Software Design Specifications

Phisher Zero

Version: [01.5]

Project Code	F25-119
Supervisor	Sir Minhal Raza
Co Supervisor	Sir Usama Antuley, Miss Saeeda Kanwal
Project Team	Zehra Jabeen Mirza (22K-4781), Muhammad Shehryar Zubair (22K-4736), Anas Ghazi (22K-5081)
Submission Date	December 9, 2025

Document History

Version	Name of Person	Date	Description of change
1.0	Zehra Jabeen Mirza	*12/4/25*	Document Created
1.1	Muhammad Shehryar Zubair Khan	*12/4/25*	Sequence design
1.2	Muhammad Shehryar Zubair Khan	*12/4/25*	Functionalities
1.3	Anas Ghazi	*12/4/25*	Design Considerations
1.4	Zehra Jabeen Mirza	*12/6/25*	Sequence Diagram ,System Architecture Diagram
1.5	Zehra Jabeen Mirza	*12/6/25*	Review

Distribution List

Name	Role
Sir Minhal Raza	Supervisor
Miss Saeeda Kanwal	Co Supervisor
Sir Usama Antuley	Co Supervisor

Document Sign-Off

Version	Sign-off Authority	Project Role	Signature	Sign-off Date
1.5	Supervisor	Supervisor	Sir Minhal Raza	9/12/2025

Document Information

Category	Information
Customer	Users of Phisher Zero
Project	Phisher Zero
Document	Software Design Specification
Document Version	1.5
Status	Complete
Author(s)	Zehra Jabeen Mirza Muhammad Shehryar Zubair Khan Anas Ghazi
Approver(s)	Sir Minhal Raza
Issue Date	December 4, 2025
Document Location	
Distribution	Supervisor

Definition of Terms, Acronyms and Abbreviations

[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.]

Term	Description
NLP	Natural Language Processing
URL	Uniform Resource Locator
API	Application Programming Interface
AI	Artificial Intelligence
TLS / HTTPS	Transport Layer Security/HyperText Transfer Protocol Secure

Table of Contents

1 Introduction

1.1 Purpose of Document

*The purpose of this Software Design Specification (SDS) document is to provide a comprehensive and detailed description of the architectural, structural, and technical design of the **Phisher Zero** system. This SDS translates all functional and non-functional requirements defined in the SRS into concrete design decisions that guide development, integration, testing, and deployment.*

The document ensures consistency, clarity, and alignment across all teams involved in the project and provides a foundation for future enhancements.

This SDS aims to:
- *Establish a clear technical communication framework among all stakeholders.*
- *Serve as a reference for developers during implementation and integration stages.*
- *Ensure that the design aligns with project objectives, security constraints, and system requirements.*
- *Provide a traceable mapping between requirements and the corresponding design components.*
- *Enable structured testing, verification, and validation of all system modules.*

1.2 Intended Audience

- *FAST-NU Faculty & Evaluation Panel*
- *Project Jury*
- *Supervisor(s)*
- *Development Team*
- *Quality Assurance & Testing Team*
- *Researchers and Security Analysts*
- *Potential users or organizations interested in phishing-detection solutions*

1.3 Document Convention

- *Font Family = Arial*
- *Font Size = 12 for headings, 10 for the rest of the content*

1.4 Project Overview

Phisher Zero is an AI-powered phishing detection system designed to identify malicious emails, suspicious URLs, and deceptive content using a multi-agent analysis pipeline. The system integrates a Chrome Extension (client-side) with a FastAPI backend that orchestrates multiple intelligent agents, including an NLP Agent, URL Reputation Agent, Ensemble Agent, and Explainability Module.

The primary objective is to provide users with real-time phishing detection, transparent explanations, and secure client-server communication, ensuring safer email interaction and early threat prevention.

