city college
plymouth

Network Traffic Profiler Dashboard

Client - Plymouth City College [Tomek Bergier]

Group 15

Team Leader - Amelia Lee Sze Yai
(write your full name & role pls:))
Scrum Master - Timothy Birtles
Tomiris
Sofia

CONTENT PAGE

IMPORTANT NOTES

- Each section/subsection should have the names of the member who wrote it.
- Try to distribute according to team contributions (equally if possible, but if not, the ones who did less technical work should contribute more here).
- Recommended page number is around 80 pages. Upper limit is 100 pages (penalty if crossing this). There is no minimum page limit. The key point is to be concise.

1) Introduction (Sofia , Amelia)

– introduce client, problem statement, brief system requirements and any other necessary details on the project

Client Introduction

The Network Traffic Profiler Dashboard project is being developed for the client, Plymouth City College, with Tomek Bergier as the client representative

Problem statement & Part of system requirements & Target Users

Modern network security tools are often complex, difficult to configure, and aimed at technical experts. Smaller organisations and non-specialist users may struggle to interpret raw PCAP data or understand network behaviour using existing tools.

This project addresses this gap by creating a lightweight, user-friendly dashboard that allows users to upload PCAP files and receive understandable insights into their network traffic.The system is designed to extract and process meaningful network flow features, apply machine learning to classify traffic activity and detect abnormal behaviour, and present results in an intuitive visual format.

The target users are non-technical staff from small and medium-sized enterprises (SMEs). By combining data processing, machine learning, and visualisation into a single workflow, the dashboard provides actionable insights without requiring advanced networking expertise.

2) Project Management (All members)

Project planning material (content from semester 1) (Tomiris)

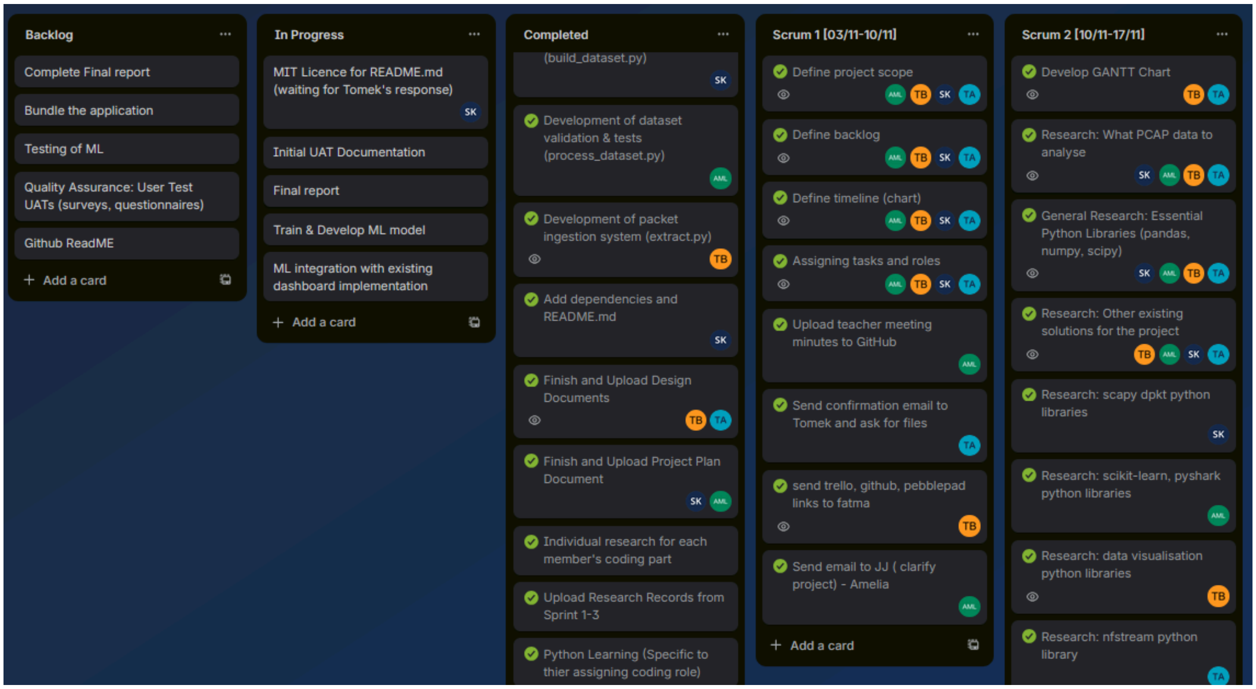
Sprint planning and execution (Tim)

