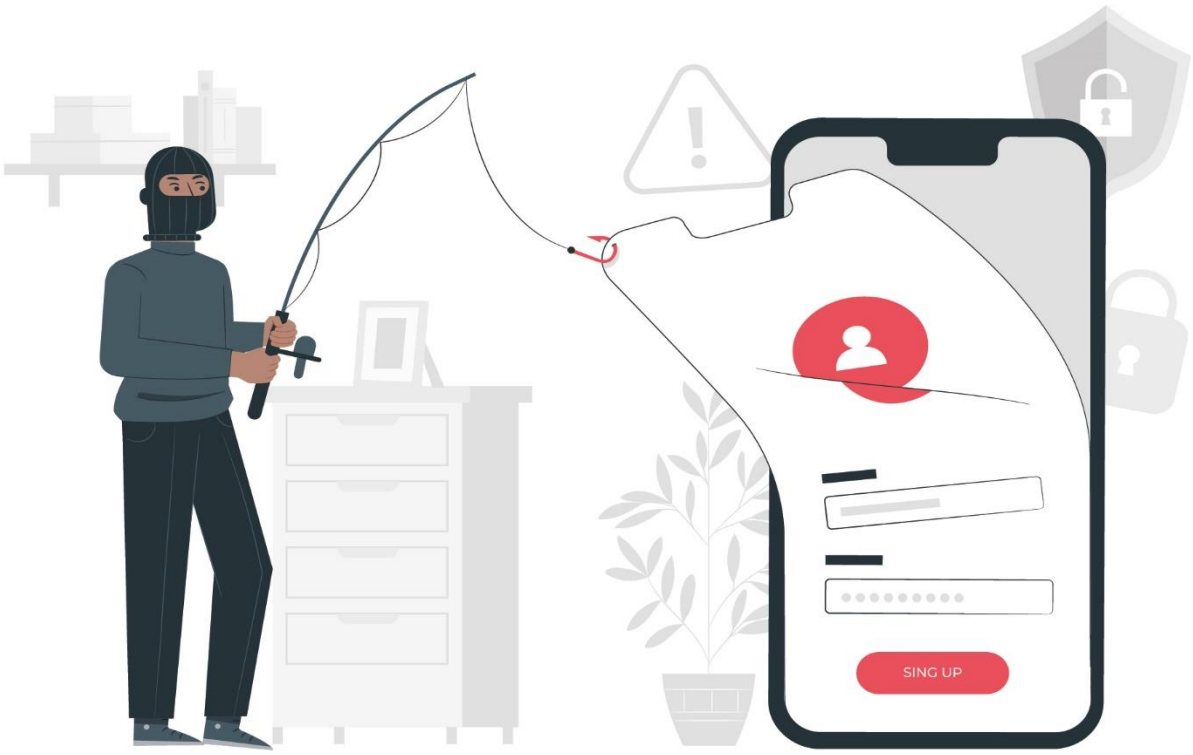
Project Proposal

Phishing Detection Browser Extension for Chrome



AUSTRALIA ECU EDITH COWAN UNIVERSITY

CSG3101.2| Applied Project
Lead Coordinator| Ms. Ann APPUHAMY
Project Supervisor| Mr. Jude Myuran KIRUPARETNAM

Group Members

NAME	SID
Sudam PULLAPERUMA	- 10660248
Dulaj KANKANAMGE	- 10659489
Tharuka GUNASEKARA	- 10659483
Tishantha ELVITIGALA	- 10663914

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF FIGURES

PROJECT GOAL: To develop a Chrome browser extension that detects and prevents phishing attacks in real-time by monitoring URLs, analyzing threats and providing users with actionable alerts to enhance online security.

1.0 BACKGROUND

Phishing attacks are among the most prevalent and harmful cybersecurity threats, exploiting human vulnerabilities to gain unauthorized access to sensitive information (James, 2022). These attacks use deceptive tactics such as fraudulent emails, fake websites or misleading URLs to trick users into disclosing personal data like passwords, financial credentials or identification information (Petrosyan, 2024). Phishing thrives because of its simplicity and effectiveness, posing severe risks to individuals and organizations (Akanbi et al., 2015).

The urgency of addressing phishing is underscored by alarming statistics. In 2023, the Anti-Phishing Working Group recorded nearly **five million phishing incidents**, the highest annual total to date (APWG, 2024). As per (FBI, 2023) Americans **lost $10.3 billion to internet scams** in 2022, with phishing being a significant contributor. Globally, phishing accounted for **36% of data breaches**, according to the IBM Security X-Force Report (IBM, 2024). High-profile cases, such as the 2021 Colonial Pipeline ransomware attack, initiated by a phishing email, disrupted critical fuel supplies across the U.S. East Coast and caused significant operational and economic damage (Bellamkonda, 2024). Similarly, the 2020 Twitter breach exploited phishing tactics to compromise high-profile accounts, demonstrating how even tech-savvy organizations remain vulnerable (Koselev, 2024).

