MAVENCODE
EARS DOCUMENT

Version <1.2>

Date	Version	Description	Author
12/04/2025	1.1	Revision based on first 2 sprints and client review	Mavencode team
04/07/2026	1.2	Comprehensive revision based on agent orchestration, agentic intelligence, complete module implementation, ROS2 integration, and digital twin simulation	Mavencode team

SYSTEM OVERVIEW

VANTAGE (Visual Autonomous Navigation and Task-driven Agentic Ground-to-air Engine) is an intelligent drone control platform combining perception AI, agentic reasoning, and digital-twin simulation. Users control drones through natural language commands via an LLM-powered agent that orchestrates multi-step missions, integrates computer vision tools, and coordinates drone movements.

Three core pillars:

1. **Perception Intelligence**: Face detection, object recognition, semantic segmentation, vision-language reasoning
2. **Agentic Communication**: LLM-powered agent interprets commands and orchestrates multi-step missions
3. **Digital Twin Simulation**: Gazebo-based drone simulator for safe testing before real-world execution

FUNCTIONAL USE CASES

Section A: Agentic Intelligence & Mission Planning

UC-A1: Natural Language Mission Command to Agent
Ubiquitous - The system shall accept natural language mission commands via text input or voice input.
Event-Driven - When the user submits a text command (e.g., "fly forward 100cm and look for people"), the agentic layer (smolagents ToolCallingAgent) shall parse the command using an LLM (Ollama llama3.2 or compatible).
Event-Driven - When the LLM parses the command, it shall identify required tools and parameters, invoke those tools in the correct sequence, and return results to the user.
Event-Driven - When a command involves multiple steps (e.g., takeoff → move → capture → process), the agent shall orchestrate each step and report progress.
Optional - Where the command is ambiguous or unsafe, the system shall ask for clarification or refuse execution with a human-readable explanation.
Related capabilities: UC-A2, UC-D1

UC-A2: Agent Tool Registry and Auto-Registration
Ubiquitous - The system shall maintain a registry of available tools that the LLM agent can invoke.
Event-Driven - When the mission agent initializes, it shall load all registered tools from the MCP server and drone tools module (drone_tools.py).
Event-Driven - When new perception tools are added or drone capabilities are enhanced, the agent shall automatically discover and use them without code changes to the agent itself.
Event-Driven - When a tool is invoked by the agent, the system shall execute the tool, capture its return value, and provide feedback to the LLM for next-step reasoning.
Optional - Where tool execution fails, the system shall catch the exception, return an error message, and allow the agent to retry or use an alternative approach.

UC-A3: Multi-Step Interactive Mission Execution
Ubiquitous - The system shall support interactive, multi-step mission execution where the user can provide commands in real-time.
Event-Driven - When the mission agent starts in interactive mode, it shall await user input from stdin, process each command through the agentic loop, and display results including status updates.
Event-Driven - When the drone executes a complex task (e.g., "survey the room"), the agent shall emit status messages to an internal mission_status field for HUD display.
Event-Driven - When the user interrupts the mission with SIGINT (Ctrl+C), the system shall trigger an emergency land command and gracefully shut down all threads.
Optional - Where the user specifies a one-off mission via --mission flag, the system shall execute that mission and exit upon completion.
Related capabilities: UC-D1, UC-D2

UC-A4: Dry-Run Mode for Mission Planning
Ubiquitous - The system shall offer a dry-run mode to validate mission plans before executing on the real drone.
Event-Driven - When the --no-fly flag is provided, the system shall route all drone commands through the digital twin (Gazebo) instead of the real hardware.
Event-Driven - When commands execute in dry-run mode, telemetry and results shall be simulated, allowing the user to verify the mission logic without hardware risk.
Optional - Where the user wants to compare dry-run vs. real results, the system shall log both execution paths for analysis.

SECTION B: DRONE OPERATION & CONTROL (DJI Tello Integration)

UC-B1: Drone Connection and Status Monitoring
Ubiquitous - The system shall manage connection state with the DJI Tello drone as a singleton resource.

