# Functional Specification

‘VeggieCheck’

| | |
|---|---|
| Student 1 | Róisín O’Rourke - 19360491 |
| Student 2 | Amelia Grigoriev - 19348241 |
| Supervisor | Paul Clarke |
| Submission Date | 10/12/2021 |

# Table of contents

# 1. Introduction

## 1.1 Overview

There are approximately 300,000 vegans or vegetarians in Ireland, with Ireland ranking third in the world for countries with the most vegans per capita. There is a definite market and need for an app that will allow shoppers to find products that cater to their dietary needs.

The product being developed is an iOS application called 'VeggieCheck' that will allow users, while shopping, to check if a food product is vegan or vegetarian friendly. The application will use the smartphone camera and text recognition to capture the ingredients of the product and then utilise a database of food ingredients to check if the item is suitable for consumption.

## 1.2 Business Context

There is no business context and the product will not be sponsored by any business organisations.

## 1.3 Glossary

| **Term** | **Definition** |
|---|---|
| Vegan | A vegan diet excludes all meat and animal derived products (meat, poultry, fish, seafood, dairy and eggs) |
| Vegetarian | A vegetarian diet excludes all meat products (meat, poultry, seafood, gelatine) |
| GANTT chart | A type of bar chart that illustrates a project schedule |
| Vision Framework | The Vision framework performs face and face landmark detection, text detection, barcode recognition, image registration, and general feature tracking |
| Swift | A programming language for iOS |
| Xcode | A development environment for macOS, used to develop software for iOS |
| Proto.io | An online prototyping tool |

*TABLE ONE: Glossary*

# 2. General Description

## 2.1 Product / System Functions

The product will be an iOS application. The product will use the phone camera to capture an image of the ingredients listed on a given food product. Text recognition will then be used to process the ingredients and they will then be compared to a food database to check if the product is vegan or vegetarian. The system will then notify the user of the status of the food check on the screen.

## 2.2 User Characteristics and Objectives

The users of our applications will mainly be vegans or vegetarians who wish to check if the food products that they are purchasing are compatible with their diet. Since we are developing a smartphone application, the user will have to be familiar with downloading and using applications and using their phone camera and also comfortable with using a touchscreen.

## 2.3 Operational Scenarios

### 2.3.1 User login

| | |
|---|---|
| **Use Case Name** | User Login |
| **Trigger** | The user wishes to log into the application. |
| **Precondition** | The user has created an account on the mobile application. |
| **Basic Path** | ● The user enters their username.<br>● The user enters their password.<br>● The system verifies the account details. |
| **Alternative Path** | The system fails to verify the account details, and the user is not granted access to the application. |
| **Postcondition** | ● The user is logged into the application. |

*TABLE TWO: Use Case 1: User Login*

## 2.3.2 Scan ingredients

| | |
|---|---|
| **Use Case Name** | Scan Ingredients |
| **Trigger** | The user wishes to check if a product is vegan or vegetarian. |
| **Precondition** | The user has a working camera on their phone. |
| **Basic Path** | ● The user holds the camera over the ingredients of the product.<br>● The user takes a picture of the ingredients.<br>● The system processes the ingredients and checks if they are suitable.<br>● The system displays the results. |
| **Alternative Path** | The system fails to process the list of ingredients and asks the user to retake the picture. |
| **Postcondition** | ● The user knows if the product is vegan or vegetarian or not. |

*TABLE THREE: Use Case 2: Scan Ingredients*

## 2.3.3 Check ingredients individually

| | |
|---|---|
| **Trigger** | The user wishes to check if a certain ingredient is vegan or vegetarian. |
| **Precondition** | The user knows the correct spelling of the ingredient they wish to check. |
| **Basic Path** | ● The user selects the check ingredient option.<br>● The user types in the ingredient they wish to check.<br>● The system compares the ingredient to the food database.<br>● The system displays the results. |
| **Alternative Path** | The system cannot find the ingredient in the database and cannot return a result. |
| **Postcondition** | ● The user knows if an ingredient is vegan or vegetarian or not. |

*TABLE FOUR: Use Case 3: Check Ingredients Individually*

## 2.4 Constraints

### Time constraints

The application must be completed by the 4th of March 2022, as this is the due date of the assignment. Time will also need to be set aside to familiarise ourselves with the technologies being used in the project.

### Hardware constraints

The smartphone that the user is using will have to have a fully functional camera in order to capture the ingredients in the item. We are developing an iOS application, so the user will have to have an iPhone.

### Speed constraints

In order for the application to be effective, it must be able to return results quickly about the status of a certain product so as not to hold the user up while they are shopping.

