**Project Test Plan**

**for**

**TTrack – Degree Tracker**

|  |  |
| --- | --- |
| **SDM404 - Assessment 3 – Project Test Plan** | |
| **Project Name** | TTrack – Degree Tracker |
| **Group #** | #1 |
| **Group Members Names** | Hussain Jameel – **A00180177**  Luis Guilherme de Barros Andrade Faria - **A00187785**  Nomayer Hossain - **A00176827**  Rosa Carolina Cortes Galvis – **A00193201**  Victor Javier Dorantes Meneses – **A00179705** |
| **Version** | 1.0.0 |
| **Date Created** | 23, July, 2025 |

Table of Contents

[1. Introduction 3](#_Toc1010490166)

[2. Test Objectives 3](#_Toc1434191742)

[3. Test Scope 3](#_Toc192170955)

[4. Test Strategy 4](#_Toc2055724862)

[5. Test Environment 5](#_Toc1006722486)

[6. Test Items 5](#_Toc281782210)

[7. Test Case Summary 6](#_Toc454508131)

[8. Test Case Results Summary 6](#_Toc1167986263)

[9. Glossary 7](#_Toc1555197491)

[10. References 8](#_Toc1029089288)

[11. Individual Contribution Log 8](#_Toc599450612)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

TTrack is a standalone desktop application designed to help Torrens University staff and students track academic progress by matching academic transcripts against prescribed degree curriculums. This test plan outlines the approach, scope, and strategy used to verify the accuracy, usability, and stability of the TTrack system before release.

# **Test Objectives**

The primary goal is to ensure TTrack meets its functional requirements and performs reliably across all supported environments. The objectives include:

* Verifying all core functions: file uploads, subject matching engine, results dashboard display and export features.
* Verify accuracy of subject matching and elective recommendations.
* Ensure user interface responsiveness and clarity for both technical and non-technical users.
* Confirm system stability when processing large datasets (~1,000 rows).
* Identify and track software defects with severity classification.

# **Test Scope**

**In Scope:**

* Transcript and curriculum file uploads (.xlsx)
* Matching engine logic and comparison
* Dashboard and result visualizations (summary tables, charts)
* Elective recommendation system
* Export functionality (CSV)
* UI preferences (light/dark mode)
* Database session storage (with *Supabase* cloud sync prototype)

**Out of Scope:**

* Real-time integration with Torrens academic systems.
* Web and mobile versions
* Multi-user collaboration or real-time sync.

# **Test Strategy**

The testing strategy combines both manual validation and automated testing to ensure comprehensive coverage of the critical features.

**Testing Levels:**

* **Unit Testing:** Validate isolated modules (e.g., file parser, matching engine)
* **Integration Testing:** Ensure that modules interact correctly (e.g., UI ↔ database)
* **System Testing:** Evaluate full application workflow and GUI responsiveness
* **User Acceptance Testing (UAT):** Feedback collected from actual Torrens students (group members)

**Test Types:**

* Functional Testing
* Usability Testing
* Non-functional Testing (performance, scalability)
* Concurrent testing
* Repeatable Test and Automated Regression Testing (after updates)

**Methods:**

* Manual testing using structured test cases
* Automated testing via pytest with >85% coverage
* Visual inspection of UI elements
* Static analysis using mypy, black, isort

**Defect Tracking:**

A bug report template is used (Excel) containing:

* Bug ID
* Severity (Low, Medium, High)
* Affected Module
* Steps to Reproduce
* Status (Open, In Progress, Resolved)

**Prioritization Focus:**

|  |  |
| --- | --- |
| **Priority** | **Area** |
| 🔴 High | File upload and matching |
| 🟡 Medium | Results visualizations |
| 🟢 Low | Export formatting, themes |

