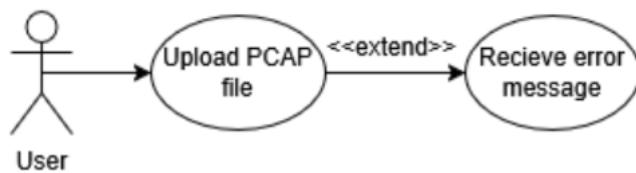


Epics and user stories

Epic One: PCAP Ingestion

User stories

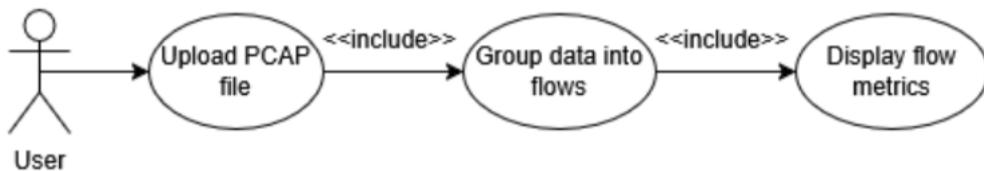
1. As a user I want to upload a PCAP file so that I can analyse my network traffic.
2. As a user I want to know if the analysis is still in progress, so I am aware the app is working.
3. As a user I want to be told if the PCAP file is corrupted or invalid so that the analysis fails safely.
4. As a **user** I want to be able to re-upload a new file so that I can analyse multiple datasets.



Epic Two: Data extraction and processing

User stories

1. As a user I want network traffic to be grouped into flows so that I can understand patterns.
2. As a user I want to view flow metrics so that I can interpret users' network behaviour.
3. As a user I want to be notified if my PCAP file contains no valid IP packet data.
4. As a user I want the system to be able to process large files without freezing.



Epic Three: Dashboard UI

User stories

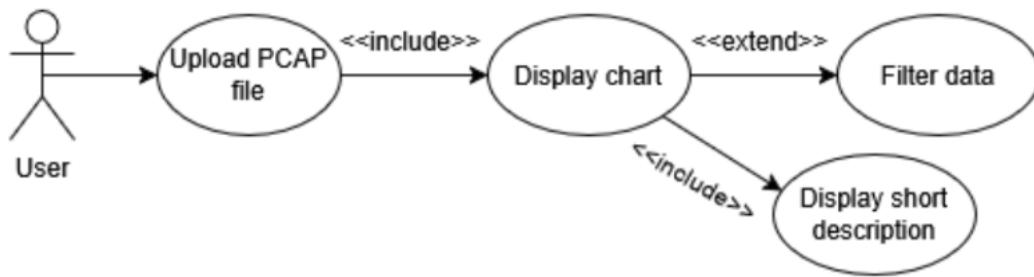
1. As a user I want a simple workflow so that I can easily analyse my data without advanced technical knowledge

2. As a user I want a clean and intuitive interface so that I can use the tool with minimal training
3. As a user I want the UI to function without crashing or freezing so that I can use the UI without hindrance.
4. As a user I want to see flow classifications in the dashboard so that I can understand traffic behaviour visually.
5. As a user I want anomalies to be highlighted in charts and tables so that I can easily spot them.
6. As a user I want to filter traffic by predicted class and anomaly status so that I can explore specific behaviours.

Epic Four: Data visualisation

User stories

1. As a user I want to be able to view interactive charts that show traffic size and flow statistics over time so that I can visually understand the data.
2. As a user I want to be able to filter the dashboard by protocol, time, and classification so that I can view specific activities.
3. As a user I want short descriptions for each chart so that I can understand what the data represents and means.
4. As a user I want summarised statistics so that I can make quick insights.



Epic Five: Documentation

User stories

1. As a user I want straightforward documentation so that I can understand how to get started.
2. As a user I want straightforward documentation of key network concepts so that I can understand the data and what it means
3. As a user I want documentation explaining how classifications are generated so that I understand system behaviour.
4. As a user I want documentation describing what the anomaly means and how it works so that I do not misinterpret results.

Epic Six: Code Maintainability

User stories:

1. As a developer I want the code to be modular so that I can update individual components without breaking others.

Epic Seven: Performance and Error Handling

User stories:

1. As a developer I want data extraction and validation to be efficient so that large PCAP files do not incur long delays.
2. As a developer I want the dashboard application to display a readable, user-friendly error if the application crashes.
3. As a developer I want to conduct tests for each pipeline stage so that changes can be validated.

Epic Eight: User Action Classification (ML)

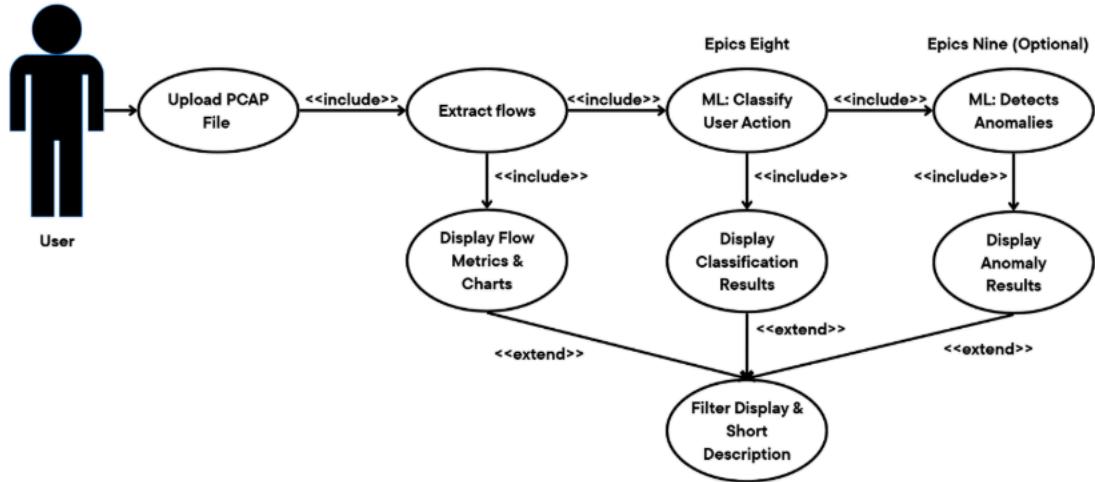
User Stories:

1. As a user I want network flows to be automatically classified so that I can understand user behaviour.
2. As a user I want each flow to have a label so that I can see what type of activity it represents.
3. As a user I want classification to run automatically after PCAP upload so that I do not need to trigger it manually.

Epic Nine: Anomaly Detection (ML – optional)

User Stories:

1. As a user I want anomalous flows to be clearly marked so that they stand out from normal traffic.
2. As a user I want the system to detect unusual or suspicious network behaviour so that I can identify potential threats or errors.



Epic Ten: ML Integration & Model

1. As a developer I want ML inference to run efficiently so that the dashboard updates without long delays.
2. As a developer I want to handle model errors gracefully so that the system does not crash if prediction fails.
3. As a developer I want to save and reuse trained models so that I do not need to retrain them every time.

