LifeBridge

Walk with Confidence, Rest with Care

Nitzan Jarus

Firas Shehady

Bara Haj

Software Requirement Specification

Methods in software engineering

Shenkar College

Faculty of Engineering

Overview

In this document we will provide our vision for this semester`s project designed to operate in a local environment aiming to give a reliable monitoring of our user's posture and detecting anomalies with an emphasis on fall detection.

Current health monitoring technology is becoming more affordable with each year, and nowadays it is fairly easy to acquire a smartwatch device and other wearable monitoring hardware that is integrated with one`s body. The demand for fall detection and posture monitoring is symbiotically rising with the longer average lifespan and population increase. However, there is a big flaw in this market plan since most of the target clientele are elderly folk and people with technological accessibility needs, needs that the generic smartphone or watch apps and hardware addons does not provide.

This fault, has created a void that lots of scam companies and products that pretend to be a "simple waterproof bracelet" with a monthly subscription, but, as it was personally experienced by one of our team members in real time, that device did not work properly and the elderly user spent some excruciating minutes in the bathtub.

The last point we made is a good segue due to the highlight it gives on the enormous responsibility designing a system like that has. In a medical anomaly detection software, bugs and other system failures has a potential to literally become life or death scenarios with severe consequences. In addition to the moral responsibility towards this user group made us very passionate about trying to solve this issue.

Our overall goal is to provide an affordable and simple option for a monitoring on-body device that will be easily operated (and will have an idle alert detection background work) by our users. This device will interface with a system dashboard features a user-friendly UI and accessibility functions. All data will be analyzed and processed locally aiming to secure sensitive health user information on the user`s hardware as well as minimizing latency for detections and alerts. and issue alerts locally thus helping us in preventing any sensitive health data to leak that can be operated easily and includes idle-alert detection running in the background.

Problem Description and Motivation

In the overview we've explored how common home-based health monitoring is becoming, as the current industrial hardship of producing a reliable fall detection support for elderly and mobility restricted users, while maintaining a friendly UI for operating and monitoring idly. From our own personal experience and what we've researched, most of the systems we found are cloud dependent with a complex smartphone application. Even worse are the subscription-based devices and services that exploit the great responsibility of taking care and look after loved ones. This issue is becoming even more severe for users that may be alone, or having an extreme difficulty operating any technological device in general, especially in medical emergencies.

Performance reliability and UI accessibility are major challenges we intend to take. Our target user group is not tech savvy, users may have mobility limitations, or cognitive constraints, making it difficult to relay on solutions that require active configuration. This has created an industry filled with poor solutions that fail to meet the needs, and sometimes even worse, scam, those individuals who depend on them most.

This project is motivated by the need for a simple, reliable, and locally operating fall detection operation. Our goal is to minimize the reliance on external services and reduce as much as possible the risk of technical failure. By using edge-based processing and detection, while aiming towards high accessibility and simple UI, we hope to enhance both response time, reliability, affordability and privacy, by offering an alternative open-sourced software that our clientele will trust.

Project Goals

The primary goal of our system is to provide reliable, real-time detection of fall, and posture related anomalies based on locally processed accelerometer data. The system will be designed for continuous monitoring while enabling users to interact with relatively complex technology as simply as possible. To achieve this, our platform aims to operate autonomously, minimize false detections, and present clear feedback through an accessible interface and exported logs.

Additional goals include maintaining full privacy by storing all processed data on the local device, supporting consistent behaviour under various motion scenarios, and offering customizable sensitivity thresholds to accommodate different mobility needs. These goals reflect both the functional requirements and the ethical responsibility of monitoring vulnerable user groups.

The Approach

Software Development

The system will be implemented primarily in Python, utilizing libraries such as NumPy and Pandas for data simulation and preprocessing. Real-time processing logic will be developed to classify movement patterns based on statistical and rule-based anomaly-detection methods.

Data

Accelerometer data will be generated automatically to create genuine motion patterns of a wearable device. The data will include normal walking, resting (in all possible positions), sudden or irrational movement, and fall-like events. All generated data will be processed in real time, enabling immediate evaluation of user-activity patterns.

