Software Quality Assurance Report

& 8 Screen Record

**QA Methodologies: The system follows various methodologies to ensure quality, including:**

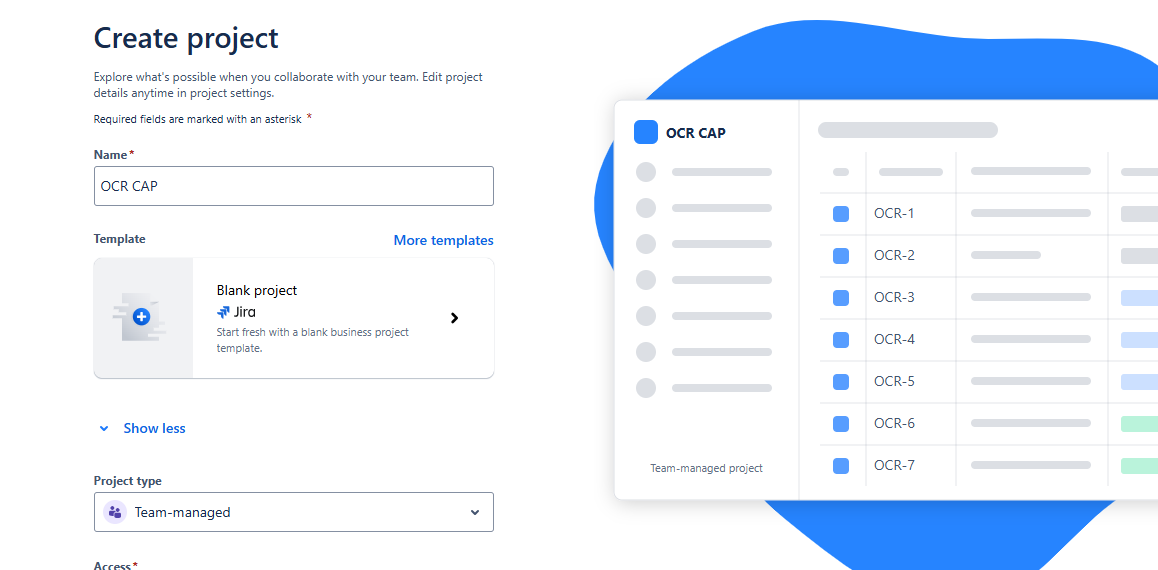
* **Behavior-Driven Development (BDD): Focuses on user behavior to ensure that features meet expectations.**

