Lumion Robots Company

Robotic Lawnmower Project

Pinja Varis, ZJAYTS25S1
Aytunc Bakir, YTC25S1
Dineshkumar Damodharan, YTI25S1
Javad Hosseini, YTP25S1

Project report
Modern Management of IT
4.12.2025
Master of Engineering

Jyväskylän ammattikorkeakoulu
University of Applied Sciences

Table of Contents

1 Introduction

In this project, we are developing a system around a Lumion robotic lawnmower that can collect, send, and analyze data in real time and proceed with mowing with the help of sensors, AI and commands. The goal is to develop a solution that combines automation, safety, and usability. This report will contain dedicated chapters for Full Stack Development, Artificial Intelligence & Data Analytics, Robotics, and Cyber Security.

2 Cyber Security

2.1 Purpose and Scope

Security is a base requirement for the Lumion Robots robotic lawnmower, not just add-on and afterthought. This product is not just a piece of hardware and firmware; it is a complex, network-connected Internet of Things (IoT) system. It communicates with the clouds, receives, sends and gathers data and is controlled by a mobile app. It interacts with the physical world. Hence a security failure has intense and broad consequences far beyond a typical IT data breach:

Physical Safety: This is our top priority. A hijacked robot could have its safety boundaries or geofencing disabled, allowing it to enter a neighbor's yard, a street, or a swimming pool area. An attacker could deliberately disable obstacle-avoidance sensors, incurring property damage or, in the worst scenario, physical harm to people or pets.

Data Privacy: The integrated robot and cloud platform store PII, precise GPS-based property maps, and operational schedules, constituting a close "pattern of life" dataset. Once this information is captured, attackers can determine that the user is not at home. The failure to protect this sensitive data is a critical breach of customer confidence and exposes the company to severe penalties under privacy regulations such as the GDPR.

Service Integrity: The robot fleet relies on our cloud backend for Over-the-Air (OTA) updates and remote commands. A compromised OTA update is a catastrophic risk; an attacker could push malicious firmware to the entire fleet, either "bricking" thousands of expensive devices and forcing a massive, costly recall, or creating a "botnet" that uses the robots' collective processing power for large-scale criminal activities (like DDoS attacks) without the user's knowledge and consent.

Brand Reputation: As a visionary in the advanced home robotics market, Lumion's brand reputation is built on trust, quality, and reliability. A single significant security incident would be widely publicized and could irreversibly damage this trust, driving customers to competitors and jeopardizing the company's market position.

2.2 Security Goals

Our cybersecurity strategy is built on protecting the Confidentiality, Integrity, and Availability (CIA) of our systems, with the addition of Safety and Privacy as a paramount and non-negotiable goal.

Confidentiality: We will protect all user data (PII, property maps, schedules) from unauthorized access. This applies to data in transit (moving between the robot, cloud, and app), at rest (stored on the robot's local storage or in our cloud databases), and in use. This protection extends to both external attackers and internal unauthorized access.

Integrity: We will ensure that all data, commands, and software are accurate, authentic, and have not been tampered with. This means ensuring that commands sent from the app (e.g., "start," "stop," "update boundary") are the exact commands received and executed by the robot. It also means protecting the integrity of all firmware and software updates to prevent malicious modification.

Availability: We will ensure that the robot, mobile app, and cloud control services are operational, resilient, and responsive when the user needs them. This includes protecting against denial-of-service (DoS) attacks that could flood our servers or the robot itself, and ensuring a user can always issue a critical command, like "emergency stop."

Safety: We will ensure the robot's physical operation does not pose a threat to people, pets, or property, even when the system is under a cyberattack. This goal is vital; security controls must be designed to fail-safe, (e.g., the robot stops and locks down if it detects malicious commands) to prevent real-world harm.

Privacy: We will ensure the management of personal data, from its acquisition to its use and ultimate retention, will be governed by stringent legal obligations and ethical principles.

2.3 Governance & Compliance

2.3.1 ISMS and ISO 27001

To manage our security posture systematically, we will adopt the principles of an Information Security Management System (ISMS), as outlined in the ISO 27001 standard. An ISMS is a systematic, risk-based approach to managing an organization's information security. For Lumion Robots, this is not just about checklists; it is about creating a culture of security. It provides us with a formal, repeatable process to:

- identify and assess security risks on an ongoing basis.
- implement, document, and manage controls to mitigate those risks.
- continuously monitor, review, and improve our security posture as new threats emerge, and our products evolve.
- demonstrate our commitment to security to customers and regulators, providing a key competitive advantage.

2.3.2 Risk Assessment & Treatment Plan

This table identifies the most relevant, high-impact risks for the robotic lawnmower project. "Likelihood" is assessed based on attacker motivation, technical difficulty, and the size of the attack surface. "Impact" is assessed as a blend of financial, reputational, data privacy, and physical safety consequences.

- **Critical:** Unacceptable risk. It must be addressed immediately.
- **High**: Serious risk. Must be addressed as a high priority.
- **Medium**: Manageable risk. Should be addressed in the next development cycle.
- **Low:** Acceptable risk. Should be monitored and addressed if resources permit.

Asset	Threat	Vulnerability		
1. Customer PII & Location Data (Stored in the cloud)	An attacker infiltrates the cloud and compromises the entire customer database.	An insecure API without proper authorization checks allows an attacker to query other users' data.	**Impact**	Critical
			Likelihood	Medium
			Risk Level	Critical
Mitigation: Implement strict cloud IAM policies (principle of least privilege). Encrypt all database contents at rest (AES-256) and in transit (TLS 1.3). Utilize cloud-native threat detection (e.g., Microsoft Sentinel). Conduct regular vulnerability scans and third-party penetration tests on the cloud environment.				
2. Robot Operational Safety (Physical device)	A remote attacker installs malicious firmware that disables security features such as blade shutdown or boundary detection	Since there is no mandatory cryptographic signature verification during the software update process, the installation of a tampered update is allowed.	**Impact**	High
			Likelihood	High
			Risk Level	Critical
Mitigation: All firmware updates must be digitally signed by the company. Implement a secure bootloader on the robot that verifies this signature before executing any code.				
3. The Physical Robot (Hardware asset)	The robot is physically stolen from the owner's property.	Lack of effective, built-in anti-theft deterrents.	**Impact**	Medium
			Likelihood	High
			Risk Level	High
Mitigation: Integrate GPS tracking with geofencing alerts sent to the owner's app. Require a unique PIN code to operate the mower if it is lifted or moved outside its designated area. Equip the device with a loud, deterrent alarm.				
4. Service	An attacker launches a Denial of	The robot's Bluetooth/Wi-Fi	**Impact**	Medium

Availability (User's ability to control the robot)	Service (DoS) attack, overwhelming the robot with connection requests and preventing the owner from connecting.	communication stack does not handle excessive connection requests gracefully.	**Likelihood**	Medium
			Risk Level	Medium
Mitigation: Implement rate limiting in the robot firmware to reject an anomalous number of connection attempts from a single source. Ensure a physical "Emergency Stop" button is present on the robot that overrides all software commands.				
5. Home Network Integrity (User's local network)	The robot is compromised and used as a pivot point to attack other sensitive devices (laptops, cameras) on the owner's Wi-Fi network.	The robot's operating system contains known, unpatched vulnerabilities (e.g., an out-dated Linux kernel or network service).	**Impact**	High
			Likelihood	Medium
			Risk Level	High
Mitigation: Maintain a Software Bill of Materials (SBOM) and establish a rapid Over-the-Air (OTA) update process to patch vulnerabilities. Advise users in the manual to segment their home network by placing IoT devices on a separate guest network.				
6. User Privacy (Location and schedule data)	A third-party analytics company links "anonymized" mowing schedule data back to specific individuals, revealing patterns of when they are away from home.	Data minimization principles are not followed; excessive granular location and timing data are collected and stored indefinitely.	**Impact**	High
			Likelihood	Medium
			Risk Level	Critical
Mitigation: Anonymize data by reducing precision (e.g., rounding GPS coordinates, and aggregating time data). Establish strict data retention policies to automatically delete detailed logs after 30 days. Provide users with clear, opt-in consent for any data usage not essential for core functionality				

Table - Risk Assessment

2.3.3 Information Security Policy

1.0 Introduction

This policy establishes formal information security objectives for the Lumion Robots. It is a "living document" that will be reviewed annually. Its purpose is to safeguard the information assets of Lumion Robots, its employees and customers from all threats, whether internal or external, deliberate or accidental.

2.0 Scope

This policy applies to all employees, contractors, and third-party vendors involved in the project. It covers all information assets, including: the robot's onboard firmware, all related mobile applications, all development and testing environments, the production cloud backend infrastructure, and all customer data processed or stored by the system.

