Software Proposal

TTrack

Group: 1

Students:

Luis Guilherme de Barros Andrade Faria - **A00187785**

Nomayer Hossain - **A00176827**

Hussain Jameel – **A00XXXXXX**

Rosa Carolina Cortes Galvis – **A00XXXXXX**

Victor Javier Dorantes Meneses – **A00XXXXXX**

Subject Code: SDM 404

Subject Name: Software Development Management

Assessment No.: 1

Title of Assessment: Proposal

Lecturer: Dr. Muhammad Atif Qureshi

Date: June 2025

Table of Contents

TTrack

SMART SUBJECT REQUIREMENT CHECKER

Torrens University Research Project

Project Overview

Problem Statement

Torrens admin staff and students face a recurring challenge: tracking degree progress and making sure no critical core or elective subjects are missing. Despite transcripts and course handbooks existing in separate silos, there's no smart, automated way to cross-check course history against what is required to graduate. The current process is manual, error-prone, and frustrating.

TTrack solves this by providing users a desktop application that parses transcripts and compares them with degree rules: no internet, no admin intervention, just clarity.

Target Users/Clients

- Academic advisors who need a faster way to audit student progress;
- University course coordinators looking to reduce manual verification load;
- Students currently enrolled in Torrens who want visibility into their degree path.

Project Goals

TTrack will consist of delivering a desktop application that performs the following:

1. Receives Academic Transcript and Academic Curriculum files (.xlsx) to be analyzed;
2. Automatically validate completed vs. pending subjects;
3. Visually summarize academic progress across subject categories;
4. Recommend electives based on unmet credit or rule gaps;
5. Package into a cross-platform offline app built in Python.

Tools & Technologies

The application will run based on:

- Python 3.10+
- PyQt5 for desktop GUI
- pandas & openpyxl for Excel processing
- matplotlib for dashboard/visuals
- SQLite (optional) for local persistence
- PyInstaller to package the app as .exe/.app

Development Methodology

We’re using Agile with 2-week sprints. This gives us breathing room for iteration, makes space for UI feedback, and fits the team’s schedule rhythm. We're sticking with Scrum’s heartbeat: backlog, sprint plan, build, demo, reflect.

References:

- Cobb, C. (2015). The Project Manager’s Guide to Mastering Agile.
- The Scrum Guide (2020). Schwaber & Sutherland.

Scope and Deliverables

In scope:

- Upload & parse transcript and curriculum Excel files
- Match completed units to core, specialisation and elective subjects
- Display subject status with 'done’, ‘missing’, ‘invalid’
- Elective suggestion engine

Out of scope:

- Real-time academic system integrations
- Web/mobile versions

Deliverables:

1. Functional desktop app withGUI
2. Matching engine with business logic
3. Visual summary/dashboard
4. Codebase & documentation

Timeline and Milestones

Milestone	Task	Owner	Start	End
Sprint 1	UI structure + upload workflow	Luis	17/06	23/06
Sprint 2	Excel parsing + table view		24/06	30/06
Sprint 3	Subject matching engine		01/07	14/07
Sprint 4	Dashboard + electives logic		15/07	25/07
Sprint 5	Test, polish and deliver		26/07	31/07

The proposed project will be implemented over 12 weeks using a Scrum-based Agile methodology (Heath, F. 2021). The development is divided into **six sprints**, each lasting two weeks. This approach supports iterative improvement, rapid feedback incorporation, and adaptability to evolving requirements from stakeholders and data insights.

Sprint 1: UI Structure, upload workflow

- Define and sketch UI wireframes for main screens
- Set up basic PyQt5 app window and layout
- Implement file upload functionality for .xlsx transcript and curriculum files
- Validate file format and display upload status in UI
- Create initial GitHub repository and project board for task tracking
- Sync with team to define MVP scope and testing expectations

Sprint 2: Excel parsing and table view

- Build parser using pandas + openpyxl to read Excel content
- Normalize transcript and curriculum data (subject code, title, credit points)
- Design in-memory schema for academic records
- Display parsed data as tables inside the app using PyQt5 TableWidgets
- Handle edge cases (e.g. missing headers, corrupted files)
- Conduct internal demo to test table rendering and upload flow

Sprint 3: Subject matching engine

- Implement logic to compare transcript subjects with required curriculum
- Define rules for subject classification: core, specialisation, elective
- Tag each subject as “done”, “missing”, or “invalid” based on business logic
- Integrate matching output with UI (status indicators)
- Test matching logic across different example inputs
- Document ruleset structure and logic used in engine