To ensure development remained aligned with both the initial design requirements and our client’s requirements, the Agile methodology was adopted. This approach was selected for its emphasis on adaptive planning and capacity for rapid response to change (Kumar and Bhati, 2012).

Project management began with the creation of a comprehensive product backlog, which was split into Semester One and Semester Two milestones. We operated in biweekly sprints which provided an excellent balance between development and evaluation. During planning sessions, tasks were subdivided based on technical prerequisites and estimated completion time, ensuring the workload assigned to each cycle was realistic.

We utilised Trello as our primary Kanban style project management tool, providing a single location to track the project status. The project's Trello board was organised into columns, 'Backlog', 'In progress', 'Completed', in addition to individual columns for each weekly Scrum.

At the start of each sprint, we met to evaluate and refine the backlog, ensure tasks were still relevant, and delegate responsibilities based on expertise and time availability. Twice-weekly standup meetings were held which focused on progress and pitfalls, ensuring progress was not stalled by technical hurdles nor lack of communication.



Backlog

- Complete Final report
- Bundle the application
- Testing of ML
- Quality Assurance: User Test UATs (surveys, questionnaires)
- Github ReadME
- + Add a card

In Progress

- MIT Licence for README.md (waiting for Tomek's response) SK
- Initial UAT Documentation
- Final report
- Train & Develop ML model
- ML integration with existing dashboard implementation
- + Add a card

Completed

- (build_dataset.py) SK
- ✓ Development of dataset validation & tests (process_dataset.py) AML
- ✓ Development of packet ingestion system (extract.py) TB
- ✓ Add dependencies and README.md SK
- ✓ Finish and Upload Design Documents TB TA
- ✓ Finish and Upload Project Plan Document SK AML
- ✓ Individual research for each member's coding part
- ✓ Upload Research Records from Sprint 1-3
- ✓ Python Learning (Specific to thier assigning coding role)

Scrum 1 [03/11-10/11]

- ✓ Define project scope AML TB SK TA
- ✓ Define backlog AML TB SK TA
- ✓ Define timeline (chart) AML TB SK TA
- ✓ Assigning tasks and roles AML TB SK TA
- ✓ Upload teacher meeting minutes to GitHub AML
- ✓ Send confirmation email to Tomek and ask for files TA
- ✓ send trello, github, pebblepad links to fatma TB
- ✓ Send email to JJ (clarify project) - Amelia AML
- + Add a card

Scrum 2 [10/11-17/11]

- ✓ Develop GANTT Chart TB TA
- ✓ Research: What PCAP data to analyse SK AML TB TA
- ✓ General Research: Essential Python Libraries (pandas, numpy, scipy) SK AML TB TA
- ✓ Research: Other existing solutions for the project TB AML SK TA
- ✓ Research: scapy dpkt python libraries SK
- ✓ Research: scikit-learn, pyshark python libraries AML
- ✓ Research: data visualisation python libraries TB
- ✓ Research: nfstream python library TA