Event-Driven - When the mission agent initializes, it shall attempt to connect to the drone and start the video stream.
Event-Driven - When the connection succeeds, the system shall set is_connected flag to True and broadcast telemetry at regular intervals.
Event-Driven - When the connection fails or the drone becomes unreachable, the system shall report the error and prevent flight commands.
Event-Driven - When the video stream is active, the system shall display the drone's camera feed in the OpenCV window with a status HUD overlay.
Optional - Where network latency is high, the system shall degrade gracefully and report reduced responsiveness.
Related capabilities: UC-B2, UC-B3, UC-D1

UC-B2: Drone Takeoff and Landing
Ubiquitous - The system shall provide commands to safely transition the drone between ground and airborne states.
Event-Driven - When the user or agent invokes drone_takeoff(), the system shall command the drone to take off and hover in place.
Event-Driven - When takeoff completes, the system shall update is_flying flag to True and set mission_status to indicate flight state.
Event-Driven - When the user or agent invokes drone_land(), the system shall command the drone to descend to ground and disarm motors.
Event-Driven - When landing completes, the system shall set is_flying to False and clear mission_status.
Event-Driven - When battery falls below safety threshold or emergency_stop is triggered, the system shall automatically execute drone_land() regardless of current state.
Optional - Where the drone detects an obstacle during landing, the system shall halt descent and request user intervention.
Related capabilities: UC-B1, UC-B3, UC-B4

UC-B3: Drone Linear Movement
Ubiquitous - The system shall allow the drone to move in cardinal directions (forward, backward, up, down) while airborne.
Event-Driven - When the agent invokes drone_move(direction, distance_cm), the system shall send Twist velocity commands to the drone.
Event-Driven - When the drone reaches the target distance, the system shall halt and hover, returning success status to the agent.
Event-Driven - All linear movements shall validate that distance_cm is between 20-500 centimeters to ensure safe operation.
Event-Driven - When the drone's trajectory would exceed workspace bounds or battery is critically low, the system shall reject the command.
Optional - Where fine-grained position control is needed, the system shall expose delta movement commands that can be chained together.
Related capabilities: UC-B2, UC-B4, UC-A1

UC-B4: Drone Rotation (Yaw Control)
Ubiquitous - The system shall allow the drone to rotate in place to change camera orientation.
Event-Driven - When the agent invokes drone_rotate(degrees, direction), the system shall command the drone to rotate around its vertical axis.
Event-Driven - The direction parameter shall accept 'cw' (clockwise) or 'ccw' (counter-clockwise) to allow bidirectional control.
Event-Driven - All rotations shall validate that degrees is between 1-360 to ensure safe, complete rotations.
Event-Driven - When rotation completes, the system shall return success and the new camera heading to the agent.
Optional - Where the user specifies absolute heading (e.g., "face north"), the system shall compute the relative rotation needed and execute it.
Related capabilities: UC-B3, UC-C1, UC-C2

UC-B5: Battery and Telemetry Monitoring
Ubiquitous - The system shall continuously monitor and report drone battery, height, flight time, and temperature.
Event-Driven - When the agent invokes get_drone_status(), the system shall query live telemetry from the drone

and return structured data.
Event-Driven - Telemetry data shall include: battery percentage, height in centimeters, flight time in seconds, temperature in Celsius, flying state, and connection state.
Event-Driven - When battery percentage falls below 20%, the system shall issue a warning to the HUD and user interface.
Event-Driven - When battery falls below 10%, the system shall automatically trigger an emergency landing sequence.
Event-Driven - When temperature exceeds safe operating range, the system shall log a warning and recommend cool-down period.
Optional - Where the user wants to extend mission duration, the system shall provide battery projection estimates based on current consumption rate.
Related capabilities: UC-B1, UC-B2

UC-B6: Emergency Stop and Thread-Safe Commands
Ubiquitous - The system shall provide fail-safe mechanisms to prevent unsafe drone operation.
Event-Driven - When the emergency_stop flag is set (via SIGINT or error condition), the system shall immediately halt all movement and execute landing.
Event-Driven - All drone commands shall be protected by threading locks to prevent race conditions when commands arrive from multiple threads simultaneously.
Event-Driven - When a command is rejected due to emergency stop or safety constraint, the system shall return a failure status with clear explanation.
Optional - Where the drone becomes unresponsive, the system shall trigger automatic emergency stop and disable further commands until connection is restored.
Related capabilities: UC-B1, UC-B2

SECTION C: PERCEPTION INTELLIGENCE & COMPUTER VISION

UC-C1: Real-Time Image Capture from Drone
Ubiquitous - The system shall enable capturing still images from the drone's camera for downstream analysis.
Event-Driven - When the agent invokes capture_frame(), the system shall grab the current frame from the video stream and save it to disk.
Event-Driven - The captured image shall be saved with a timestamp-based filename to avoid overwrites.
Event-Driven - The capture_frame tool shall return the absolute file path of the saved image for use in subsequent vision tool calls.
Event-Driven - When video stream is unavailable, the system shall return a failure status and suggest reconnecting the drone.
Optional - Where high-frequency capture is needed, the system shall support batch image capture at specified frame rates.
Related capabilities: UC-C2, UC-C3, UC-C4, UC-C5