Hardware Utilization

As an edge-operated system, all numeric calculations will occur locally on the device running the application. By simulating an embedded environment identical to smart wristbands and watches or personal health monitors, no external cloud resources will be required.

Data and Information Management

Because continuous data will be generated, the system will utilize an SQL database. Only processed classifications or minimal user configurations will be stored; raw accelerometer data will not be saved in any form, simplifying privacy requirements and maintenance.

User Interface

A simple and accessible dashboard will be built using standard HTML/CSS with Flask. The UI will present a customizable dashboard for two types of users (the monitored user and the family member/caregiver), real-time motion graphs, system-status indicators, alert notifications, and exported monitoring logs upon request. Accessibility for elderly users will guide all design decisions.

Integration and Extensibility

The software architecture will integrate with external alert systems, such as an SMS-sending service's API (funded by our team) and critical software notifications issued through a designated app or dynamic dashboard page. The design will support future expansion to additional sensors or physiological measurements.

Stakeholders and Scenarios

Stakeholders

Stakeholder	Role / Involvement	Main Needs / Scenarios
Monitored user	Primary system user	Requires passive, continuous monitoring with minimal interaction. Needs clear alerts and accessibility-friendly UI.
Caregiver / family member	Secondary user	Needs accurate alerts, simple status overview, and quick understanding of user activity or fall events.
System Developers	Software developers	Responsible for data accuracy, tuning detection sensitivity, minimize and fix false alerts, system stability, anomaly-detection logic, all while ensuring local processing.
Healthcare Consultant (Nitzan`s sister; MD)	Advisor	May provide insights on realistic fall signatures, safety thresholds, and health related movement anomalies.

Scenarios

Scenario	Title	Description
SC1	Routine Activity Monitoring	The system continuously processes accelerometer data, classifies movement, and displays normal activity on the dashboard. No alerts are triggered.
SC2	Potential Fall Event	A sudden acceleration spike is detected, followed by inactivity. The system flags a fall, updates the dashboard, and records the event timestamp.

Scenario	Title	Description
SC3	Prolonged Inactivity	The system identifies unusual patterns (e.g., prolonged inactivity or irregular movement). Status changes to “Abnormal Activity Detected,” prompting caregiver review.

Functional Requirements

Data

ID	Name	Description
FR1	Data Simulation	The system will generate accelerometer data automatically to create genuine motion patterns of a wearable device.
FR2	Real-Time Processing	All generated motion data will be processed in real time with no external servers involved.
FR3	Simulation Configuration	The system will allow configuration of simulation parameters such as intensity, frequency, and noise levels.
FR4	Local Storage	The system may store recent classified data locally using a simple SQL database.
FR5	Raw Data Privacy	Raw accelerometer data will not be saved in any form.
FR6	Data Clearing	Local logs can be cleared through a developer option or configuration.

Activity Classification

ID	Name	Description
FR7	Activity Classification	The system will classify the user’s movement into resting, walking/movement, sudden impact, and fall-like events.

ID	Name	Description
FR8	Continuous Classification	The system will update the activity classification continuously as new data arrives.
FR9	Activity Logging	The system will log classification outcomes for short-term analysis.

Alerts & Notifications

ID	Name	Description
FR10	Fall Detection	The system will detect patterns consistent with a fall signature (spike + inactivity).
FR11	Fall Alert	The system will issue an alert when a potential fall is detected.
FR12	Adjustable Sensitivity	Sensitivity thresholds for fall detection will be customizable.
FR13	False-Positive Control	The system will minimize false detections by combining multiple characteristics such as impact and inactivity windows.

User Interface

ID	Name	Description
FR14	Dashboard Display	The UI will present real-time motion graphs, system-status indicators, and active alerts.
FR15	Status Colours	The UI will use clear, color-coded indicators for statuses (Normal, Warning, Fall Suspected).

ID	Name	Description
FR16	Passive Operation	The interface will require minimal interaction and operate in a passive background mode.

Non- Functional Requirements

Performance

ID	Name	Description
NFR1	Processing Delay	The system shall process accelerometer data and update activity classification with a maximum delay of 200 ms.
NFR2	Fall-Detection Speed	Fall-detection logic shall respond to detected anomalies in under 1 second.
NFR3	Long-Term Stability	The system shall maintain stable operation during continuous data processing for extended periods (at least 2 hours of uninterrupted runtime).

