Software Requirements Specification (SRS)

Project: Bandwidth Monitoring System

Team Members:

- V. Nithin Reddy – SE22UCSE278
- M. Panvi Tej – SE22UCSE194
- N. Kushwanth Reddy – SE22UCSE177
- C. Aryan – SE22UCSE035
- E. Anvith Tej – SE22UCSE089
- B. Jathin Reddy – SE22UCSE118
- A. Hemanth – SE22UCSE109

1. Introduction

1.1 Purpose

This document defines the functional and non-functional requirements for the Bandwidth Monitoring System, a lightweight web-based tool that monitors network bandwidth usage in real time. It tracks device-level and application-level traffic and generates alerts when usage exceeds specified thresholds.

1.2 Scope

The system is intended for small offices, educational labs, or home networks. It provides real-time usage data, logs historical bandwidth, tracks connected devices, and classifies bandwidth usage by application. A web dashboard allows users to visualize and analyze the data.

1.3 Intended Audience

- Network Administrators
- Students and Educational Institutions
- Developers interested in network analytics
- Project Reviewers and Faculty

1.4 Definitions

- Bandwidth: The amount of data sent or received over a network interface.
- Flask: A Python-based web framework used for developing the backend API.
- psutil: A Python library for retrieving system and network resource usage.

2. Overall Description

2.1 Product Perspective

This system is a standalone application that uses local machine resources and SQLite for storage. It provides REST APIs and a web-based UI to present data to the user.

2.2 Product Functions

- Real-time bandwidth monitoring
- Bandwidth usage logging (per interface, per device, per application)
- Threshold-based alerting
- Device and application categorization
- Web-based dashboard for visualization
- Historical usage reporting

2.3 User Classes and Characteristics

Admin: Can start/stop the monitoring, configure thresholds
Viewer: Can view current and historical data via the dashboard

2.4 Operating Environment

- Python 3.10+
- SQLite database
- Web browser (for accessing dashboard)
- Works on Windows, Linux

2.5 Design and Implementation Constraints

- Uses psutil, which may require elevated permissions
- Tracks local device/application usage only (not across routers)
- Not designed for high-traffic enterprise-grade networks

3. System Features

Real-Time Bandwidth Monitoring

Continuously fetch bandwidth data using psutil.net_io_counters().

Device Tracking

Records IP, MAC, bytes sent/received, and top application per device.

Application Usage Monitoring

Monitors bandwidth usage per application.

Alerts and Thresholds

Generates alert if total bandwidth usage exceeds a specified threshold.

Dashboard Visualization

Provides a web-based dashboard (index.html) for real-time and historical data.

4. External Interface Requirements

4.1 User Interface

- Accessible via web browser on http://localhost:5000
- Provides charts for total bandwidth, top interfaces, devices, and apps

4.2 Hardware Interfaces

- System should have at least one network interface
- No external hardware required

4.3 Software Interfaces

- Flask: Serves the API and HTML pages
- SQLite: Local database storage
- Matplotlib: For generating plots in PNG

4.4 Communications Interfaces

RESTful APIs served over HTTP

5. Non-functional Requirements

- Performance: Updates bandwidth stats every 5 seconds
- Reliability: Background thread for monitoring with exception handling
- Usability: Simple web interface with charts, no login needed
- Maintainability: Modular, documented Python code
- Portability: Platform-independent, works on Windows and Linux

6. Appendices

A. Database Tables

- bandwidth_usage(timestamp, interface, bytes_sent, bytes_received, total_bytes)
- device_usage(timestamp, ip_address, mac_address, bytes_sent, bytes_received, top_application)
- application_usage(timestamp, application_name, bytes_sent, bytes_received)
- alerts(timestamp, message, threshold, actual_value, resolved)

B. API Endpoints

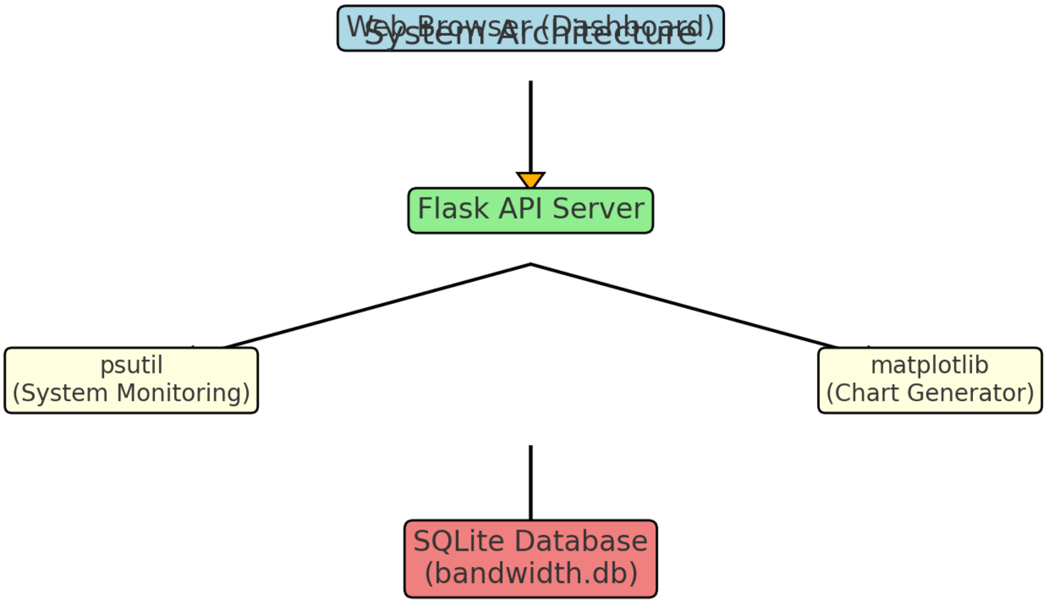
- /api/current_usage
- /api/historical
- /api/device_usage

- /api/application_usage
- /api/alerts
- /api/usage_report

C. Diagrams

1. System Architecture

The diagram below shows the architecture of the Bandwidth Monitoring System, including the web UI, backend API, monitoring services, and database.


System Architecture
Web Browser (Dashboard)
Flask API Server
psutil
(System Monitoring)
matplotlib
(Chart Generator)
SQLite Database
(bandwidth.db)