3.0 Security Objectives

Our core security objectives are to ensure the **Confidentiality**, **Integrity**, **Availability**, and **Safety** of our product and its associated data and our customers **Privacy**.

- All customer data is classified as **confidential**. This data must be protected at rest and in transit using industry-best-practice encryption and access control.

- Software and data integrity must be maintained to ensure reliable and safe robot operation. All code and commands must be verifiable and protected from unauthorized modification.

- Systems must be resilient to ensure high availability for our customers. This includes robust, scalable architecture and defenses against denial-of-service attacks.

4.0 Key Principles

Security by Design: Security is a non-negotiable component, not an add-on and afterthought. It must be integrated into every phase of the development lifecycle, from initial concept to end-of-life. This means security requirements are defined first, and threat modeling is a standard part of our design process.

Privacy by Design: We will proactively embed privacy into our systems, adhering to GDPR principles of data minimization. We will only collect the minimum data necessary for the robot's function, and we will be fully transparent with users about what data is collected and how it is used.

Principle of Least Privilege: All systems, services, and users (including developers) will be granted only the minimum level of access (permissions) required to perform their function, and only for the duration necessary. For example, a developer's credentials should not work on a production database.

Zero Trust Architecture: We will "never trust, always verify." Home Wi-Fi networks are often insecure, so we must treat them as a vulnerability point. All network communication, whether internal or external, must be authenticated, authorized, and encrypted.

5.0 Responsibilities

CTO: The executive sponsor for product security, responsible for providing necessary resources, aligning security goals with business objectives, and accepting residual risk.

Head of Software Engineering: Responsible for implementing and enforcing secure development practices (SSDLC), managing the integration of security tools into the CI/CD pipeline, and ensuring teams are trained.

Cybersecurity Lead: Responsible for defining security requirements, conducting risk assessments, guiding security tool implementation, acting as the primary security consultant for development teams, and leading incident response.

All Developers & Engineers: Responsible for writing secure code, following security best practices, promptly reporting any identified vulnerabilities, and actively participating in remediating vulnerabilities found in their code.

2.4 Secure Development Processes

To integrate security into our software development, we will adopt a Secure Software Development Lifecycle (SSDLC). This means security is not a final checkpoint, but a continuous, iterative part of our process:

Requirements: We will define specific security and privacy requirements beside functional ones at the beginning of a feature. This includes technical requirements (e.g., "All user PII must be encrypted with AES-256") and regulatory requirements (e.g., GDPR compliance).

Design: We will conduct formal threat Modeling during the design phase to identify potential security flaws before any code is written. By using frameworks like STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege), we can brainstorm "what-if" attack scenarios and design mitigations from the ground up.

Implementation: Developers will follow secure coding standards (e.g.,OWASP Secure Coding Practices, OWASP IoT Top 10) and use verified, approved libraries. Security-focused code reviews will be a mandatory part of our pull request process, checking for common pitfalls like injection flaws or broken authentication.

Testing: We will perform comprehensive, automated security testing. This includes Static Application Security Testing (SAST) to scan source code for flaws, Dynamic Application Security Testing (DAST) to test the running application's APIs, and Software Composition Analysis (SCA) to check our open-source dependencies for known vulnerabilities. This will be supplemented by periodic manual penetration testing by external experts.

Deployment & Maintenance: We will use secure configurations for all cloud services (scanned using Infrastructure as Code tools) and have a formal incident response and patch management plan to handle vulnerabilities discovered after release. This includes a public-facing vulnerability disclosure program.

This SSDLC will be automated using a DevSecOps approach. Given that Lumion already uses GitLab and CI/CD, we will integrate security tools directly into this pipeline. This "shifts security left," provides developers with immediate feedback on vulnerabilities within their normal workflow. This is not just a tooling change; it's a cultural shift that breaks borders between Development, Security, and Operations, making security a shared responsibility. This approach empowers developers to be the first line of defense, finding and fixing security issues when they are cheapest and fastest to resolve.

Key principles for this project include:

Zero Trust: As stated in our policy, we will not trust any network by default. Home networks are outside our control. Therefore, all communication from the robot to the cloud, and from the app to the cloud, must be mutually authenticated (e.g., using client-side and server-side certificates, known as mTLS) and encrypted.

OTA Update Security: The Over-the-Air update process is a critical attack vector, as a single compromised update could hijack the entire fleet. All firmware updates must be cryptographically signed by a secure, offline, hardware-based key. The robot must validate this signature in a secure, trusted environment on the device before the update is even decrypted or applied, making it impossible to load malicious firmware.

2.5 Tools for Secure Development

No single tool can find all vulnerabilities. We will use a multi-layered, "defense-in-depth" toolset to support our SSDLC, integrating directly with Lumion's existing technology stack (GitLab, private cloud, diverse programming languages).

TOOLS	
GitLab Ultimate	**DevSecOps Platform**
Provides built-in SAST, DAST, Dependency, and Container Scanning directly within the CI/CD pipeline. This gives developers immediate feedback in their merge requests, correlating vulnerabilities directly with the code commit that introduced them.	
Snyk	**Software Composition Analysis (SCA), (SAST)**
Finds, prioritizes, and helps fix vulnerabilities in the open-source libraries used in our Python, Go, and JavaScript code. It provides actionable remediation advice, often with one-click pull requests to apply suggested fixes. Developer-focused platform that includes SAST, SCA, and Infrastructure as Code (IaC) security.	
OWASP ZAP	**Dynamic Application Security Testing (DAST)**
Actively scans our running web applications and cloud APIs (which control the robot) for common vulnerabilities. This simulates real-world attacks against our API endpoints *before* they are exposed to the public.	
Ansible	**Configuration Management**
Enforces secure, hardened configurations (e.g., disabling unused ports, enforcing file permissions) consistently across all cloud servers. This provides a repeatable, auditable "Infrastructure as Code" (IaC) process, which prevents manual configuration errors.	
Prometheus & Grafana	**Security Monitoring**
Used for monitoring and visualizing security-relevant metrics in real-time. We will create dashboards and alerts for security-critical events to detect potential attacks as they happen.	
Azure Key Vault	**Secrets Management**
It is a cloud service from Microsoft Azure used for securely storing and managing secrets used by cloud applications and services. It provides a centralized, secure location for three main types of sensitive information: **Secrets:** Passwords, database connection strings, API keys, tokens, and other sensitive configuration data. **Keys:** Cryptographic keys (both software-protected and **Hardware Security Module (HSM)-protected**) used for data encryption, decryption, signing, and verification.	

Certificates: SSL/TLS certificates and their private keys for secure communication.	
HashiCorp Vault	Secrets Management
Securely stores, manages, rotates, and audits access to all secrets (API keys, database passwords, TLS certificates) instead of hardcoding them in code or config files. This prevents secrets from being leaked in our Git repositories.	

Table - Tools for Secure Development

2.6 Minimum Security Controls

Here are the essential security measures that must be in place for this robotic lawnmower project to ensure a baseline of security and user trust. These controls directly map to our highest-priority risks.

Mandate Multi-Factor Authentication (MFA): Enforce MFA for all customer accounts (via the mobile app) and for all internal employees accessing development (GitLab) and cloud (AWS/Azure/GCP) environments. This is the single most effective control to prevent account takeover resulting from stolen or weak credentials.

Full-Chain Data Encryption: All user data (PII, property maps) must be encrypted **at rest** (on the robot's local storage and in the cloud database) and **in transit** (between the robot, cloud, and app) using strong, modern standards (e.g., AES-256, TLS 1.3). This ensures that even if data is intercepted or the physical hardware is stolen, the data remains unreadable.

Secure, Signed OTA Updates: All firmware and software updates must be cryptographically signed with a secure, offline key. The robot must validate this signature before applying for any update. This control directly mitigates the catastrophic, fleet-wide risk of a malicious firmware-based attack.

Implement a Zero Trust Network Model: Treat all networks (especially the user's home Wi-Fi) as untrusted. All communication between the robot, cloud, and mobile app must be mutually authenticated (e.g., via mTLS) and encrypted. This protects the robot from other potentially compromised devices on the user's local network.

Automated Vulnerability Scanning in CI/CD: Integrate automated security testing (SAST & SCA) directly into the GitLab pipeline. This "shifts security left," empowering developers to find and fix common security bugs early in the lifecycle, which is significantly faster, cheaper, and safer than fixing them in production.

