# Requirement Gathering

## 1. Business Problem

Currently, the process of extracting device compliance data from Microsoft Intune is **manual, repetitive, and inefficient**, relying on IT staff to log in, navigate the portal, and export reports manually. This slows down triage efforts, limits visibility of non-compliant devices, and introduces risk through inconsistent reporting intervals.

## 2. Objective

To design and build a Python-based tool that automates the retrieval, storage, and reporting of compliance data for Windows and iOS devices managed by Microsoft Intune, providing consistent and timely insights for IT operations.

## 3. Stakeholder Engagement

To shape the requirements and ensure the project delivers value:

* I consulted with our Head of Digital & Transformation and IT Support Manager regarding reporting needs and process gaps.
* I discussed report structure and compliance categories with members of the IT team.
* The project scope and plan were reviewed and approved by my Digital Learning Consultant (DLC), Rachel Cunningham.

## 4. Functional Requirements

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| Retrieve Compliance Data | Use Microsoft Graph API to fetch device compliance data (platform, status, last check-in) |
| Store Data | Insert structured records into a SQL database for long-term reference |
| Generate Reports | Provide summaries such as number of compliant vs non-compliant devices, platform breakdown |
| Automate | (Optional) Schedule periodic execution to keep data up to date |
| Alerting | (Stretch goal) Email notifications if non-compliance thresholds are exceeded |

## 5. Technical Requirements

|  |  |
| --- | --- |
| **Area** | **Technology / Decision** |
| Language | Python 3 |
| API Access | Microsoft Graph API with OAuth 2.0 |
| Data Storage | SQLite (lightweight, file-based SQL database) |
| Data Processing | pandas for transformation and aggregation |
| Visual Reporting | matplotlib or seaborn for charts (if time allows) |
| Security | Use .env file to store sensitive API credentials securely |

## 6. Constraints and Assumptions

* API access is assumed to be pre-authorised for device compliance endpoints.
* The script will run in a local or controlled internal environment with access to the Graph API.
* Delivery time is capped at ~30 hours across 4 weeks, with 7.5 hours per week.

With a clear understanding of the current challenges, requirements, and limitations, I was ready to begin the Analysis and Design planning phase.