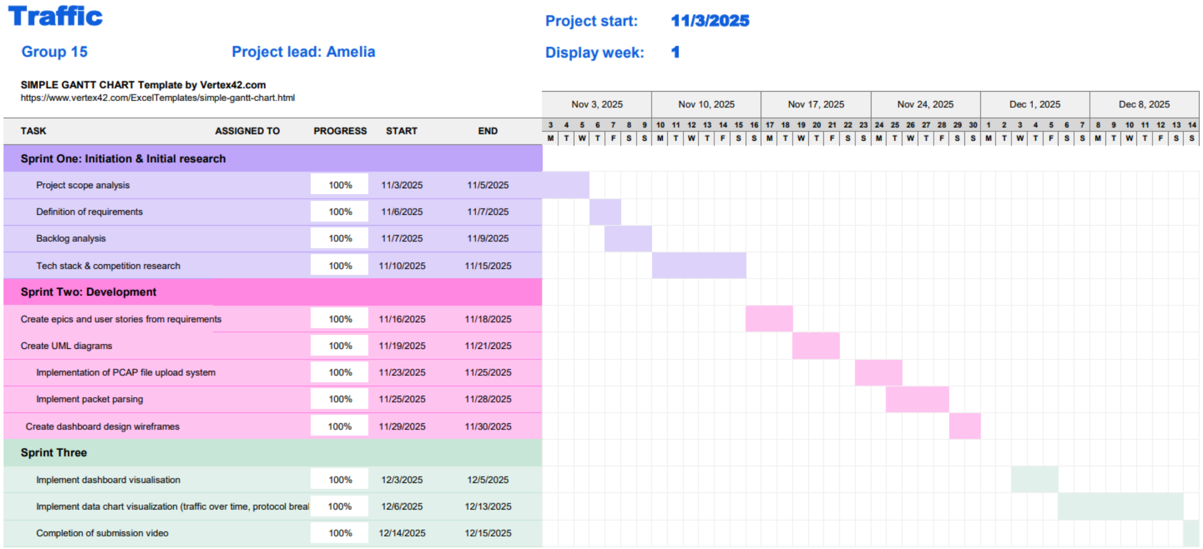
Figure 1 - Trello board snapshot

Risk assessment and mitigation (Sofia)

Project schedule (Tomiris)

The project spanned 12 weeks across two semesters. The schedule was designed to ensure a logical progression from foundational system development to advanced machine learning integration, while maintaining alignment with client requirements and submission deadlines.

To support time management and task coordination, detailed Gantt charts were developed for each semester. These charts outlined task durations, sequencing, and key dependencies, enabling clear visualisation of the project timeline and ensuring transparency in workload allocation.



Traffic

Group 15 Project lead: Amelia

Project start: 11/3/2025

Display week: 1

SIMPLE GANTT CHART Template by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

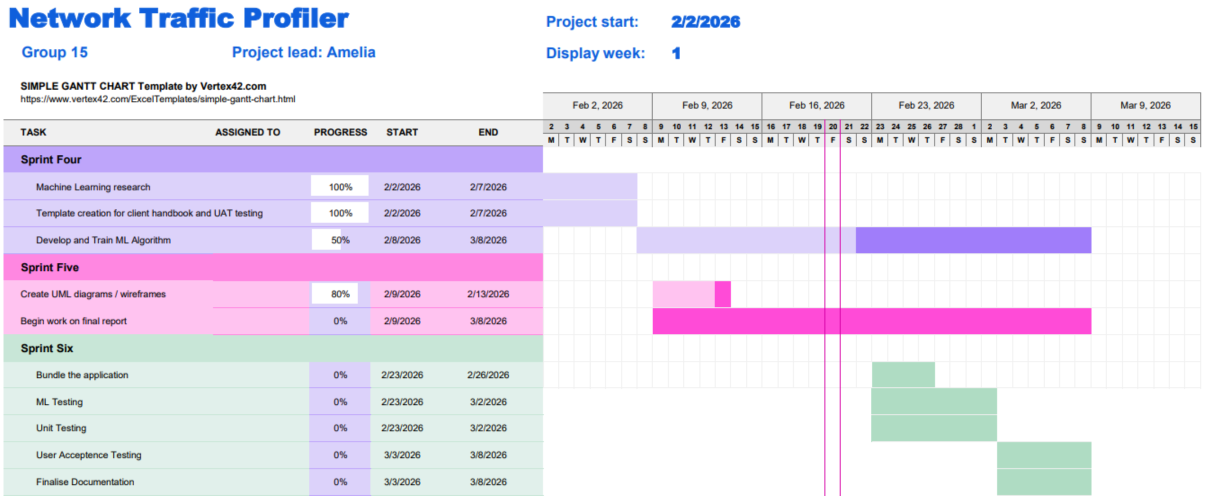
TASK	ASSIGNED TO	PROGRESS	START	END
Sprint One: Initiation & Initial research				
Project scope analysis		100%	11/3/2025	11/5/2025
Definition of requirements		100%	11/6/2025	11/7/2025
Backlog analysis		100%	11/7/2025	11/9/2025
Tech stack & competition research		100%	11/10/2025	11/15/2025
Sprint Two: Development				
Create epics and user stories from requirements		100%	11/16/2025	11/18/2025
Create UML diagrams		100%	11/19/2025	11/21/2025
Implementation of PCAP file upload system		100%	11/23/2025	11/25/2025
Implement packet parsing		100%	11/25/2025	11/28/2025
Create dashboard design wireframes		100%	11/29/2025	11/30/2025
Sprint Three				
Implement dashboard visualisation		100%	12/3/2025	12/5/2025
Implement data chart visualization (traffic over time, protocol breal		100%	12/6/2025	12/13/2025
Completion of submission video		100%	12/14/2025	12/15/2025