1.1 CURRENT CHALLENGES AND GAPS IN EXISTING SOLUTIONS

Phishing attacks continue to outpace traditional defense mechanisms despite advancements in cybersecurity, owing to their evolving sophistication. Attackers employ advanced techniques such as typosquatting (e.g., "g00gle.com"), dynamic URL generation and social engineering to bypass detection systems (Spaulding et al., 2016). Modern phishing campaigns exploit multiple channels, including SMS, social media and chat platforms, further complicating detection and prevention efforts (APWG, 2024). Existing tools, like spam filters and antivirus programs, primarily focus on email filtering or post attack mitigation, offering limited real-time protection during live web browsing (Pickard, 2023). Additionally, these tools lack user-friendly features, such as customizable whitelist/blacklist management and actionable real-time alerts, leaving non-technical users vulnerable to phishing threats (Pickard, 2023).

These challenges expose critical gaps in current solutions. One major issue is the lack of real-time browsing protection, as most tools fail to dynamically monitor and analyze URLs during active sessions, leaving users exposed to phishing attempts (Bawa et al., 2024). Another significant gap is the absence of user friendly features, such as customizable threat management options, essential for empowering non-technical users. Moreover, existing solutions often struggle to detect advanced phishing tactics, including domain spoofing, embedded malware and multi-channel attacks (Pickard, 2023). Addressing these gaps is vital to improving phishing prevention strategies and reducing the success rate of attacks.

1.2 IMPORTANCE OF ADDRESSING THE PROBLEM

Phishing attacks have far-reaching consequences that extend beyond financial losses. They erode trust in digital interactions, damage organizational reputations and disrupt critical operations (Do et al., 2022). For industries and governments, the economic toll includes increased recovery costs, higher insurance premiums and disrupted workflows. Addressing phishing is not only a matter of protecting individuals but also of ensuring the stability of the global digital ecosystem. Implementing real-time protection can minimize financial and operational impacts, while empowering users with actionable tools will enhance their sense of security (Editor, 2024). By closing the gaps in phishing prevention, this project will strengthen digital trust and contribute to building a safer and more reliable online environment for all.

2.0 SCOPE - A BRIEF DESCRIPTION OF THE PROJECT

This project aims to develop a lightweight Chrome browser extension to detect phishing and malware threats in real time. It will analyze visited URLs using APIs like VirusTotal and provide immediate alerts for suspicious links, allowing users to block or proceed cautiously. Features include customizable whitelist/blacklist management and an email phishing detection tool to analyze email content for malicious links or suspicious senders. Browser notifications will categorize threats by severity, ensuring clear and timely feedback. The extension addresses gaps in existing tools by offering real-time analysis, user-friendly features and actionable alerts to enhance online security and confidence.

2.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS USING MOSCOW FRAMEWORK

Priority	Requirement Type	Requirement	Description
Must-Have	Functional	Real-Time URL Monitoring	Monitor all visited URLs in the browser and analyze them for phishing or malware indicators using VirusTotal API.
		Phishing URL Detection	Detects and flag suspicious or malicious URLs based on API analysis results.
		Basic User Alerts	Provide real-time notifications or pop-ups when malicious URLs are detected, with actionable options like "Block" or "Proceed with caution."
		Whitelist/Blacklist Management	Allow users to add trusted websites (whitelist) and blocked websites (blacklist).
		Basic Malware Detection	Use VirusTotal API to analyze URLs for potential malware threats.
		Email Phishing Detection	Allow users to paste email content or headers for analysis and detect phishing indicators in embedded links or sender details
		Browser Notifications	Send categorized browser notifications with severity levels, such as safe, suspicious or malicious.
	Non-Functional	Lightweight and Efficient Design	The extension must have a lightweight design to ensure minimal impact on browser performance.
		User-Friendly Interface	Provide an intuitive interface that is accessible to non-technical users.
		Compliance with Chrome Policies	Adhere to Chrome Web Store policies for extension development and security.
		Secure Data Handling	Ensure user data is handled securely, complying with data protection standards.
Should-Have	Functional	Severity-Based Alerts	Provide additional context in alerts, such as the threat level and recommended actions for users.
		Email Content Parsing	Automatically parse and analyze structured email content for embedded phishing links.
Could-Have	Functional	Advanced Machine Learning for Phishing Detection	Integrate pre-trained models to identify phishing patterns and implement basic machine learning models for improved detection accuracy.
		Heuristic URL Analysis	Add support for detecting suspicious patterns in URLs, such as typosquatting, IP-based URLs, and uncommon TLDs.

