Assignment 2
Building A Barcode-enabled Android App for Displaying Product Information

Objectives

In a department store, a barcode label affixed to a product's packaging or container serves various purposes for users, such as inventory checking, payment, and product information inquiry. To support these functionalities, product information associated with a specific barcode must be created and managed using an information system that incorporates a database. Subsequently, users can conveniently utilize a computer application, such as a web or mobile app, to scan the barcode label and access the relevant product information.

This assignment aims to provide students with a hands-on experience in designing and building a mobile app for displaying food product information by scanning the barcode label on the food packaging.

Upon completion of this individual assignment, students are expected to develop the following skills:

1. Designing and building an Android app using the Microsoft .NET MAUI.
2. Scanning the barcode label attached to the food packaging.
3. Retrieving product information from a public Web API within the Android app. The Web API "https://world.openfoodfacts.org/api/v0/product/<barcode>.json" will be utilized.
4. Generating an APK file for the Android app.

Major Tasks

Each student is required to design and develop an Android mobile app with the following features:

1. A user interface (UI) that allows users to enter a barcode using an on-screen keyboard.
2. A UI that enables users to scan food packaging's barcodes, such as 1D EAN barcode and 2D QR CODE, and then the app can retrieve related food product information by making a request to the Web API "https://world.openfoodfacts.org/api/v0/product/<barcode>.json".
3. The UI should display common product information, such as the brand name, product name, product images, ingredients, and more.

Additionally, students are expected to prepare a comprehensive user manual with clear instructions for guiding users how to use the mobile app.

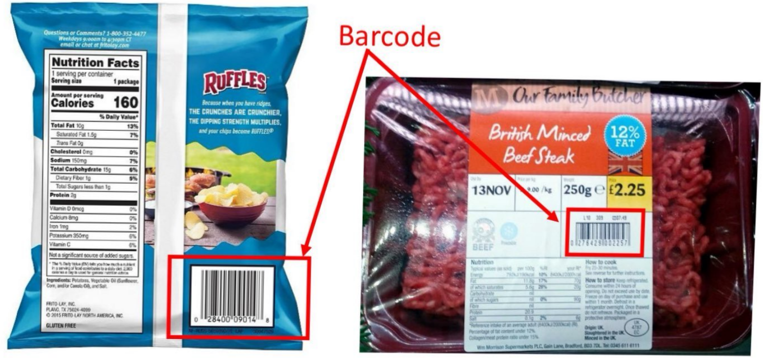
Items for Submission

Each student is required to submit the following:

1. A mobile app APK file and its corresponding source code.
2. A comprehensive user manual with detailed instructions guides users how to operate the mobile app.
3. A short video demonstrating the step-by-step process of operating the mobile app.

Information about barcode labels and web API

Barcode labels attached to products offer valuable assistance to users across various applications, including inventory management, supply chain tracking, product identification, and point-of-sale automation. The provided figures showcase examples of barcode labels affixed to the packaging of food products.



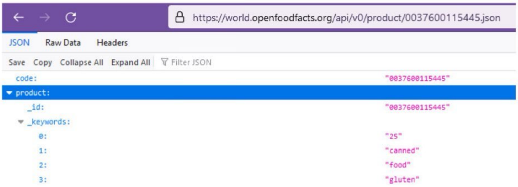
EAN-13 and **UPC-A** are two widely used barcode formats commonly found on food product packaging. The EAN-13 barcode format is designed to encode 13-digit numbers, while the UPC-A barcode format encodes 12-digit numbers. These formats are primarily utilized in specific regions, with UPC-A being predominantly used in the United States and Canada, while EAN-13 is adopted globally. Barcode data is typically stored as textual information in databases and serves as a key for product identification.

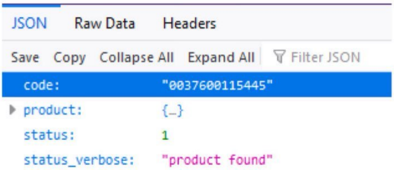
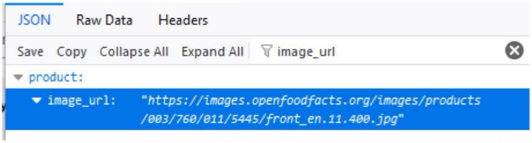
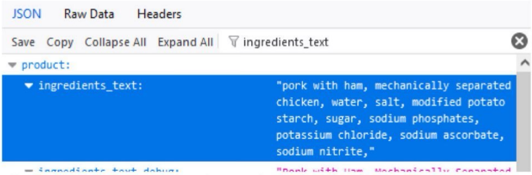
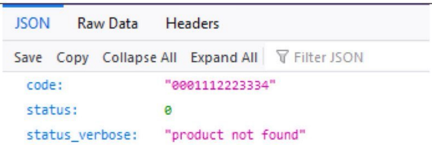
The table below presents several examples of barcodes:

Product	Format	Barcode (Numbers stored in a database)	Barcode (Printed on a label)
Nutella Ferrero 750 g	EAN-13	3017620421006	3 017620 421006 >
Coca Cola 1.25L	EAN-13	5449000267412	5 449000 000439 >
Spam, 25% less sodium 336g	UPC-A	037600115445	0 37600 11544 5

The Android mobile app will integrate a free Web API service offered by "Open Food Facts." This platform operates as a collaborative, free, and open database for food products worldwide. To fetch product data, students will utilize the Web API using the following URL format: https://world.openfoodfacts.org/api/v0/product/[barcode].json. When using this URL, students should replace the placeholder [barcode] with either an EAN-13 or a UPC-A barcode. The Web API will respond with the requested data in JSON format.

The table below presents an example of using this Web API:

An HTTP Request sent to the Web API	https://world.openfoodfacts.org/api/v0/product/0037600115445.json
An HTTP Response received from the Web API if the product is found	JSON data of the product would be found in the response.  https://world.openfoodfacts.org/api/v0/product/0037600115445.json JSON Raw Data Headers Save Copy Collapse All Expand All Filter JSON code: "0037600115445" ▾ product: _id: "0037600115445" ▾ _keywords: 0: "25" 1: "canned" 2: "food" 3: "gluten" The above image is a screenshot captured from the Firefox browser. Notably, Firefox incorporates a built-in JSON data browser, which facilitates the display of JSON data in a format that is both easy to read and comprehend.

	 JSON Raw Data Headers Save Copy Collapse All Expand All Filter JSON code: "0037600115445" product: {…} status: 1 status_verbose: "product found" The JSON data contains an element named 'status' that serves as an indicator of the web API's status. When the value of 'status' is equal to 1, it signifies that a product has been located in the database. In such cases, the product details can be accessed within the 'product' element.  JSON Raw Data Headers Save Copy Collapse All Expand All image_url product: image_url: "https://images.openfoodfacts.org/images/products/003/760/011/5445/front_en.11.400.jpg" Within the 'product' element, there is an element called 'image_url' that specifically stores the URL of the product photo.  JSON Raw Data Headers Save Copy Collapse All Expand All ingredients_text product: ingredients_text: "pork with ham, mechanically separated chicken, water, salt, modified potato starch, sugar, sodium phosphates, potassium chloride, sodium ascorbate, sodium nitrite," ingredients_text_debug: "Pork with Ham, Mechanically Separated The 'product' element contains an element called 'ingredients_text' that stores a list of ingredients.
An HTTP Response received from the Web API if the product is not found	 JSON Raw Data Headers Save Copy Collapse All Expand All Filter JSON code: "0001112223334" status: 0 status_verbose: "product not found"