📄 Scope of Work (SoW): Software Development Team

Project Title: Intelligent Wagon Load Analysis & Reporting System

Timeline: 2 Months (8 Weeks)

Team Focus: Data Acquisition, Processing, Cloud Integration, Analytics, and Reporting

---

🧠 Project Objective

Develop a cloud-connected software system to:

- Acquire real-time LiDAR + IMU data via Ethernet
- Process it to analyze material balance, volume, weight estimation
- Count the number of wagons
- Generate and deliver automated PDF reports for each train trip

---

📌 Key Responsibilities

1. System Integration & Data Acquisition

- Interface with:
  - **LiDAR sensor** via Ethernet (e.g., UDP/TCP or ROS-based)
  - **IMU** (via serial/Ethernet)
  - **Flight Computer** (e.g., Jetson, Raspberry Pi, or custom)
- Time-sync and store raw LiDAR & IMU data with metadata (train ID, time, etc.)
- Develop a **Data Acquisition Module** with:
  - Live feed monitor
  - Local buffer and push-to-cloud functionality
  - Data integrity check (e.g., checksum, packet loss detection)

---

2. Data Preprocessing

- Clean and filter raw LiDAR point clouds
- Align LiDAR and IMU data for stable frame of reference
- Apply basic transformations to align wagons and filter ground plane
- Generate structured datasets:
  - Wagon-wise point clouds
  - Bounding boxes
  - Time stamps and unique identifiers

📄 Scope of Work (SoW): Software Development Team

3. Analytics & Algorithms

(Your team will provide training data; software team implements these features.)

- **Wagon Balancing**:
    - Analyze symmetry of material distribution inside wagons
    - Output left-right/top-bottom deviation in centimeters or degrees
- **Volume Estimation**:
    - Use 3D surface reconstruction (e.g., Poisson Surface Reconstruction or Convex Hull) to calculate volume
    - Validate against known empty wagon geometry
- **Weight Estimation**:
    - Convert volume to estimated weight using predefined material density
    - Add compensation factor based on IMU tilt (if wagon is tilted)
- **Wagon Detection & Counting**:
    - Segment wagons from LiDAR stream using edge detection or sudden gap detection
    - Assign each wagon a unique identifier
    - Handle edge cases like partial scans or overlapping wagons

4. Cloud Integration

- Real-time or batch upload of:
    - Raw data (compressed format)
    - Processed metrics (JSON or CSV)
    - Final report (PDF)
- Use secure API to:
    - Push data to cloud (e.g., AWS S3, Firebase, or custom server)
    - Retrieve analytics and visualize status

5. Automated PDF Report Generation

- For each train passing:
    - Generate a single PDF containing:
        - Timestamp, Train ID
        - Wagon count
        - Per-wagon metrics:
            - Balance info
            - Volume (m^3)
            - Estimated weight (tons)
        - Summary graphs/charts

        - Anomalies or alerts (e.g., overload, imbalance)
- Include:
    - Auto-signed authority footer
    - QR code for verification
    - Cloud link embedded

6. Dashboard (Optional but Recommended)

- Simple web dashboard for:
    - Live feed of scanning process
    - History of scanned trains
    - PDF archive with download links
    - Analytics summary (weekly/monthly trends)

7. Testing & Validation

- Dry-run testing with provided datasets
- Stress testing for real-time data flow
- Integration test for complete pipeline
- Edge-case testing (incomplete scans, tilted wagons, rain/obstruction)

🛠️ Tools & Tech Stack (Suggested)

Layer	Technology
LiDAR Interface	ROS / Python + UDP Socket / C++
Data Processing	Open3D / PCL / NumPy / Pandas
Analytics & Volume	SciPy / PyTorch (for AI)
Backend + Cloud Sync	Python (Flask/FastAPI) + AWS/Firebase
Report Generation	Python (ReportLab / WeasyPrint)
Dashboard (Optional)	React / Vue + Node.js / Django

📅 Timeline: Software Development Milestones

Week	Deliverable
1	Sensor interfacing modules complete
2	Data acquisition & storage system working

📄 Scope of Work (SoW): Software Development Team

Week	Deliverable
3	Basic LiDAR+IMU preprocessor ready
4	Wagon segmentation + initial analytics prototype
5	Volume & weight estimation algorithms integrated
6	Balancing computation + PDF format finalized
7	Cloud upload API + report generator
8	Integration test with your live dataset
9	Bug fixes + edge case handling
10	Final deployment & documentation

📋 Deliverables

1. Complete data acquisition & processing pipeline
2. Wagon detection + metrics calculation
3. Fully functional report generation system
4. Cloud upload and retrieval interface
5. Codebase with documentation
6. Deployment scripts and testing logs