Included Functionalities:
- *Real-time extraction of email content, headers, and URLs through a Chrome Extension.*
- *NLP-based content analysis using LLM-powered features with fallback lightweight models.*
- *URL analysis including lexical features, domain reputation checks, and threat scoring.*
- *Ensemble fusion model combining outputs from all agents to generate a final risk score.*
- *LLM-powered explainability to justify why an email or URL is flagged as suspicious.*
- *Logging, audit tracing, and secure storage of scan metadata.*
- *API-based communication using FastAPI with encrypted data transmission.*

Excluded Functionalities:

- *Phisher Zero does not provide email sending or forwarding capabilities.*
- *It does not modify the user's inbox or alter existing emails.*
- *It does not integrate with enterprise SIEM/SOC systems in this version.*
- *It does not perform network-level phishing detection (only email/content-based).*
- *It does not provide automated incident response actions (blocking, reporting, etc.).*

1.5 Scope

Included Functionalities:

- *Real-time extraction of email subject, body text, metadata, and URLs via Chrome Extension.*
- *NLP-based phishing content analysis using LLMs and fallback transformer models.*
- *URL analysis using lexical features and external reputation services (e.g., VirusTotal).*
- *Ensemble scoring that combines NLP, URL, and metadata signals to produce a final risk level.*
- *Explainability generation that highlights suspicious phrases, URLs, or patterns.*
- *Secure communication between extension and backend using HTTPS and token-based authentication.*
- *Logging and storage of scan metadata, URL analysis results, and agent outputs.*
- *Configurable backend settings for threshold tuning, model weights, and agent behavior.*

Excluded Functionalities:

- *No full email client features (sending, deleting, archiving emails).*
- *No attachment analysis (PDF/image malware scanning is out of scope).*
- *No phishing simulation or training modules.*
- *No enterprise-wide dashboard or admin panel in this version.*
- *No deep browser automation or scraping outside the user's active tab.*
- *No access to user inbox directly unless the email is visible in the browser tab*.

2 Design Considerations

This section establishes the foundational elements required before finalizing the system design. It addresses environmental constraints, external system dependencies, technical assumptions, and areas that may require architectural flexibility due to evolving requirements or technological changes.

2.1 Assumptions and Dependencies

The following assumptions and dependencies influence the design of the Phisher Zero system and must be valid for correct operation:

● ***Chrome Manifest V3 Support:***
The Chrome Extension relies on Manifest V3 APIs for background service workers, content scripts, and secure message passing.

● ***Stable Internet Connectivity:***
The system depends on backend services and external APIs (e.g., LLM provider, URL reputation APIs), therefore uninterrupted connectivity is assumed.

● ***Availability of LLM Provider (Gemini or equivalent):***
The NLP Agent requires access to an LLM. If unavailable, the design includes fallback to a lightweight local transformer model.

● ***VirusTotal/SafeBrowsing API Availability:***
The URL Agent depends on third-party reputation lookups. Design must accommodate rate limits and quotas.

● ***HTTPS Enforcement:***
All communication between the Chrome Extension and Backend is assumed to occur over secure HTTPS channels.

● ***Client Browser Environment:***
Assumes email content is accessible within the browser tab to allow DOM parsing by the content script.

2.2 Risks and Volatile Areas

The following risks and highly variable factors may impact the system design. The architecture must accommodate these uncertainties to maintain reliability:

● ***Third-Party API Instability:***
External services (LLMs, VirusTotal, SafeBrowsing) may introduce downtime, latency, quota restrictions, or pricing changes.
Mitigation: *Fallback models, caching, and circuit breaker patterns integrated into the design.*

● ***Model Drift and Evolving Phishing Techniques:***
As phishing patterns evolve, NLP and URL models may become less effective over time.
Mitigation: *Configurable model weights, modular agent updates, and retraining support.*

● ***Browser API Deprecation or Policy Changes:***
Chrome may update or restrict extension capabilities under Manifest V3 and future versions.
Mitigation: *Decoupled extension architecture and limited reliance on restricted APIs.*

● ***LLM Provider Changes:***
Providers may modify output formats, APIs, or pricing tiers.
Mitigation: *Abstraction layer around LLM interactions to allow easy provider replacement.*

● ***Security & Privacy Risks:***
Phishing detection involves handling sensitive email content.
Mitigation: *Strict data minimization, hashing of stored content, and secure transmission policies.*