Sprint 4: Dashboard and electives logic

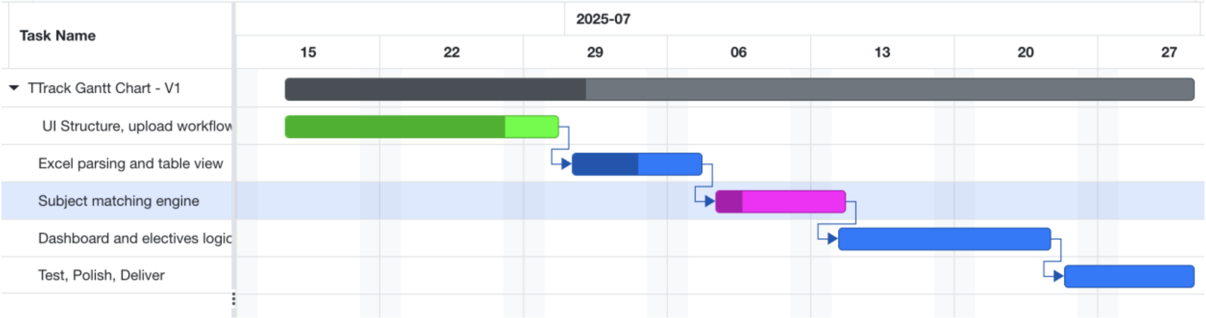
- Design visual dashboard showing academic progress (e.g. pie chart, bar chart)
- Integrate matplotlib charts into the PyQt5 GUI
- Build elective suggestion engine based on unmet credits or required categories
- Test UI responsiveness with different dataset sizes
- Refine UX: colors, labels, and tooltips for better clarity
- Record user feedback from mock testing (peers or advisors)

Sprint 5: Test, Polish, Deliver

- Conduct usability testing with mock users or stakeholders
- Validate prediction outputs against holdout test set
- Ensure system responsiveness for non-technical users
- Prepare final documentation, user guide, and deployment script
- Record walkthrough demo video of app’s functionalities

- Present findings and demo to facilitator and peers
- Final polish and retrospective evaluation of project outcomes

Gantt Chart



Team Roles

Name	Role	Responsibilities
Luis	Fullstack developer	System integration, QA
Rosa	System’s designer	
Hussain	Backend Developer	
Victor	DBM	
Nomayer	Backend developer, tester	

Cost and Effort

Human resources needed to make the project happen:

Role	Hours	Rate	Cost
Project Manager	30	AUD 70	AUD 2,100
Frontend Dev	40	AUD 60	AUD 2,400
Backend Dev	45	AUD 60	AUD 2,700
DBM	35	AUD 45	AUD 1,575
Fullstack Dev	40	AUD 60	AUD 2,400
TOTAL	**180**		**AUD 11,175**

Risk Management

Every software development project faces uncertainties that can affect scope, quality, timeline or even user satisfaction. Identifying and mitigating risks early is key for delivering a stable and valuable solution. In the context of **TTrack**, potential risks range from data-related issues (such as incomplete or inconsistent survey inputs) to technical challenges in machine learning model performance and stakeholder management.

This section outlines the most significant risks identified during project planning, their likelihood, potential impact and strategies that we propose to mitigate them.

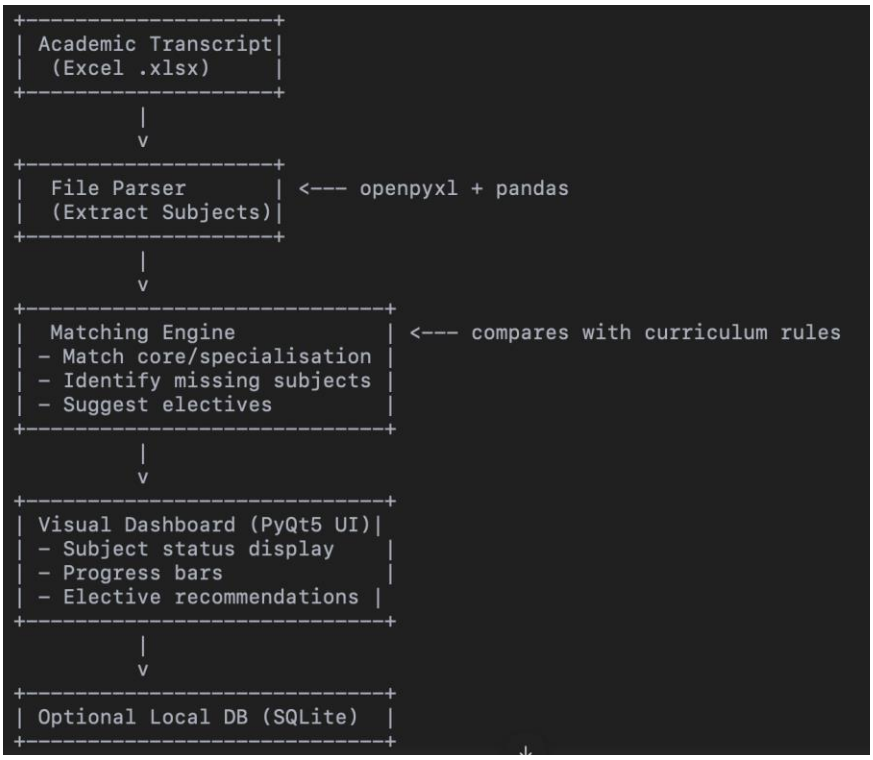
Risk	Likelihood	Impact	Mitigation
Curriculum format inconsistency	High	High	Validate against official Torrens format
Over engineering rule logic	Medium	Medium	Start with MVP rule layer and build from it
UI bloat or confusion	Low	Medium	Keep it user-first and minimal

