**Topic 2 – Benchmark - Milestone 1: Final Draft**

Steven Merkling

College of Engineering and Technology, Grand Canyon University

CST-590-O500: Computer Science Capstone Project

Dr. Aiman Darwiche

August 27, 2025

**Raspberry Pi 5 Edge ML Traffic Monitoring System**

**Project:** Raspberry Pi 5 Edge ML Traffic Monitoring System

**Authors:** Technical Team / Project Team / Documentation Team / Research Team

**Organization:** College of Engineering and Technology, Grand Canyon University

**Project Manager:** Steven Merkling

**Date:** August 27, 2025

**Project Overview and Project Objectives (C2.3)**

**Project Overview**

The Raspberry Pi 5 Edge ML Traffic Monitoring System addresses the growing concern of speeding and unsafe driving in residential neighborhoods. Traditional traffic monitoring and speed detection systems are often prohibitively expensive, making them inaccessible to many communities. This project aims to develop a cost-effective, accessible, and reliable solution for real-time vehicle speed monitoring using affordable hardware and open-source software.

The system leverages a Raspberry Pi 5, Sony IMX500 AI camera, and OmniPreSense OPS243-C radar sensor to detect, measure, and log vehicle speeds at the edge. By processing data locally, the system reduces bandwidth requirements, enhances privacy, and enables rapid responses to traffic events. The project is designed to empower communities and local authorities to proactively address speeding issues and improve road safety without significant financial burden.

**Project Objectives**

1. **Accuracy:** Achieve vehicle speed detection within ±5% of manual measurement tools.
2. **Cost-Effectiveness:** Deliver a solution at least 50% less expensive than traditional traffic monitoring systems.
3. **Reliability:** Ensure consistent operation under varied environmental conditions.
4. **Data Quality:** Provide accurate, analyzable data for community and authority use.
5. **Community Impact:** Enable positive changes in driver behavior and improve perceptions of safety in residential areas.

**Significance and Impact**

By making advanced traffic monitoring technology affordable and accessible, this project has the potential to transform how communities address speeding and road safety. The use of open-source hardware and software allows for widespread adoption and customization. The project also contributes new knowledge to the field of edge-based traffic monitoring, demonstrating the feasibility and benefits of local data processing for privacy and responsiveness.

The Raspberry Pi 5 Edge ML Traffic Monitoring System aims to create safer streets, empower citizen-led safety initiatives, and serve as a model for other communities seeking practical, scalable solutions to traffic safety challenges.

**Background**

The increasing number of vehicles and the growing need for enhanced road safety in residential areas have become significant concerns for communities and local authorities. Speeding in residential neighborhoods poses serious risks, potentially leading to accidents, injuries, and fatalities among pedestrians, cyclists, and other vulnerable road users. Traditional traffic monitoring and speed detection systems can be effective, but they often come with excessive costs, making them inaccessible to many communities.

Advancements in affordable and accessible technology, such as the Raspberry Pi, offer new possibilities for developing cost-effective solutions for traffic speed monitoring. Communities are increasingly seeking innovative ways to enhance road safety without imposing significant financial burdens. Affordable, DIY solutions leveraging open-source technology and readily available components can empower local governments and community groups to proactively address speeding issues.

This project, named "Raspberry Pi 5 Edge ML Traffic Monitoring System," aims to develop and test a proof-of-concept for a cost-effective residential traffic speed monitoring system using Raspberry Pi. By prioritizing affordability, ease of implementation, and effectiveness, this project seeks to address the gap in accessible traffic monitoring solutions and contribute to safer residential environments.

**Research Questions and Hypotheses**

**Research Questions:**

* How accurately can a Raspberry Pi-based system detect vehicle speeds compared to manual tools?
* Is the system cost-effective compared to traditional solutions?
* Is the system reliable under various environmental conditions?
* What is the impact of the system on community safety and perceptions?

**Hypotheses:**

* The system will accurately detect vehicle speeds within ±5% of manual tools.
* The overall cost will be at least 50% lower than traditional systems.
* The system will demonstrate consistent performance and reliability under various conditions.
* The system will lead to measurable reductions in speed and improved community safety perceptions.

