a simple interface for users to upload food images.

2.3 User Characteristics

- **End Users**: Health-conscious individuals and dietitians.
- **Researchers/Developers**: Individuals interested in improving or expanding the system.

2.4 Constraints

- Requires GPU for optimal performance.
- Food categories are limited to the 101 classes in the Food-101 dataset.
- Calorie estimates are based on standard nutritional data and do not account for portion size variation.

2.5 Assumptions and Dependencies

- Assumes users will upload clear images of food items.
- Relies on pre-trained deep learning models for accurate predictions.
- Future integrations may depend on APIs for real-time calorie data updates.

3. Specific Requirements

3.1 Functional Requirements

a. Upload Image:

- **Description**: Users can upload images of food items using the system's interface. The system prompts the user to upload an image and processes it through a machine learning model.
- **Functionality**:
  - Prompts the user to select an image file for upload.
  - Processes the uploaded image using EfficientNet or ResNet.
  - Displays the predicted food name(s) within 3-5 seconds.

b. View Nutrition Information:

- **Description**: Users can view detailed nutritional information for a predicted food item. The system fetches this data from a preloaded database and displays it in an easy-to-understand format.
- **Functionality**:
  - Allows the user to select a predicted food item.
  - Retrieves the nutritional data (e.g., calories, macronutrients) for the selected item.
  - Displays the information on the interface promptly.

c. Search Food:

- **Description**: Users can search for specific food items by entering a food name in the search bar. The system retrieves matching results and displays relevant details.
- **Functionality**:
  - Provides a search bar for entering a food name.
  - Verifies the input against the database.
  - Retrieves and displays food details, including images and nutritional values.

3.2 Non-Functional Requirements

- **Performance**: The system should process each image in less than 5 seconds on a GPU.
- **Usability**: The interface must be user-friendly, requiring minimal training to use.
- **Reliability**: The system must handle up to 100 concurrent users without downtime.
- **Scalability**: Supports addition of new food categories in future versions.

3.3 External Interface Requirements

1. **User Interfaces**:
   - Drag-and-drop functionality for image uploads.
   - Results are displayed in a clear and concise format.
2. **Hardware Interfaces**:
   - The system should run on devices with at least 8 GB RAM and a GPU (e.g., NVIDIA GTX 950M or higher).
3. **Software Interfaces**:
   - TensorFlow and Keras for model training and prediction.
   - Python libraries such as NumPy and Pandas for data preprocessing.

3.4 System Features

- **Seamless Image Upload and Classification**
  Provides an intuitive drag-and-drop interface for uploading food images and ensures accurate classification into one of 101 predefined categories.

- **Comprehensive Nutritional Analysis**
  Offers detailed nutritional information, including calorie count and macronutrient breakdown, for identified food items.

- **Advanced Search Functionality**
  Includes a powerful search feature enabling users to find nutritional data for specific food items effortlessly.

4. Other Non-Functional Requirements

4.1 Safety Requirements

- User-uploaded data should be automatically deleted after processing to maintain privacy.

4.2 Legal and Regulatory Requirements

- The system must comply with data privacy regulations like GDPR for user data protection.

4.3 Environmental Requirements

- Should operate on devices with standard configurations in both Windows and Linux environments.

