**Individual Contribution Report**

**for**

**TTrack – Degree Tracker**

|  |  |
| --- | --- |
| **SDM404 - Assessment 4 – Individual Contribution Report** | |
| **Project Name** | TTrack – Degree Tracker |
| **Group #** | #1 |
| **Group Members Names** | Luis Guilherme de Barros Andrade Faria - **A00187785** |
| **Version** | 1.0.0 |
| **Date Created** | 19, August, 2025 |

Table of Contents

[1. Introduction 2](#_Toc206484064)

[2. Tasks Worked On 2](#_Toc206484065)

[3. Problems Faced and Reflections 3](#_Toc206484066)

[4. Solutions Implemented 4](#_Toc206484067)

[5. Outcome 4](#_Toc206484068)

[6. Personal Reflection 4](#_Toc206484069)

# **Introduction**

This project was my first opportunity to carry dual responsibility as both Software Engineer and Project Manager. In my professional background, I have played these roles separately, but never simultaneously. Acting in both capacities changed my perspective: I had to constantly balance engineering trade-offs (e.g., which library or database to adopt) against management imperatives (timelines, risk, stakeholder validation).

Looking back, my workflow consistently followed the loop of *plan → build → validate → adjust*. Each iteration forced me to face technical unknowns while maintaining delivery discipline.

The project became a mirror of the Software Development Management principles studied in class. I’d like to point out that not abstract theories, but living practices that made or broke our progress!

# **Tasks Worked On**

A quick summary of the progress on the application development can be found here and you’ll also be able to see the complete changelog within the directory */docs* of the TTrack’s source code.

|  |  |
| --- | --- |
| **Date** | **Description** |
| 27/06/2025 | First version of TTrack GUI with pandas integration |
| 04/07/2025 | Integrated sample data and file upload module |
| 12/07/2025 | Built macOS and Windows distributions |
| 30/07/2025 | Integrated Supabase database for persistent storage |
| 05/08/2025 | Layered Architecture rollout (Core, Services, Controllers) |
| 07/08/2025 | Added Student Records tab to show processed history |
| 16/08/2025 | Engine Matching 2.0 (MSIT curriculum) |
| 18/08/2025 | Engine Matching for ADIT21 (multi-curriculum support) |
| 19/08/2025 | Course selection feature (switch between MSIT/ADIT) |

# **Problems Faced and Reflections**

As this was my first time building an offline app using PyQt5, many challenges appeared during the progress, below you’ll be able to find a few highlights/lowlights:

1. **CSV parsing errors with dirty headers**
   * *Challenge:* Early runs failed due to trailing spaces and inconsistent column names.
   * *Deeper lesson:* A system is only as strong as its assumptions. I had to harden the pipeline against messy, real-world data and created the feature of giving final users a sample data to be used as default input content.
2. **UI discrepancies between macOS and Windows**
   * *Challenge:* The same PyQt layouts rendered misaligned on different platforms.
   * *Deeper lesson:* Cross-platform engineering is never “free.” Anticipating variability should be built into the schedule.
3. **Light/Dark mode toggle bug**
   * *Challenge:* State logic broke when switching themes mid-session.
   * *Deeper lesson:* Small UX issues reveal architectural weaknesses. This pushed me to centralize theme state in a *ThemeManager* class.
4. **Database decision (MongoDB vs Supabase)**
   * *Challenge:* Choosing between a flexible but heavy stack (Mongo) and a lean but managed solution (Supabase).
   * *Deeper lesson:* Good management is often *saying no*. I learned to prioritize simplicity and integration over “what looks powerful.” Also first time using Authentication from Supabase.
5. **Encryption of .env in distributed builds**
   * *Challenge:* Security vs usability: distributing credentials safely while keeping builds functional offline.
   * *Deeper lesson:* Security isn’t an add-on; it must be engineered into the release pipeline from day one.
6. **Late-breaking elective rules (ADIT)**
   * *Challenge:* ADIT introduced slightly different elective/credit logic.
   * *Deeper lesson:* A well-designed engine can adapt. The dict-based loader I created made this pivot possible without rewriting everything. You’ll be able to see that in class *DataProcessor*.

# **Solutions Implemented**

In summary, the approach to the previous issues were:

* **Robust preprocessing**: Stripping whitespace, normalizing case → eliminated fragile CSV parsing.
* **Responsive layouts**: Dynamic PyQt grids replaced static positioning → stable cross-platform UX.
* **Centralized theming**: ThemeManager handled state → light/dark mode became predictable.
* **Database integration**: Chose Supabase → avoided over-engineering, but gained reliable cloud sync.
* **Security-conscious builds**: Automated .env encryption → shipped binaries safely to end users.
* **Multi-curriculum engine**: Modular data loaders for MSIT & ADIT → extensibility proven.

# **Outcome**

TTrack evolved from a simple matching script into a full-fledged academic progress tracker that can:

* Upload student transcripts and compare against different curricula
* Validate prerequisites and fill elective slots dynamically
* Save sessions locally or to the cloud (Supabase), with retrieval by student ID
* Run cross-platform on macOS/Windows with consistent UI/UX
* Export progress reports for offline or administrative review

The engine is no longer bound to a single course. It now supports both MSIT and ADIT structures, positioning TTrack as a scalable solution for Torrens University.

# **Personal Reflection**

Dr. Atif, thank you for the invaluable guidance and for shaping the TTrack project into both a technical and personal learning journey. Working on this blurred the boundaries between engineering execution and management responsibility. Through the process, I grew as a professional by:

* Building functional and scalable code
* Balancing trade-offs between scope, schedule, and complexity
* Translating technical decisions for non-technical stakeholders
* Experiencing the cost of poor assumptions and the relief of resilient design
* Leading myself through iterative cycles of challenge, adjustment, and delivery

The most meaningful outcome was not just the software itself, but a shift in mindset: I now view projects less as “code to be written” and more as **systems of decisions, risks, and people**.

To extend this journey, I have documented the entire process in an open-source repository, my personal website, and the dev.to platform, sharing not only the code, but also reflections, documentation and insights from the project. If you have the time to check:

* **Repository:** [github.com/lfariabr/masters-swe-ai](https://github.com/lfariabr/masters-swe-ai)
* **Dev.to:** [dev.to/lfariaus](https://dev.to/lfariaus)
* **Portfolio:** [luisfaria.dev](https://luisfaria.dev/)

Looking forward to when our paths cross again in the future. Thank you!