**Expected Outcomes**

* Accurate speed measurement within specified tolerance
* Significant cost savings compared to traditional systems
* Reliable operation under various conditions
* Positive changes in driver behavior and community safety perceptions

**Project Scope and Objectives**

The Raspberry Pi 5 Edge ML Traffic Monitoring System is designed to provide real-time vehicle detection, speed measurement, and traffic analytics at the edge. The system leverages a Raspberry Pi 5, an AI-enabled camera, and an OmniPreSense OPS243-C radar sensor to process video and radar data locally, sending only processed results to cloud services for aggregation and reporting. This approach reduces bandwidth, increases privacy, and enables rapid response to traffic events.

Objectives:

* Deploy a low-cost, scalable, and reliable edge-based traffic monitoring solution
* Integrate ML/AI for vehicle detection, classification, and anomaly detection
* Fuse camera and radar data for accurate speed and event measurement
* Provide real-time dashboards and cloud-based analytics

Key challenges include hardware integration, algorithm performance, and ensuring data privacy. Benefits include reduced bandwidth usage, improved privacy, and rapid response to traffic events.

**Project Scope Details**

Work Breakdown (see Project Management Summary for full details):

* Hardware setup (Raspberry Pi, camera, radar, storage)
* Software environment setup (OS, Python, ML libraries)
* Basic vehicle detection (TensorFlow Lite, OpenCV)
* Radar integration and data fusion
* Web interface and API development
* System integration, testing, and documentation

Teams/Stakeholders: Technical Team, Project Team, Documentation Team, Research Team. Roles: Hardware lead, software lead, documentation lead, project manager (see Project Management Summary for details).

Resources: Hardware components, software tools, cloud services (optional).

**Project Completion**

**Completion Criteria:**

1. **Accuracy**: The system consistently and accurately detects and measures vehicle speeds within a specified tolerance (±5%).
2. **Cost-Effectiveness**: The project remains within the budget using affordable, readily available components and is at least 50% less expensive than traditional systems.
3. **Reliability**: The system operates consistently under various environmental conditions over an extended period.
4. **Data Quality**: Effective data logging and storage solutions are implemented, with accurate and analyzable data collected.
5. **Community Impact**: Positive feedback from community members and local authorities, and measurable improvements in traffic safety and driver behavior in monitored areas.

**Assumptions:**

* Affordable and reliable components (e.g., Raspberry Pi, radar sensors) are available for purchase.
* The system can be seamlessly integrated without major compatibility issues.
* Sufficient technical expertise is available to develop and test the system.
* Community and local authorities are willing to support and provide feedback on the project.
* Internet access is available for real-time data display and cloud storage.

**Constraints:**

* Budget: The project must remain within the specified budget constraints.
* Time: The project must be completed within the specified timeline, with all tasks finished as scheduled.
* Resources: Limited availability of technical resources and personnel.
* Environmental Conditions: The system must function reliably under varied weather and lighting conditions.
* Regulatory Compliance: The project must comply with local traffic and surveillance laws.

**Project Controls**

**Risk Management Table:**

| **Event Risk** | **Probability** | **Impact** | **Mitigation** | **Contingency Plan** |
| --- | --- | --- | --- | --- |
| Sensor Accuracy - The sensors might not accurately detect vehicle speeds under different conditions | Low | High | Conduct thorough testing under various conditions to calibrate and validate sensor accuracy. Implement software filters and adjustments to improve reliability. | Research and source multiple sensors to ensure at least one will work. Find sensors that have worked for this type of application in the past. |
| Incompatibility between hardware components (sensors, Raspberry Pi, power sources, Internet). | Low | High | Choose components with proven compatibility. Perform integration tests early in the development process to identify and address any issues. | Identify issues early and replace incompatible components. |
| Algorithms may not process data in real-time, causing delays. | Medium | High | Optimize algorithms for efficiency and test their performance. Use parallel processing techniques if necessary. | Persist data locally and send persisted data on CPU downtime limiting the system to near real time data posting. |
| Data might be lost due to hardware failure or software errors. | Low | Mid | Implement regular data backups, both locally and to the cloud. Use redundant storage systems to ensure data integrity. | Identify the issues and resolve them as they happen |
| Unauthorized access to sensitive data. | Low | Low | Encrypt data both in transit and at rest. Implement strong access controls and regularly update security protocols. | This will be a low security system that does not store sensitive data |
| Community members may have concerns about data collection and privacy. | Low | Low | Clearly communicate the purpose and scope of data collection. Ensure compliance with privacy regulations and implement data anonymization where possible | The first phase of data collection will be anonymous. Only speed, time, and direction will be collected. |
| The project may exceed the allocated budget due to unforeseen costs. | Medium | High | Carefully plan and monitor the budget. Allocate a contingency fund for unexpected expenses. |  |
| The system may not comply with local traffic and surveillance laws. | Medium | High | Conduct thorough research on relevant regulations and ensure the system adheres to legal requirements. Consult with legal experts if necessary. |  |
| Potential intellectual property conflicts related to the use of specific technologies or methods. | Medium | Medium | Conduct a thorough review of intellectual property rights and obtain necessary permissions or licenses. Consider developing original solutions if conflicts arise. | Redesign any portions that have conflicts. |