UC-C2: Object Detection and Localization
Ubiquitous - The system shall detect and localize objects in images with bounding boxes and confidence scores.
Event-Driven - When the agent invokes detect_objects(source, confidence_threshold=0.25, iou_threshold=0.45), the system shall run YOLOv8 object detection.
Event-Driven - The detection results shall include: bounding box coordinates (x1, y1, x2, y2), confidence score (0.0-1.0), class ID, and human-readable class name (e.g., 'person', 'car', 'dog').
Event-Driven - Common use cases include answering "how many people are visible?" or "is there a vehicle in the frame?".
Event-Driven - When save=True, the system shall annotate the image with bounding boxes and save to disk.

Event-Driven - Users can adjust confidence_threshold (lower = more detections but more false positives) and iou_threshold (controls overlap suppression).
Optional - Where performance is critical, the system shall cache the detection model in GPU memory between calls.
Optional - Where custom object classes are needed, the system shall support loading fine-tuned YOLOv8 models from disk.
Related capabilities: UC-C1, UC-A1

UC-C3: Semantic Segmentation for Scene Understanding
Ubiquitous - The system shall analyze images at the pixel level to identify and classify regions (road, sky, person, car, building, vegetation, etc.).
Event-Driven - When the agent invokes segment_image(source, variant='deeplabv3_resnet50', save_overlay=True), the system shall run semantic segmentation inference.
Event-Driven - The segmentation results shall include: list of detected classes with percentage coverage, pixel count per class, class ID, and image dimensions.
Event-Driven - Total number of distinct classes (excluding background) shall be reported for scene complexity analysis.
Event-Driven - When save_overlay=True, the system shall save a color-coded visualization showing which regions belong to which class.
Event-Driven - Common use cases include "estimate distance to person by checking coverage percentage" or "analyze road composition".
Optional - Where different model variants are needed (speed vs. accuracy tradeoff), the system shall support multiple DeepLabV3 backbone options.
Optional - Where real-time segmentation is critical, the system shall cache the model in GPU memory.
Related capabilities: UC-C1, UC-A1

UC-C4: Image Captioning and Visual Description
Ubiquitous - The system shall generate natural language descriptions of images automatically.
Event-Driven - When the agent invokes caption_image(image_url, device='cpu', model='Salesforce/blip-image-captioning-base'), the system shall produce a 1-2 sentence caption.
Event-Driven - The caption shall describe the main objects, actions, and scene context in the image.
Event-Driven - Captions are general high-level descriptions (not detailed inventories); for specific details, use ask_question_about_image instead.
Event-Driven - Caption generation shall work on both local file paths and HTTP(S) URLs as input.
Event-Driven - Users can specify device ('cpu' or 'cuda') to control performance vs. memory tradeoff.
Optional - Where accessibility is critical, the system shall automatically generate alt-text via captioning when requested.
Related capabilities: UC-C1, UC-C5

UC-C5: Visual Question Answering (VQA) - Image Understanding
Ubiquitous - The system shall answer natural language questions about image content using vision-language models.
Event-Driven - When the agent invokes ask_question_about_image(image_url, question, model='blip'), the system shall process the image and question through a VQA model.
Event-Driven - The VQA system shall support diverse question types: "What color is X?", "How many Y are there?", "What is the person doing?", "Is this indoors or outdoors?", "What time of day?".
Event-Driven - Common use cases include real estate drone survey ("Are there any people visible?"), safety assessment ("Is this area cluttered?"), and scene understanding ("How many windows does the building have?").
Event-Driven - When save=True, the system shall save the annotated image and question-answer pair to disk for logging.
Event-Driven - The answer shall be returned as natural language text that the agent can reason about.
Optional - Where multiple VQA models are available, the system shall allow model selection to balance quality vs. inference speed.
Optional - Where follow-up questions are asked, the system shall maintain image context between calls for conversational VQA.
Related capabilities: UC-C1, UC-C4, UC-A1