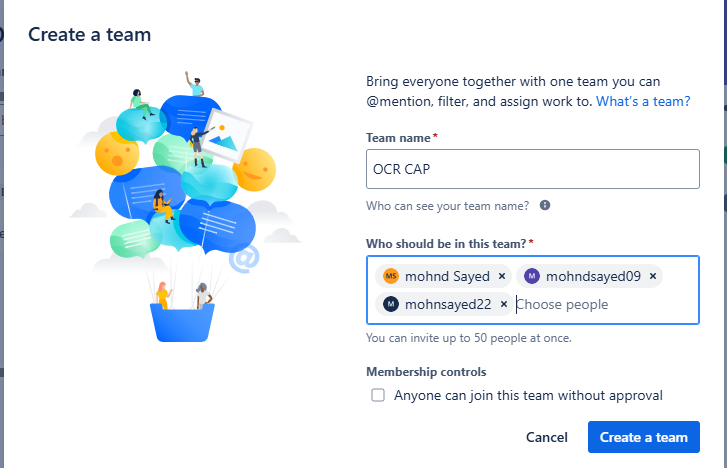
**Quality Management Plan (QMP):**

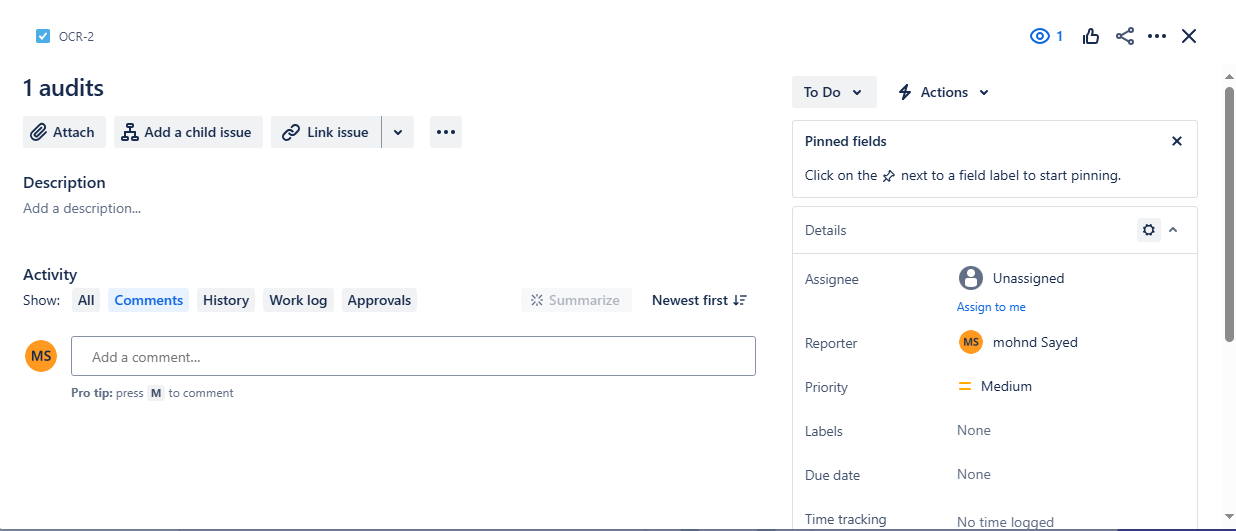
* **The objective of the plan is to ensure the system accurately detects car plate numbers from images.**
* **Project Manager: Ensures that quality goals are met.**
* **QA Manager: Conducts audits and testing.**
* **Developers/Testers: Develop and test the system.**
* **QA Engineer: Monitors tests and documents the results.**

**Quality Assurance Procedures:**

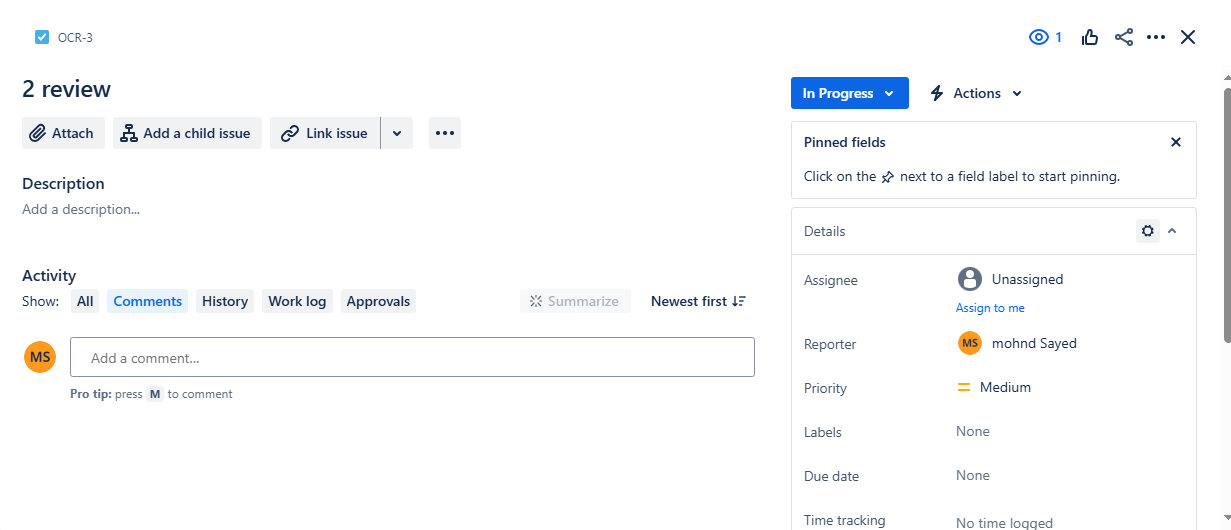
**- Audits: regular checks on the detections model performance  
- Reviews: verify documentation and test result   
- Inspections: ensuring the software match the initial requirements**



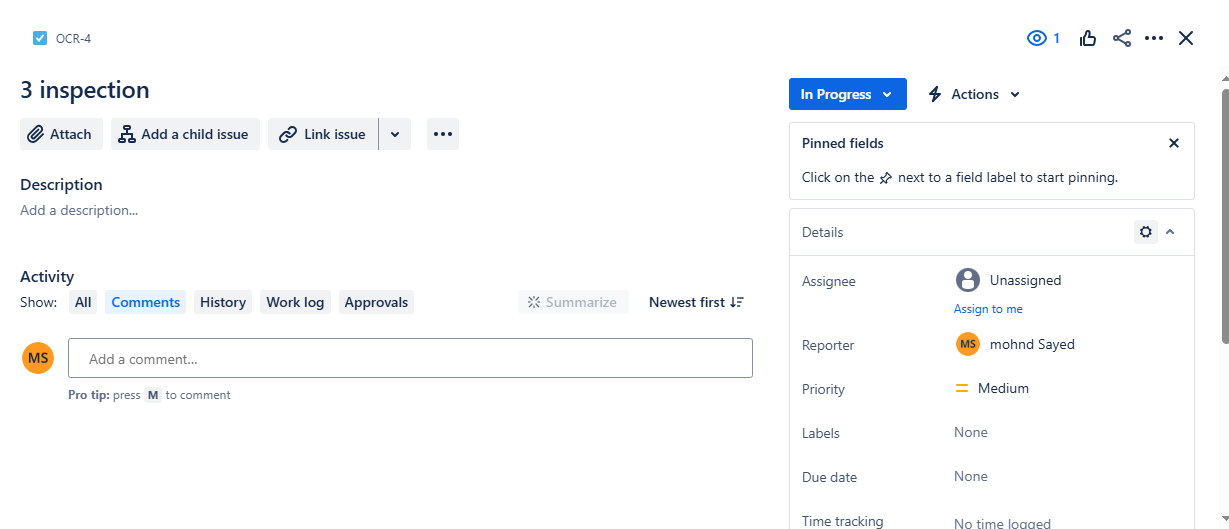


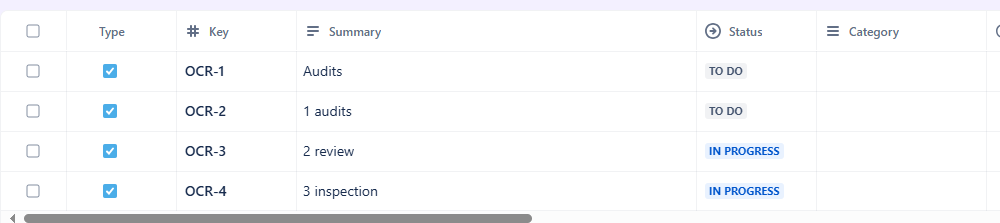
Task 1 **audits send the task to someone to make it**

**Task 2 :** **review**



Task **3 inspection**





**4. Test Plans and Test Cases:**

**Test Case 1: Detect Car Plate Under Various Lighting**

* **Test Case Code: CD\_01**
* **Description/Scope: Tests the system's ability to detect car plates under different lighting and weather conditions.**
* **Steps:**
  + **Upload the image on the Hugging Face platform.**
  + **Process the image.**
  + **Detect the car plate number under different lighting and weather conditions.**
* **Input: Car image.**
* **Expected Result: Correct detection of the car number plate.**
* **Acceptance Criteria: 90% detection accuracy.**
* **Result: Passed.**
* **Comment: The system successfully detected the number plate under various lighting and weather conditions.**

**Test Case 2: Detect Car Plate Number from Far Away**

* **Test Case Code: CD\_02**
* **Description/Scope: Tests the system's ability to detect car plate numbers when the car is at a distance.**
* **Steps:**
  + **Upload the car image on Hugging Face.**
  + **Process the image.**
  + **Detect the car plate number from far away.**
* **Input: Car image.**
* **Expected Result: Correct detection of the car number plate.**
* **Acceptance Criteria: 88% detection accuracy.**
* **Result: Passed.**
* **Comment: The system successfully detected the number plate when the car was far away.**

**Test Case 3: Detect the Car's Plate Number Written in Arabic**

* **Test Case Code: CD\_03**
* **Description/Scope: Tests the system's ability to detect car plates that are written in Arabic.**
* **Steps:**
  + **Upload the car image on Hugging Face.**
  + **Process the image.**
  + **Detect the car plate number in Arabic.**
* **Input: Car image.**
* **Expected Result: Incorrect detection of the car number.**
* **Acceptance Criteria: 65% detection accuracy.**
* **Result: Failed.**
* **Comment: The system could not correctly detect car numbers written in Arabic.**

**5. Defect Reports**

**Total Test Cases: 3 test**

**Success Test Case: 2 testcase**

**Failed Test Case: 1   
Issue: the system can not detect car plate number when it Is written in Arabic (incorrect detection for Arabic number plate image   
Severity: high   
Test case code : DC\_03**

**solution: Adjust model with more diverse training data**

**6. Change Requests**

**Change: Add support for plates number of car image Arabic .( for non-standard license plates)**

**Impact: Requires more data collection and model retraining.**

**7. Version Control System (VCS) Logs**

**Track code changes using Git.**

**Example Log: " add new collected data to retrain mode – 20\_10 \2024"**

**8. Configuration Management Documentation**

**Configurations:**

**Versions of necessary Libraries**

**- Python 3.8, OpenCV 4.7, YOLOv8 Model**

**9. Quality Metrics and Reports / KPIs**

**Defect Density: 1 per 3 test case**

**Test Coverage : 85% of possible scenarios tested.**