Change management: Track all changes/decisions in a change log (see Documentation\_TODO.md). End users are involved via feedback and testing of the web dashboard and system features.

**Feasibility and Timeline Insights**

With the use of AI coding tools, a solid MVP (basic vehicle detection, speed logging, SQLite storage, REST API, Docker, simple dashboard) is achievable in 8 weeks. However, advanced features such as robust multi-object tracking, accurate camera-radar fusion, and production-level optimization will require additional time and significant hands-on problem-solving. AI tools excel at basic integrations, API endpoints, and configuration, but real-time optimization and complex sensor fusion will require more manual effort.

**Testing and Peer Feedback**

* Software testing should include unit, integration, and system tests.
* Hardware testing should include environmental/weather testing of the enclosure for durability.
* Security: Ensure all documentation and tables are complete, especially regarding data privacy and system access.

**Significance/Implications of Study**

Imagine a neighborhood where children can play freely, and residents can walk safely without the constant fear of speeding vehicles. This project aims to make that vision a reality by developing an affordable and easy-to-build traffic speed detection system using the Raspberry Pi. By empowering communities with accessible technology, we can create safer streets for everyone, fostering a sense of peace and security within residential areas.

This project is valuable because it addresses a critical safety concern: the prevalence of speeding in residential neighborhoods. Traditional traffic monitoring systems are often prohibitively expensive, making them inaccessible to many communities. By developing a low-cost alternative, we aim to provide a practical solution that can be widely adopted.

The use of Raspberry Pi technology represents new knowledge in the field of traffic speed monitoring. Unlike existing systems, this approach leverages the affordability and versatility of open-source hardware and software to create an effective speed detection system. This innovation has the potential to transform how communities address speeding, providing an accessible and scalable solution.

The major implications of this study are far-reaching. By demonstrating the feasibility of a low-cost traffic speed detection system, we can inspire other communities to adopt similar solutions, leading to safer streets nationwide. This project highlights the potential for citizen-led initiatives to drive meaningful changes, highlighting the power of open-source technology to enhance the quality of life in our communities.

In summary, this project offers a comprehensive, affordable solution to a pressing safety issue, introduces new knowledge in the form of accessible technology, and has significant implications for community-led safety initiatives. The information and rationales provided are accurate and relevant, making the case for why this study is both necessary and impactful.

**Literature Review**

**Traditional traffic speed monitoring systems**

Traditional traffic speed monitoring systems, while effective, often come with excessive costs that make them inaccessible to many communities. Fixed speed cameras, radar speed guns, lidar systems, inductive loop sensors, and ANPR (Automatic Number Plate Recognition) systems are some common methods used to monitor traffic speed. Fixed speed cameras are installed at specific locations to capture images of speeding vehicles and issue fines automatically. While they provide legal evidence for enforcement, they are expensive to install and maintain and are limited to fixed locations. Radar speed guns, which law enforcement officers use to measure vehicle speed using radar waves, offer portability and immediate feedback but require manual operation and cover a limited area. Lidar systems use laser beams to measure vehicle speed and distance accurately but are costly and require manual operation. Inductive loop sensors, embedded in the roadway to detect vehicle speed and count by measuring changes in inductance as vehicles pass over, are durable and reliable but have high installation costs and are disruptive to install. ANPR systems capture and analyze vehicle license plates to calculate speed over a known distance, providing continuous monitoring but requiring extensive infrastructure and raising privacy concerns.

The high initial costs of equipment and installation, along with ongoing maintenance expenses and the need for technical expertise, make traditional traffic speed monitoring systems financially challenging for many communities. Additionally, these systems are often limited to specific locations, may require significant resources for manual operation, and can be vulnerable to vandalism and environmental factors. These limitations highlight the need for more affordable and accessible alternatives, such as the Raspberry Pi-based traffic speed detection system proposed in the "Raspberry Pi 5 Edge ML Traffic Monitoring System" project. This project aims to provide a practical, low-cost solution to enhance road safety in residential neighborhoods by leveraging modern, accessible technology.

