

**INDY-9 RED — MyFoodScan App**  
**Software Requirements Specification**  
**CS 4850, Section 01/02, Spring 2024**  
**Feb 13, 2024**

Ibrahima



Jedae



Bri



Victoria



# Table of Contents

1.0	Introduction .....	4
1.1	Overview .....	4
1.2	Project Goals .....	4
1.3	Definitions and Acronyms .....	4
1.4	Assumptions .....	5
2.0	Design Constraints.....	5
2.1	Environment.....	5
2.2	User Characteristics.....	5
2.3	System .....	5
3.0	Functional Requirements .....	5
<b>3.1</b>	<b>Phase 1</b> .....	<b>6</b>
3.1.1	Login and Register .....	6
3.1.1	Questionnaire.....	6
3.1.2	Display Home Page with Scan .....	6
3.1.3	Barcode Scanning .....	6
3.1.4	Navigate from Home Page to Data Pop-up After Scanning.....	6
3.1.5	Navigation Bar .....	6
3.1.6	Navigate from Home Page to User Profile Page.....	6
3.1.7	Navigate from Home Page to Previous History Page .....	7
3.1.8	Machine Learning for Recommendation of Similar Items.....	7
<b>3.2</b>	<b>Phase 2</b> .....	<b>7</b>
3.2.1	Machine Learning for Compliance Checking .....	7
3.2.2	Additional Menu Options .....	7
3.2.2	Notification System .....	7
4.0	Non-Functional Requirements .....	7
4.1	Performance .....	7
4.2	Security.....	7
4.3	Usability.....	8
4.4	Reliability .....	8
4.5	Scalability.....	8
5.0	External Interface Requirements.....	8
5.1	User Interface Requirements .....	8

5.1.1 Sign-Up/Login Interface..... 8

5.1.2. Questionnaire Interface ..... 8

5.1.3. Main Interface ..... 9

5.2 Hardware Interface Requirements..... 9

5.3 Software Interface Requirements ..... 9

5.4 Communication Interface Requirements ..... 9

# 1.0 Introduction

## 1.1 Overview

In response to the growing need for personalized dietary management, this project proposes the development of an advanced food/snack scanning application. The application will utilize cutting-edge image recognition and AI technology to analyze food products and identify their compliance with various dietary restrictions such as kosher, Halal, vegan, vegetarian, and any allergen-specific needs.

Additionally, the application will feature user profiles where individuals can specify their dietary restrictions, allowing for a customized experience. To enhance engagement, the app will also use a recommendation algorithm to provide users with similar products that follow their dietary needs. The process will involve scanning the barcodes of the user's purchased food items, which will serve queries to the OpenFoodFacts API, providing the user with all the necessary information. Moreover, the application will conveniently store the user's previous scans to provide a concise user archive and prevent redundancy.

For the front-end development of the app, React Native is chosen for its comprehensive and cross-platform capabilities. Additionally, Canva and Figma are going to be utilized for UX/UI prototyping, ensuring a user-friendly design. In terms of backend development, Google Firebase will be used, particularly for its smooth compatibility with React Native. Lastly, the implementation of Machine Learning algorithms is crucial for the image recognition and analysis component of the application, ensuring accurate product scanning and analysis.

## 1.2 Project Goals

The application will feature user profiles where individuals can specify their dietary restrictions, allowing for a customized experience. The goal of the application is to provide users with an opportunity to scan their purchased products to see if they align with their dietary needs. Moreover, the application encourages users to find more products that comply with their dietary restrictions. To enhance engagement, the app will also provide educational content about different dietary practices and health tips. By offering a user-friendly and informative platform, this project aims to empower individuals with dietary restrictions, making their shopping experiences safer, more convenient, and aligning with their health and beliefs.

## 1.3 Definitions and Acronyms

The following definitions and acronyms are provided in this SRS for proper document interpretation.

<i>Definitions</i>
<b>Kosher:</b> Refers to Jewish dietary restrictions. These laws dictate what foods are allowed to be consumed, how the food should be prepared, and which foods to avoid
<b>Vegan:</b> A person who does not consume any food sourced from animals or use products derived from animals

<b>Halal:</b> Refers to Islamic dietary restrictions. These laws dictate how food is prepared, what foods are allowed to be consumed, and what practices to avoid
<b>Vegetarian:</b> A person who does not eat meat, but may consume other products derived from animals
<i>Acronyms</i>
<b>API:</b> Application Programming Interface - allows two applications to communicate with each other
<b>UX:</b> User Experience
<b>UI:</b> User Interface

## [1.4 Assumptions](#)

The assumptions may include technical and user assumptions. Technical assumptions can include the availability of particular hardware components, compatibility with specific databases and operating systems, and the existence of required network connectivity. It is assumed that the user has previous experience with similar technology and familiarity with functions of similar mobile applications. User assumptions also include that the user has access to an iOS or Android device.