3 Robotics

The Robotics section defines the physical hardware, its control logic, and the essential data pipeline for the Lumion robotic lawnmower. Our focus is on ensuring the mower's autonomous operation is reliable, safe, and fully integrated with the cloud-based management system (Full-Stack) while providing rich data for predictive intelligence (AI/DA).

3.1 Data from Robots: The Foundation for Smart Management

The lawnmower functions as a sophisticated **IoT device**, generating a continuous stream of operational and environmental telemetry. This data is the critical input for the entire platform, enabling remote control, preventative maintenance, and product evolution.

3.1.1 Sensor Data Categorization

To ensure effective utilization by the AI/DA and Full-Stack teams, the generated data falls into three core categories:

- **Positional and Navigation Data:** The mower uses **GPS/GNSS** for accurate geofencing and path planning, supplemented by **IMU (Inertial Measurement Unit)** for dead reckoning and tilt detection, and **Odometry** from wheel encoders for local movement tracking.
- **Operational Performance Data:** This includes **Battery Management System (BMS) data** (level, voltage, current) and **Motor Feedback**. Monitoring the **Blade Motor Current/RPM** is vital, as this unique data point allows the AI team to estimate grass density and predict blade wear, directly enabling **Predictive Maintenance**.
- **Environmental and Safety Data:** Sensors like **Ultrasonic**, **Bump**, and **LiDAR** handle immediate obstacle avoidance. **Lift/Tilt sensors** ensure safety by immediately cutting power to the blade upon physical disturbance.

3.1.2 Lawnmower Telemetry Fields

The data fields below are specifically chosen to align with the Full-Stack team's defined database schema, ensuring seamless ingestion into the telemetry and events tables.

Data Category	Field Name	Data Type	Unit/Format	Description
Location & Time	ts, lat, lon, speed_mps	Time/ Float/ Numeric	ISO 8601, Degrees, m/s	Real-time position and velocity for mapping and tracking.
Battery Health	battery_pct, battery_voltage, charging_status	Int/ Numeric/ Text	%, Volts, State	Critical status for scheduling and energy efficiency reporting.
Performance	blade_rpm, blade_motor_current_ma, total_mowing_hours	Int/ Int/ Numeric	RPM, mA, Hours	Direct input for AI-driven maintenance models and performance reporting.

Status/Faults	status, activity, error_code, firmware_version	Text/ Text/ Text	MOWING, ERROR, E003	Operational state and diagnostic information for remote support.
Safety	lift_tilt_sensor_active, rain_sensor_active	Boolean	True/False	Immediate trigger for safety stop and environmental protection.

3.2 Interfaces and Integration with Other Systems

Effective remote management depends on reliable communication interfaces. We employ a hybrid protocol strategy to balance real-time control, data volume efficiency, and security.

3.2.1 Device-to-Cloud (D2C) and Command-and-Control (C2D)

The communication architecture is designed to handle continuous telemetry (D2C) and instantaneous user commands (C2D):

- **D2C Telemetry:** We primarily use **MQTT** over a cellular network. MQTT's lightweight, publish/subscribe model is ideal for continuous, low-bandwidth sensor data, reducing power consumption compared to constant HTTP requests. Secure **HTTPS POST** is maintained as a fallback for large data uploads (e.g., diagnostic logs).
- **C2D Commands:** Commands from the Full-Stack application (e.g., START, PAUSE, UPDATE_GEOFENCE) are delivered in real-time via an **MQTT Subscription**. The mower maintains a persistent connection to the cloud, allowing commands to be "pushed" immediately, which is essential for low-latency actions like an **EMERGENCY_STOP**. Command **Polling** on the API is used only as a last-resort fallback.

3.2.2 Enterprise System Integration

To maximize the value of the collected data across the organization, the backend aggregates telemetry for integration with key enterprise systems:

- **ERP (Enterprise Resource Planning):** Data on usage hours and predicted component life (**Predictive Maintenance**) is processed by a backend microservice and pushed to the ERP via **REST/SOAP APIs**. This enables automated inventory management for spare parts (e.g., replacement blades) and streamlines the supply chain.
- **CRM (Customer Relationship Management):** Critical alerts, such as an E003: Blade Jam or a theft attempt, trigger webhooks from the PostgreSQL events table to the CRM system. This allows for **Proactive Customer Support**, creating a support ticket automatically before the customer calls.

3.3 Robot Control and Connectivity Selection

The lawnmower's mobile, outdoor environment necessitates robust wireless connectivity, which is a major design and business decision.

3.3.1 Wireless Solution Analysis and Justification

We compared Wi-Fi, LPWAN, and Cellular (4G/5G). The **Public Cellular (4G LTE/5G)** network is the definitive choice for the Lumion Mower:

Requirement	Justification for Cellular (4G/5G)	Management Impact
Reliability & Mobility	Provides guaranteed service quality and nationwide coverage, essential for remote control and tracking (e.g., when the mower is stolen or out of home Wi-Fi range). Wi-Fi is inadequate for this scope.	**Reduced Support Costs:** Higher reliability significantly cuts down on connectivity-related service calls.
Bandwidth	Supports periodic telemetry, low-latency C2D control, and large **Over-The-Air (OTA) firmware updates**.	**Enables Product Evolution:** Guaranteed bandwidth allows for richer data features and continuous software improvement.
Business Model	The use of a SIM module and monthly data costs translate directly into a recurring **OPEX** that must be incorporated into the product's pricing or a subscription model.	**Strategic Pricing:** Forces the business to adopt a subscription-based revenue stream aligned with modern IoT service offerings.

3.4 Robot Safety: Principles and Risk Management

Safety is not just a feature but a design principle that requires both hardware and software intervention.

3.4.1 Core Safety Mechanisms

1. **Safety-Rated Monitored Stop (SRMS):** The immediate priority. The **Lift/Tilt Sensors** are hardwired to an independent safety controller that triggers an instantaneous, non-negotiable power cutoff to the blade motor, preventing injury during physical disturbance.
2. **Safe Separation:** Software-controlled boundaries defined by the Full-Stack app (**Geofencing**) issue automatic PARK or STOP commands if the mower breaches its safe area.

3.4.2 Addressing Security and Operational Risks

The robotics component presents specific risks that require mitigation, providing clear action items for the Cyber Security team:

Risk Focus	Example Risk	Mitigation Strategy (Robotics and Firmware)

Theft & Unauthorized Use	The mower is lifted and moved out of range.	**Physical Alarm/Locking:** GPS-based geofence breach triggers a physical alarm; requires authenticated **C2D token** via the app to unlock the drive system.
Hacked Commands	Malicious injection of START or MOVE commands.	**Secure Communication:** Enforce **TLS 1.2+** encryption on all MQTT/HTTPS command channels. **Device Authentication:** Use unique device certificates for all cloud connection handshakes.
Firmware Integrity	Loading unauthorized firmware.	**Secure Boot & OTA:** Use digital signatures and secure boot processes to validate the authenticity of all Over-The-Air (OTA) firmware updates before execution.

3.5 Summary of Robotics Contribution

The Robotics section delivers the functional blueprint for the lawnmower's operation, explicitly defining the **high-value data streams** for AI/DA, the **MQTT/Cellular protocol stack** for Full-Stack integration, and the **core safety requirements** for Cyber Security. This integration ensures the Lumion system is managed efficiently and reliably.

4 Artificial Intelligence & Data Analytics

The AIDA section defines the tasks performed using data for the Lawn Mower. In this project, we will also clean the data, perform exploratory data analysis (EDA), and include some visualization metrics.

For AIDA, I’ve chosen the following tasks:

- Battery & Runtime Prediction
- Path Optimization & Navigation
- Environmental Awareness

We are using synthetic data for this project, as real-time datasets are not available. The synthetic data is generated using GPT. At the end, we will also apply some machine learning techniques to make predictions (for example: Runtime Prediction).

4.1 Tasks

4.1.1 Battery & Runtime Prediction:

In this task, we will analyze the data obtained from the battery to estimate the remaining power stored.

- Based on this data, we will calculate the remaining runtime for the lawn mower.
- While charging, we can estimate the remaining time required to reach 100% charge.

4.1.2 Path Optimization & Navigation:

Using the dataset that includes path information, obstacles, and the target area, we will attempt to optimize the path to minimize the time taken to cover the entire target area—thereby indirectly reducing battery consumption.

4.1.3 Environmental Awareness:

We will integrate weather data to prevent the lawn mower from operating during rain or extreme heat conditions.

4.2 Data Sets used for this project

All the data sets used for this project are synthetic data created using AI assistant. We tried to cover the data required to achieve the tasks taken for this project as part of AIDA. We decided to have separate data sets for three tasks chosen to ensure clear data analysis and visualization. It will also simplify the data cleaning process and filtering of the columns required for visualization. Each dataset has around 200 rows. Datasets will be uploaded to the SharePoint inside a separate folder.