**Capabilities and advantages of the Raspberry Pi**

The Raspberry Pi is a versatile, credit-card-sized single-board computer that has gained immense popularity due to its affordability, compact size, and ease of use. Its cost-effectiveness makes it accessible to a wider audience, and its small form factor allows for easy integration into various devices and projects. The Raspberry Pi is user-friendly, supported by a large community and extensive documentation, which makes it an ideal choice for both beginners and advanced users. As an open-source platform, it supports various Linux-based operating systems, encouraging customization and innovation. One of the standout features of the Raspberry Pi is its General-Purpose Input/Output (GPIO) pins, which enable easy interfacing with sensors, actuators, and other hardware components. Additionally, its low power consumption makes it suitable for continuous operation in numerous applications.

In the realm of traffic monitoring, the Raspberry Pi has proven to be highly capable and advantageous. It can be used to develop cost-effective speed detection systems by integrating cameras and sensors to capture and analyze vehicle speeds. Moreover, it can be employed in smart traffic light control systems to optimize traffic flow and reduce congestion using real-time data. The Raspberry Pi also excels in vehicle counting applications, where it uses sensors and cameras to count the number of vehicles passing through a specific point, providing valuable data for traffic analysis. Furthermore, with the help of Automatic Number Plate Recognition (ANPR) software, the Raspberry Pi can recognize and log license plates, aiding in traffic law enforcement. Real-time traffic monitoring systems can also be created using the Raspberry Pi, offering live updates on traffic conditions. Its flexibility and ease of use allow for a wide range of traffic monitoring applications, from simple speed detection to complex traffic management systems, making it an ideal platform for developing innovative and cost-effective solutions.

**DIY traffic speed detection system case studies**

Numerous studies and case examples demonstrate the feasibility and effectiveness of using Raspberry Pi-based systems for traffic speed detection. These projects highlight the importance of leveraging open-source technology and accessible components to empower communities in proactively addressing speeding issues. However, there remains a gap in the comprehensive evaluation of such systems, particularly in terms of accuracy, cost-effectiveness, reliability, and community impact. The "Raspberry Pi 5 Edge ML Traffic Monitoring System" project seeks to address this gap by developing and testing a proof-of-concept for a residential traffic speed monitoring system using Raspberry Pi.

By anchoring the proposed project within the context of existing research, this literature review has established a solid foundation for the "Raspberry Pi 5 Edge ML Traffic Monitoring System" initiative. It has highlighted the potential impact and significance of deploying affordable traffic speed detection systems and set the stage for further exploration and development in this field. The findings from this project could serve as a model for other communities, enabling widespread adoption of accessible and effective road safety solutions.

**Project Schedule**

The project schedule is structured in four major phases, as defined in the Capstone Completion Plan (authoritative):

**Phase 1 (Weeks 1-2): Get Basic Detection Working**

* Set up Raspberry Pi 5 hardware environment
* Install and configure Sony IMX500 AI camera
* Implement basic vehicle detection using TensorFlow Lite
* Establish data collection and storage foundation
* Create initial system health monitoring

**Phase 2 (Weeks 3-4): Add Radar Integration and Simple Correlation**

* Integrate OPS243-C FMCW Doppler radar sensor
* Implement GPIO/UART communication for radar data
* Develop basic data fusion between camera and radar
* Create simple speed correlation algorithms
* Establish real-time data processing pipeline

**Phase 3 (Weeks 5-6): Build Web Interface and API**

* Develop Flask-SocketIO API gateway
* Create real-time WebSocket communication
* Build web-based dashboard for monitoring
* Implement REST endpoints for data access
* Add system configuration management

**Phase 4 (Weeks 7-8): Integration Testing, Documentation and Basic Optimization**

* Conduct comprehensive system integration testing
* Optimize performance for edge deployment
* Complete technical documentation and user guides
* Implement Docker containerization
* Perform final validation and deployment preparation

Buffer time is included in each phase for contingencies. Each milestone/component is mapped to its corresponding phase for clarity and traceability. For full details, see the Capstone Completion Plan.