Reliability

ID	Name	Description
NFR4	Offline Operation	The system shall operate consistently in offline mode with no dependency on internet connectivity.
NFR5	Balanced Accuracy	Anomaly-detection accuracy shall be balanced to maintain a low false-positive rate while ensuring critical events are not missed.
NFR6	Error Recovery	The system shall recover gracefully from minor execution errors without requiring a full restart.

Security & Privacy

ID	Name	Description
NFR7	Local Processing	All data shall remain stored and processed locally on the user's device
NFR8	No External Transmission	No identifiable or sensitive user information shall be transmitted externally.
NFR9	Restricted Access	Access to system data or logs shall be restricted to authorized users only.

Usability & Accessibility

ID	Name	Description
NFR10	User-Friendly UI	The user interface shall be simple, clear, and readable for elderly users and individuals with low technological familiarity.
NFR11	Clear Status Indicators	Status indicators shall use distinct colours, large visuals, and minimal text.
NFR12	Autonomous Operation	The system shall require minimal user interaction and operate autonomously in the background.

Maintainability

ID	Name	Description
NFR13	Modular Codebase	The codebase shall be modular and separated into clear components (data simulation, anomaly engine, UI).
NFR14	Editable Configuration	Configuration values (thresholds, sensitivity, display preferences) shall be

ID	Name	Description
		accessible for adjustment without modifying core logic.
NFR15	Environment Compatibility	Developers shall be able to run the system on any standard local environment supporting Python 3.x.

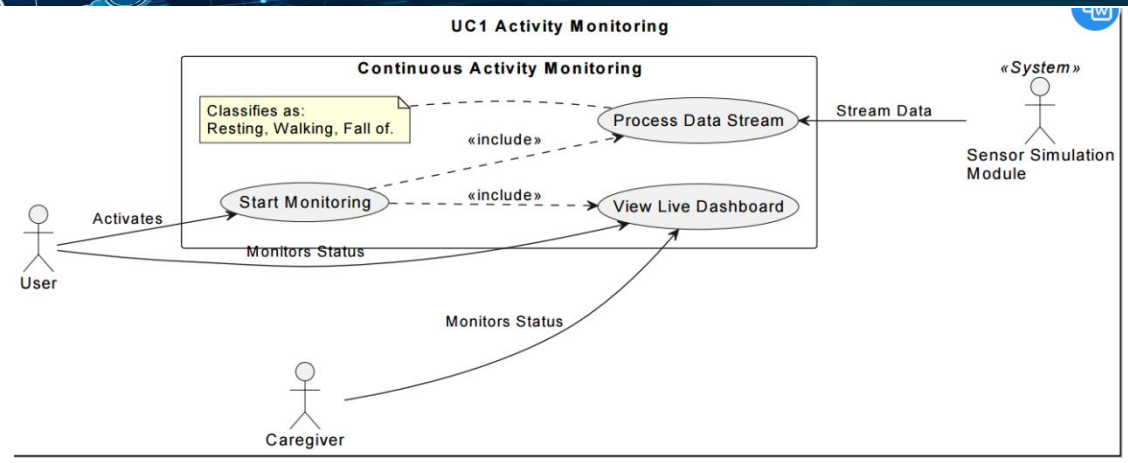
Portability

ID	Name	Description
NFR16	Multi-OS Support	The system shall be able to run on Windows, macOS, or Linux with minimal adjustments.
NFR17	Browser-Based UI	The UI component shall be operable in a standard browser or within a lightweight application wrapper.