● ***User Interface Variability Across Email Providers:***
DOM structure of email clients (Gmail, Outlook web) may change and break extraction logic.
Mitigation: *Content script abstraction layer and frequent UI parsing updates.*

● ***Performance Constraints:***
Real-time scanning must not degrade user experience.
Mitigation: *Parallel agent execution and caching of recurrent domains.*

3 System Architecture

3.1 System Level Architecture

Phisher Zero is designed as a modular, agent-based system that separates responsibilities across multiple components to ensure scalability, maintainability, and security. The high-level system architecture decomposes the system into the following major elements:

- ***Chrome Extension (Client Layer):***
  Provides the user-facing interface for scanning emails. Captures email content, URLs, and metadata from the user's browser and communicates securely with the backend API.
- ***Backend API Layer:***
  Hosts RESTful endpoints that orchestrate communication between the Chrome extension and AI agents. Responsible for preprocessing email content, invoking agents, and aggregating results.
- ***AI Agents:***
  - ***NLP Agent:*** *Analyzes email text for phishing indicators (urgency, impersonation, threats).*
  - ***URL Agent:*** *Evaluates links for malicious or suspicious behavior using ML models and external services like VirusTotal.*
  - ***Ensemble Agent:*** *Fuses results from NLP and URL agents to produce a final phishing classification.*
  - ***Explainer Agent:*** *Generates human-readable explanations of detection outcomes.*
- ***Database Layer:***
  Stores email scan logs, historical analysis results, and system metadata. Supports quick retrieval for dashboard display and auditing purposes.
- ***External Services:***
  Integrates optional services such as VirusTotal, PhishTank, OpenPhish, or Gmail API for URL reputation checks and email fetching.

Relationships and Execution:

- *The Chrome extension communicates exclusively with the backend API over secure HTTPS.*
- *Backend orchestrates agent execution asynchronously, ensuring low latency.*
- *AI agents operate on the backend server (local or cloud) with optional GPU acceleration for NLP inference.*
- *Error handling is centralized at the backend, ensuring graceful fallback if an agent or external API is unavailable (e.g., using DistilBERT if Gemini API fails).*

Global Design Strategies:

- *Modular architecture allows independent updates to AI agents without affecting the client layer.*
- *Secure communication and encryption protocols are enforced across all interfaces.*
- *Scalability is supported by separating agent workloads and employing asynchronous processing.*

3.2 Software Architecture

The software architecture follows a three-tier layered design, ensuring separation of concerns, maintainability, and scalability.

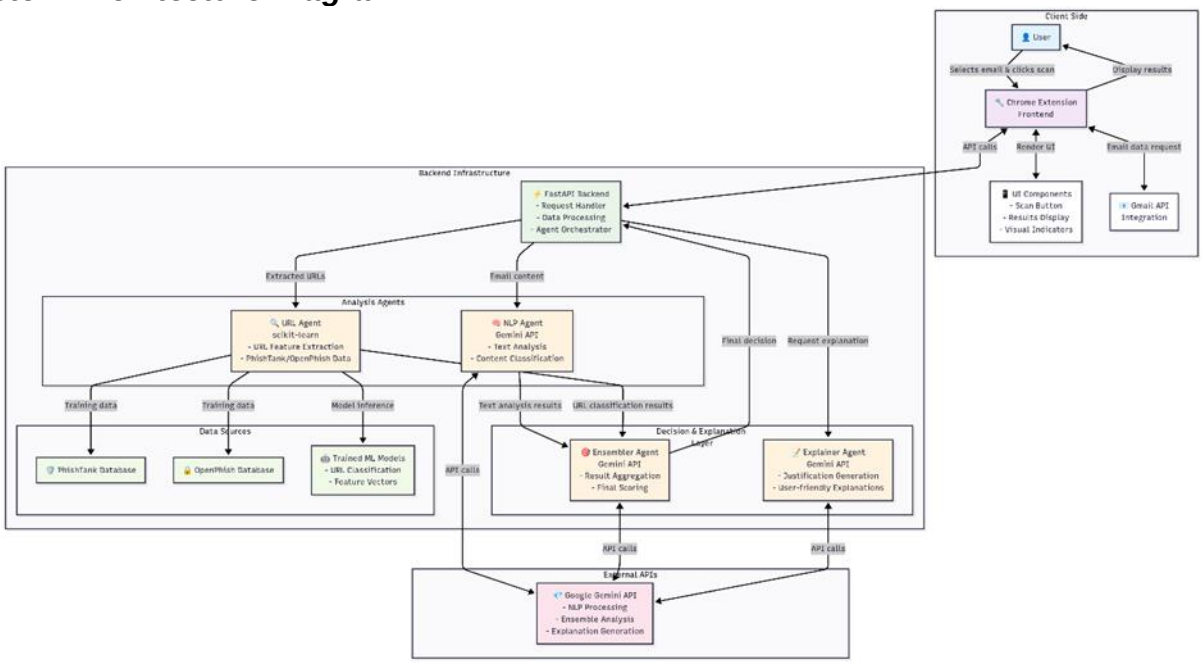
1. ***User Interface Layer (Presentation Layer):***
   - *Chrome extension interface displaying phishing scan results and explanations.*
   - *Provides controls for email scanning and visualization of risk levels.*
2. ***Middle Tier (Application/Logic Layer):***