**Project Cost Estimate (or Alternative Criteria)**

The following cost breakdown is sourced from the Project Management Summary and reflects the most current and detailed estimates:

**Hardware Costs (Per Unit)**

| **Component** | **Model/Description** | **Unit Cost (USD)** | **Quantity** | **Total Cost** |
| --- | --- | --- | --- | --- |
| Raspberry Pi 5 | CanaKit Raspberry Pi 5 Starter Kit MAX - Turbine White (256 GB Edition) (16GB RAM) | $120 | 1 | $219 |
| AI Camera | Sony IMX500 (Raspberry Pi AI Camera) | $70 | 1 | $70 |
| AI Camera extended cable | Official CSI FPC Flexible Cable Compatible with Raspberry Pi 5, 22Pin to 15Pin, Suitable for CSI Camera Modules, 500MM/50CM/1.64 FT | $9.79 | 1 | $9.79 |
| Radar Sensor | OmniPreSense OPS243-C | $255 | 1 | $255 |
| Storage | Samsung T7 Shield 2TB, Portable SSD, up-to 1050MB/s, USB 3.2 Gen2, Rugged,IP65 Water & Dust Resistant, Extenal Solid State Drive (MU-PE2T0S/AM), Black | $149 | 1 | $149 |
| MicroSD Card Extender | LANMU Micro SD to Micro SD Card Extension Cable Adapter Flexible Extender Compatible with Ender 3 Pro/Ender 3/Ender 3 V2/Ender 5 Plus/Ender 5 Pro/CR-10S Pro/Raspberry Pi(5.9in/15cm) | $5.99 | 1 | $5.99 |
| Power Supply | DC 12V/24V to 5V 15A Step Down Converter - Voltage Regulator Buck Converter Power Supply Transformer | $10.99 | 1 | $10.99 |
| USB Power Supply | DC 12V/24V to 5V USB C Step Down Converter Type-C Interface 5A 25W Waterproof Buck Module Power Adapter Compatible with Raspberry Pi 4, Cell Phones 1-Pack | $9.99 | 1 | $9.99 |
| Main power adapter | 24V DC Power Adapter Doorbell Transformer for C WireThermostat,Heywell Nest,Ring Pro doorbell,Ring Wired doorbell,Wyze Video Doorbell,24V LED Strip Light,Comes with CH-2 Connector, Cord 19.6FT Black | $15.99 | 1 | $15.99 |
| Ethernet cable | adaol Cat 6 Ethernet Cable 100 ft, Outdoor & Indoor 10Gbps Support Cat8 Cat7 Network, Flat RJ45 Internet LAN Computer Patch Cable for Router, Modem, Switch, Gaming Consoles, Streaming Devices, White | $16.14 | 1 | $16.14 |
| Enclosure | Junction Box, IP67 Waterproof Plastic Enclosure for Electrical Project, Hinged Grey Cover, Includes Mounting Plate and Wall Bracket 290×190×140mm (11.4"×7.5"×5.5") | $24.99 | 1 | $24.99 |
| Mounting Hardware | Pole/wall mount and assorted hardware | $50 | 1 | $50 |
| **Hardware Subtotal** |  |  |  | **$804.89** |

**Software & Development Costs**

| **Category** | **Description** | **Cost (USD)** |
| --- | --- | --- |
| Software Licenses | Open source (PostgreSQL, Python, TensorFlow) | $0 |
| Cloud Services | Optional cloud hosting (monthly) | $20-50 |
| Development Tools | VS Code, Git, testing tools | $0 |
| **Software Subtotal** |  | **$0-50** |

**Implementation & Deployment Costs**

| **Activity** | **Description** | **Hours** | **Rate** | **Total Cost** |
| --- | --- | --- | --- | --- |
| System Integration | Hardware assembly and configuration | 8 | $75 | $600 |
| Software Deployment | OS setup, application installation | 4 | $75 | $300 |
| Testing & Validation | Field testing and calibration | 6 | $75 | $450 |
| Documentation | User training and documentation | 4 | $75 | $300 |
| **Implementation Subtotal** |  |  |  | **$1,650** |

**Total Project Cost**

* **Per Unit Hardware:** $804.89
* **Implementation (One-time):** $1,650
* **Monthly Operating:** $20-50 (if using cloud services)
* **Total First Unit:** $2,454.89 + monthly costs

*Note: Costs may vary based on supplier, location, and bulk purchasing agreements.*

**Project Issue Log**

The log issue tracking mechanism is clearly designed, and a comprehensive list of preventive steps/issues is presented; identifying potential project issues, explaining the potential impact scope, schedule, and/or cost, as well as including an action plan for a resolution is included. TODO: See Documentation\_TODO.md for outstanding tasks and issue log template. Issue tracking is ongoing and will be updated as the project progresses.

