Progress Report 1: Smart Door Security Framework with Anomaly Detection & Cybersecurity

Student Name: Eric Sanjo
Student ID: 300395898
Project: Smart Door Security Framework with Anomaly Detection & Cybersecurity for Elderly Care
Course: CSIS 4495 – Applied Research Project, Section 3
Reporting Period: Jan 19 – Feb 9, 2026
Report Date: February 9, 2026

Work Date/Hours Logs

Jan 27, 2026	1.0	Looked up introductory material on anomaly detection (Isolation Forest, simple statistical approaches) and how they are used in IoT/smart home settings; saved links and notes.
Jan 29, 2026	1.5	Helped review the email draft to Armin summarizing our planned direction (door + anomaly detection + security) and checked that the anomaly section matched what we discussed.
Jan 30, 2026	1.5	Read a few articles and blog posts about common “anomalous” patterns in access logs (e.g., long inactivity, frequent failed attempts) and wrote short notes on features we might use later.
Feb 3, 2026	2.0	In-person meeting with Armin: saw the Raspberry Pi, lock hardware, and Ring-style camera prototype video; asked questions about what data would realistically be available for anomaly detection.
Feb 3, 2026	1.5	Team debrief after hardware meeting: discussed what kind of events we can actually log (entries, exits, failures) and how that would shape the synthetic dataset and anomaly detection approach.
Feb 4, 2026	2.5	Read introductory/tutorial material on Isolation Forest (parameters, contamination rate) and thought about how we could represent door events as feature vectors in the future.
Feb 5, 2026	1.5	Team call: discussed how rule-based checks and anomaly scores might be combined; wrote down some ideas for simple thresholds we can use before adding any heavier models.

Feb 6, 2026	2.0	Sketched a first rough plan for a **synthetic access log** (normal daily patterns + a few simple anomalies like late-night access and many failed attempts); no code written yet, just planning.
Feb 7, 2026	2.0	Read a few pages of documentation/examples on how to work with tabular time-series data in Python (Pandas) to prepare for generating and analysing access logs.
Feb 8, 2026	1.5	Short team sync: confirmed that I will focus on generating the synthetic access dataset and later a basic anomaly detection prototype once logging structure is clearer.

Total Hours This Period: ~17 hours

Summary Description of Work Done During This Reporting Period

For this first reporting period, my focus was on **helping shape the project direction from an anomaly-detection point of view** and understanding what kind of data we will realistically have. In January, I participated in the initial idea filtering, proposal writing, and clarifying that we want a smart door project where anomaly detection supports security and elderly care use cases.

The key step was the **Feb 3 in-person meeting with Armin**, where we saw the Raspberry Pi, the lock, and a Ring-style camera prototype. This helped me understand that our anomaly detection will likely be based on fairly simple door events (time, user role, success/fail), not huge complex datasets. After that, I spent time reading about **Isolation Forest** and general anomaly detection basics, and I started planning what a **synthetic access log** for our system should look like (normal daily patterns plus a few types of unusual behaviour).

At this stage, I have not implemented any ML models yet. Instead, I am preparing the ground: deciding what fields we need in the logs, what "normal" vs "suspicious" might look like in data terms, and how simple rule-based checks can work alongside anomaly detection later.

We are still in the planning and design phase. The current priority for me is to design a synthetic dataset and feature structure that will make sense once the logging and basic face-recognition flow exists, without over-committing to a complex model too early.

Issues Encountered and How We Handled Them

- **Unclear data structure at the start**
  Initially, we didn't know exactly what data would come from the door system

(e.g., just timestamps and “open/close”, or more detailed info). The hardware meeting and discussion with Armin clarified that we can plan around: time of event, user/role, success/failure of access, and maybe basic context (e.g., reason or mode).

- **Balancing ML complexity with project scope**
  I was interested in trying more advanced models, but after talking as a team and with Armin, we agreed to keep the ML side **lightweight and realistic** for a course project. The idea is to start with simpler anomaly detection (Isolation Forest or even basic statistics) and not promise anything we can’t deliver.
- **Avoiding designing a dataset that doesn’t match reality**
  There is a risk of designing a synthetic dataset that looks nice on paper but doesn’t match what the hardware can actually generate. To address this, I’ve based the planned fields and patterns directly on what we saw (Pi + lock + camera) and on what Advitiya and Reubin plan to log from the system.

Changes to Proposal / Plan

No major changes have been made to the official proposal, but we clarified internally that:

- Anomaly detection will be **supporting** the core security features (logging, rules, alerts), not the entire focus of the project.
- The synthetic dataset will be **modest in size and complexity**, enough to test basic ideas rather than aiming for a research-level benchmark.
- The anomaly logic will initially run with simple features (time of day, frequency, failed attempts, inactivity) that map directly to what our system can log.

Updated Individual Timeline

Next 1–2 weeks (up to Midterm):

- Finalize the structure of the synthetic access dataset (fields and example values).
- Start writing a small script to generate a first version of the synthetic log (normal vs a few simple anomalies).
- Coordinate with Advitiya and Reubin so the dataset structure matches the way they intend to log real events later.

- If time allows before midterm, try a very small test with a simple anomaly detection approach (even if just basic statistics) on the synthetic data to see if the patterns are visible.

Repo Check-In of Implementation Completed

So far, my contributions are mainly notes and planning, not full implementation. Examples:

- **Docs / Planning**
    - Added notes in the documents folder describing a possible **dataset schema** (timestamp, user/role, event type, success/fail).
    - Wrote a short document listing "normal behaviour examples" (typical daily routine) and "odd behaviour examples" (late-night access, repeated failures, long inactivity).
- **No ML code yet**
    - I have not committed any anomaly detection code yet, as we agreed first to finalise the logging and event structure before building models.
    - Initial dataset generation and prototype code will be added in the next reporting period.

AI Use Section

Tools I Used and How

- **Chat-based AI (e.g., ChatGPT / Gemini)**
    - To get simple, high-level explanations of **Isolation Forest** and other anomaly detection ideas, so I could then read more detailed sources.
    - To brainstorm examples of **"odd" access patterns** (e.g., very late visits, too many failed attempts) and see if we were missing any obvious ones.
    - To help rephrase some parts of the proposal and my notes so they were clearer.
- **Code suggestion tool (e.g., Copilot)**
    - I have not used this much yet; I may use it later when writing simple data-generation scripts (for boilerplate loops and structures), but any logic for anomalies will be written and checked manually.

What I Did Myself (Value Beyond AI)

- Chose which anomaly ideas were realistic by comparing AI suggestions with what we saw at Armin's office and what the Pi/lock/camera setup can reasonably support.
- Designed the planned dataset fields based on our own project needs and discussions, not just on generic AI examples.
- Read additional tutorials and documentation (blogs, library docs) to check that the AI explanations matched actual tooling and functions.

Examples of Actual Prompts (Appendix)

Some examples of the prompts I used:

- "Explain Isolation Forest in simple terms for a small IoT/door access project."
- "What are some common examples of weird or suspicious patterns in access logs?"
- "Give a few ideas for synthetic anomalies in a door access dataset (time of day, frequency, failures)."
- "What kind of columns should a basic door access log table have?"

These were used as starting points; I then adapted the ideas to our actual project and hardware constraints.

Short Reflection

This period helped me understand **what data we will actually have** and how far we should go with anomaly detection in a single term. The meeting with Armin and seeing the hardware were the most helpful parts, because they grounded our ideas in reality. In the next phase, I want to move from planning to actually generating a small synthetic dataset and then testing simple anomaly detection ideas on top of it.