# **Test Environment**

|  |  |
| --- | --- |
| **Component** | **Specification** |
| OS | Windows 10, macOS Sonoma |
| Hardware | 8 GB RAM, Intel i5/M1 CPU |
| Python | 3.10+ with venv environment |
| GUI | PyQt5 |
| Libraries | pandas, openpyxl, matplotlib, scikit-learn |
| Packaging | PyInstaller for .exe and .app |
| Data | Excel files with valid curriculum and transcript structure |

# **Test Items**

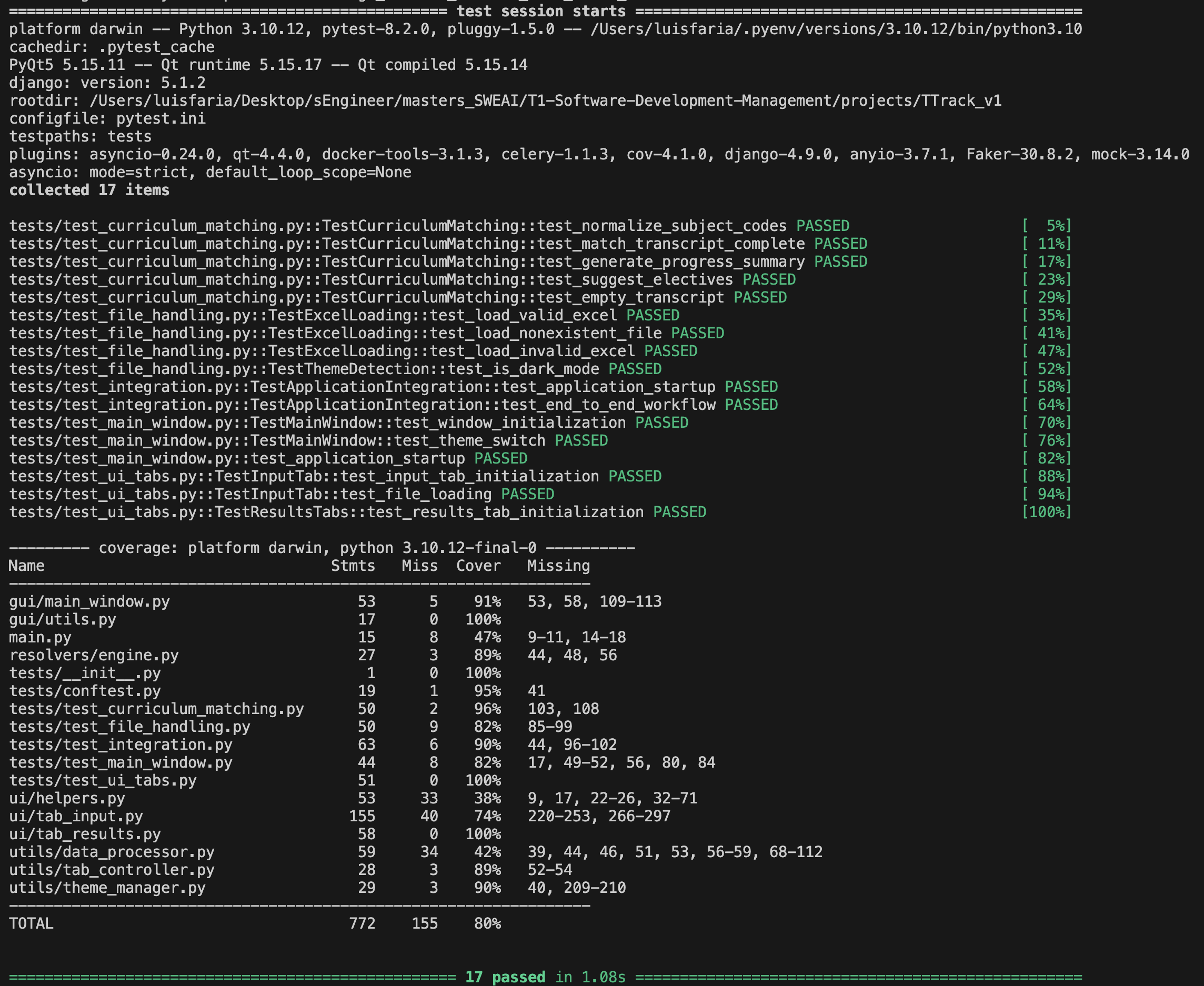
|  |  |
| --- | --- |
| **Module** | **Description** |
| Upload Transcript | File validation, data parsing, preview table |
| Upload Curriculum | Structure validation, data extraction |
| Matching Engine | Subject status calculation, fuzzy matching |
| Elective Suggestions | Based on unmatched electives in curriculum |
| Dashboard Analytics | Pie and bar charts for progress tracking |
| UI Theme | Light/Dark mode toggle |
| Export Features | PDF/CSV exports of matched data and dashboard |
| Upload Transcript | File validation, data parsing, preview table |
| Upload Curriculum | Structure validation, data extraction |
| Matching Engine | Subject status calculation, fuzzy matching |
|  |  |
|  |  |