5. Appendices

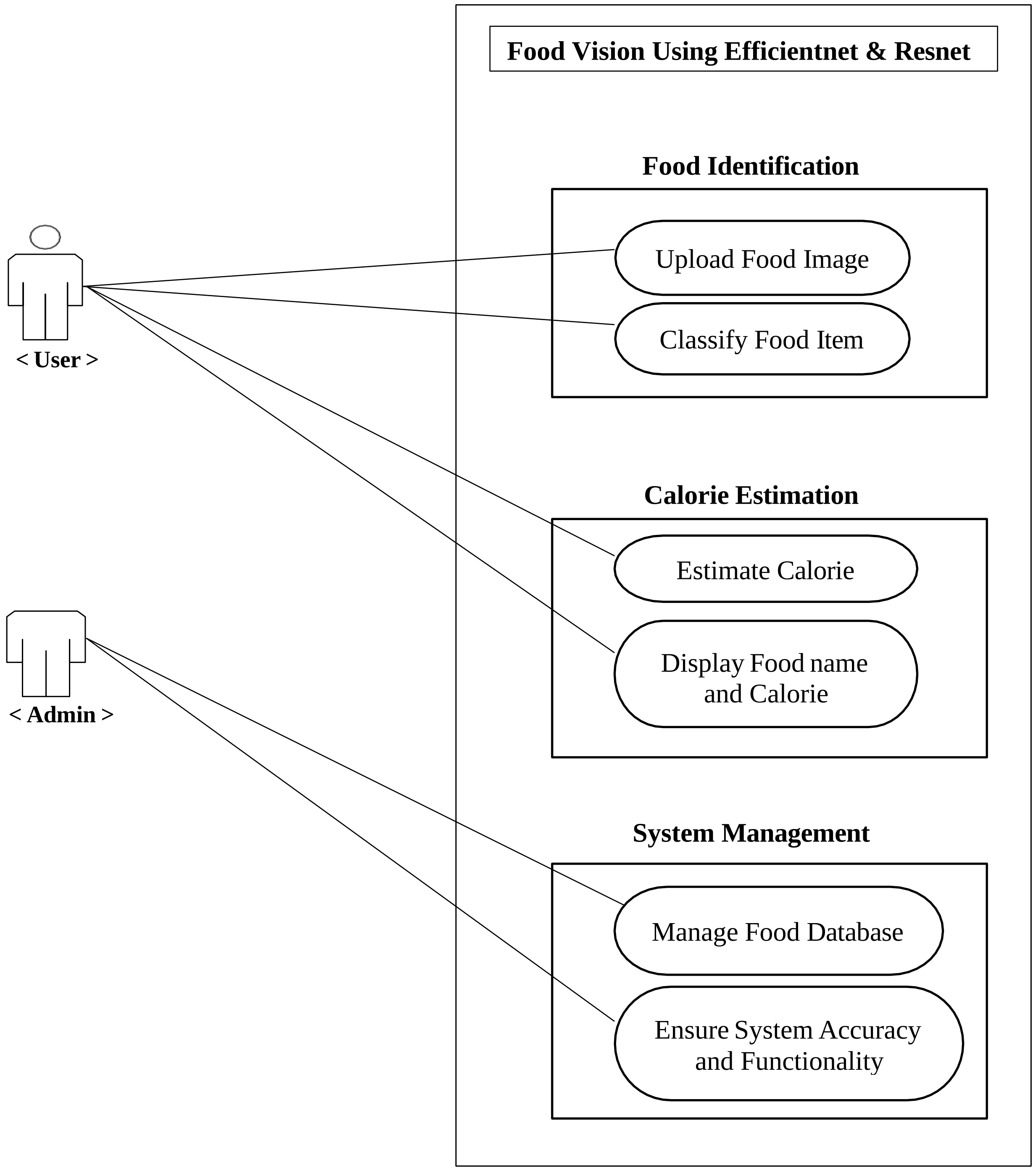
5.1 Glossary

Term/Acronym	Definition
CNN	Convolutional Neural Network.
Food-101	A dataset containing 101 categories of food images.
TensorFlow	A machine learning library used for building and deploying models.

5.2 Supporting Information

- **Dataset Information**: Food-101 Dataset, consisting of 101,000 images across 101 categories.
- **Model Information**: EfficientNet and ResNet are pre-trained models used for transfer learning

Use Case Diagram



Fully Dressed Use Case

Use Case 1: Upload Image

Use Case ID:

UC-01

Use Case Name:

Upload Food Image

Use Case Prepared By:

[Maria Ali]

Use Case Prepared On:

28th December 2024

Use Case Updated By:

[Ali Haider]

Use Case Updated On:

29th December 2024

Use Case Description:

This use case describes how a user uploads an image of food to the system for classification.

Primary Actor:

<User (Customer/Food Enthusiast)>

Stakeholders & Interests:

- **User**: Wants to upload an image and get the food item classified accurately.
- **Admin**: Ensures the image upload and classification work seamlessly.

Pre-Conditions:

1. The system must be running.
2. The user must have an image of a food item ready to upload.

Main Success Scenario:

User Action	System Response
1. User uploads a food image.	System prompts the user to confirm the upload.
2. User confirms the upload.	System processes the image using machine learning models.
3. System classifies the food item.	System displays the name of the classified food.

