COMP3900 – Info Tech Project

P23 – Predicting the freshness and best before of foods using AI-based analysis of colour and shape change

Software Design Document

COMP3900H11ADigitalHaven

This document presents the software design document of the project. This includes the system architectural diagram presenting the workflow and dataflow in a diagram format, followed by a textual description. To better visualise the diagrams, a separate document named 'System Architectural Diagrams' containing the full-scale diagrams in this document has been attached in the handover pack.

Overall System Architecture

To use this system, a user can authenticate themselves through the frontend interface to gain access to all application features. Once authenticated, they can view their previous prediction records, which are securely stored in a database. Additionally, users can upload an image to generate a new prediction. Once a successful prediction is made the application will save a copy of that uploaded image in its database.

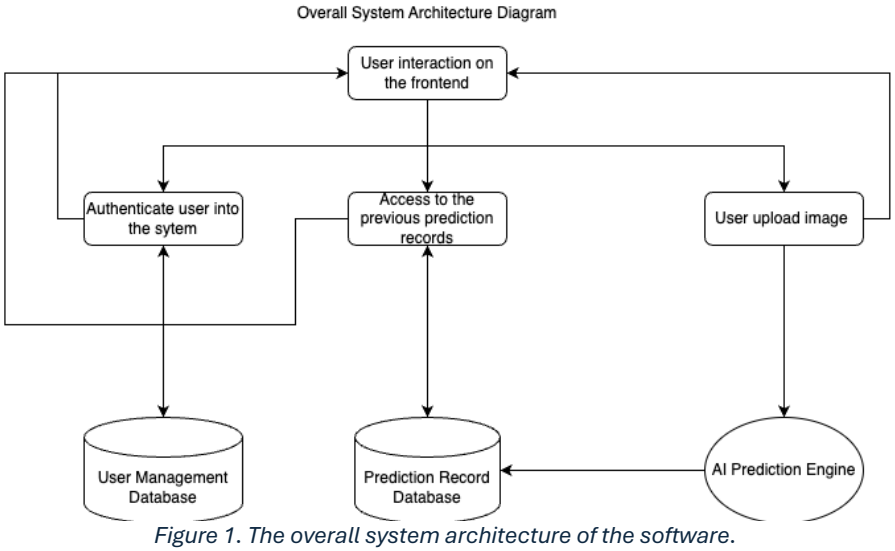

Figure 1. The overall system architecture of the software.

Frontend System Architecture

When a user arrives at the welcome page, an option to log in or register into the system will be provided. If the user has an existing account, they will be directed to the history page after login, where all previous prediction histories and inventories will be presented. If the user registers for the service, they will be directed to the profile page after registering, where the user can upload their profile picture, modify their password, and update their preferred default notification date. After being authenticated, the user could switch between the prediction page, where the user uploads an inventory to generate the expiration prediction, the history page, and the profile page. A login button would also be provided to authenticate the user from the system. To manage the exciting predictions, the user would be navigated to the history page where options have been provided to mark a product being consumed or disposed of, change the notification dates for a product, and view the product details. A notification alert would also be provided if a product is not consumed but within the notification period before the expiration date.