4.2.1 Battery & Runtime/Charging Dataset

Columns

- timestamp (minute/hour of day)
- battery_level (%)
- voltage (V)
- current_draw (A)
- charging_current (A)
- charging_status (Discharging / Charging)
- temperature (°C)
- runtime_remaining (min) (target when discharging)
- time_to_full_charge (min) (target when charging)

Sample Data

```
timestamp,battery_level (%),voltage (V),current_draw (A),charging_current (A),charging_status,temperature (°C),runtime_remaining (min),time_to_full_charge (min)
2025-01-01 00:00:00,99.0,11.13,2.16,,Discharging,24,75.0,
2025-01-01 00:05:00,96.6,11.52,2.90,,Discharging,25,53.0,
2025-01-01 00:10:00,101.2,12.64,2.89,,Discharging,25,79.0,
2025-01-01 00:15:00,98.8,10.56,2.16,,Discharging,28,28.0,
2025-01-01 00:20:00,97.4,11.25,2.05,,Discharging,29,69.0,
```

4.2.2 Path Optimization & Navigation Dataset

Columns

- path_id
- grid_x (X-coordinate)
- grid_y (Y-coordinate)
- obstacle_detected (Yes/No)
- distance_to_target (m)
- time_taken (s)
- battery_consumed (%)

Sample Data

```
path_id,grid_x,grid_y,obstacle_detected,distance_to_target (m),time_taken (s),battery_consumed (%)
P000,10,6,Yes,25,26,2
P001,0,2,Yes,6,29,4
P002,7,11,Yes,15,27,1
P003,12,16,No,43,19,4
P004,11,5,No,45,53,1
```

4.2.3 Environmental Awareness Dataset

Columns

- timestamp
- location
- season
- temperature (°C)
- humidity (%)
- wind_speed (m/s)
- precipitation (mm)
- uv_index
- cloud_cover (%)
- weather_label (Rain / ExtremeHeat / Normal)

Sample Data

```
timestamp,location,season,temperature_C,humidity_pct,wind_speed_mps,precipitation_mm,uv_index,cloud_cover_pct,weather_label
2025-07-01T00:00:00,"Espoo, Finland",Summer,19.6,66.5,4.67,2.76,8.3,40.3,Normal
2025-07-01T01:00:00,"Espoo, Finland",Summer,16.2,62.9,4.48,0.0,5.3,35.7,Normal
2025-07-01T02:00:00,"Espoo, Finland",Summer,9.3,61.5,1.68,0.0,6.5,26.8,Normal
2025-07-01T03:00:00,"Espoo, Finland",Summer,11.4,67.2,0.94,0.0,5.2,47.2,Normal
2025-07-01T04:00:00,"Espoo, Finland",Summer,16.2,62.9,2.42,0.9,8.8,44.7,Normal
```

4.2.4 Exploratory Data Analysis, Prediction and Visualization

All the data sets will be read in pandas and cleaned before any processing. This whole process of EDA and visualization heavily relies on the AI assistant. But the prompts are created by us based on the requirements. Responses are carefully reviewed and altered to achieve the output required. If the responses from the AI assistant are not satisfactory, then the prompts are re-created until the response is satisfied.

Data analysis on Battery dataset

Based on the data we get from the battery logs (Natarajan, A. n.d.), we can process the information to perform EDA. In the example, we have done the basic EDA, and from the output we can analyse the battery health by comparing the battery charge percentage and the runtime of the mower (Ryu, M. n.d.).

We can also predict the remaining runtime based on the history of the current mower, which we change dynamically depending on battery health. For charging, we can predict the time remaining to reach full battery percentage.

We can perform additional analysis that is important for mower performance. For example, from the battery logs we can analyse the voltage and current draws during runtime; if there is any abnormal behaviour, we can notify the user to take the mower to the service center. By doing this, we can identify issues with the battery or other electronic components related to the battery at an early stage.

We can also analyse the temperature of the battery during charging and discharging; if something is abnormal, we can stop the function and notify the user. This can help avoid accidents caused by battery overheating.

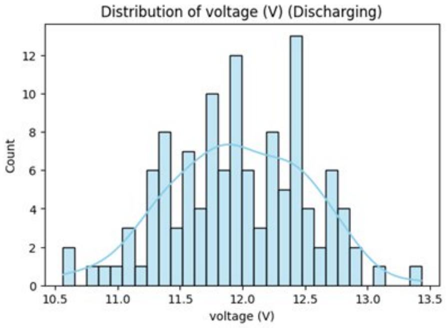
Data Analysis on Sample Battery Data set:

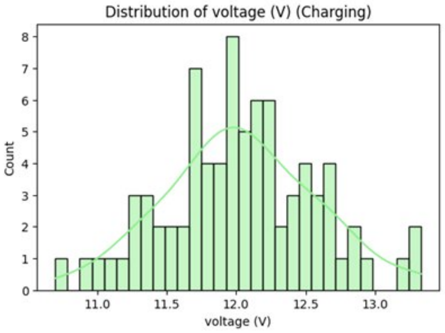
We performed the basic EDA and got the output displays the number of rows we got in the datasets and also to understand the voltage, current draw, Temperature.

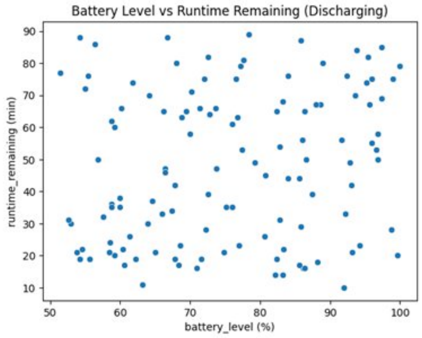
For prediction we used models such as Linear regression, random forest, Gradient boosting, support vector regression, Neural network (MLP) to compare the values and to see which model suits our data.

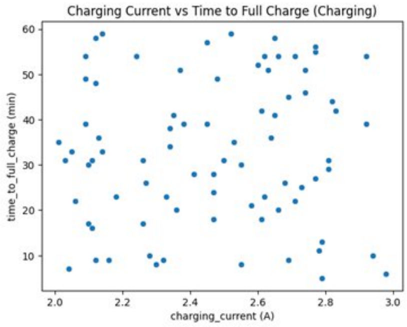
Due to synthetic, we are getting high Mean absolute error. But from the result we understood, Neural Network model gives the better result for Runtime prediction, and Linear regression model gives the better result for Charging Time Prediction.

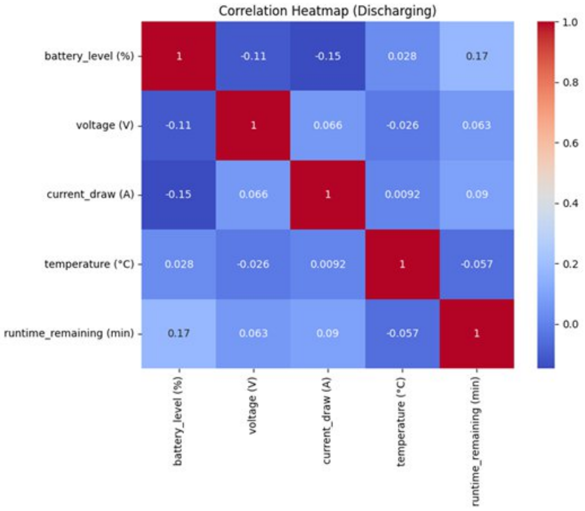
Virtualization for our synthetic dataset:

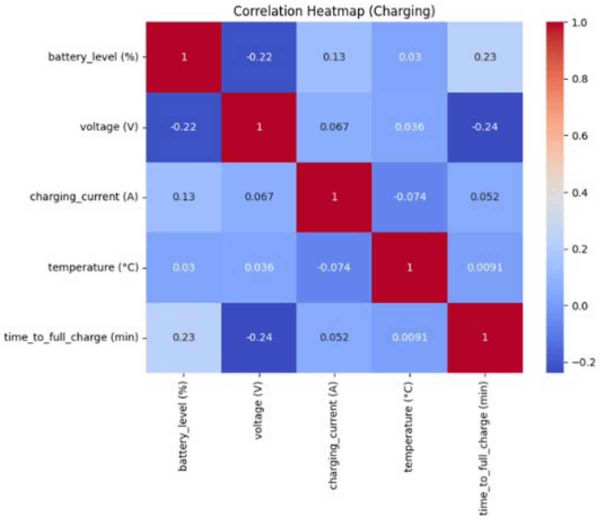

Distribution of voltage (V) (Discharging)


Distribution of voltage (V) (Charging)


Battery Level vs Runtime Remaining (Discharging)


Charging Current vs Time to Full Charge (Charging)


Correlation Heatmap (Discharging)


Correlation Heatmap (Charging)

Data Analysis on Path Optimization dataset

Our idea here is to perform the EDA (Behrens, J. T., DiCerbo, K. E., Yel, N., & Levy, R. (2013)) with the available data to understand the battery and time consumption on different paths, to find the best possible path which will be quicker and save battery consumption.