Post-Conditions:

- Food classification results are displayed to the user.
- The uploaded image is deleted after processing.

Extension Points:

1. **If the file type is unsupported**:
   - System prompts the user to upload a valid image file.
2. **If the image is unclear**:
   - System displays an error message asking for a clearer image.

Priority: High.

Frequency: High (frequently used functionality).

Cross-References: SSD-1, SRS-1.

Use Case 2: View Nutrition Information

Use Case ID:

UC-02

Use Case Name:

View Nutrition Information

Use Case Prepared By:

[Maria Ali]

Use Case Prepared On:

28th December 2024

Use Case Updated By:

[Ali Haider]

Use Case Updated On:

29th December 2024

Use Case Description:

This use case describes how a user views detailed nutritional information for a classified food item.

Primary Actor:

<User (Customer/Food Enthusiast)>

Stakeholders & Interests:

- **User**: Wants to view comprehensive nutritional details for a food item.
- **Admin**: Ensures accurate nutritional data is retrieved.

Pre-Conditions:

1. Food must be classified by the system.
2. Nutritional data must be available in the database.

Main Success Scenario:

User Action	System Response
1. User selects a classified food item.	System fetches the nutritional data from the database.
2. System retrieves the nutritional data.	System displays details such as calories, proteins, fats, and carbohydrates.

Post-Conditions:

- Nutritional data for the food item is displayed successfully.

Extension Points:

1. **If no nutritional data is available**:
   - System displays a message informing the user about the missing data.
2. **If the user selects the wrong item**:
   - System prompts the user to reselect the item.

Priority: Medium.

Frequency: Moderate (used depending on user interest).

Cross-References: SSD-2, SRS-2.

Use Case 3: Search Food

Use Case ID:

UC-03

Use Case Name:

Search Food

Use Case Prepared By:

[Maria Ali]

Use Case Prepared On:

28th December 2024

Use Case Updated By:

[Ali Haider]

Use Case Updated On:

29th December 2024

Use Case Description:

This use case describes how a user searches for a food item in the database to view its details.

Primary Actor:

<User (Customer/Food Enthusiast)>

Stakeholders & Interests:

- **User**: Wants to search for food items quickly and view their details.
- **Admin**: Ensures the search functionality is efficient and accurate.

Pre-Conditions:

1. The database must have the food item available.
2. The system must be online and running.

Main Success Scenario:

User Action	System Response
1. User enters a food name in the search bar.	System verifies the entered food name.
2. System retrieves matching results.	System displays food details, including images and nutritional data.

Post-Conditions:

- The search results, including nutritional details, are displayed successfully.

Extension Points:

1. **If no matching results are found**:
   - System displays a message indicating no results were found.
2. **If the search query is unclear or incomplete**:
   - System suggests alternative food names or asks the user to refine the search.

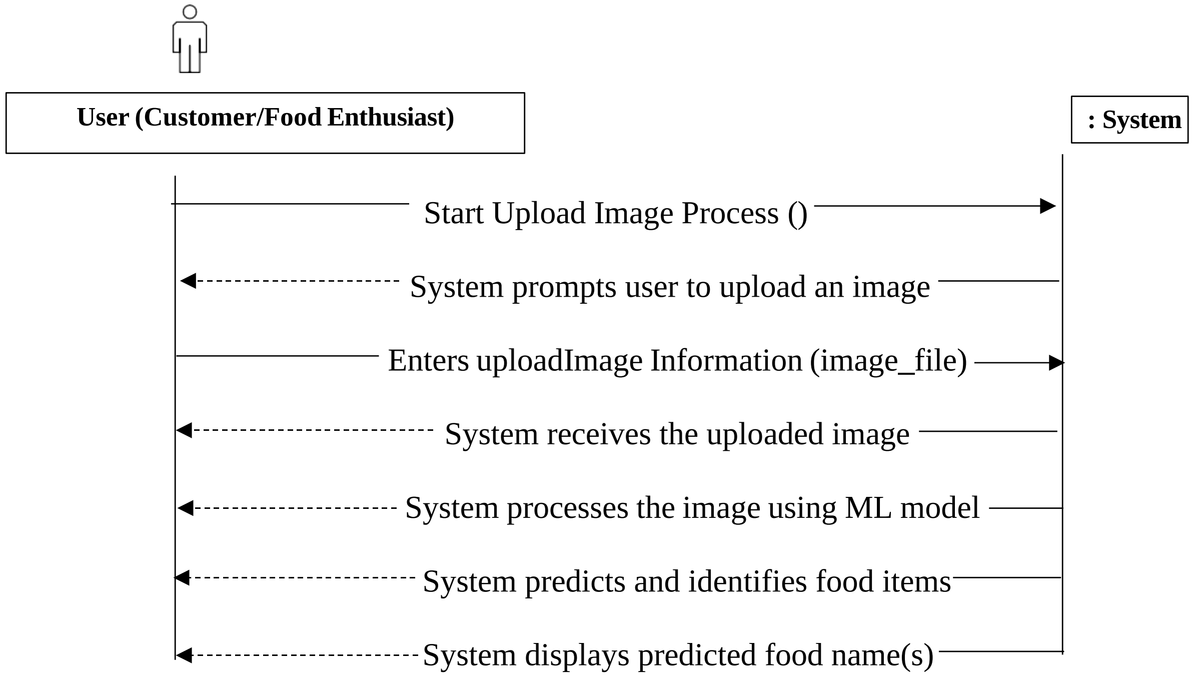
Priority: High.