- *FastAPI backend orchestrating the execution of AI agents.*
- *Implements business logic, including request validation, preprocessing, ensemble aggregation, and error handling.*
- *Handles communication with external services like VirusTotal and Gemini API.*

Interaction Workflow:

- *The **UI Layer** sends email content and URL data to the **Middle Tier** via RESTful API calls.*
- *The **Middle Tier** invokes AI agents to analyze content and URLs.*
- *Results are aggregated by the **Ensemble Agent** and explanations are generated by the **Explainer Agent**.*
- *Final results are returned to the **UI Layer** for display.*

System Architecture Diagram:



4 Design Strategy

Phisher Zero is designed for modularity, scalability, and security, focusing on fast and accurate phishing detection.

4.1 Modularity & Extensibility

- *Independent AI agents (NLP, URL, Ensemble, Explainer) enable easy updates or the addition of new detection methods.*
- *Backend APIs decouple the Chrome extension from agent logic, simplifying maintenance and future expansion.*
- *Modular design allows individual agents to be swapped or upgraded without affecting the entire system.*

4.2 System Reuse

- *Core modules such as text preprocessing, URL analysis, and ensemble fusion are reusable across different platforms or projects.*
- *Agents' standardized input/output interfaces allow integration into other email clients or security tools.*

4.3 User Interface

- *Minimalist, intuitive Chrome extension design with color-coded risk levels for quick comprehension.*
- *Provides real-time, context-sensitive feedback for non-technical users.*
- *Explanations are displayed in a human-readable format without overwhelming the user with technical details.*

4.4 Data Handling

- *No persistent storage; scan results exist temporarily in memory to ensure privacy.*
- *Agents process data asynchronously, allowing multiple scans to run concurrently without conflicts.*
- *Sensitive information is handled locally whenever possible, with only non-identifying data sent to external services.*

4.5 Concurrency & Synchronization

- *NLP and URL agents run in parallel to reduce scan time.*
- *Ensemble agent synchronizes results and produces a unified classification.*
- *Gracefully handles partial or failed analyses, ensuring the user always receives a safe fallback status.*

4.6 Trade-offs

- ***Performance vs. Explainability:** Fast AI inference is prioritized, while explanations are generated separately by the Explainer Agent.*
- ***Local vs. Cloud:** Sensitive data is processed locally to protect privacy, though some URL checks use external threat intelligence services.*

- ***Simplicity vs. Extensibility:*** *The UI remains simple for usability, limiting detailed information upfront, but the system remains adaptable for future enhancements.*

Overall:
Modular, secure, and responsive design ensures effective phishing detection while remaining flexible and extensible for future updates or integration with additional detection methods.

