BIL 481 – Design Document

TOBB ETÜ

University of Economics & Technology

16 Mar 2025

TASK MATRIX

Section	Description	Contributor
1. Overview	Overview of the quality assurance plan and its objectives.	Efe Görkem
2. Testing Methodologies	Description of testing techniques (unit testing, integration testing, usability testing) and automated vs. manual testing strategy.	Efe Görkem
3. Quality Factors & Metrics	Definition and explanation of performance, accuracy, false positive rate, and UI responsiveness.	Ali Şahin
4. Test Plan	Definition of test cases, expected outcomes, and system behavior.	Ege Rasim
5. Bug Tracking	Description of bug reporting and tracking process using GitHub Issues.	Joint Contribution

1. Overview

This document describes the approach to ensuring software quality in the PickDish project. The quality assurance plan is designed to minimize errors, improve user experience, and ensure system sustainability.

2. Testing Methodologies

The testing methodologies to be used in the project include:

2.1. Unit Testing: Ensuring the accuracy of independent components such as the YOLO model and OpenAI API integration.

2.2. Integration Testing: Validating that different components (YOLO model, API calls, user interface) work together seamlessly.

2.3. Usability Testing: Assessing how users experience the interface and its ease of use. Functional Testing: Verifying that the core features of the application work as expected.

2.4. Automated vs. Manual Testing:

Automated Tests: Unit tests will be used to measure the accuracy of the YOLO model and validate API calls.

Manual Tests: User interface, error messages, and usability testing will be conducted manually.

3- Quality Factors & Metrics

Quality Factor	Description	Measurement Metric
Performance	Speed of ingredient detection and recipe generation.	Average response time (ms) from image upload to recipe display.
Accuracy	Accuracy of detected ingredients and relevance of generated recipes.	Percentage of correctly detected ingredients and user rating on recipe relevance.
False Positive Rate	Rate of incorrectly detected ingredients.	False positive count / total detections (%)
UI Responsiveness	Speed and smoothness of user interface interactions.	Average response time for user actions (ms)

4- TEST PLAN

Test Cases:

Test Case 1. User opens program, uploads a picture of 2 apples, 1 banana and 1 orange, which was within the training picture dataset. Then, user clicks the "Generate Recipe" button. System returns with a recipe for a fruit salad.

Test Case 2. User opens program, clicks the "Generate Recipe" button. System returns a warning message saying "Photo of the ingredients is required for recipe generation.".

Test Case 3. User opens program, uploads a picture of a pack of pasta, a clove of garlic and a jar of tomato paste, which was not in the training picture dataset. Then, user clicks the "Generate Recipe" button. System returns with a recipe for pasta with tomato sauce.

Test Case 4. User opens program, uploads a picture of an empty table, then clicks the "Generate Recipe" button. System returns a warning message saying "No ingredients are found within the picture.".

Test Case 5. User opens program, uploads a picture of a bread and some cheese, then clicks the "Generate Recipe" button. System returns with a recipe for toast with cheese on

top. Then, user clicks “Generate Recipe” button for a second time. System returns with a recipe for grilled cheese sandwich.

Bug Tracking:

Bug reporting and tracking will be managed via GitHub Issues. Each bug report will include:

- Bug description
- Steps to reproduce
- Expected vs. actual results
- Priority level
- Bug resolution status