# **Test Case Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected Result** |
| TC-01 | Upload valid transcript | Valid .xlsx | Data previewed without error |
| TC-02 | Upload invalid transcript | File with missing headers | Not possible to select file |
| TC-03 | Run matching engine | Uploaded transcript + curriculum | Subjects categorized correctly |
| TC-04 | Generate elective recommendations | Missing electives | Suggested list displayed |
| TC-05 | Export results | User clicks export | Processed data saved as CSV |
| TC-06 | View dashboard charts | Uploaded files + match results | Visual tables clearly indicating progress shown |
| TC-07 | Switch dark/light mode | Toggle in settings | Theme changes immediately |
| TC-08 | Large file test | 1,000 row Excel file | Processed within 5 seconds |

# **Test Case Results Summary**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Result** | **Comments** |
| TC-01 | Passed | Correct structure and preview shown |
| TC-02 | Passed | Error handling works as expected |
| TC-03 | Passed | Matching logic accurate |
| TC-04 | Passed | Elective list correctly generated |
| TC-05 | Passed | Exports functioned properly |
| TC-06 | Passed | Charts responsive and dynamic |
| TC-07 | Passed | Theme toggle worked as designed |
| TC-08 | Passed | Performance within acceptable range |



*Figure 1: Console log of robust test coverage using Pytest.*

Code Quality Standards

* Test Coverage: >85% with pytest
* Type Safety: mypy static analysis
* Code Style: black + isort formatting
* Documentation: Comprehensive docstrings + technical specs
* Performance: Profiled with cProfile for optimization

# **Glossary**

|  |  |
| --- | --- |
| **Term** | **Description** |
| **REQ** | Requirement |
| **SRS** | Software Requirements Specification |
| **PTP** | Project Test Plan |
| **PyQt5** | Python GUI Framework |
| **GUI** | Graphical User Interface |
| **BDD** | Behavior Driven Development |
| **CSV** | Comma-separated values |

# **References**

* Cobb, C. G. (2015). The Project Manager’s Guide to Mastering Agile. Wiley.
* Stephens, R. (2015). Beginning Software Engineering. Wrox.
* Ewusi-Mensah, K. (2003). Software Development Failures. MIT Press.
* Schwaber, K. & Sutherland, J. (2020). The Scrum Guide.
* Torrens University. (2025). SDM404 Assessment 2 Brief.