5 Detailed System Design

5.1 Database Design

Not applicable

5.1.1 ER Diagram

not applicable

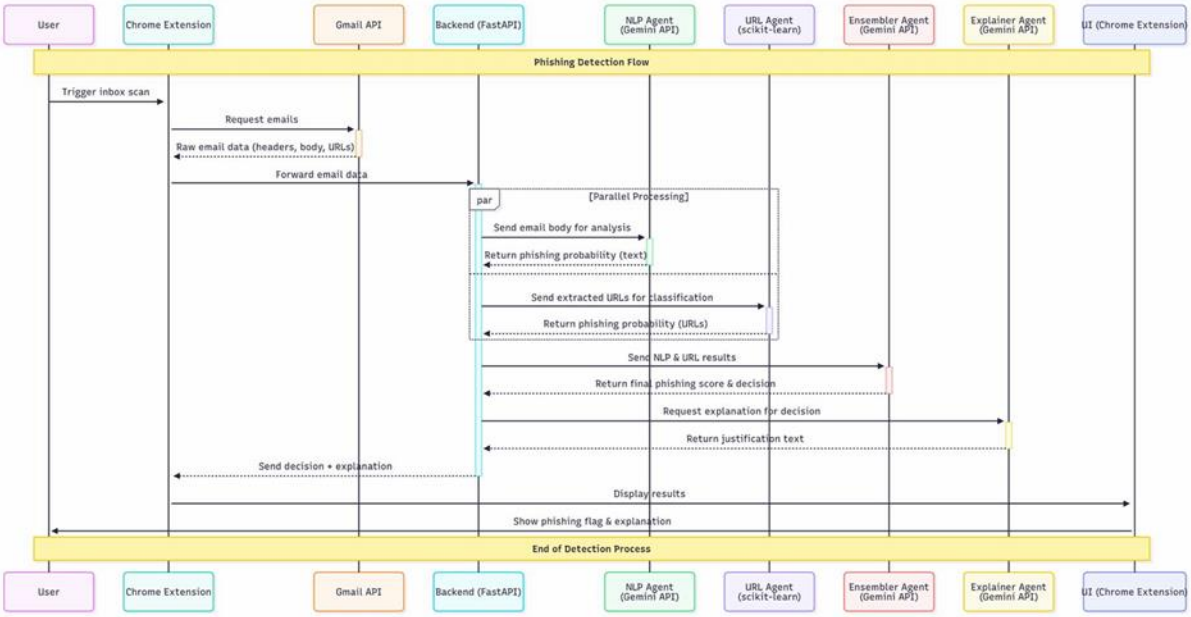
5.1.2 Data Dictionary

Not applicable.