Figure 2 - Semester 1 Gantt Chart

Semester 1 focused on establishing the core system architecture and delivering a viable prototype.

- The first sprint of semester 1 was dedicated to defining the project scope, allocating team roles and conducting technical research related to network traffic analysis and PCAP data processing.
- The second sprint involved preparing the design documentation and project plan. During this phase, scripts were developed to extract and validate features from PCAP files, and the dataset was constructed.
- The third sprint was dedicated to implementing the dashboard interface and conducting unit testing to validate system functionality. This phase concluded with preparation for the interim submission milestone, ensuring that a functional prototype was delivered within the allocated timeframe.

All Semester 1 deliverables were completed according to the planned schedule.



Network Traffic Profiler

Group 15 Project lead: Amelia

Project start: 2/2/2026

Display week: 1

SIMPLE GANTT CHART Template by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

TASK	ASSIGNED TO	PROGRESS	START	END
Sprint Four				
Machine Learning research		100%	2/2/2026	2/7/2026
Template creation for client handbook and UAT testing		100%	2/2/2026	2/7/2026
Develop and Train ML Algorithm		50%	2/8/2026	3/8/2026
Sprint Five				
Create UML diagrams / wireframes		80%	2/9/2026	2/13/2026
Begin work on final report		0%	2/9/2026	3/8/2026
Sprint Six				
Bundle the application		0%	2/23/2026	2/26/2026
ML Testing		0%	2/23/2026	3/2/2026
Unit Testing		0%	2/23/2026	3/2/2026
User Acceptence Testing		0%	3/3/2026	3/8/2026
Finalise Documentation		0%	3/3/2026	3/8/2026

Figure 3 - Semester 2 Gantt Chart

Semester 2 shifted focus toward enhancing the system with machine learning capabilities and preparing the final deliverable.

- The first sprint involved research and selection of appropriate machine learning algorithms, alongside early preparation of final report documentation.
- The second sprint focused on updating UML diagrams to reflect architectural changes, developing and training the machine learning model using the prepared dataset, and integrating it into the existing dashboard framework.
- The final sprint was allocated to model evaluation, user acceptance testing (UAT), performance validation, and system refinement. Documentation was finalised in preparation for submission.

Project Milestones

Milestone	Description	Status
Viable Prototype Developed	Core system implementation and dashboard functional	Completed
Interim Submission	Submission of prototype and documentation	Completed
Machine Learning Integration	ML model developed and integrated into system	In Progress
Final Submission	Complete system, testing, and documentation	Pending

These milestones provided structured checkpoints that supported progress monitoring and accountability throughout the project lifecycle.

All Semester 1 deliverables were completed within their allocated timeframes, and no critical deadlines were missed. The phased structure of the schedule ensured a smooth transition from system development to machine learning integration, reducing rework and maintaining development momentum. The use of sprint-based planning supported incremental progress, clear task ownership, and continuous monitoring of deliverables.