Use Cases

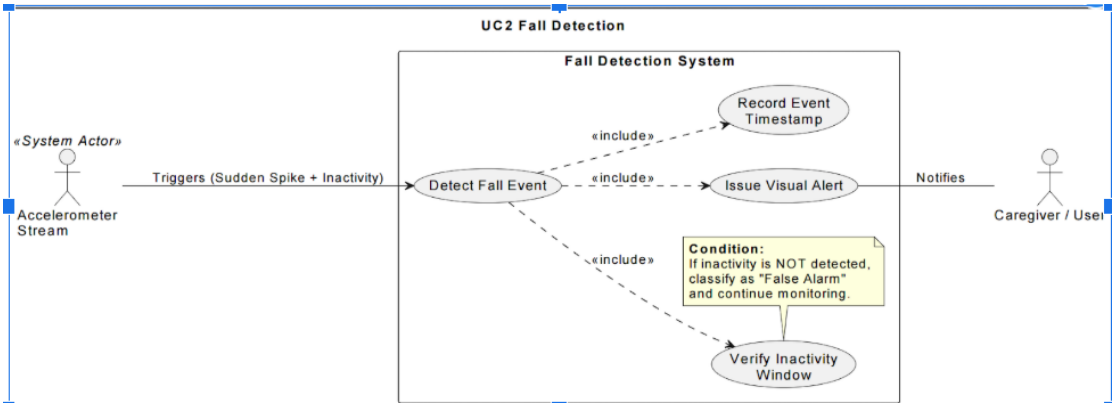
Continuous Activity Monitoring

UC ID	Name	Main Actors	Brief Description
UC1	Continuous Activity Monitoring	User (wearing or carrying the device)	The user activates the monitoring system. The system generates or receives accelerometer data, classifies activity in real time, and updates the dashboard continuously until monitoring stops. Recent activity samples remain available for short-term review.



Detect Fall Event

UC ID	Name	Main Actors	Brief Description
UC2	Detect Fall Event	Accelerometer Stream (system actor) and Caregiver / User (alert recipients)	The system detects a sudden high-impact acceleration spike followed by inactivity, classifies the event as a potential fall, issues an alert in the UI, and records the event timestamp. If inactivity is not detected, the event is treated as a false alarm.



View Status and Alerts

UC ID	Name	Main Actors	Brief Description
UC3	View Status and Alerts	User / Caregiver	The user opens the dashboard to view the current activity status, review any logged fall events, and acknowledge or dismiss alerts. Monitoring continues normally after the dashboard is closed.

Export Monitoring Logs

UC ID	Name	Main Actors	Brief Description
UC4	Export Monitoring Logs	Family Member / Caregiver	The caregiver accesses the dashboard and exports recent activity classifications, fall alerts, and timestamps for review or documentation. Logs are generated from stored classified data without exposing raw sensor information.

External Alert Notification

UC ID	Name	Main Actors	Brief Description
UC5	External Alert Notification	System → Caregiver	When a potential fall is detected, the system triggers an external notification (e.g., SMS or push alert) using an integrated API. The caregiver receives the alert even if they are not actively viewing the dashboard.