Application Design

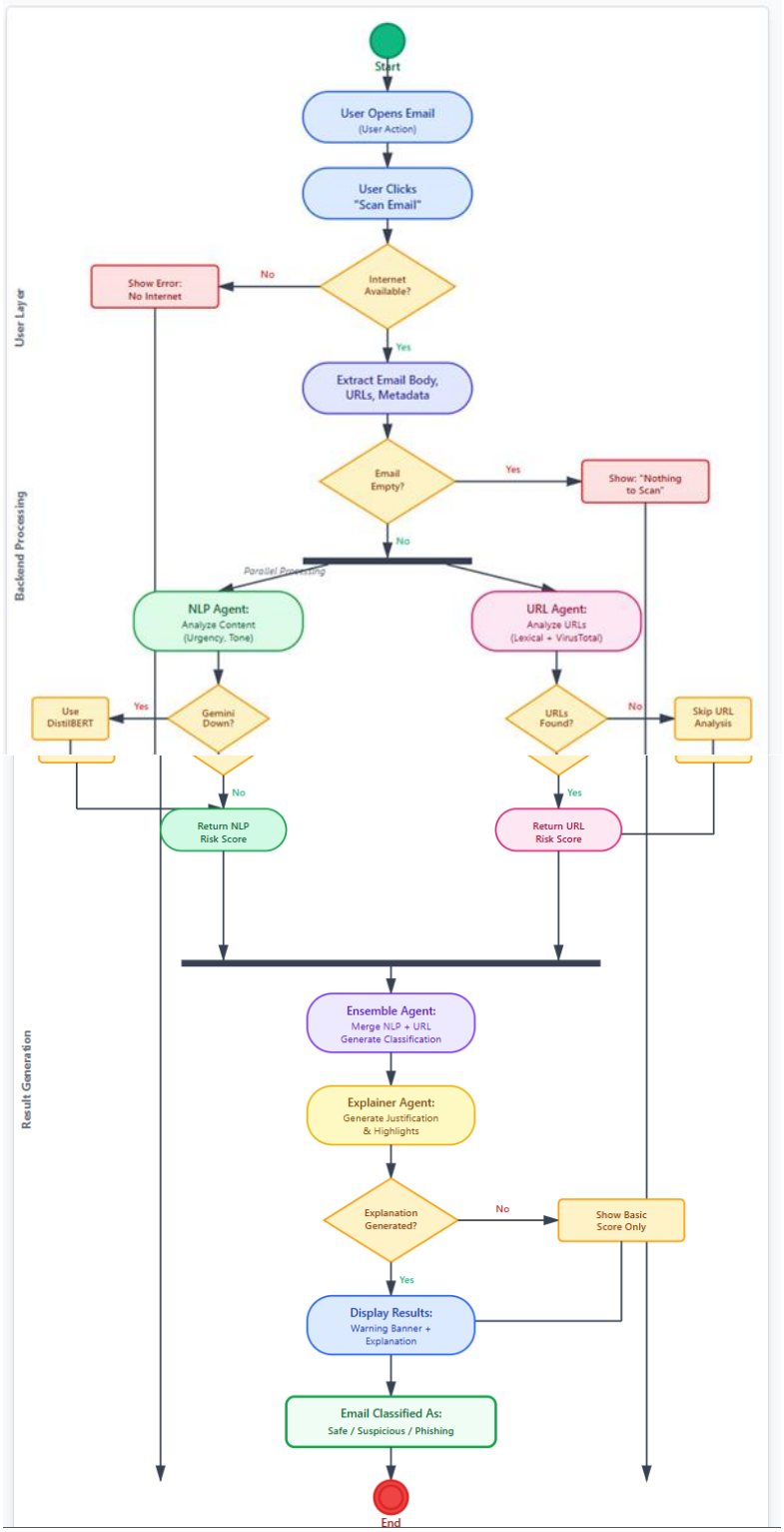
5.1.3 Sequence Diagram

5.1.3.1 <Sequence Diagram>



5.1.4 State Diagram

5.1.4.1 <State Diagram>



6 References

Verizon, 2023 Data Breach Investigations Report *(DBIR). Verizon Enterprise, 2023. Available: https://www.verizon.com/business/resources/Tf09/reports/2023-data-breach-investigations-report-dbir.pdf?*

Amir Herzberg and Ahmad Jbara, "Security and identification indicators for browsers against spoofing and phishing attacks," ACM Transactions on Internet Technology*, vol. 8, no. 4, 2008. DOI: https://doi.org/10.1145/1391949.1391950*

Pranav Maneriker, Jack W. Stokes, Edir Garcia Lazo, Diana Carutasu, Farid Tajaddodianfar, Arun Gururajan, "URLTran: Improving Phishing URL Detection Using Transformers," arXiv, 2021. Available: https://arxiv.org/abs/2106.05256

Ayesha Arshad, Attique Ur Rehman, Sabeen Javaid, Tahir Muhammad Ali, Javed Anjum Sheikh & Muhammad Azeem, "A Systematic Literature Review on Phishing and Anti-Phishing Techniques," arXiv, 2021. Available: https://arxiv.org/abs/2104.01255

B. B. Gupta, Nalin Asanka Gamagedara Arachchilage & Konstantinos E. Psannis, "Defending against Phishing Attacks: Taxonomy of Methods, Current Issues and Future Directions," arXiv, 2017. Available: https://arxiv.org/abs/1705.09819

Government / Policy / Industry Reports

Verizon, "2023 Data Breach Investigations Report," Verizon Enterprise, 2023. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

Website

Google, "Safe Browsing: Protecting users from phishing," 2023. [Online]. Available: https://safebrowsing.google.com