### Inexperience constraints

Most of the technologies being used in this project, such as Vision Framework and Xcode, are unfamiliar to the developers.

# 3. Functional Requirements

## 3.1 Image Capture

**Description -** The system must be able to capture a clear picture of the ingredients listed on the food product in order for text recognition to accurately recognise the individual ingredients.

**Criticality -** The ability to take an accurate image of the ingredients is highly critical to the functionality of the system as without it, the ingredients cannot be checked.

**Technical Issues -** The user's phone must have a camera that takes images that are of high enough resolution for text recognition to process the ingredients. The image taken must be able to capture all the ingredients listed on the product.

**Dependencies -** This functional requirement is not reliant on any other functional requirements.

## 3.2 Text Recognition

**Description -** The system must be capable of taking an image taken of the food product and using text recognition, process the list of ingredients.

**Criticality -** This functional requirement is also highly critical. The ingredients must be recognised and processed in order for the ingredient checker to be implemented.

**Technical Issues -** An issue that could arise could be that the system cannot process the list of the ingredients if the image is blurry or unclear. The text recognition must also be capable of reading very small print and print that is not written on a flat surface.

**Dependencies -** This functional requirement relies on the previous functional requirement, 3.1. It is dependent on a clear image being taken that text recognition can be applied to.

## 3.3 Ingredient Checker

**Description -** The system must be able to take the list of ingredients that have been processed and compare them with a food database to determine if the product is suitable for vegans or vegetarians.

**Criticality -** This functional requirement is the most critical to the overall system. If the system is capable of processing the ingredients of a product, but is incapable of checking if they are vegetarian or vegan friendly, the application is redundant.

**Technical Issues -** A technical issue that might occur is if any of the ingredients listed on product cannot be found in the food database and therefore cannot be checked if it is suitable.

**Dependencies -** This functional requirement is dependent on the functional requirements 3.1 and 3.2. It relies on a clear image being taken and the text recognition processing the ingredients in order to check them.

## 3.4 Display Results

**Description -** Once the system has determined if the food product is suitable for the user, the system must be able to display the results on the screen of the application.

**Criticality -** This requirement is imperative to the functionality of the application. The user needs to be informed about the status of the food.

**Technical Issues -** If the system incorrectly processes an ingredient or the database fails to verify if it is vegan/vegetarian or not, the incorrect results may be displayed.

**Dependencies -** This functional requirement is dependent on the functional requirement, 3.3. It must return the result that has been found through the ingredient checker functionality.

## 3.5 User Login

**Description -** The user must create an account on the application. Once the user has created an account, they must log in and then can use the system.

**Criticality -** This requirement is not critical to the functionality of the system. If it cannot be implemented, the rest of the system will still work accordingly.

**Technical Issues -** An issue could arise if the system cannot verify the username or password of the user.

**Dependencies -** This functional requirement is not reliant on any other functional requirements.

## 3.6 Individual Ingredient Checker

**Description -** The user will be able to check individual ingredients to see if they are vegan or vegetarian. The system must be capable of taking the inputted ingredient and checking it with the database to see if it is suitable.

**Criticality -** This requirement is of medium criticality to the system. The main functionality of the application will revolve around scanning the products, so the system will still be operational if we cannot implement this requirement.

**Technical Issues -** If the spelling of the ingredient is incorrect, the system will not be able to check if it is suitable for the user.