**Requirements Analysis: Use Cases**

See Use\_Cases.md for full list and diagrams.

A screenshot of a computer

AI-generated content may be incorrect.

Example use cases:

1. System detects and logs speeding vehicles (camera/radar fusion, event logging, alert generation)
2. Remote user views live traffic dashboard (VPN, web UI, authentication)
3. Edge device handles network outage (local storage, queued upload)
4. User receives real-time speeding alert (dashboard/notification)
5. Admin remotely updates system software (SSH, update, restart)
6. Data analyst reviews historical traffic data (cloud dashboard)
7. Device self-monitoring and automated recovery (health monitor, auto-restart)
8. Scheduled data backup to external SSD See Use\_Cases.md for full sequences and ASCII diagrams.

**Requirements Analysis: System Design**

See Technical\_Design.md for top-down design and architecture diagrams. The system is organized in layers:

* Cloud Services Layer (optional): Aggregation, analytics, long-term storage
* Network & Communication Layer: WebSocket, REST API, network resilience
* Edge Processing Layer: Vehicle detection, speed analysis, data fusion, API, UI, health monitoring
* Physical Sensing Layer: AI camera, radar, storage, power, housing Each component is included for modularity, reliability, and real-time analytics at the edge.

**Requirements Analysis: Technical Requirements**

Technical requirements:

* Hardware: Raspberry Pi 5 (16GB RAM), Sony IMX500 AI camera, OmniPreSense OPS243-C radar, Samsung T7 SSD, PoE/UPS, IP65 housing
* Software: Raspberry Pi OS, Python 3.11+, TensorFlow Lite, OpenCV, Flask, Flask-SocketIO, PostgreSQL (cloud), Docker
* Network: Tailscale VPN, WiFi/Ethernet/Cellular
* Security: HTTPS/TLS, authentication, firewall, disk encryption See Technical\_Design.md for full details.

**Requirements Analysis: System Logical Model**

See Technical\_Design.md and References\_Appendices.md for logical architecture diagrams and data flow diagrams.

The system flows from sensors (camera/radar) to edge processing (detection, fusion, storage) to cloud (optional) and user dashboards.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.  


**Requirements Analysis: Reports**

Reports will be delivered via the Cloud and local UIs. Please see the examples in the Screen Definitions and Layouts section.

**Requirements Analysis: Screen Definitions and Layouts (or Alternative Criteria)**

The system includes both cloud-based and local user interfaces, each designed for specific operational and monitoring needs. Below is a summary of the main UI screens, their features, and links to mockup images stored in the project archive:

**Cloud Dashboard UI**

The cloud dashboard provides remote access to real-time and historical traffic data, system status, and analytics. Key features include:

* Live traffic data visualization (charts, tables)
* System health and status indicators
* Historical data analytics and export options
* User authentication and access control

Mockup images:

A screenshot of a computer dashboard

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a dashboard

AI-generated content may be incorrect.

A screenshot of a dashboard

AI-generated content may be incorrect.

**Local Device UI**

The local dashboard is accessible directly from the edge device (e.g., Raspberry Pi) and is intended for on-site monitoring and configuration. Key features include:

* Live camera and radar feed display
* Local system health and diagnostics
* Device configuration and network setup
* Manual data export and troubleshooting tools

Mockup images:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

These mockups provide a visual reference for the intended layout and functionality of the system's user interfaces. For further details, see the Technical\_Design.md and the image files in the archive folder.

**Requirements Analysis: Security (or Alternative Criteria)**

See Security\_TODO.md for full security checklist and open issues.

Key security requirements:

* Enforce authentication/authorization on all endpoints
* Use HTTPS/TLS for all network traffic
* Disk encryption for sensitive data
* Input validation and sanitization
* Secure session management
* Regular dependency updates and vulnerability scans
* Limit OS user permissions, firewall configuration
* Data privacy compliance (GDPR/CCPA) See Security\_TODO.md for full list and status.

**References**

Anthropic. (2025). Conversation with Claude [AI conversation]. Claude. https://claude.ai

Gemini LLM (n.d.). Gemini - chat to supercharge your ideas. (n.d.). Gemini.google.com. https://gemini.google.com

Google. (2024). Google Colaboratory. https://colab.research.google.com/.

Microsoft Corporation. (2024). Microsoft Copilot [Software as a Service]. https://copilot.microsoft.com