Frequency: High (commonly used feature).

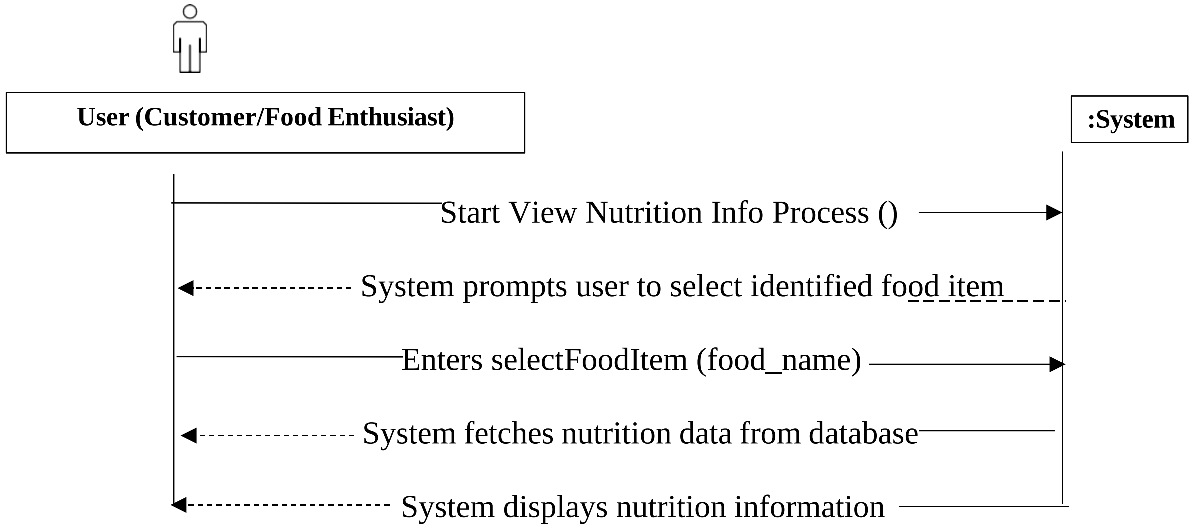
Cross-References: SSD-3, SRS-3.