Adjust Sensitivity and System Settings

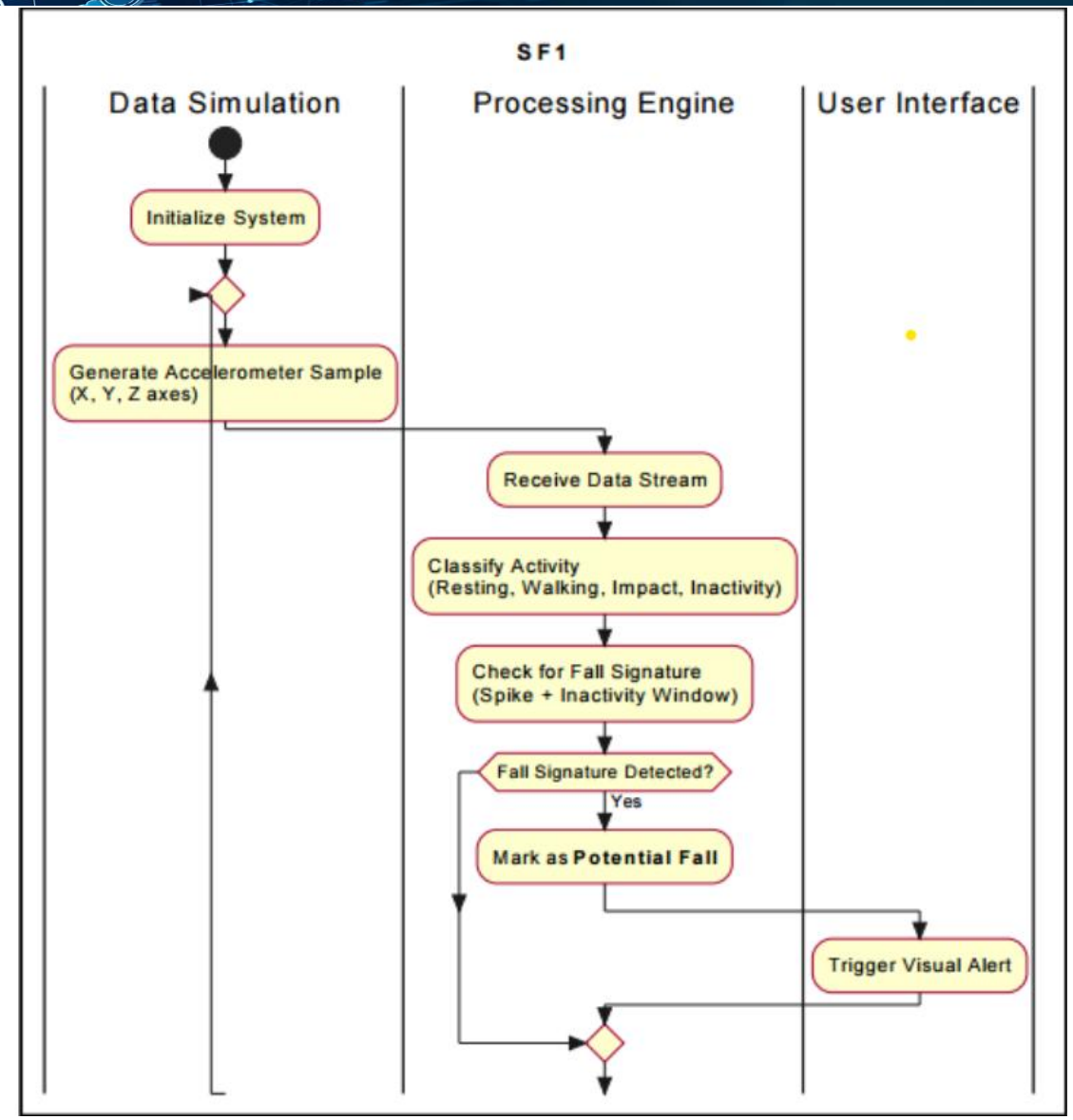
UC ID	Name	Main Actors	Brief Description
UC6	Adjust Sensitivity and System Settings	User / Caregiver	The user accesses the settings panel to adjust fall-sensitivity thresholds, display preferences, or monitoring parameters. Changes take effect immediately without modifying core system logic.

System Flows

This section describes the end-to-end flows of data and interactions within the fall-detection system. The flows are illustrated in diagrams SF1 and SF2 and summarized in the following table. SF3 is an additional textual flow without a diagram.