For coding, we took the help of an AI assistant, but we came up with the core concepts and the same has been used for prompting.

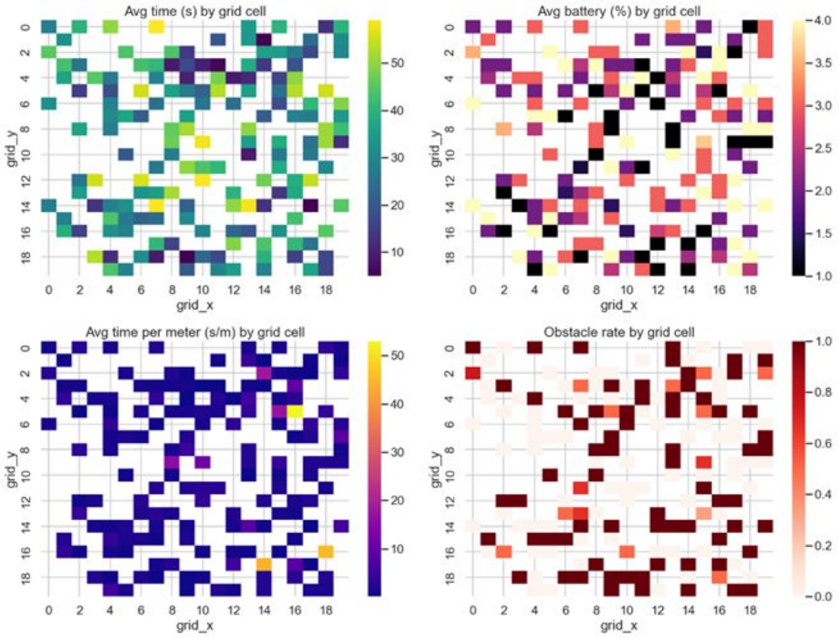
Our concept here is to load the path dataset into pandas and clean the data for further analysis. Basic EDA explains the time and battery impacts with and without obstacles. Calculating the correlations is used to identify the relationships to optimize the navigations.

We can analyse the efficient and inefficient paths and provide the user the option to select for frequent and infrequent schedules (Streiner, D. L. -2005). It will also help the user to plan the lawn in the future.
Visualization helps us understand navigation insights. Distributions help us to understand the variability of time and battery. Top/Bottom paths help us to identify the efficient and inefficient paths for better optimization

Data Analysis on Sample Path Data set:

Our objective here is to analyze the data to find the efficient path with the available data from the logs. This helps the lawn mower to be more efficient by saving time, which directly increases the runtime of the mower. We used pandas and numpy for all the data analysis. For visualization, we used matplotlib and seaborn libraries.

Virtualizations for our synthetic dataset:



Data Analysis on Weather dataset

Weather data plays the role of environmental awareness in the lawn mower project. If the user is scheduling the lawn mower regularly, then this analysis helps to restrict the operation based on the weather prediction.

For example, we can skip the schedule if our prediction says it will rain or if there is extreme heat. In this way, we can prevent damage to the lawn mower due to weather factors.

For data analysis coding, we took the help of an AI assistant, and we performed the prediction using regression models and visualization with matplotlib and seaborn.

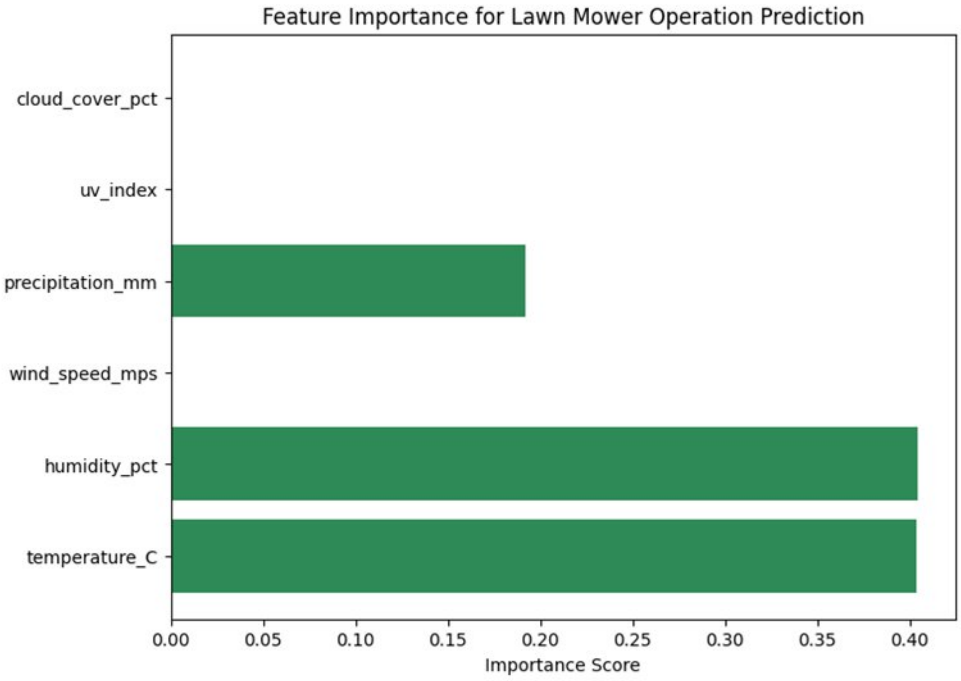
Data Analysis on Sample Weather Data set:

Our objective here is to predict whether the operation is allowed or not for our lawn mower. We will build models that use temperature, humidity, precipitation, UV index, and cover.

We have used two models, Logistic Regression and Decision Tree, to compare the results (Tantri, B. R., & Bhat, S. -2025). From our dataset, we found that both models predict very well. The Decision Tree is slightly better compared to Logistic Regression (Sagu, A., & Gill, N. S. -2020).

Also, we are visualising to get the insights of trend awareness over time, distribution of conditions, classification balance, and operational decision timeline

Virtualisation for our synthetic dataset:



4.3 Summary on AI/DA contribution:

When we started, the main purpose is to predict the data based on the logs available. So, we decided to choose the three main tasks. For predictions, we have compared multiple models based on the datasets. It helps us to understand how data prediction is helpful in designing the tasks for the lawn mower. To provide the support for these tasks, AI/DA provides:

- Performing the data cleaning process
- EDA to know more about the received data
- Visualization to get the insights of data in various scenarios
- Prediction of data as decided in the beginning of the tasks using different data and ML models

5 Full Stack Development

This chapter describes the system architecture and technical choices made in the development of the Lumion application. The goal is to make the application user-friendly for everyone to use. Users of the Lumion lawnmower application can be at any age, so the interface should be as clean and understandable as possible. The application will be provided with multiple languages, so it has multi-language support.

The application interface should also consider people with certain disabilities and be compliant with accessibility guidelines. This means that the color contrast should be within the WCAG standards, the application is fully functional with screen reader technology, and the technical choices support accessibility in all functionalities.

5.1 System Architecture Overview

The architecture consists of a React frontend, a Node.js backend, and PostgreSQL database with REST routes, all deployed within a secure cloud environment. The application front end provides the user interface, the backend coordinates device communication, and the database stores telemetry, configuration, and event data. The cloud-based architecture enables real-time communication with the robotic lawnmower and provides a scalable foundation for analytics, monitoring, and future feature expansion.

5.2 API Design

The public API is the main integration layer between the application, the cloud backend, and the robotic lawnmower. These endpoints are used by the application and by internal services such as AI/DA pipelines, ERP and CRM integrations, and monitoring tools. AI/DA components consume telemetry and event data from the backend to train and run models for battery/runtime prediction, path efficiency analysis, and weather-aware scheduling.

Clear Lifecycle: Queued -> In Progress -> Completed/Failed.