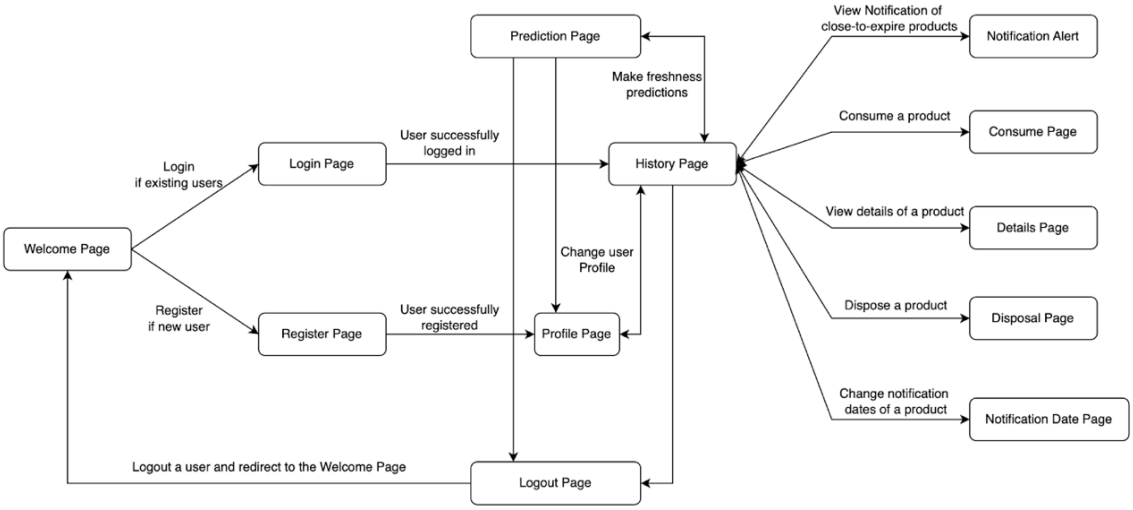

Figure 2. The frontend system architecture of the software.

Backend System Architecture

The backend uses flask app routing to receive requests from the frontend. Upon registration, their information is stored on a Flask SQL alchemy database, and they receive a JSON Web Token for authentication. For login, their details are checked with information from the database, and if successful, are given an access token.

When a user inputs an image into the system through the prediction page, the image as well as its metadata, is stored on the flask SQL alchemy database. This includes the weather information, which is obtained through the Open Weather Map API, by getting the user's location information through their web browser. All images inputted by the user can be viewed through the history page which queries the database. Also, every 6 hours, the system sends an email to the user, notifying them of a product that is near to its expiry date.

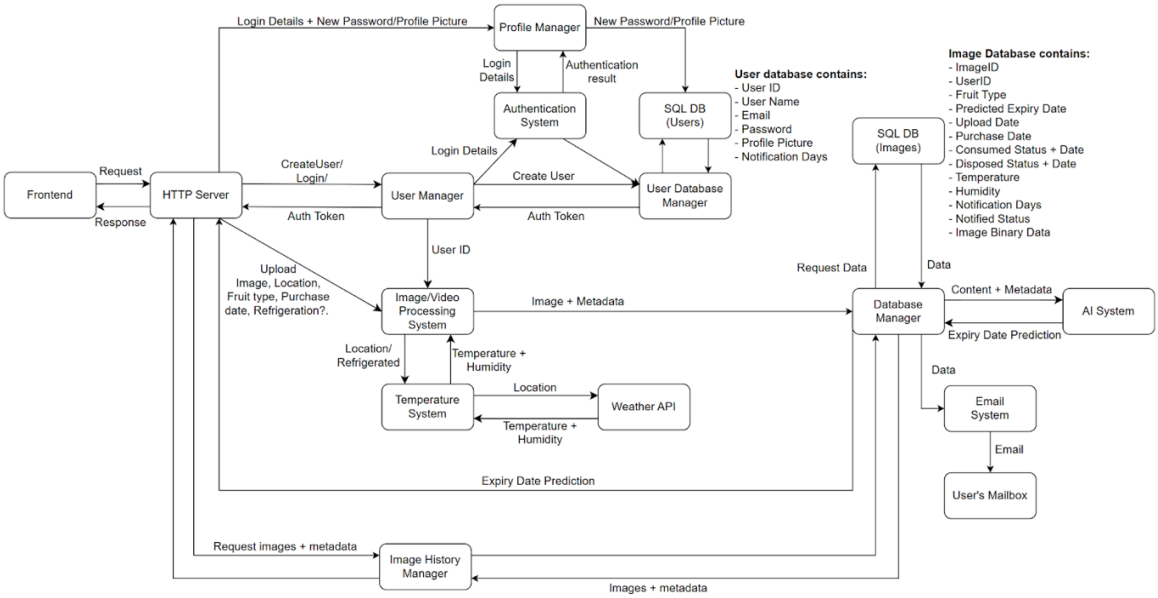

Figure 3. The backend system architecture of the software.

AI Engine System Architecture

To predict the freshness of the product, the system takes an image input and processes it through the fruit detection model (YOLOv5), which identifies all instances of fruits in the image. Once fruit(s) are detected, the detected fruits are cropped out of the image and passed into the best-before classification model specific to each fruit type (e.g., apple, banana, mango, orange, strawberry). Each classification model then assesses the detected fruit's freshness to determine its best-before status. Finally, the system outputs a prediction, indicating the best-before classification for the identified fruit.