Data Simulation and Fall Detection

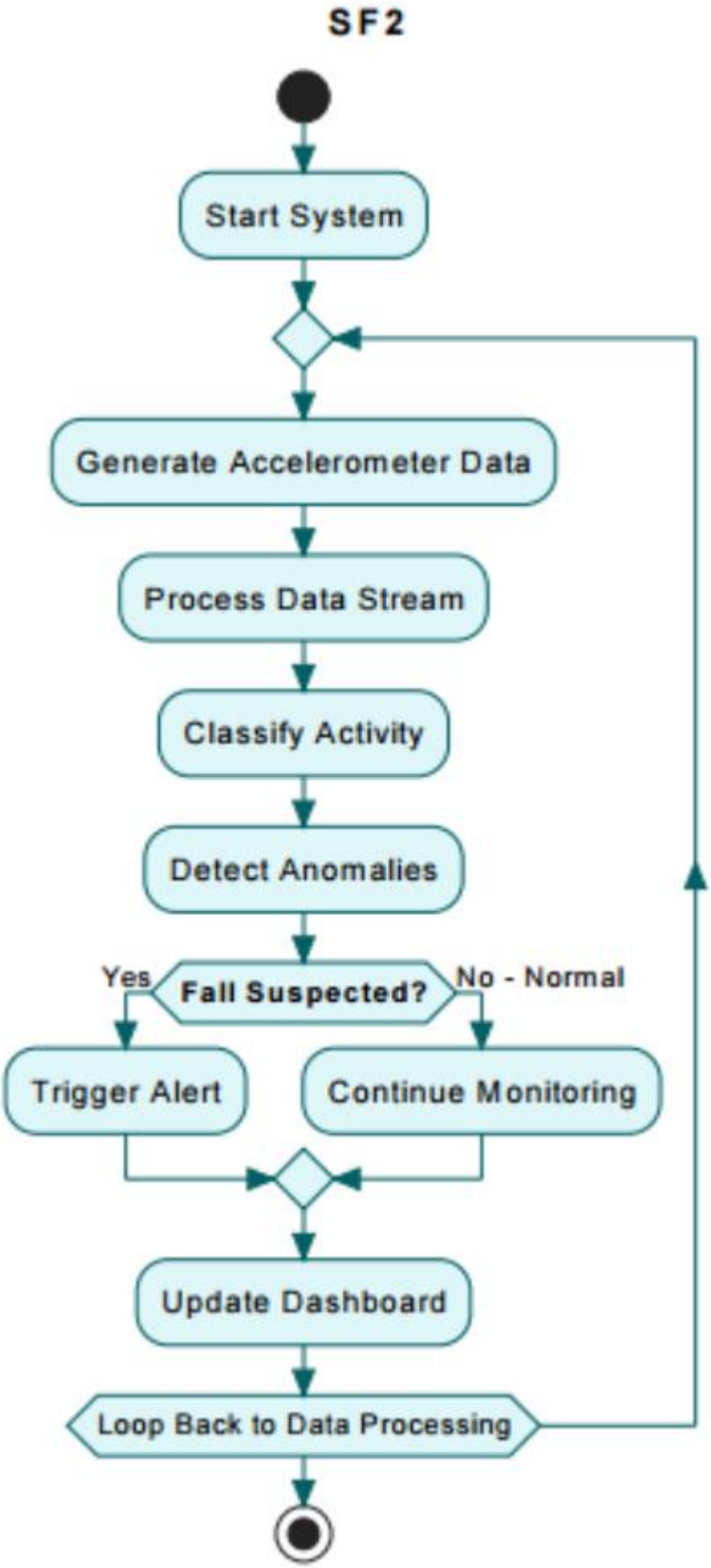
Flow ID	Name	Steps (High-Level)
SF1	Data Simulation and Fall Detection	1. Initialize system and start accelerometer-data simulation. 2. Generate continuous accelerometer samples (X, Y, Z axes). 3. Send the data stream to the processing engine. 4. Classify activity (Resting, Walking, Impact, Inactivity). 5. Check for fall signature (Sudden Spike + Inactivity Window). 6. If fall signature detected → mark as *Potential Fall*. 7. Trigger a visual alert in the UI.



Continuous Monitoring and Alert Loop

Flow ID	Name	Steps (High-Level)
SF2	Continuous Monitoring and Alert Loop	1. Start system and begin monitoring mode. 2. Generate accelerometer data continuously. 3. Process the data stream in real time. 4. Classify incoming activity. 5. Detect anomalies and check if fall is suspected.

Flow ID	Name	Steps (High-Level)
		6. If No → continue monitoring. 7. If Yes → trigger alert and update dashboard. 8. Loop back to data processing and continue monitoring.


SF2
Start System
Generate Accelerometer Data
Process Data Stream
Classify Activity
Detect Anomalies
Yes
Fall Suspected?
No - Normal
Trigger Alert
Continue Monitoring
Update Dashboard
Loop Back to Data Processing

Log Export and Review Flow

Flow ID	Name	Steps (High-Level)
SF3	Log Export and Review Flow	1. User/caregiver opens dashboard. 2. Requests export of recent monitoring logs. 3. System retrieves stored classifications and fall events. 4. Formats data for export (no raw sensor data). 5. Provides downloadable/viewable log file for review.

Appendix

1. An expansive solution with a complex UI presentation
2. Ynet article describing "אמבולנט" company as its exploiting elderly users and over chage for faulty service
3. Ynet article from 2025 about the same topic, three companies are described as they participated in fraud operation causing the loss of millions new Israeli shekels
4. Israel`s authorities official warning aboiut health related scams