NDY-9 Blue — MyFoodScan Web App
CS 4850, Section 01, Spring 2024
Feb 13, 2024

Ibrahima 	Mohammed
Mohamad 	

Table of Contents

1. Introduction

1.1. Document Outline

- Introduction
- System Overview
- Design Considerations
  - Assumptions and Dependencies
  - General Constraints
  - Goals and Guidelines
  - Development Methods
- Architectural Strategies

    - strategy-1 name or description
    - strategy-2 name or description
    - ...
- System Architecture
    - component-1 name or description
    - component-2 name or description
    - ...
- Policies and Tactics
    - policy/tactic-1 name or description
    - policy/tactic-2 name or description
    - ...
- Detailed System Design
    - module-1 name or description
    - module-2 name or description
    - ...
- Glossary
- Bibliography

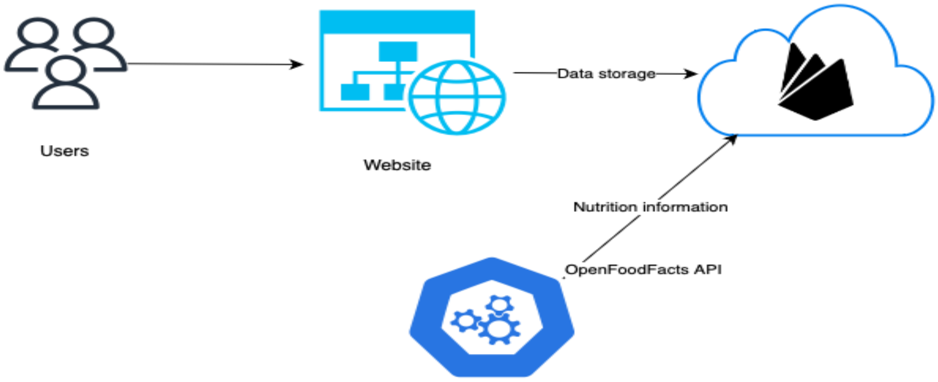
reorganized according to this outline. However, if this is done then you need to choose

1.2. Document Description

1.2.1. Introduction

Finding foods that cater to specific dietary laws like Halal and Kosher can be a challenge, especially for travelers or those in unfamiliar locales. MyfoodScan web app will allow users to lookup ceratin products based off what they look up and the application will give them brands and items that specifically follow those restrictions. For the frontend development of the application, JavaScript (JS). JS is a programming language primarily for web development allowing for dynamic and interactive web pages. For the backend side of things, we are going to be utilizing Firebase for hosting and its cloud capabilities.

1.2.2. System Overview



2. Design Considerations

2.1. Assumptions and Dependencies

The technological set of assumptions will actually be in place and the user assumptions will also be relevant. Technically, there come features such as taking into consideration only certain specific hardware of the server, specific compatibility of databases and web browsers, and the undoubtedly crucial feature of being always connected to the internet. User expectations convey the belief of the users prior to any type of exposure to the site having a history of the successful usage or at least some level of familiarity with the operations of similar platforms. Thus, is implied that a device that has the capability of accessing the internet, such as a computer, tablet, or smartphone, is also available to the users.

2.2. General Constraints

In order to gain acceptance of the project in its operating environment, we must identify, mitigate and overcome the obstacles of physical, and deployment constraints presenting. As to the features, the website should support all the latest browsers and be reachable on desktop and mobile devices. Regarding operations, the system should work best under a high-speed internet setup, and therefore 256 kbps or higher connection is highly recommended for optimal performance. In line with the limits imposed on the deployment of this site, is chosen as the clouds provider where it can be hosted and interaction with OpenFood Facts API carried through.

The target audience covers people above 18 years of age, and especially those who want to identify their packaged food with the type of nut within. A census indicates that buyers are people with special diets, as well as those who shop for others, and those who wish to learn about various dietary needs.

In the matter of the system limits, the website should be able to intuitively acess all the API inputs from OpenFoodFacts for product data. The data collected through OpenFoodFacts will be kept by and entered into the system in Google database together with consumer information provided.

2.3. Goals and Guidelines

The main goal is to enhance the speed via website about getting instances from OpenFoodFacts API. The main goals include providing users with low latency and addressing the problem of scalability as the user number grows. Read the given sentence and suggest another sentence that means the same but has a different structure. Substitute the underlined sentence with its own variant.

2.4. Development Methods

The broad overview of development will incorporate both Waterfall and Prototyping methodologies that are chosen and modified to web development. The Waterfall method will guide the project through sequential phases: dedicated to the tasks of collecting customers' requests, design, execution, verification, and maintenance.Such a method of organization makes it possible to sharpens goals and therefore, the process of development becomes more linear.

3. Architectural Strategies

Choice of Technology

Having weighed in a number of distinguishable approaches under consideration, we opted for Java script in building the frontend of MyFoodSearch -- a web-based platform. The body of the text will continue with the statement where Java script will be presented with its wide ecosystem and with the ability to build the websites which Java script to all the information coming from users in a very responsive way. MyFoodSearch development account for the crucial libraries and tools including Java script Webcam for camera access in a browser, Java script for backend integration, Java script barcode scanning library which is compatible with Java script for the purpose of processing the product barcodes, and TailwindCSS for a fast and easy-to-use styling of the frontend. Choosing Java script ; a familiar language that is JavaScript, offers so many different features as well as it makes app development faster and more efficient, making it perfect for our project.The of Google is chosen as the option for its secure environment, real time database capability, rapid query speed and easy manner of integration with Java script during development process thus creating a scalable solution. Reducing the development time via recycling existing components of the software. To ensure the development process is optimized and remains compliant with the standard in all platforms, we plan to recode software components wherever possible for reuse. For the last approach is to apply and design elements on the button components that can create reuse of it across different pages, and as for the functional aspects, the form handling, or the user authentication user modules. This strategy which merges different processes, speeds up progress as well as provides an opportunity for the users to enjoy a consistent experience while navigating the system.

Future Plans

Among the objectives that we have, is to re-design the platform enriching it with interactive elements such a favorites section where the consumers can save and directly access their favorite products and an education section that offers useful information for dietary restrictions These are imperative factors to be considered, and these things conform to our endeavor to incorporate a complete product selection assistance platform which also educates the users.

Performance Testing

Measuring the application's speed, reliability, and responsiveness properly in light of the threshold standards needed for the software to perform effectively. End-to-End Testing: Creating situations where the user has real life interaction with the platform to test if the system really works for the user. This full testing methodology makes sure that users experience smooth operation and an intuitive interface when using the application.

Once the end-to-end testing has been done, there may be issues found, the developer will then work on the codebase fixing them and once more the testing will be done to assure errors are no longer present. The cyclical nature of this process ensures that MyFoodSearch offers a valuable tool that is effective and easily understood by users, who do not experience any unexpected hurdles during its use.

4. System Architecture

MyFoodScan is intended to handle data retrieval, user interaction, data management, location services, and user interaction. The user interface handles all user input, output, and web navigation. As for the data retrieval, it collects, and process food products based on OpenFoodFacts and additional sources. The data management component will oversee user profiles, dietary preferences, and historical data for previously viewed items.

Our app, seamless cooperation between different parts is essential: the authenticity of the applicable Javascript in the application is controlled by the Firebase for the hearty e-front and back-end activities.Firestore is used to store the user data and the preferences while MongoDB is used to fetch the posts. As well, we deploy Google's Cloud Functions to process the sending apps' barcodes that produce a very good ranking of the consumption depending on the data from the OpenFoodFacts API.

Firebase Cloud takes a center stage in the real-time by automating the requests for the product information from OpenFoodFacts API and also data transmission to the frontend. It is featured like this an intermediary at the backend side that plays the role of a protector of data between our backend and the external APIs.

5. Policies and Tactics

The project team decided that the backend of the MyFoodScan web app would run on Firebase. Firebase provides a broad spectrum of backend services such as real-time database, hosting, authentication and cloud services. Firebase has the advantage of being easy to implement, scalable, and seamlessly integrable with the JavaScript-based frontend framework. The team examined a myriad of backend options like Firebase, AWS, and Google Cloud Platform. Firebase was selected due to its simplicity and ability to integrate the frontend technology without difficulty, although AWS is known for providing the most options in the cloud computing area. Therefore, the online application-made rapid development and implementation a prime goal. The organization has created coding standards and practices related to the process of Firebase development. One of them is defining security requirements, organizing cloud features, and arranging database schema. The code standards aim at making the backend more user-friendly, consistent, and maintainable at the same time. Firebase's backend services can be used through a REST API. The frontend client will use standard HTTP protocols to talk to Firebase services like Firestore for data storage and Firebase Auth for user authentication. It is through the medium of this interface that data is integrated and transferred from the frontend to the backend. Firebase's CLI tools and deployment pipelines will be applied to build and deploy the web app to Firebase Hosting. Continuous integration and deployment (CI/CD) pipelines will be adopted to automate the build, test, and deploy processes, thereby enabling a speedy delivery of updates and new features to the users.

6. Detailed System Design

Most components described in the System Architecture section will require a more detailed discussion. Other lower-level components and subcomponents may need to be described as well. Each subsection of this section will refer to or contain a detailed description of a system software component. The discussion provided should cover the following software component attributes:

6.1. Frontend

The website will provide manual and social authentication options to make users sign in effortlessly and fast. With regards to manual logins, we securely deal with and keep user credential information, including usernames and passwords. For those who like to log in via social networking, our site is integrated with the most common platforms, such as Google and Yahoo, so users can access our services with their existing social media accounts. Manual Authentication Process: Users who choose to log in manually will be verified by the Firebase Authentication service directly. Once the users are verified, they are given access to various features of the website. Our website makes use of the OAuth protocols for the social media logins so that users can log in through their social network accounts. This method guarantees a secure transfer of user data between social platforms and our website, enabling simpler logging in processes but at the same time providing high security.

6.2. Backend

Our site is supported by a Firestore NoSQL database which is a very useful tool for storing and managing the user profile information including the personal data and a scanned products history. Firestore guarantees real-time synchronization across each user’s device, giving access to the information at any time. Firestore enables creating, updating, and deleting the user profile information. Additionally, it takes the responsibility for authentication of new data, including email and phone number updates, and then incorporates these changes into user profiles. The site has the history of the last ten scanned items stored in the Firestore. The scanned products are shown with their picture, title, and dietary alignment status (check or "x"), making it very easy for the user to review them. Cloud functions specifically designed for this purpose evaluate the product data against the the user's dietary preferences. The compliance is then signified by a check or an "X."

6.3. API

The process involves sending HTTP requests to the OpenFoodFacts API, decoding the data returned from the API, and empowering the user experience with product details and features that enhance dietary compliance. Once the OpenFoodFacts API responds, the function at the back end uses the JSON data to isolate the crucial product information. This process guarantees that users get the data that is precise and relevant to the barcodes they scan, which end up being used to make informed decision based on their dietary needs.