Client management (Amelia)

Effective client management was a key priority and critical to the success of this project, ensuring continuous alignment with the client's requirements, expectations, and timelines. A strong, collaborative relationship was maintained throughout the project lifecycle through the following methods:

1. **Communication and Collaboration**

Regular communication channels were established to maintain transparency and collaboration:

- Bi weekly meetings were held every Thursday during the first semester and every Wednesday during the second semester, either in person at Portland Square, Room A03, or via Microsoft Teams. These meetings focused on project timeline and progress, upcoming tasks, short and long term plans, client feedback and product demonstration. All requirements, comments, and feedback were recorded in client meeting minutes.
- Consistent email updates were used to provide documentation, approvals, and follow ups on key decisions.
- Interim dashboard demonstrations were presented to the client for review to ensure alignment with expectations.

2. **Feedback and Change Management**

Client feedback was actively incorporated into development. This included:

- Adjusting dashboard visualizations based on client feedback
- Refining AI model output presentation according to client preferences
- Prioritising requested features or modifications within sprint cycles

Any suggested changes were formally recorded, evaluated for their impact on timeline and scope, and either approved or deferred based on mutual agreement. Continuous monitoring of deliverables against initial requirements helped manage expectations.

3. **Documentation and Approval**

All major milestones, including project requirements, design choices and the final product, required client approval. Approvals were documented through emails, signed forms, or recorded meeting minutes. The client's formal sign off also ensured that the delivered dashboard met agreed specifications and expectations.

3) Detailed System Design (Tim & Tomiris)

– user story, ERD, prototype screenshots, HIPO chart, state diagrams, DFDs, sequence diagrams flowcharts, UML etc. Screenshots of snippets that you think are important can be placed here as well, but no need for full source code, just link to git repo.

Potential code snippets: Extract pcap data, ML training?

4) Quality Assurance (Sofia & Amelia)

– (post test: surveys & questionnaires), documentation on unit test ,UAT on methodology, results (Sofia)

- post-project support and any other quality assurance measures taken (Amelia)

Ensuring high quality was a priority throughout the development of the Network Traffic Profiler Dashboard. A combination of rigorous testing, structured reviews, and post-project support measures was implemented to maintain reliability, usability, and alignment with the client's requirements.

Post- project Support

After project completion, the following support measures were planned to ensure smooth adoption and continued functionality to ensure that the client could successfully use and maintain the dashboard:

Client Handover: A comprehensive walkthrough of the dashboard was conducted, demonstrating installation, configuration, operation, and troubleshooting procedures.

Documentation Provision: Provided the client with up to date README files, user manuals, design diagrams, annotated source code, and testing reports. This ensures that both current users and future developers can understand and extend the system if needed.

Knowledge Transfer: Detailed technical explanations and diagrams were shared to enable future developers or IT teams to integrate the dashboard into larger systems or perform further maintenance.

Other quality assurance measures taken

Several quality assurance practices were applied throughout the project lifecycle to maintain high standards:

Version Control and Peer Code Reviews: GitHub was used to track all code changes. Team members reviewed each other's code to prevent errors and ensure consistent code quality. Clear commit messages and structured branching improved traceability.

Continuous Testing: Beyond unit tests, manual testing of dashboard functions, including uploads, interactive charts, tables, and ML model outputs, was conducted to validate real-world performance and ensure accuracy and robustness.

- Client check in General group review (Every sprint)
- UI/UX Evaluation
- Performance and Data Quality Monitoring
- Guide (Testing & validation guide)

5) Conclusion (Tim, Tomiris)

– summarize your project achievements and merits.

6) References (3 References Each)

– any referenced sites/papers you cited within your report. Recommended 10-15 References.

Kumar, G. and Bhati, P.K. (2012) 'Impact of Agile Methodology on Software Development Process,' International Journal of Computer Technology and Electronics Engineeri, 2(4). https://www.researchgate.net/profile/Gaurav-Kumar-175/publication/255707851_Impact_of_Agile_Methodology_on_Software_Development_Process/links/00b49520489442e12d000000/Impact-of-Agile-Methodology-on-Software-Development-Process.pdf.

7) Appendix

– attach the 1000 word CW Submission Evidence and Evaluation file for each member as appendix. This will not count towards the page limit.