API endpoints:

- GET /v1/devices - list all devices
- GET /v1/devices/{id} - get lawnmower details
- POST /v1/devices/{id}/commands - send commands
  - type: “START/STOP/PAUSE/PARK”
  - type: “EMERGENCY_STOP”
  - type: “LOCK/UNLOCK”
  - type: “ALARM_ON/ALARM_OFF”
  - type: “UPDATE_GEOFENCE”
- GET /v1/devices/{id}/telemetry - telemetry
- GET /v1/devices/{id}/events - event logs
- GET /v1/devices/{id}/security - security state of the lawnmower (physical)

5.2.1 Rate Limiting and Command Idempotency

All public APIs (e.g. /v1/devices/{id}/commands and /v1/devices/{id}/telemetry) are protected by rate limits per user, per IP, and per device. This prevents accidental overload and mitigates simple DOS attempts by restricting abnormal traffic before it reaches core services.

Each command sent to /v1/devices/{id}/commands (START, STOP, PARK, EMERGENCY_STOP, UPDATE_GEOFENCE, etc.) owns a unique identifier. The application backend stores recent command IDs and treats duplicate IDs as the same request. This prevents accidental double execution of critical commands (for example, from user double-clicks or network retries).

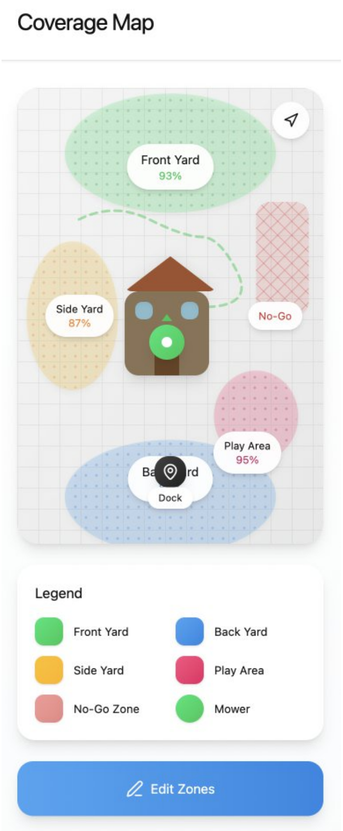
5.3 Scheduler

The application includes a scheduling system that allows users to define specific mowing times. Through the application users can create schedules for selected weekdays and time intervals, for ex. every Tuesday at 9.00 AM and every Saturday at 15 PM. These schedules are stored in the database and mower gets the scheduling information through API.

Backend automatically sends START-command to the lawnmower at the scheduled time. If the user has defined a specific ending time, PARK-command is used, and lawnmower proceeds to dock and end the mowing. It’s also possible to trigger the START, STOP, and PARK-commands manually if the user wants to use the robotic lawnmower partially or fully in manual mode. In case the robotic lawnmower almost runs out of battery during the scheduled mowing, it will automatically dock itself. The device also includes rain sensors, and it will stop mowing during rainy hours. In case of rain during the mowing process, the lawnmower will dock itself.

5.4 Operation Zones and Moving

Lawnmower sends data through /v1/devices/{id}/telemetry, including its current latitude and longitude. Users can visualize mowing zones on a map interface, and the coordinates are saved to the database. Users can create and edit the zones by selecting or drawing the boundaries on the map. Users can also create multiple maps in case there is a need for different areas to be mowed at different times.



Above is a simple mockup made by Figma Make AI-tool. Users can define no-go-zones for driveways or other areas that shouldn’t be accessible for the lawnmower. Different zones can be named accordingly, and they are distinguished by colors.

The geofencing logic makes sure that the lawnmower stays within the correct area and automatically stops or triggers an alert if it moves or is being moved outside of the zone. Based on the operation zones, users can also turn on physical alert sound for the lawnmower as a theft-prevention measure. The robotic lawnmower includes physical sensors that detect lifting of the machine. If the lawnmower is being lifted while operating, it will immediately stop mowing as a safety measure.

Inside of the defined zones there might be some permanent or temporary obstacles. For obstacles, the lawnmower has physical sensors that detect the occurring obstacles. The obstacles will be recorded during the mowing process and can be checked from the application afterwards.

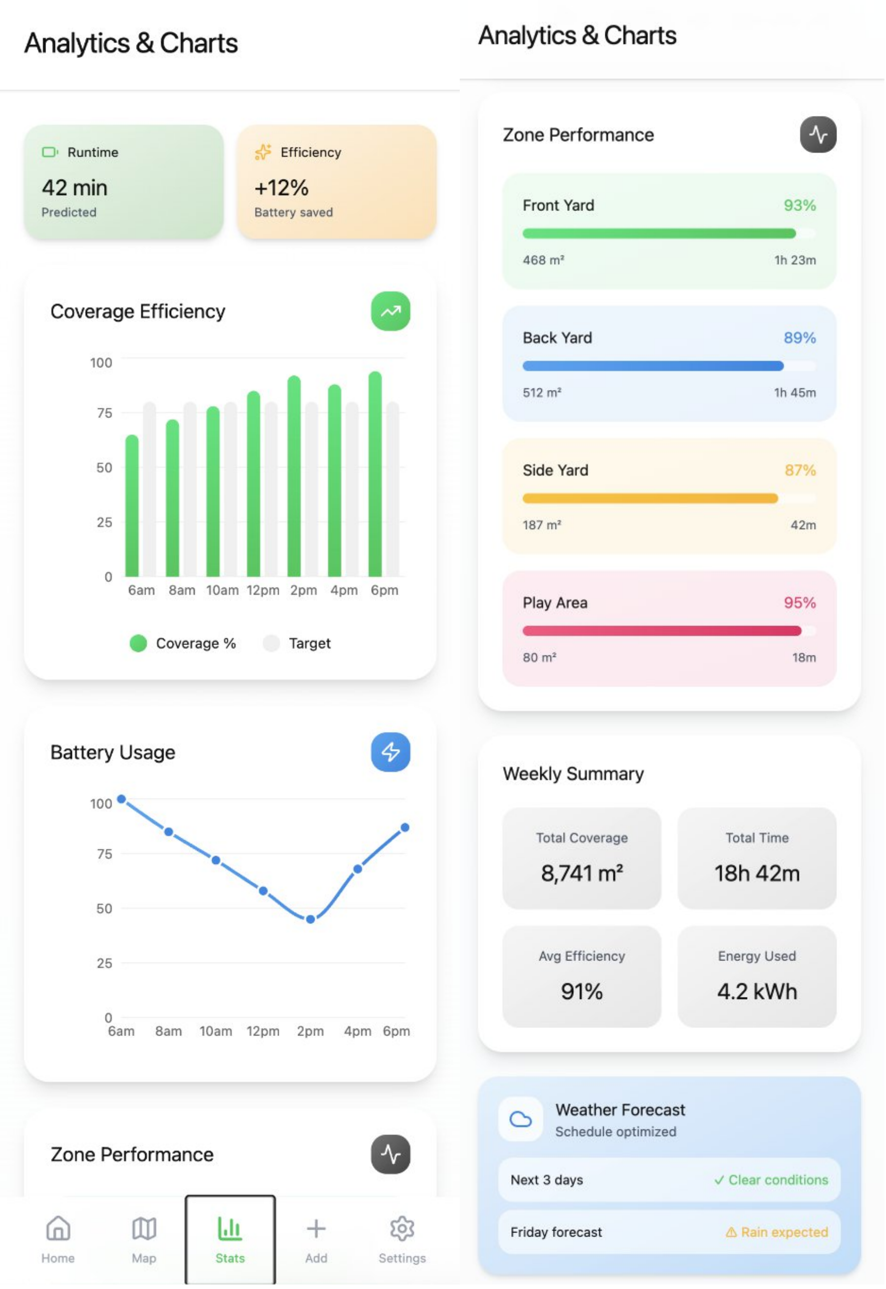
Operating zones can be prioritized, and it can be useful in case there is rain or other reason all the zones can’t be mowed in the normal schedule. This way users can prioritize, for example, the front yard that is a more visible part of the yard.

5.5 Reporting and Statistics

The application includes a report dashboard built with charts which provides users with a clear visual overview of the lawnmower’s activity and performance. The report data is generated from the lawnmower’s telemetry and event information which has been stored in the database. This reporting information allows users to track their lawnmower’s daily and long-term performance, energy efficiency and maintenance.

Users can see how many hours or minutes the lawnmower has operated per day, week or month, how the battery level changes over time, and the total area of lawnmower has mowed each week or month. The data is fetched from the database in simple JSON format, which is then dynamically visualized by different types of charts.

In cooperation with AI/DA team the dashboard can show predicted remaining runtime based on usual battery behaviour, estimated time for charging, suggested optimal mowing times based on weather patterns, and comparisons between efficient and inefficient mowing paths.