# **Individual Contribution Log**

|  |  |
| --- | --- |
| **Student Name** | **Contribution** |
| **Luis Faria** | Throughout the development lifecycle of TTrack, I played a central role in both the implementation and quality assurance processes of the system. As a core developer and team coordinator, my contributions covered full-stack features, user interface enhancements, backend logic, database integration, testing strategy, and documentation.  On the **frontend** side, I led the design and implementation of multiple user interface components using PyQt5, ensuring the application was intuitive, responsive, and visually aligned with both light and dark themes. I developed the “Student Records” tab, enabling users to view, filter, and export historical sessions. I integrated export-to-CSV functionality across various result tables, with timestamped filenames and compact UI layouts. These features improved user experience and allowed academic staff to save and analyze data outside the application.  On the **backend**, I implemented and refactored major services under a layered architecture, including the DatabaseManager for persistent storage via Supabase and the DataProcessor for data cleaning and subject matching. I maintained a clear separation of concerns between business logic and UI layers to support maintainability and future scalability.  Regarding quality assurance, I authored the Test Plan for Assessment 3, outlining objectives, scope, strategy, environment setup, and detailed test cases. I designed functional and non-functional tests to validate critical components such as file uploads, matching engine logic, dashboard visualization, and export features. I carried out these test cases manually across macOS and Windows environments to ensure consistent behavior. I documented all findings in the Bug Report Template, categorizing each issue by severity and providing suggested fixes.  I also ensured proper code quality by following formatting standards and introduced type hints across modules to support mypy static analysis. I emphasized documentation by writing clear docstrings and inline comments across the codebase, which helped align team members and facilitate handovers.  As project lead, I managed our GitHub repository using semantic versioning and GitFlow strategies. I maintained changelogs, tagged releases, and coordinated sprints using a version-controlled approach. I regularly presented progress to faculty, responded to stakeholder feedback, and iterated on features such as use case narratives, diagrams, and system architecture decisions based on professor guidance.  Lastly, I contributed significantly to the SRS and SDS documentation. I rewrote Section 3.2 to reflect accurate software interface integration and added detailed user stories and use case diagrams to match academic expectations. These enhancements ensured that our deliverables aligned with SDM404 standards and learning outcomes.  In summary, my contributions were made across planning, development, testing, and delivery. I ensured the system was functionally complete and also user-friendly, well-documented, and technically robust for real-world use within an academic context. |
| **Nomayer Hossain** | For the group project TTrack – Degree Tracker, my contributions for Assessment 3 – Project Test Plan focused on ensuring that our testing process was fully aligned with the functional requirements defined in Assessment 2.    Key contributions included:  - Test Items & Scope: I mapped the functional requirements from our SRS to corresponding test items, ensuring that each core feature (file uploads, matching engine, dashboard analytics, elective recommendations, export functionality, and UI themes) was included in the testing scope.  - Test Case Development: I authored detailed test cases for features such as Upload Transcript, Upload Curriculum, Run Matching Engine, and Generate Elective Recommendations. Each case included inputs, execution steps, expected results, and pass/fail criteria, and I ensured inclusion of both positive and negative scenarios.  - Test Execution: I carried out manual testing on both Windows 10 and macOS environments to verify cross-platform performance and usability. This included validating that the application processed large files (up to 1,000 rows) within the required performance benchmark of 5 seconds.  - Defect Reporting: I documented defects in our shared bug register, including one issue where validation messages were not consistently displayed for malformed curriculum files. I provided reproduction steps, screenshots, and assigned severity levels.  - Test Results Summary: I helped compile and verify the pass/fail outcomes in the Test Case Results Summary table, ensuring results were clear, concise, and linked to their respective test IDs.  - Traceability: I assisted in confirming that REQ IDs from the SRS matched the correct Test Case IDs, ensuring full requirement coverage.  One challenge was maintaining alignment between evolving requirements and our test design. Certain features underwent refinement after our initial draft, requiring adjustments to test cases to ensure accuracy.    Another difficulty was designing realistic negative test scenarios without introducing unrealistic edge cases that would not occur in normal user behaviour. Additionally, time constraints made it challenging to execute every test case across both OS platforms without overlap or duplication of effort among team members.  