System Sequence Diagram

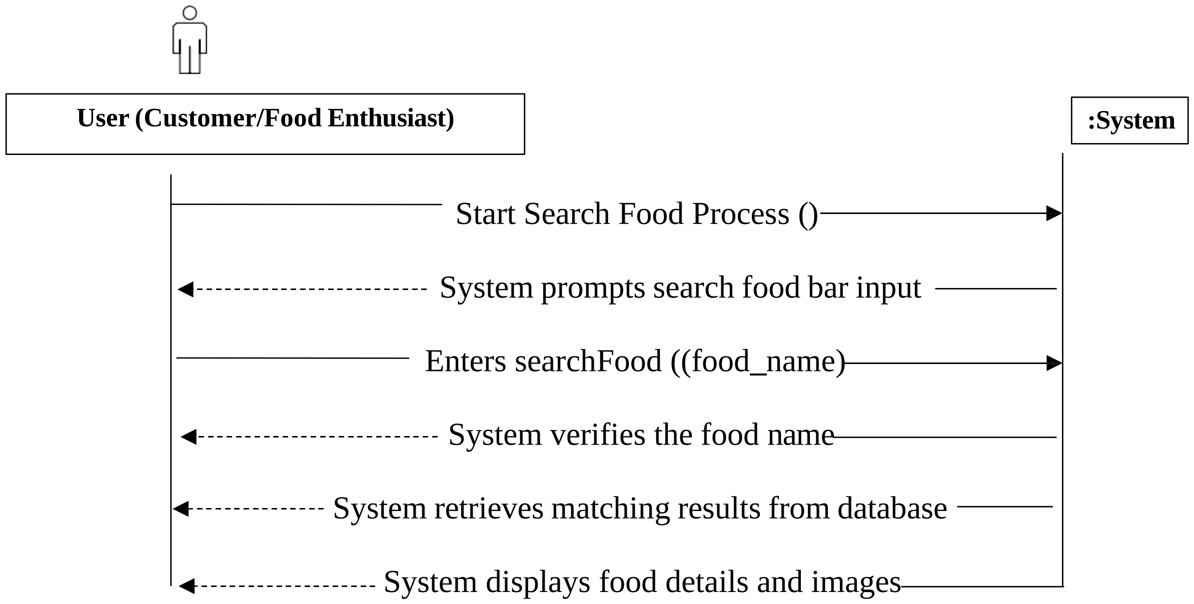
SSD 01: Upload Image


User (Customer/Food Enthusiast)
: System
Start Upload Image Process ()
System prompts user to upload an image
Enters uploadImage Information (image_file)
System receives the uploaded image
System processes the image using ML model
System predicts and identifies food items
System displays predicted food name(s)

