

# リーフ株式会社 (製造業) の取組み事例： 在宅・介護領域におけるセンシングDXプラットフォーム開発

## 参加者

企業	社名	リーフ株式会社
	業種	システム開発
	事業概要	医療機器・リハビリ支援機器・介護福祉機器などの（システム開発、製造、販売）
	所在地 (都道府県・市区町村)	福岡県北九州市
参加した 受講生 の概要	提供データ 種類/蓄積量	<ul style="list-style-type: none"> <li>体重データ</li> <li>姿勢状態データ等</li> </ul>
	チーム名	マナビの庭
	チーム人数	5名
	スキル・PR	ITやプログラミング初心者
	進める中で工夫したこと	チーム内で意志の疎通をしっかりと図り、役割分担を明確化し、各自が参加意識を持てるようにした。

## 実施概要

### 課題内容

#### ベッドセンサーからの情報を表示するUI開発

- 高齢者の体位変換・体重・体調変化を誰でも直観的に理解できるWEBアプリで可視化

### 実施した検討内容

#### 見やすいUI開発のための検討と改善を実施した

- それぞれのベッドに横たわる高齢者の状態の把握と介入のための閾値を検討
- 協働企業に実施いただいた現場介護施設のPOCを踏まえて表示内容を見直した。

### 検討の成果

#### 閾値を4段階に変更、離床・起き上がりの表示機能を追加

- 当初案では、警告対象に離床や起き上がりは含まれなかったが追加をした。
- 第2画面を設けて、それぞれの変化を表示・確認できるように変更をした。

# リーフ株式会社（製造業）の取組み事例：在宅・介護領域におけるセンシングDXプラットフォーム開発

## 実施内容の詳細

PoC用ローカル環境の構築により、ユーザーの声を受けて開発、検証を高速で回し、品質面のテストも実施

### ローカルPC環境での高速開発

本来は物理環境を必要とするセンサー起点のアプリ開発において、機材調達に縛られず、ニーズ対応重視で開発

### ユーザーの負荷軽減を考えた仕様調整

アラートが発報し続けないように、対応完了時の姿勢を記憶し、姿勢が変わらない限り再アラートを抑制

### 証跡管理機能の搭載

スタッフのアクション（対応開始・完了）を秒単位のタイムスタンプとともにデータベースへ保存し、業務の透明性と分析可能なDBを構築

### 管理対象ベッド数の拡張性を考えたUI設計

PoC段階で12台のベッド状況をリアルタイムに俯瞰できるダッシュボードと、過去120日分のデータを分析できる詳細画面を構築

## 成果（成果物は別シート）

迅速な開発でアラート抑制と証跡管理を実現し、12台同時監視の安定稼働と品質を実証

### 開発スピードの向上と機材依存の解消

短期間のうちにソフトウェアロジックを実運用レベルへと昇華させ、検証のPDCAサイクルを劇的に高速化した

### スタッフの精神的・実務的負担の軽減

現場の「アラート疲労」を抑制し、本来のケア業務への集中を支援できる可能性を見出した

### 客観的なデータに基づく証跡（エビデンス）の確保

手書きの記録業務を削減しつつ、事故時の証明や業務負荷の定量的な分析も可能な仕組みへ昇華できた

### 多台数監視における安定性と視認性の実証

シミュレーションによる120日分の長期データを用いた負荷検証に成功、全23項目の機能テストをすべてクリアし、実運用に耐えうるシステム品質を実証できた

# リーフ株式会社 (製造業) の取組み事例：在宅・介護領域におけるセンシングDXプラットフォーム開発

## 成果物一覧

### 1. フロントエンド (サービス操作UI画面)

- ダッシュボード (第一画面)：施設内の12台のベッド状況をリアルタイムに俯瞰できる画面です。緊急度を4段階の色分け (緑・黄・橙・赤) とアニメーションで表現し、「今、誰に何をすべきか」を直感的に判断可能
- 詳細分析画面 (第二画面)：特定の入居者の詳細な履歴を確認する画面です。過去120日分の体重推移グラフや姿勢変化のタイムラインを表示し、サルコペニア (筋力減衰) などの健康リスクの早期発見に寄与

### 2. バックエンド (システム・プログラム構成物)

- APIサーバー (app.py / api\_server.py)  
システムの司令塔として、フロントエンドへのデータ提供とスタッフの対応記録 (証跡) の保存
- データ処理ロジック (data\_processor.py)  
システムの「頭脳」であり、4段階のアラート判定、体重スパイク検知、およびスタッフ対応後の不要なアラートを抑止機能の制御
- シミュレーター (simulator.py)  
12台の仮想ベッドの動きを再現するテストエンジン。起動時にランダムなトラブル状態を注入、テスト用にUIの動作を検証可能
- データ生成ツール (generate\_data.py)  
分析画面のテスト用に、120日分の長期履歴データをデータベースに自動投入
- 一括起動バッチ (start\_system.bat)  
データベースの初期化、シミュレーター、サーバーの起動、ブラウザの立ち上げをダブルクリックのみで自動実行し、デモ準備を高速化
- SQLiteデータベース (beds.db)  
姿勢・体重履歴に加えて、スタッフが「いつ、誰に、何をしたか」を秒単位で記録するテーブルなど、4つのテーブルで構成される永続化基盤

### 3. ドキュメント

- 技術仕様書 (システムアーキテクチャ、データベース設計、API仕様、データ処理ロジックの詳細)
- 環境構築手順書 (非エンジニア対象検証環境構築マニュアル)      • デモ環境構築手順補足資料 (手順書をスライド化)
- テスト結果報告書 (23のテスト項目の検証結果)      • 企業様用サービス提案資料      • プロジェクト管理ドキュメント (WBS、課題管理表)