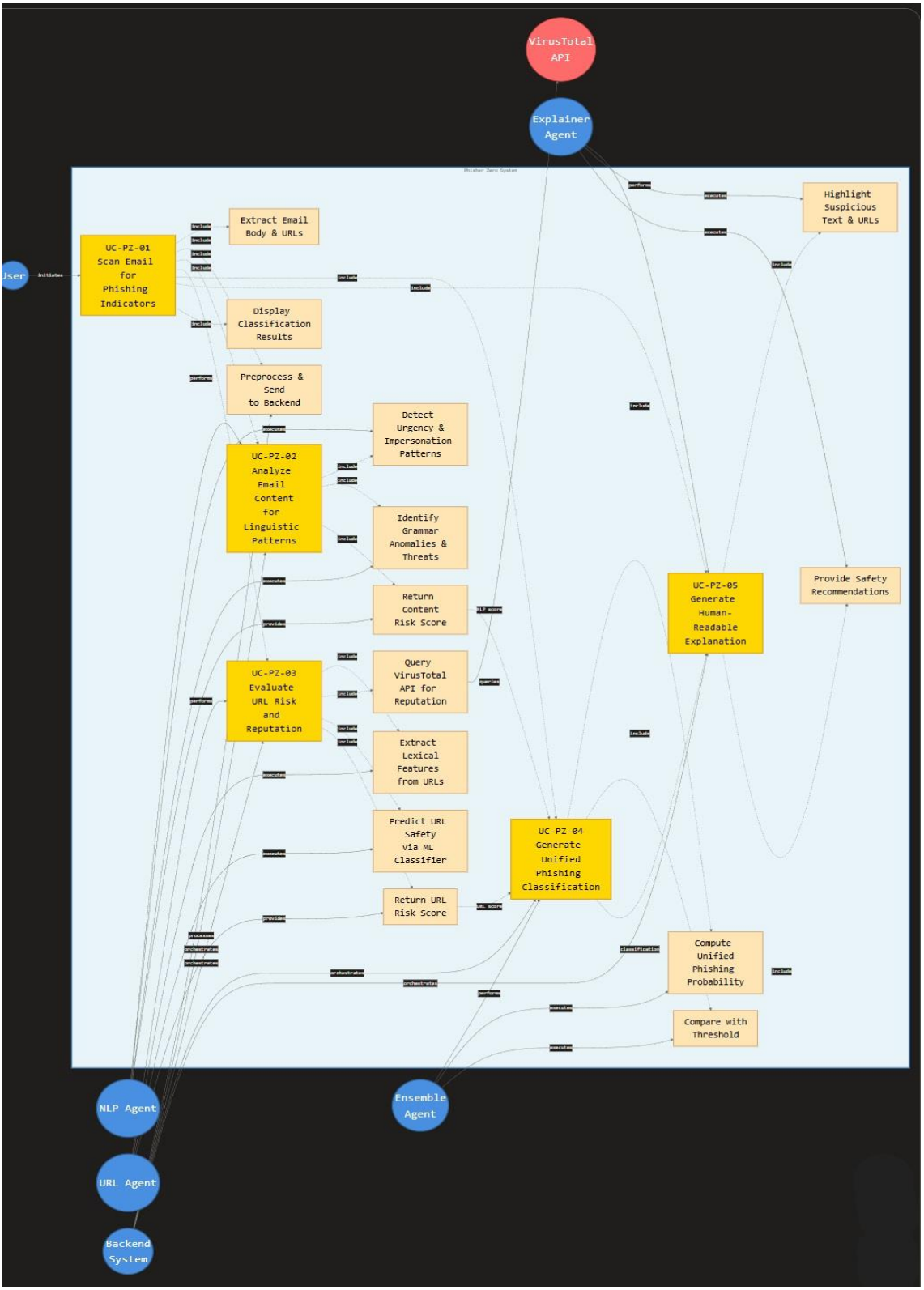
Google, "Gemini API documentation," 2024.[Online]. Available: https://ai.google.dev/gemini-api/docs

Classic / Foundational Sources

T. Jagatic, N. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," *Communications of the ACM*, vol. 50, no. 10, pp. 94–100, 2007. [Online]. Available: https://doi.org/10.1145/1290958.1290968

7 Appendices

7.1.1 Use Case Diagram:



Simplified Use-case Diagram: