# 📝 NLP Project – Technical Lead Report

**Name**: Erin David Cullen

**Date**: 21 July

**Date**: 08 August

## ✅ 1. Work Completed (Since Last Report)

\_What you’ve built, implemented, tested or reviewed and an estimate as to how many hours it took to complete

* AI module development and integration (~12-16 hr)
* ROS2 video feed implementation (~4 hr)
* Docker container setup and configuration (~4 hr)
* Feature/Module:
  + Implemented individual AI processing modules (object detection, speech-to-text, intent processing, depth analysis, RTAB-Map SLAM)
  + Built main AI.py system to subscribe to ROS2 video feeds and process frames
  + Created ROS2 video publisher to stream camera feeds into ROS2 container
* Tools/Libs Used:
  + ROS2 Humble with rosbridge websocket
  + OpenCV for video processing
  + YOLOv8 (ultralytics) for object detection
  + Whisper (transformers) for speech-to-text
  + WebSocket for ROS2 communication
  + Docker containers for modular deployment
* Outcome/Results:
  + Successfully implemented video feed pipeline: Camera → ROS2 Container → AI Processing → Display
  + AI modules can run independently and process video/audio inputs
  + Established WebSocket-based communication between containers
  + System can receive and display video frames from ROS2 with processing overlays
* Contributions (if applicable):
  + Built AI.py main system with frame processing pipeline
  + Implemented object detection node using YOLOv8
  + Created speech-to-text node using Whisper
  + Developed ROS2 video publisher for camera feed integration
  + Established container communication architecture

## 📌 2. Current Tasks in Progress

\_What you’re actively working on. Include blockers if any.\_

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Description | ETA | Blockers |
| Building AI processing modules | Implementing individual modules (object detection, STT, intent processing, etc…) | ~12-16 hours | Module dependencies, model loading |
| Static module testing | Getting each module to work independently with test data | ~4 hours | Built modules |
| Input integration | Connecting video and audio inputs to the ROS2 container | ~4 hours | ROS2 WebSocket reliability |

## 📅 3. Upcoming Tasks

\_Planned work for the next sprint or phase.\_

• **Task:** Module Consolidation

* **Purpose/Goal:**
  + Integrate all AI processing modules into unified pipeline
  + Ensure seamless data flow between object detection, STT, intent processing, and SLAM
* **Dependencies:**
  + All individual modules working independently
  + ROS2 input streams established

• **Task:** Live Stream Processing Pipeline

* **Purpose/Goal:**
  + Feed live video/audio data streams into the consolidated AI pipeline
  + Process real-time data and output results for downstream systems
* **Dependencies:**
  + Consolidated module pipeline completed
  + Stable ROS2 WebSocket connections
  + Performance optimization for real-time processing

## 🚨 4. Issues & Risks

\_Bugs, technical debt, resourcing, or anything threatening progress.\_

|  |  |  |  |
| --- | --- | --- | --- |
| Issue | Impact | Suggested Action | Owner |
| Low latency video inputs | Video feed unreliability affects real-time processing | Lower the latency and improve the stability and reliability | Ed |
| ROS2 container isolation | Difficult integration with Python-based project components | Design ROS2 bridge/API for Python communication | Ed |
| Live video processing challenges | Object detection on continuous feed vs single images | Optimize processing pipeline and implement frame skipping | Hayley |

## 📈 5. Key Insights / Recommendations

\_Lessons learned, suggestions, architecture notes, or optimizations.\_

* WebSocket-based communication between Docker containers proved effective for modular AI system architecture
* Individual AI modules (YOLO, Whisper) work well in isolation but integration complexity increases with real-time constraints
* ROS2 Humble container serves as reliable central hub for video/audio data distribution across system components