## [2.0 Design Constraints](#)

### [2.1 Environment](#)

There are three environments that should be considered: physical, operational, and deployment. Regarding the physical environment, the software should comply with the latest version of iOS and android. In terms of the operational environment, the system should operate best in a high-speed internet environment with the lowest requirement being 256 kbps to run. For the deployment environment, the application needs to be deployable on Google Firebase as a cloud infrastructure to be utilized for database management.

### [2.2 User Characteristics](#)

The application is intended for users approximately 18 years of age and older, specifically with a demand to discover if their purchased products comply with a specific dietary need. Users can be categorized into various groups, such as users who personally have dietary restrictions, users who know someone with dietary restrictions, users who are curious about specific dietary restrictions, etc.

### [2.3 System](#)

The software must integrate with the OpenFoodFacts API for product consumption services. The data from OpenFoodFacts will be used in the Google Firebase database along with additional consumer data. The implementation of a RESTful API is necessary to extract the data from the Google Firebase database.

## [3.0 Functional Requirements](#)

## 3.1 Phase 1

### 3.1.1 Login and Register

- After opening the app, the user is prompted to register for security concerns. It will be required to sign up using their email and password or through OAuth from different providers such as Google, Facebook, Apple, etc.

### 3.1.1 Questionnaire

- After the login/register page, the user will be prompted to complete a questionnaire about which dietary restrictions apply to them. This information will be automatically updated to the user's profile, which they will be able to modify later within the user profile page.

### 3.1.2 Display Home Page with Scan

- When signed into the app, the user is presented with the home page. The primary section of the home page will be the barcode scanner that can be utilized by the user for scanning a food item. This scanner will be seen in the middle of the page and will take up most of the space. The navigation bar at the bottom of the screen will contain a history button, a scan button, and a user profile button. Since the scanner is the home page and the home page is the first page displayed, the user will be on the scan option of the navigation bar.

### 3.1.3 Barcode Scanning

- Access to the user's camera will be required. A pop-up will show up when the app request access to the users' camera. Users should be able to use the camera function to scan barcodes of different products to identify additional information.

### 3.1.4 Navigate from Home Page to Data Pop-up After Scanning

- After scanning the item, the user will see a pop-up that displays the information regarding the product. This page will display the scanned product's title, nutrition information, a message about if the product complies with the user's dietary restrictions based on their profile, the product image from the database, and recommendations for similar products that comply with the user's restrictions in cases where the current product does not comply. Furthermore, the recommended similar products will be displayed at the bottom of the pop-up.

### 3.1.5 Navigation Bar

- The navigation bar previously mentioned will be displayed at the bottom of the application. The home page will automatically begin on the scan page since this page is the home page. The other two options on the navigation are the history button and the user profile. The history button will be on the bottom left, and the user profile will be on the bottom right.

### 3.1.6 Navigate from Home Page to User Profile Page

- When users select the button for the user profile on the navigation bar, they will be prompted to a page that allows users to view and edit their information. Users will have a profile page with their personalized preferences and changes can be made anytime. This page will include their name and other personal information and a restrictions section. This section will have buttons with various dietary restrictions that are highlighted according to which dietary restrictions apply to the user from the answers of the questionnaire. On this page, users will be able to see

these dietary restrictions highlighted and can modify them, if needed. This page will also display a section for access to camera, where users modify this access.

#### 3.1.7 Navigate from Home Page to Previous History Page

- When users select the button on the navigation bar for the previous history, they will be prompted to a page that users can utilize to view the past previous items. This page will display the last ten previously scanned items for user convenience.

#### 3.1.8 Machine Learning for Recommendation of Similar Items

- Machine learning algorithms will be used to provide users with recommendations to similar products in cases where the current product does not comply with the selected dietary restrictions of the user.

### 3.2 Phase 2

#### 3.2.1 Machine Learning for Compliance Checking

- Machine learning will be utilized for ingredient analysis and compliance checking to classify scans more efficiently.

#### 3.2.2 Additional Menu Options

- There will be a favorites section where users can save scans of different products and an education section providing tips and information on different dietary needs. The favorites section will allow further convenience for users to have their most common or liked items in one easy-to-access selection. Moreover, the education section will advance one of the goals of the application, which is to educate users on various dietary restrictions.

#### 3.2.2 Notification System

- Notifications will be sent to the user periodically to encourage/remind them to return to the application for further scanning of products and to provide additional educational fun facts to increase user involvement.

## 4.0 Non-Functional Requirements

### 4.1 Performance

The system should be able to run within 5 seconds after opening the app. If multiple users are using the app concurrently, the system should still be able to run efficiently without slowing down or crashing. Processes such as loading and retrieving data should occur as quickly as possible.

### 4.2 Security

The app will have a secure authentication system that requires users to make a username and password to create their profile. Each user that creates a profile is limited to view the information that they put into the app. The users are not allowed to access or edit other profiles. In addition, the app does not share any data with other users.

### 4.3 Usability

The application will be user-friendly as it will display an intuitive interface that appeals to our target audience. It will be fully responsive for both iOS and android devices and should deliver a smooth user experience where the user encounters little to no delays or errors.

### 4.4 Reliability

The system should be constantly available as long as the user has network connectivity. The user should be able to consistently open the app without facing any errors. When prompted to create a user profile, the app should save the information to be available to view and edit. Each option presented to the user should be constantly available and running properly when selected. When the user allows the app to access the camera, the barcode scanner should be functioning properly and reading the barcodes on selected food items.

### 4.5 Scalability

As more users join the app and make their own profiles, the system should be able to handle the increasing amount of data without decreasing the overall performance. The system should allow updates to include new features without causing any problems to the users or reducing the app's efficiency.

## 5.0 External Interface Requirements

### 5.1 User Interface Requirements

#### 5.1.1 Sign-Up/Login Interface

This is the first interface that the user will see when user uses MyFoodScan for the first time.

##### *5.1.1.1 Login*

In this screen, there will be the name of the application, the option to either login with email and password or login by providers like Google, Facebook, Apple, etc.

##### *5.1.1.2 Register*

In this screen, MyFoodScan will enable users to register. After entering the necessary information, the user is registered to use the application.

#### 5.1.2. Questionnaire Interface

##### *5.1.1.1 Choosing Dietary Restrictions*

In this screen, MyFoodScan will allow the user to choose from many dietary restrictions provided by the application that mostly correlates to the users.



### 5.1.3. Main Interface

The interface will contain three screens and one pop-up overlay. All screens/overlay will have a consistent layout.

#### 5.1.3.1 Scan Barcode

In this screen, the application shows a scanner overlay, providing visual cues within the product barcode that should be placed for scanning.

#### 5.1.3.2 Product/Compliance Details & Recommendations

After scanning a product, the user will have a screen pop-up showing the product image, name, and simplified version of its nutrition facts. Underneath, the screen contains a list of dietary restrictions that comply with the user's preferences. In terms of the messages about if the product complies with the user's dietary restrictions, the user's dietary restrictions that do not comply, if any, will be prioritized to the top of the list of dietary restrictions listed, and they will be listed with a "x" next to them. Next, the user's dietary restrictions that do comply, if any, will be followed, but the font will be greyed out since the information has less of a priority compared to the user finding out that the product does *not* comply with a certain restriction. These will be listed with a check next to them. Below that will be a section where similar product recommendations that'll be provided to the users.

#### 5.1.3.3 View History

The screen will present a list of the last ten previously scanned items. This screen will display these products and their images in a two-columns list.

#### 5.1.3.4 Show/Edit Profile

In this screen, a user can see their name, information, profile photo, and settings to change information or profile photo. Additionally, users will be able to change their dietary restrictions and access the camera.

## 5.2 Hardware Interface Requirements

There is no external hardware required other than a mobile device.

## 5.3 Software Interface Requirements

For database services, application shall use Google Firebase latest version. Application will run Android version above or equal to Red Velvet Cake (version 11) or iOS version 8 or higher. Application shall use React Native library '*react-native-camera*' to utilize the device's camera to scan and interpret barcodes.

## 5.4 Communication Interface Requirements

Users fill up the necessary forms to register or login into MyFoodScan application. These details i.e. email, and password are encrypted before they are passed on to the database. Email communication is required if users forget their password. HTTP protocol is pertinent with mobile devices in communicating with Google Firebase.