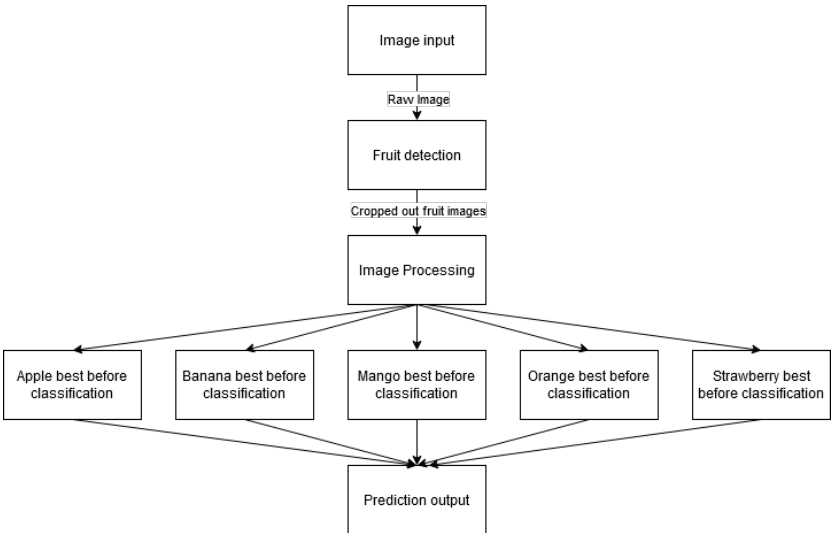

Figure 4. The AI architecture of the software.

The Flow of data for fruit prediction

When a user wants to make a fruit prediction, they upload an image through the frontend interface, which then sends the image to the backend. The backend then forwards this image to the AI engine, where the detection model first identifies and crops out instances of the fruits. Each cropped image is then passed through a confidence filter to ensure accuracy: the system retains only the highest confidence fruit type detected (e.g., if the highest confidence is for apples, only apples will be considered, while other fruit types are discarded). Once filtered, the classification model is applied to each of the cropped images of fruits, and the results are sent back to the backend. The backend averages out these predictions and applies an algorithm to adjust the best-before range based on whether the user stores their fruits in a refrigerator or outside. Finally, this refined best-before range is stored in the database and displayed on the frontend for the user.

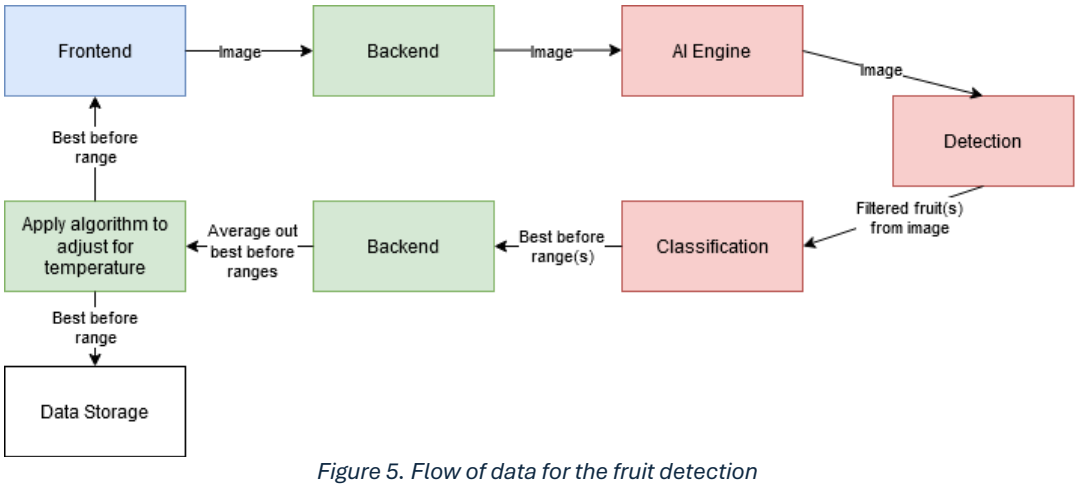

Figure 5. Flow of data for the fruit detection