App Architecture and Code Workflow

DASS Team-33

IIIT Hyderabad - April 19, 2024

Contents

1 Overview

This application is designed to be a cross-platform mobile app with a scalable scope. It consists of four main components: Frontend, Backend, Database and Web Servers.

2 Components

2.1 Frontend

2.1.1 Framework

React Native

2.1.2 Justification

React Native is chosen for its cross-platform compatibility and scalability. With a large community and extensive documentation, it provides flexibility for modifying the codebase as the project evolves.

2.2 Backend

2.2.1 Framework

Flask

2.2.2 Justification

Flask is chosen for its flexibility and suitability for rapid prototypes. It allows for better control over the database and coding flexibility, making it ideal for initial development on a small platform.

2.3 Database

2.3.1 Framework

MongoDB

2.3.2 Justification

MongoDB is chosen for its ease of interaction with React and web servers. Its flexible document model is suitable for storing diverse data types, and it promotes seamless integration with other components of the architecture.

2.4 Web Servers

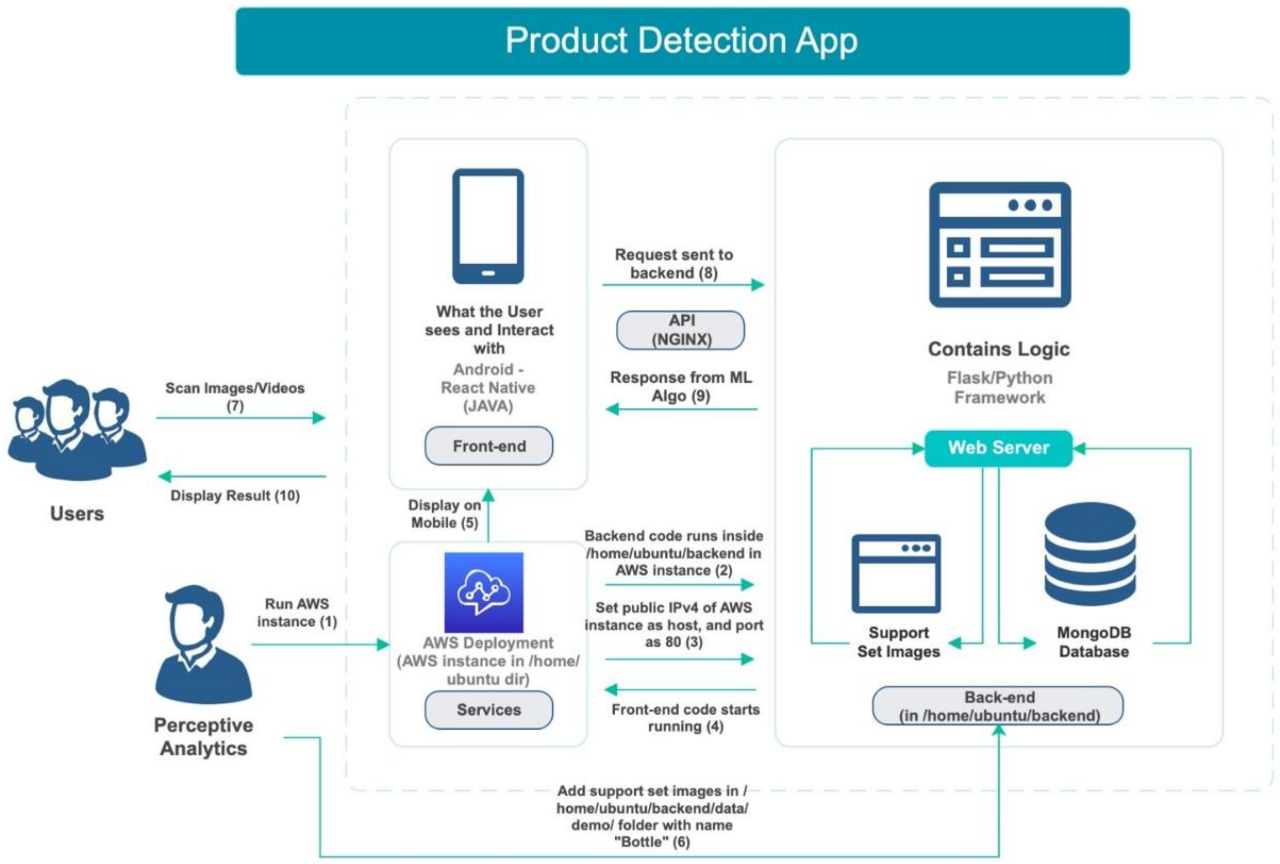
2.4.1 Framework

NGINX

2.4.2 Justification

NGINX is chosen to serve as the web server for its capability to handle concurrent connections efficiently. It promotes quicker concurrency in API and other data transfer across the network, enhancing the overall performance of the application.

3 Application WorkFlow


Product Detection App
What the User sees and Interact with
Android - React Native (JAVA)
Front-end
Request sent to backend (8)
API (NGINX)
Response from ML Algo (9)
Contains Logic
Flask/Python Framework
Web Server
Support Set Images
MongoDB Database
Back-end (in /home/ubuntu/backend)
Scan Images/Videos (7)
Display Result (10)
Users
Display on Mobile (5)
Backend code runs inside /home/ubuntu/backend in AWS instance (2)
Set public IPv4 of AWS instance as host, and port as 80 (3)
Front-end code starts running (4)
Run AWS instance (1)
AWS Deployment (AWS instance in /home/ubuntu dir)
Services
Perceptive Analytics
Add support set images in /home/ubuntu/backend/data/demo/ folder with name "Bottle" (6)

3.1 Description

The above flow chart illustrates the flow of data and processes by providing a visual representation of the architecture and the interaction between different components of the application. Developers can refer to this diagram to understand the code workflow in depth.

4 Updating Class Images

Administrators must take note of the following steps for adding or updating the existing class images:

1. Navigate to the `backend/raw_inputs` folder.
2. To modify an existing class image, change the files in the folder.
3. Please note that the images should be in `.png` format.
4. Please note that while giving an input to a new order the item name should always match with the corresponding entities in the `backend/raw_inputs` folder.
5. If there is a change in the IP address of the server, you can compensate for the changes by changing the host and port accordingly in the `frontend/Constants.js` file.