Analytics & Charts

Runtime
42 min
Predicted

Efficiency
+12%
Battery saved

Coverage Efficiency

Battery Usage

Zone Performance

Analytics & Charts

Zone Performance

Zone	Efficiency	Area	Time
Front Yard	93%	468 m²	1h 23m
Back Yard	89%	512 m²	1h 45m
Side Yard	87%	187 m²	42m
Play Area	95%	80 m²	18m

Weekly Summary

Total Coverage 8,741 m²	Total Time 18h 42m
Avg Efficiency 91%	Energy Used 4.2 kWh

Weather Forecast
Schedule optimized

Next 3 days	✓ Clear conditions
Friday forecast	⚠ Rain expected

Above is a simple mockup made by Figma Make AI-tool. There are some examples of possible chart data displayed in the application.

5.6 Security & Privacy (Full-Stack Perspective)

The Cyber Security team will be mostly responsible for the overall security, but the Full Stack team must make sure the application uses strong enough authentications and other security implementations.

All user and device interaction is occurring over secured HTTPS channels to prevent any data manipulation or interception. Authentication is using short-lived API tokens and JWT-based sessions which lowers the risk of user data being compromised. Input sanitization is applied for the frontend and backend of the application. This ensures that the data entering the system is safe and valid.

Backend needs some safety features to ensure users' safety. Below are some of the backend features that are being applied together with the robotic lawnmower's physical sensors.

- If robot lawnmower is being lifted, the backend sends EMERGENCY_STOP even if the user does nothing.
- If the robot sends an error code (e.g., blade jam), the backend refuses START commands.
- If GPS is outside geofence, backend auto-issues PARK.

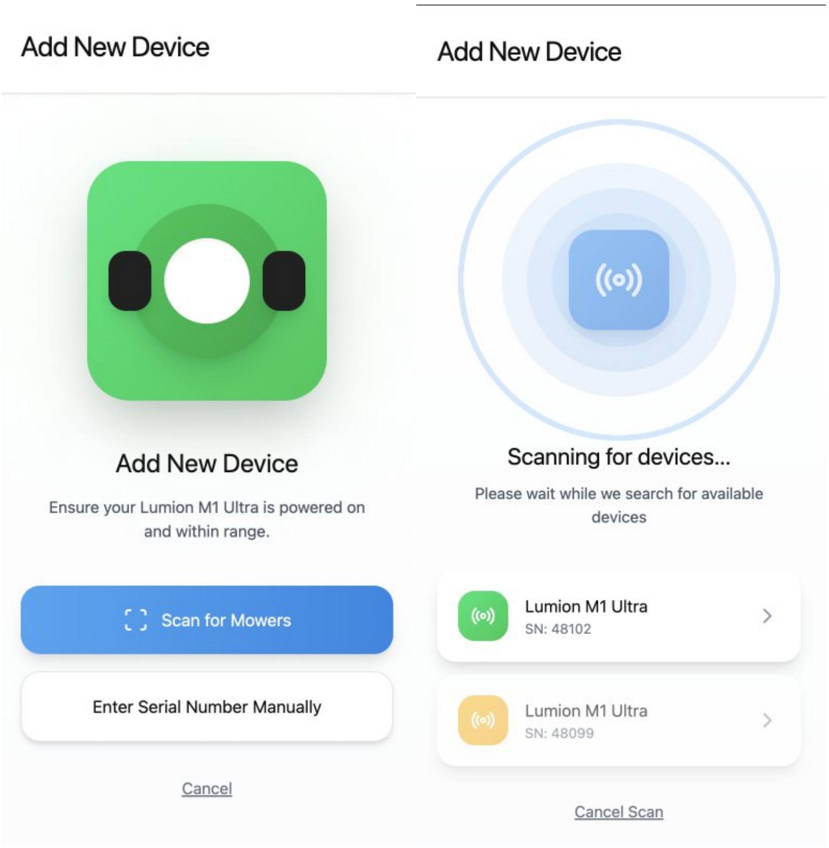
If the lawnmower or network connection goes temporarily offline, the application continues to display the last known telemetry and status, labeled with the timestamp of the last update.

At the front end, for e.g. all the form fields are restricted to specific formats by validation rules. On the backend, incoming JSON is being checked with validation to make sure it has all the correct data types, ranges, and values.

5.6.1 Secure pairing process

For the user to be able to add the robotic lawnmower to their application, there should be a secure pairing process to ensure the safety of the users. During pairing, the lawnmower presents its factory-provisioned device certificate to the backend over authenticated (mTLS) channel.

Application's backend verifies the device identity and securely binds the device to the user's account. All telemetry and commands are only accepted from devices with a valid certificate and an active user binding. With these security measurements together with robotics and cyber security plans, there is a smaller chance for the robotic lawnmower to be compromised.



Add New Device

Ensure your Lumion M1 Ultra is powered on and within range.

Scanning for devices...

Please wait while we search for available devices

Lumion M1 Ultra	SN: 48102
Lumion M1 Ultra	SN: 48099

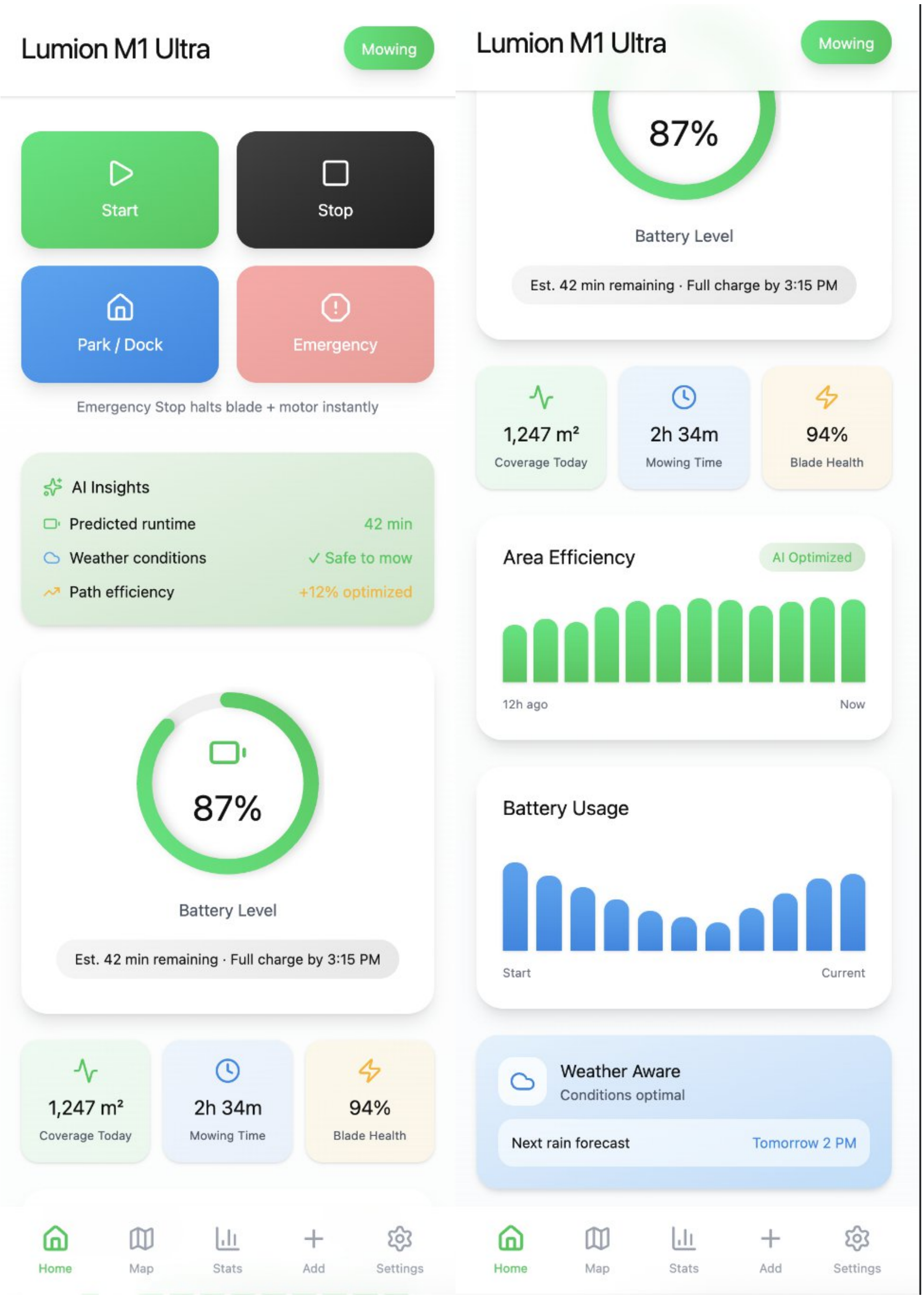
Above is a simple mockup of the device pairing screen made by Figma Make AI-tool.