Communication Plan

- Daily async updates via Microsoft Teams
- Weekly sync meetings with mentor/peers
- GitHub for version control and documentation
- OneDrive for mockups, diagrams and references

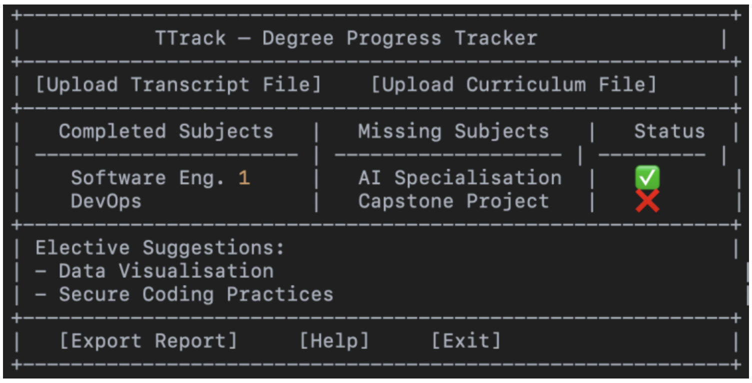
Appendices

a) Architecture Diagram



```
+--------------------+
| Academic Transcript|
|  (Excel .xlsx)     |
+--------------------+
          |
          v
+--------------------+
|  File Parser       | <--- openpyxl + pandas
|  (Extract Subjects)|
+--------------------+
          |
          v
+-----------------------------+
|  Matching Engine            | <--- compares with curriculum rules
| - Match core/specialisation |
| - Identify missing subjects |
| - Suggest electives         |
+-----------------------------+
          |
          v
+-----------------------------+
| Visual Dashboard (PyQt5 UI)|
| - Subject status display    |
| - Progress bars             |
| - Elective recommendations |
+-----------------------------+
          |
          v
+-----------------------------+
| Optional Local DB (SQLite)  |
+-----------------------------+
```

b) UI Wireframes / Mockups



```
+--------------------------------------------------------------+
|           TTrack — Degree Progress Tracker                  |
+--------------------------------------------------------------+
| [Upload Transcript File]    [Upload Curriculum File]         |
+--------------------------------------------------------------+
|   Completed Subjects    |   Missing Subjects    |   Status   |
| ----------------------- | -------------------- | ---------- |
|    Software Eng. 1      |   AI Specialisation   |    ✅       |
|    DevOps               |   Capstone Project    |    ❌       |
+--------------------------------------------------------------+
| Elective Suggestions:                                        |
| - Data Visualisation                                          |
| - Secure Coding Practices                                     |
+--------------------------------------------------------------+
|   [Export Report]     [Help]     [Exit]                      |
+--------------------------------------------------------------+
```

c) Dev Notes and Meeting Logs

Individual Report

Tasks Performed

As a full-stack developer, I led the architecture design and core implementation of TTrack. My responsibilities included setting up the file parsing logic using pandas and openpyxl, configuring the PyQt5 UI for local use and orchestrating the subject matching engine logic to compare completed subjects with curriculum requirements. I also handled integration between backend logic and frontend visualization, including progress tracking and elective suggestion rendering.

Challenges

The biggest challenge was mapping course codes accurately between transcript and curriculum sheets, especially considering inconsistent formatting in real-world transcripts. I also faced complexity when trying to keep the UI simple while still delivering rich information, such as course status and suggestions.

Lessons Learned

This project helped me apply software engineering principles in a real-life context. I deepened my skills in Agile project management, version control (via GitHub) and GUI frameworks. Working in sprints helped us deliver incrementally and handle scope adjustments without losing momentum.

I also learned how critical it is to clarify data assumptions early, especially when working with unstructured inputs like transcripts.

Finally, collaborating in a multidisciplinary group sharpened my communication and coordination skills, which I consider key in software delivery.

References

Ahmad, M. O., Markkula, J., Oivo, M. (2013). Factors affecting software maintainability from customer perspective. Journal of Systems and Software.

Cobb, C. G. (2015). The Project Manager's Guide to Mastering Agile. Wiley.

Ewusi-Mensah, K. (2003). Software Development Failures. MIT Press.

Heath, F. (2021). The Professional Scrum Master Guide. Packt Publishing.

Medallia. (n.d.). The #1 Experience Management Platform. Retrieved June 16, 2025, from https://www.medallia.com/platform

Qualtrics. (n.d.). Qualtrics by use case. Qualtrics. Retrieved June 16, 2025, from https://www.qualtrics.com/use-case/

SurveyMonkey. (n.d.). SurveyMonkey: The world's most popular survey platform. https://www.surveymonkey.com

Stephens, R. (2015). Beginning Software Engineering. Wrox Press.