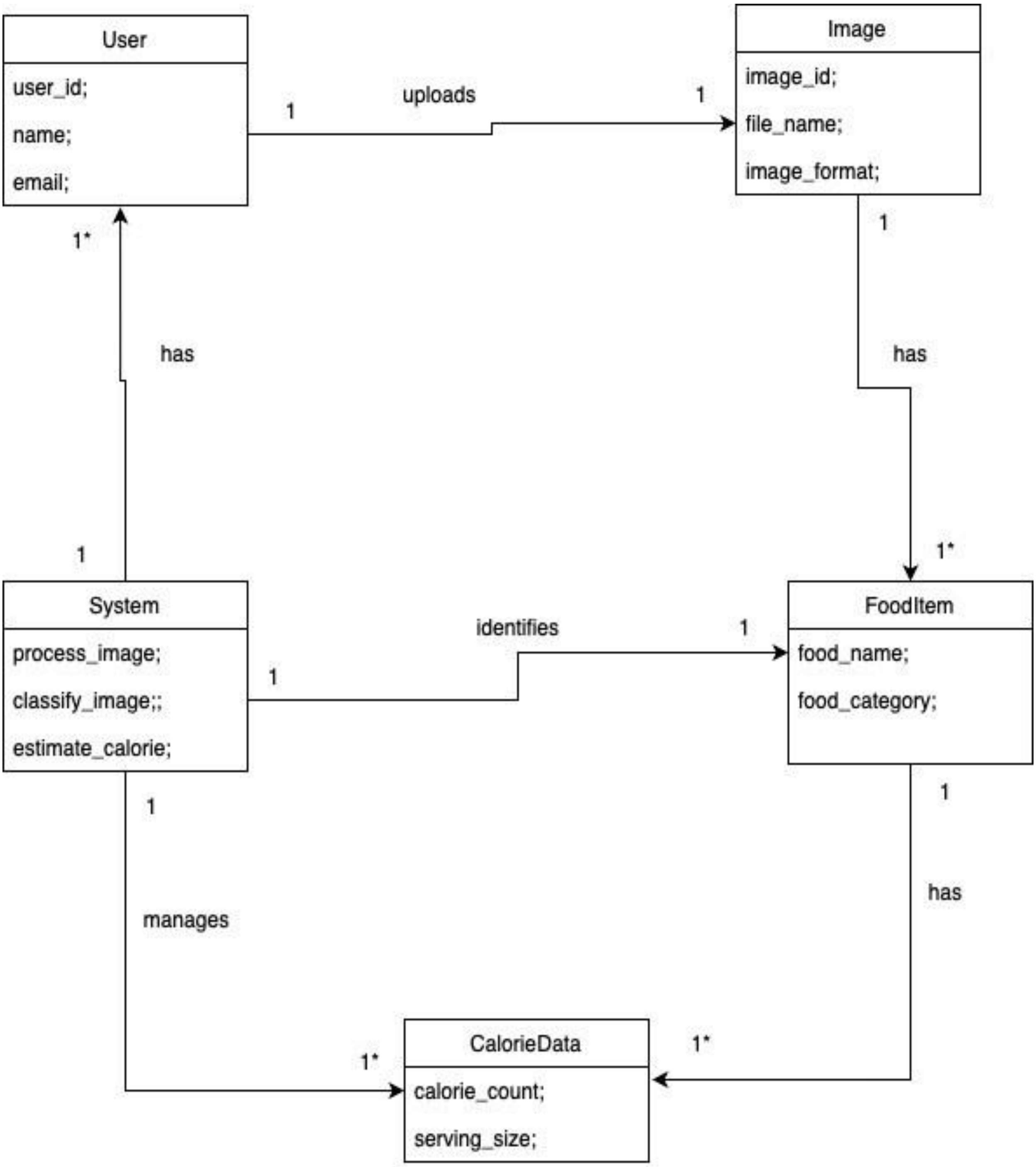
SSD 02: View Nutrition Information


User (Customer/Food Enthusiast)
:System
Start View Nutrition Info Process ()
System prompts user to select identified food item
Enters selectFoodItem (food_name)
System fetches nutrition data from database
System displays nutrition information