UC-C6: Face Detection and Recognition
Ubiquitous - The system shall detect faces in images and provide bounding boxes and confidence scores.
Event-Driven - When the agent invokes face detection through the perception tools, the drone can assess whether people are present in the scene.
Event-Driven - Face detection output shall include bounding box coordinates and confidence metrics for each detected face.
Event-Driven - Common use cases: "Are there any people in this building?", "How many individuals are visible from this angle?", "Is the area occupied?".
Event-Driven - When save=True, face bounding boxes shall be annotated and saved for visual confirmation.
Optional - Where privacy is critical, the system shall offer face blurring on captured images.
Optional - Where person tracking is needed, the system shall correlate faces across sequential frames.
Related capabilities: UC-C1, UC-C2

SECTION D: AUDIO PROCESSING & VOICE INTERACTION

UC-D1: Speech-to-Text Transcription (Audio Input)
Ubiquitous - The system shall convert spoken audio to text for voice command interpretation.
Event-Driven - When the user clicks "Record" in the UI or provides an audio file via CLI, the system shall capture or load the audio.
Event-Driven - When transcription is requested, the system shall invoke WhisperSTT (OpenAI Whisper) to decode the audio file.
Event-Driven - The transcription result shall include the detected text, optional word-level timestamps, and confidence metrics.
Event-Driven - Users can specify model size ('tiny', 'small', 'medium', 'large') to balance speed vs. accuracy.
Event-Driven - Device selection ('cpu' or 'cuda') allows GPU acceleration for faster transcription.
Event-Driven - When transcription completes, the text shall be passed to the agentic layer for command parsing.
Event-Driven - When STT fails (e.g., audio too quiet or corrupted), the system shall report the error and request re-recording.
Optional - Where real-time transcription is needed, the system shall support streaming transcription with incremental results.
Optional - Where background noise is high, the system shall offer pre-processing filters for audio enhancement.
Related capabilities: UC-D2, UC-A1

UC-D2: Text-to-Speech Synthesis (Audio Output)
Ubiquitous - The system shall convert text responses into natural-sounding speech audio.
Event-Driven - When the agent generates a text response (e.g., "detected 3 people") and the user has enabled TTS, the system shall queue the text for synthesis.
Event-Driven - When text_to_speech(text, lang='en', device='cpu', cps=14.0) is invoked, the system shall synthesize audio using WhisperSpeech or Parler TTS.
Event-Driven - The synthesis shall produce a WAV file containing the spoken text with natural prosody and intonation.
Event-Driven - Synthesis parameters shall include: language selection (default: English), device (cpu/cuda), and characters-per-second (cps) for speech speed control.
Event-Driven - When synthesis completes, the system shall return the output file path and duration in seconds.
Event-Driven - Common use cases: mission progress updates ("taking off now"), status alerts ("battery low"), and query results ("found 5 objects").
Event-Driven - When TTS synthesis fails, the system shall log the error and fall back to displaying the text response.

Event-Driven - Users can adjust speech speed via cps parameter (lower = slower, higher = faster speech).
Optional - Where voice cloning is desired, the system shall support reference audio for speaker adaptation.
Optional - Where multiple languages are needed, the system shall support language auto-detection from text.
Related capabilities: UC-D1, UC-A1

UC-D3: Voice Command Interpretation and Execution
Ubiquitous - The system shall interpret voice commands as first-class mission input.
Event-Driven - When the user provides a voice command (via UI or CLI), the system shall transcribe it to text via UC-D1.
Event-Driven - The transcribed text shall be provided to the agentic layer, which parses intent and executes the corresponding tools.
Event-Driven - Voice commands shall support the same mission syntax as text commands (e.g., "fly forward 100 centimeters and look for people").
Event-Driven - When voice command execution completes, the system shall provide audio feedback via UC-D2.
Event-Driven - When voice command is ambiguous, the system shall ask for clarification via TTS.
Optional - Where custom voice commands are needed, the system shall support training phrase recognition for domain-specific terms.
Related capabilities: UC-D1, UC-D2, UC-A1

SECTION E: DIGITAL TWIN SIMULATION & ROS2 INTEGRATION

UC-E1: Gazebo Simulation Environment
Ubiquitous - The system shall provide a digital twin environment (Gazebo) for safe mission testing before real-world execution.
Event-Driven - When the --no-fly flag is provided, the system shall route all drone commands to the Gazebo simulator instead of real hardware.
Event-Driven - The simulator shall model drone physics, including gravity, aerodynamics, and sensor noise, to replicate real-world behavior.
Event-Driven - Drone model (multicopter_velocity_control.sdf) defines the simulated drone structure and physical properties.
Event-Driven - When a movement command is issued in simulation, the system shall update drone position according to physics engine calculations.
Event-Driven - Simulation shall support virtual obstacle spawning for path planning validation.
Optional - Where sensor simulation is needed, the system shall provide synthetic camera feeds with realistic degradation (blur, noise, occlusion).
Related capabilities: UC-B3, UC-A4