5.6.2 Central logging

All API requests, command executions, and error conditions are logged and monitored using the stack described in the Cyber Security chapter (e.g., Prometheus and Grafana). These logs and metrics are used both for operations (detection of performance issues or outages) and for security (incident investigations and unusual traffic patterns with possible attacks).

5.7 Frontend & Accessibility

The lawnmower app could be used outdoors, and it needs to be considered for the interface of the application. Sunlight makes it harder for users to use the application, so the interface has high contrasts and clear designs. This way users can still use the application even in challenging outdoor situations. These types of color contrasts are also important for the application to comply with WCAG AA color contrast requirements.



Lumion M1 Ultra

Mowing

Start

Stop

Park / Dock

Emergency

Emergency Stop halts blade + motor instantly

AI Insights

Predicted runtime	42 min
Weather conditions	✓ Safe to mow
Path efficiency	+12% optimized

87%

Battery Level

Est. 42 min remaining · Full charge by 3:15 PM

1,247 m²	2h 34m	94%
Coverage Today	Mowing Time	Blade Health

Home Map Stats Add Settings

Lumion M1 Ultra

Mowing

87%

Battery Level

Est. 42 min remaining · Full charge by 3:15 PM

1,247 m²	2h 34m	94%
Coverage Today	Mowing Time	Blade Health

Area Efficiency

AI Optimized

12h ago

Now

Battery Usage

Start

Current

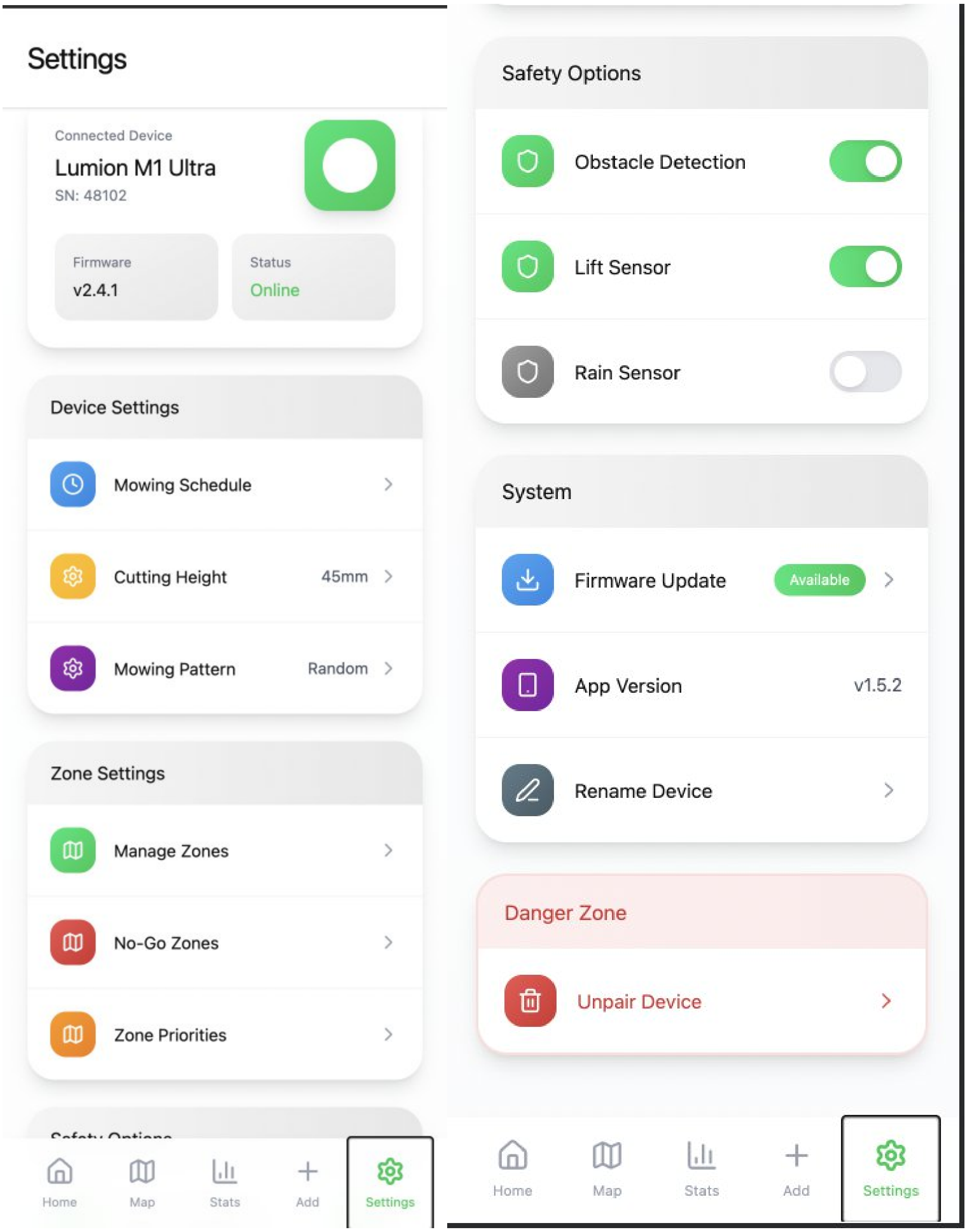
Weather Aware

Conditions optimal

Next rain forecast Tomorrow 2 PM

Home Map Stats Add Settings

Above is a simple mockup of the home page made by Figma Make AI-tool. The home page includes manual controls for the lawnmower, in case users need to quickly take control of the device. The home page also includes the battery level and some recent statistics of the lawnmower's actions. In the top of the home page, users can also see the status of the device (moving, charging, etc.).



Settings

Connected Device
Lumion M1 Ultra
SN: 48102

Firmware	Status
v2.4.1	Online

Device Settings

- Mowing Schedule >
- Cutting Height 45mm >
- Mowing Pattern Random >

Zone Settings

- Manage Zones >
- No-Go Zones >
- Zone Priorities >

Safety Options

- Obstacle Detection: on
- Lift Sensor: on
- Rain Sensor: off

System

- Firmware Update Available >
- App Version v1.5.2
- Rename Device >

Danger Zone

- Unpair Device >

Above is a simple mockup of the settings page made by Figma Make AI-tool. In the settings page, users can define device settings (schedule, cutting height, and mowing pattern). Additional controls allow users to initiate firmware updates, unpair devices, and enable or disable safety features.

References

- **ISO/IEC 27001:2022:** Information technology — Security techniques — Information security management systems — Requirements
- **ISO/IEC 27005:2022:** Information technology — Security techniques — Information security risk management
- **ISO 31010:2019:** Risk Management – Risk Assessment Techniques
- **NIST:** Secure Software Development Framework
- **OWASP Foundation:** OWASP Internet of Things
- **Natarajan, A. n.d.:** Battery data analysis [Data set & code repository]. GitHub. https://github.com/anatarajank/Battery-Data-Analysis
- **Ryu, M. n.d.:** NASA battery data exploratory data analysis [Data set & notebooks]. GitHub. https://github.com/RyuMyunggi/NASA-battery-dataset-eda
- **Behrens, J. T., DiCerbo, K. E., Yel, N., & Levy, R. (2013): Exploratory data analysis. In J. A. Schinka, W. F. Velicer, & I. B. Weiner (Eds.), Handbook of psychology: Research methods in psychology (2nd ed., pp. 34–70). John Wiley & Sons, Inc. https://psycnet.apa.org/record/2012-27075-002**
- **Streiner, D. L. -2005:** Finding our way: An introduction to path analysis. Canadian Journal of Psychiatry, 50(2), 115–122. https://doi.org/10.1177/070674370505000207
- **Sagu, A., & Gill, N. S. – 2020:** Machine learning decision tree classifier and logistics regression model. International Journal of Advanced Trends in Computer Science and Engineering, 9(1.4), 2491–42020. https://doi.org/10.30534/ijatcse/2020/2491.42020
- **Tantri, B. R., & Bhat, S. -2025:** Accuracy comparison of logistic regression and decision tree prediction models using machine learning technique. In Proceedings of the 4th International Conference on Mathematical Modeling and Computational Science (pp. 452–460). Springer. https://doi.org/10.1007/978-3-031-90998-6_41

Appendices

Appendix 1. AIDA - Attachments (Sharepoint Link)

Appendix 2. Lawnmower JSON (Sharepoint Link)