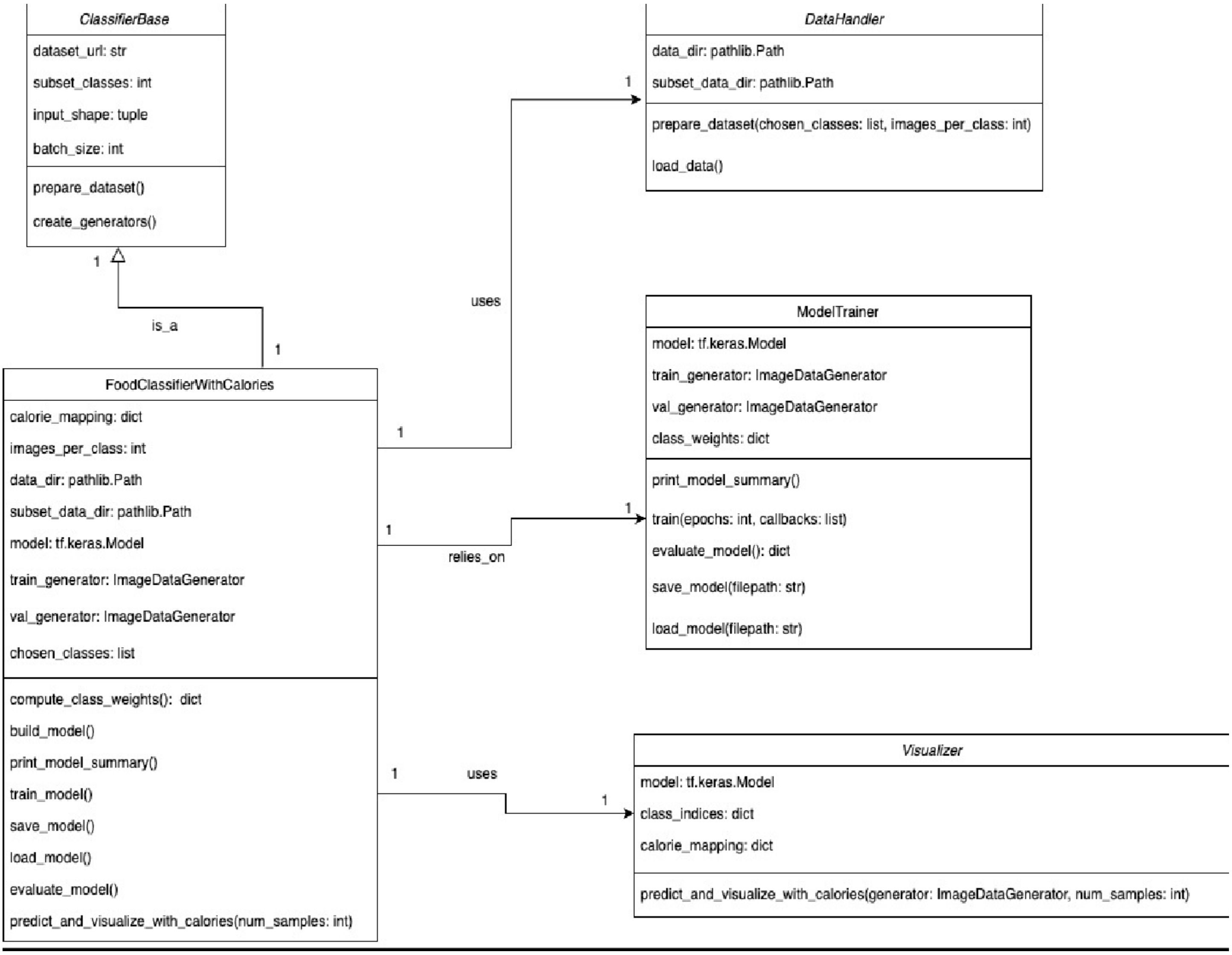
SSD 03: Search Food


User (Customer/Food Enthusiast)
:System
Start Search Food Process ()
System prompts search food bar input
Enters searchFood ((food_name)
System verifies the food name
System retrieves matching results from database
System displays food details and images

Domain Model



Class Diagram



Test Case

Test Case 1: Upload Food Image

Test Case ID: TC-01

Test Case Name: Upload Food Image

Test Case Prepared By: [Maria Ali]

Test Case Prepared On: 2nd January 2025

Test Case Updated By: [Ali Haider]

Test Case Updated On: 2nd January 2025

Test Case Description:
This test case verifies that the system allows users to upload food images for classification.

Primary Actor: <User>

Main Success Scenario:

User Action	System Response
1. User selects a food image for upload.	System prompts the user to confirm the upload.
2. User clicks the confirm upload button.	System processes the image and classifies the food item.
3. User views the classification result.	System displays the classified food item name.

Testing Requirement:

Testing Condition: System must be in a running state.

Input Data: Image file (valid format: PNG, JPG).

Expected Result: Food image is successfully uploaded and classified.

Actual Result: Food image uploaded and classification result displayed.

Priority: High

Frequency: Frequent

Test Acceptance: Passed

Test Case 2: View Nutrition Information

Test Case ID: TC-02

Test Case Name: View Nutrition Information

Test Case Prepared By: [Maria Ali]

Test Case Prepared On: 2nd January 2025

Test Case Updated By: [Ali Haider]

Test Case Updated On: 2nd January 2025

Test Case Description:
This test case verifies that the system displays detailed nutritional information for a classified food item.

Primary Actor: <User>

Main Success Scenario:

User Action	System Response
1. User selects a classified food item.	System retrieves the nutritional data for the selected item.
2. User views the nutritional details.	System displays calorie and nutrient information.

Testing Requirement:

Testing Condition: System must have a classified food item with nutritional data available.

Input Data: Classified food item.

Expected Result: Nutritional information is displayed correctly.

Actual Result: Nutritional information retrieved and displayed.

Priority: High

Frequency: Moderate

Test Acceptance: Passed

Test Case 3: Search Food

Test Case ID: TC-03

Test Case Name: Search Food

Test Case Prepared By: [Maria Ali]

Test Case Prepared On: 2nd January 2025

Test Case Updated By: [Ali Haider]

Test Case Updated On: 2nd January 2025

Test Case Description:
This test case verifies that the system allows users to search for food items and view relevant details.

Primary Actor: <User>

Main Success Scenario:

User Action	System Response
1. User enters a food item name in the search bar.	System validates the search input.
2. User clicks the search button.	System retrieves matching food items from the database.
3. User views the search results.	System displays the food items along with their details.

Testing Requirement:

Testing Condition: The database must contain food items matching the search query.

Input Data: Food item name (text input).

Expected Result: Search results with food details displayed.

Actual Result: Search results retrieved and shown successfully.

Priority: Medium

Frequency: Moderate

Test Acceptance: Passed