UC-E2: ROS2 Core and Bridge Integration
Ubiquitous - The system shall integrate with ROS2 messaging infrastructure for distributed drone operations.
Event-Driven - When the drone launches, the ROS2 core (roscore equivalent) shall initialize and establish DDS networking.
Event-Driven - ROS2 topics shall carry drone telemetry (sensor data), user commands (twist velocities), and system state.
Event-Driven - ROSBridge shall provide a WebSocket interface used by the backend to publish to ROS2/Gazebo.
Event-Driven - When a drone command is sent from the web frontend, it shall traverse the frontend → FastAPI backend → ROSBridge → ROS2 DDS → Gazebo or hardware driver.
Event-Driven - When drone telemetry is updated, it shall be published to ROS2 topics and forwarded to connected clients via the backend.
Optional - Where multiple drones are needed, the system shall support ROS2 namespacing for independent drone

control.
Related capabilities: UC-E1, UC-B1

UC-E3: Perception Tools Integration with MCP
Ubiquitous - The system shall expose all perception tools (vision, audio) through the Model Context Protocol (MCP) server.
Event-Driven - When the mission agent starts, it shall connect to the MCP server (entrypoint: main_mcp_with_stt.py) and discover available tools.
Event-Driven - When the agent calls a tool (e.g., detect_objects), the MCP client shall marshal the request to the server and return structured results.
Event-Driven - Tool results shall be interpreted by the LLM agent for next-step reasoning.
Event-Driven - When tool execution fails, the MCP server shall return a structured error message for agent error handling.
Optional - Where custom tools are added, the system shall auto-register them with the MCP server via FastMCP decorators (e.g., @main_mcp.tool).
Related capabilities: UC-C2, UC-C3, UC-C4, UC-C5, UC-D2

==========

SECTION F: WEB FRONTEND & USER INTERFACE

==========

UC-F1: Web-Based Mission Control Interface
Ubiquitous - The system shall provide a web interface for mission control and real-time monitoring.
Event-Driven - Frontend connects to FastAPI WebSockets for video and chat/mission features. The backend can send commands to ROSBridge for simulation or hardware.
Event-Driven - The interface shall display live video feed from the drone camera with a heads-up display (HUD) showing battery, height, temperature, and mission status.
Event-Driven - Users shall be able to enter text commands or voice commands directly in the interface.
Event-Driven - When a command is submitted, the frontend shall forward it to the backend mission agent for execution.
Event-Driven - Real-time telemetry updates shall be received via ROSBridge and displayed on the HUD.
Event-Driven - When drone connection status changes, the interface shall update connection indicators and disable/enable controls appropriately.
Optional - Where VNC viewer access is needed (Gazebo desktop display), the system shall provide VNC endpoint for remote desktop viewing.
Related capabilities: UC-B1, UC-A1, UC-A3

==========

SECTION G: SAFETY & CONSTRAINTS

==========

UC-G1: Safety Constraints and Bounds Checking
Ubiquitous - The system shall enforce physical and operational safety constraints on all drone commands.
Event-Driven - When a movement command exceeds workspace bounds or battery is below threshold, the system shall reject the command with a clear error.
Event-Driven - All linear movement distances shall be validated to be between 20-500 centimeters.

Event-Driven - All rotation angles shall be validated to be between 1-360 degrees.
Event-Driven - Battery-based constraints: At 20%, system warns user; at 10%, automatic emergency landing is triggered.
Event-Driven - Height-based constraints: Drone shall not exceed maximum safe altitude.
Optional - Where GPS is available, the system shall provide geofencing to prevent flight outside designated areas.
Related capabilities: UC-B2, UC-B3, UC-B4

UC-G2: Error Handling and Recovery
Ubiquitous - The system shall gracefully handle errors and provide clear feedback to the user.
Event-Driven - When any tool execution fails, the system shall catch the exception, log it, and return an error message to the agent.
Event-Driven - When drone connection is lost, the system shall attempt automatic reconnection and alert the user.
Event-Driven - When an agent encounters an invalid command, it shall ask for clarification rather than executing undefined behavior.
Event-Driven - When a critical error occurs (e.g., loss of telemetry), the system shall trigger emergency landing.
Optional - Where error recovery is possible, the system shall retry failed operations with exponential backoff before alerting the user.
Related capabilities: UC-B1, UC-B6

==========