**Dependencies -** This functional requirement is not reliant on any other functional requirements.

# 4. System Architecture

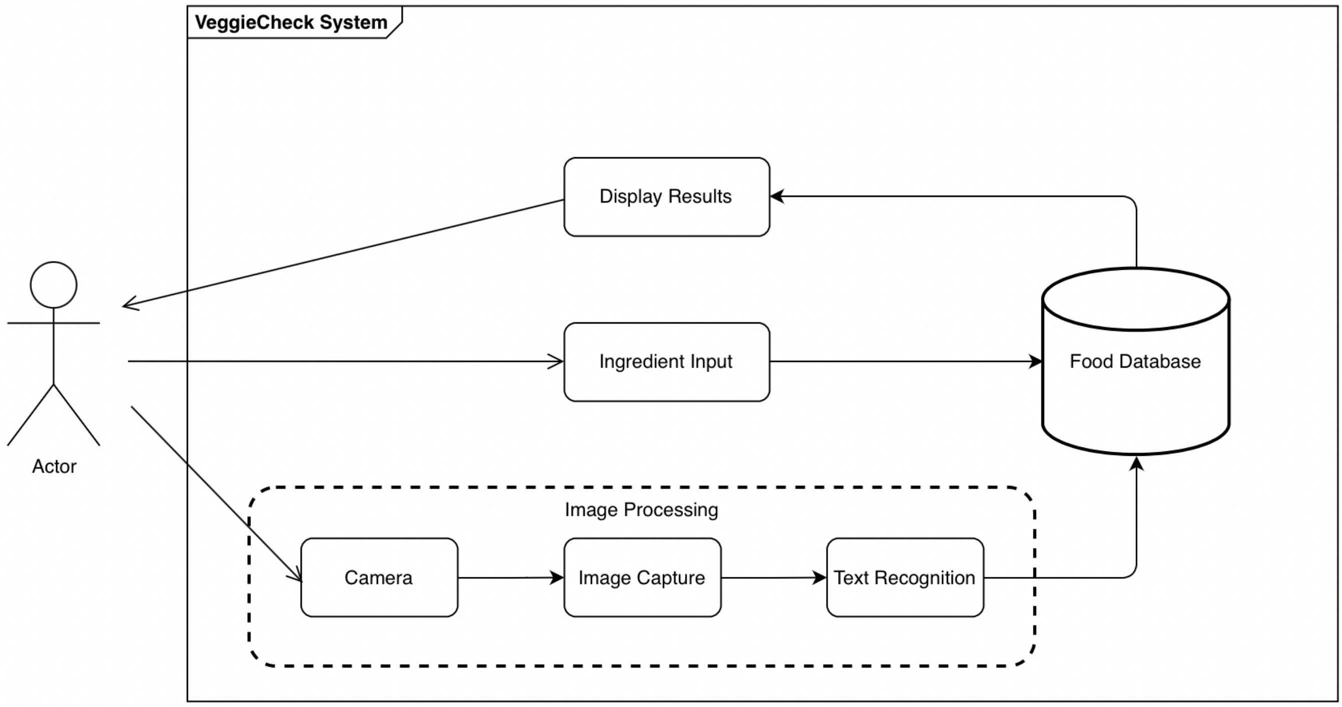


*FIGURE 1: System Architecture Diagram*

# 5. High-Level Design

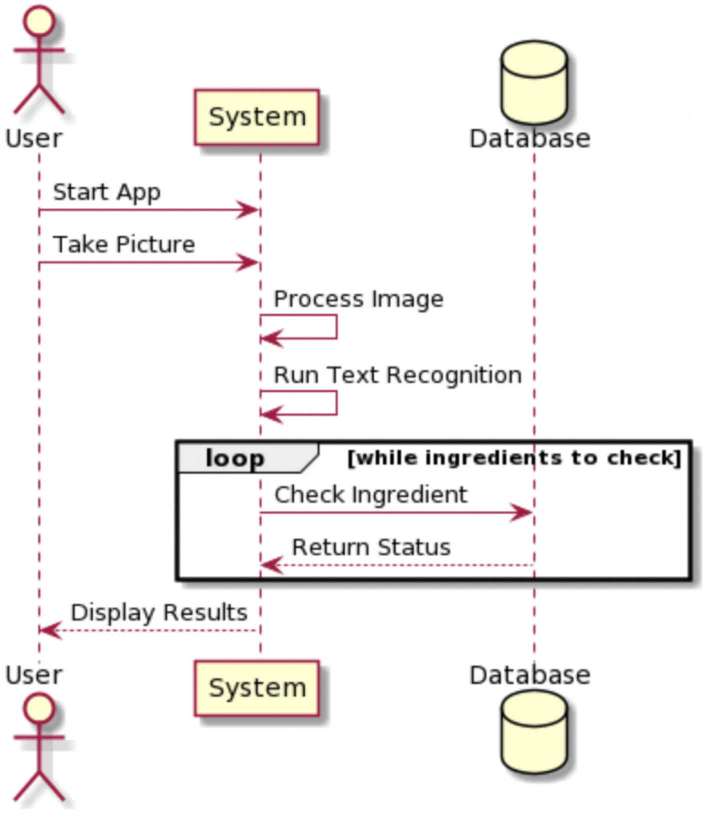


*FIGURE 2: Sequence Diagram 1*

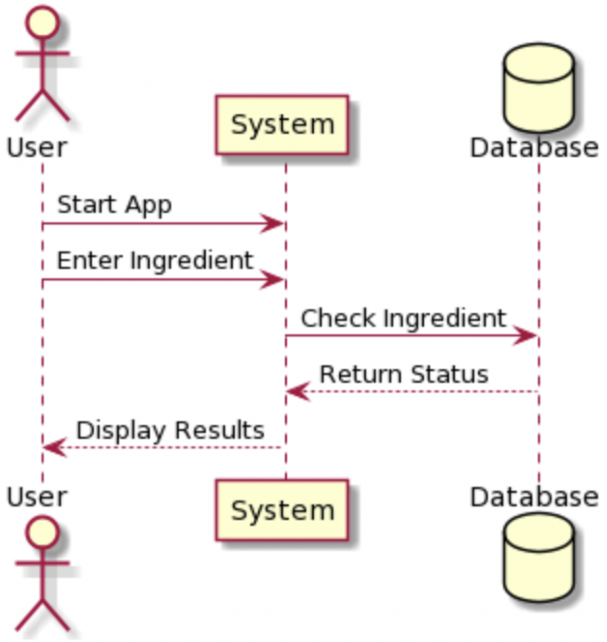
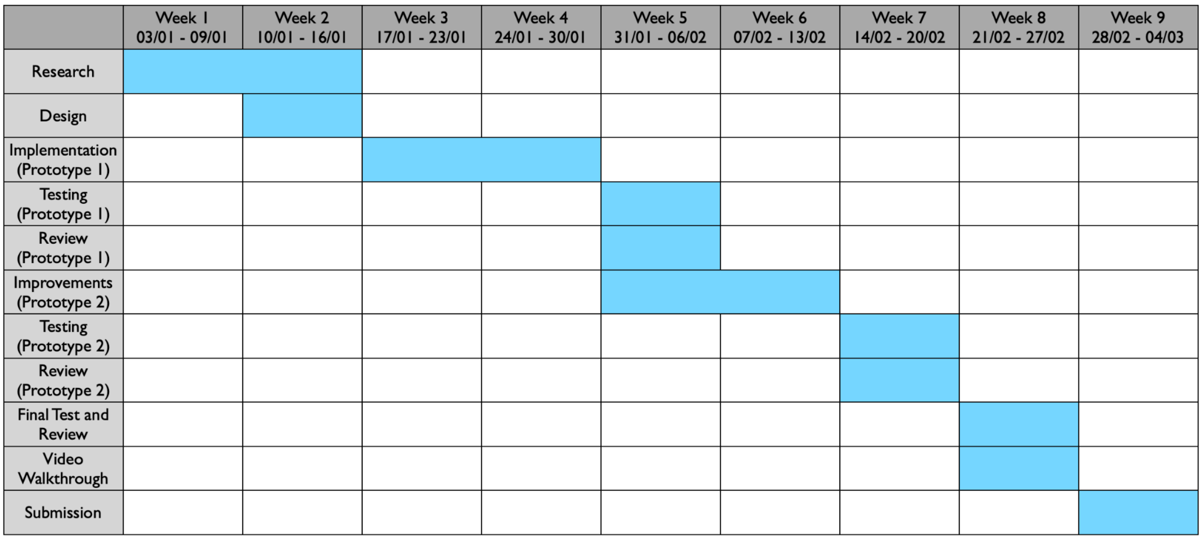


*FIGURE 3: Sequence Diagram 2*

# 6. Preliminary Schedule



| | Week 1<br>03/01 - 09/01 | Week 2<br>10/01 - 16/01 | Week 3<br>17/01 - 23/01 | Week 4<br>24/01 - 30/01 | Week 5<br>31/01 - 06/02 | Week 6<br>07/02 - 13/02 | Week 7<br>14/02 - 20/02 | Week 8<br>21/02 - 27/02 | Week 9<br>28/02 - 04/03 |
|---|---|---|---|---|---|---|---|---|---|
| Research | 🟦 | 🟦 | | | | | | | |
| Design | | 🟦 | | | | | | | |
| Implementation (Prototype 1) | | | 🟦 | 🟦 | | | | | |
| Testing (Prototype 1) | | | | | 🟦 | | | | |
| Review (Prototype 1) | | | | | 🟦 | | | | |
| Improvements (Prototype 2) | | | | | 🟦 | 🟦 | | | |
| Testing (Prototype 2) | | | | | | | 🟦 | | |
| Review (Prototype 2) | | | | | | | 🟦 | | |
| Final Test and Review | | | | | | | | 🟦 | |
| Video Walkthrough | | | | | | | | 🟦 | |
| Submission | | | | | | | | | 🟦 |

*FIGURE 4: GANTT Chart*

During the research phase we will be researching different food databases, finding one which suits our needs the best. We will be researching and learning how to use technologies that we will be using in our project, such as Swift, the Vision framework and XCode. We will also be researching other technologies that might be useful and that we may want to use during our implementation of the project. During the design phase we will be designing the user interface for our application. To do this we will be using proto.io.