To ensure alignment, I created a requirements-to-test coverage matrix so that each functional requirement could be traced to at least one test case. This made it easier to identify gaps and update test cases when requirements changed.    To optimise execution time, I developed module checklists to track which cases were completed per platform. I also coordinated with team members via short MS Teams calls to review test design before execution.    For defect reporting, I adopted a standardised format with Bug ID, severity, affected module, reproduction steps, and status. This approach improved clarity and helped the lead developer prioritise fixes.  This project deepened my understanding of how clear, traceable test cases form the backbone of effective quality assurance. I improved my skills in:  - Understanding and refining test types.  - Writing structured test cases aligned to requirements.  - Applying both functional and non-functional testing approaches.  - Documenting defects with precise reproduction steps and impact analysis.  - Collaborating in a coordinated testing process across platforms.    Most importantly, I learned how to bridge the gap between requirements engineering (from Assessment 2) and quality assurance planning (Assessment 3), ensuring that testing fully validates intended system behaviour.    Through this contribution, I gained confidence in applying systematic testing practices and recognised the importance of test planning in delivering reliable, user-ready software. |
| **Hussain Jameel** | To develop this assessment, I focused on validating critical TTrack functionalities and identifying defects that could affect data accuracy and usability. My testing approach ensured that results aligned with both functional requirements and user expectations.  I executed test case TC-05 (Export results to CSV) to confirm that exported data matched the on-screen results and followed the correct format. I verified the column structure, data sequencing, and formatting, ensuring that exported files were accurate and ready for use without requiring additional manual adjustments.  During execution, I identified bug B-004 (Mismatch count doesn’t update). When running the matching engine with incorrect subjects, the mismatch count failed to refresh. I documented the issue with reproduction steps and screenshots, rated it low severity, and confirmed the resolution after it was fixed.  I also ran test case TC-02 (Invalid curriculum format), uploading a file with missing required fields. The system correctly rejected the file, and the test passed. This validated that input validation mechanisms prevented incomplete or malformed data from entering the system.  Another usability issue I reported was bug B-005 (Filename not suggested in export). Without a default filename, users had to manually type one during each export. I suggested an enhancement to auto-generate meaningful filenames, which was implemented, improving workflow efficiency.  Throughout the process, I worked closely with the lead developer and test designer, providing feedback to ensure test coverage addressed both functionality and user experience. I participated in discussions on defect resolution and validated that fixes met design expectations.  Lessons learned: This testing cycle reinforced the value of detailed defect documentation for efficient resolution and the importance of aligning functional validation with usability improvements. I strengthened my skills in functional testing, defect reporting, and collaborative problem-solving while contributing to a smoother, more reliable TTrack user experience. |
| **Rosa Galvis** | To develop this assessment, I focused on validating key TTrack functionalities and detecting defects related to the accuracy of the desired results:  I ran test case TC-05 (Export results to CSV), verifying that the exported file accurately reflected the information displayed in TTrack and adhered to the expected format. The test included reviewing the column structure and preserving the data order.  I identified and logged bug B-004 (Mismatch count doesn't update), detecting that, after running the matching engine with incorrect subjects, the mismatch count did not update as expected. I documented the issue with reproduction steps and classified it as low severity. The defect was subsequently corrected.  I worked closely with the lead developer and test designer for TTrack, providing observations and feedback throughout the execution to ensure that validations covered both functional requirements and user experience aspects.  Lessons learned: This role allowed me to strengthen my skills in functional testing and clear defect documentation, as well as understand how QA efforts are enhanced when supported by a detailed and well-structured technical plan. |
| **Victor Dorantes** | Provided early-stage feedback during the planning phase of the Test Plan, contributing to discussions on the selection of test cases and ensuring they aligned with the functional requirements of the TTrack application.  Participated in initial review meetings, helping refine the scope and approach for the testing strategy.  While not directly involved in the execution of test cases or bug tracking, my input during team discussions helped confirm the overall direction of the testing process.  I also reviewed parts of the draft document for clarity and consistency, offering minor suggestions to improve wording. I supported the conceptual development phase, ensuring alignment between the test plan and the original project objectives. |