		Customizable User Settings	Enable users to adjust detection sensitivity (e.g., strict vs. moderate modes) and toggle features like malware scanning or heuristic analysis.
Won't-Have	Functional	Multi-Browser Compatibility	Extend the extension to support browsers like Firefox, Edge, and Brave.
		Sandbox Integration for File Analysis	Add a feature to upload and scan files for malware using APIs like VirusTotal.

Table 1: Functional and Non-functional Requirements

2.2 DEVELOPMENT METHODOLOGY: AGILE APPROACH

We will utilize the Agile methodology for project development, ensuring flexibility, iterative progress and continuous feedback to meet evolving requirements effectively.

- **Note:** Justification for selecting agile methodology will be provided in appendix A.

2.3 DELIVERABLES

Deliverable	Description	Date
Initial Proposal	Basic proposal with a project idea	23/112024
Project Proposal Draft	Create a draft for a project proposal based on the instructions and guidance	30/11/2024
Project Proposal	Finalized project proposal with supervisor approval	14/12/2024
Final Executable	A fully functional Chrome browser extension that can monitor URLs in real time and detect phishing and malware threats.	25/01/2025
Source Code	Well-documented source code, provided in a Git repository or compressed archive, for future reference and updates.	25/01/2025
User Guide	A comprehensive manual for end-users detailing installation steps, configuration, usage, and troubleshooting.	25/01/2025
Developer Manual	Technical documentation including system architecture, API integration details, code structure, and setup instructions.	25/01/2025
Requirements Report	A report explaining how the project satisfies all specified functional and non-functional requirements.	25/01/2025
Testing and Evaluation Report	A summary of test cases, methodologies, results, and identified issues or limitations discovered during testing.	25/01/2025
Final Presentation	A presentation summarizing project objective, methodology, features, testing outcomes, and deliverable status	25/01/2025

Table 2: Deliverables

- **Note:** All identified potential risks, along with their detailed descriptions, mitigation strategies and a corresponding risk matrix, are provided in Appendix B

3.0 TOOLS AND TECHNICAL REQUIREMENTS

Category	Item	Purpose	Source	Access Method
Datasets	Phishing Site URLs	Training and validating phishing detection logic	Kaggle	Download using university-provided corporate email.
	Malicious URLs Dataset	Enhancing detection of malicious patterns	Kaggle	Download using university-provided corporate email.
APIs	VirusTotal API	Real-time phishing and malware URL detection	VirusTotal API	Register with university-provided corporate email and obtain API key.
Development Tools	Visual Studio Code	Code editor for extension development	Visual Studio Code	Free download from official website.
	Chrome Developer Tools	Testing and debugging the extension	Built into Google Chrome	Access through the Chrome browser.
	Anaconda	Environment for Python-based development and dependency management	Anaconda.com	Download and install from official website using personal or institutional resources.
Programming Languages and Frameworks	HTML, CSS, JavaScript	Building the Chrome extension's front-end	Open Source	Access and write code directly in Visual Studio Code.
	JSON	Data handling and integration	Open Source	Incorporated as part of JavaScript and API responses.
	Python and Flask	Back-end development and API integration	Open Source	Install Python and Flask via Python.org and pip.
Collaboration Tools	GitHub	Version control and team collaboration	GitHub	Use free account with personal emails.
Documentation Tools	Microsoft Word and Google Docs	Preparing user manuals and developer guides	Microsoft Office or Google Docs	Use free or institutional licenses.
	Canva or PowerPoint	Creating presentation materials	Canva or Microsoft Office	Access free Canva plans or use institutional Microsoft PowerPoint.
	Adobe Illustrator	Designing visual assets and diagrams for documentation and presentations	Adobe.com	Institutional or personal license required for access.
Hardware	Laptops/Desktops	Development and testing	Personal	Provided by team members or through university resources.
	Internet Connection	Access APIs, download datasets, and collaborate	Personal or institutional	Access via personal or university networks.

Table 3: Tools and Technical Requirements

4.0 TEAM CAPABILITY ALIGNMENT

Student Name	Student ID	Discipline	Role(s)	Key Task(s)	Learning Outcomes
Dulaj Walgama	10659489	CyberSec (Y89)	Machine Learning Specialist & Developer	• Research and prototype machine learning models • Assist with heuristic URL analysis.	• Apply machine learning for phishing detection. • Develop innovative cybersecurity solutions.
Sudam Pullaperuma	10660248	CyberSec (Y89)	Incident Response Manager & Project manager	• Oversee real-time URL monitoring. • Perform risk assessments and define detection workflows.	• Manage real-time incident detection. • Implement secure workflows.
Tharuka Gunasekara	10659483	CyberSec (Y89)	Lead Developer & UI/UX Designer	• Develop core features and integrate APIs. • Design an intuitive user interface.	• Build practical web applications. • Enhance user experience.
Tanushka Elvitigala	10663914	CyberSec (Y89)	Security Analyst & QA Specialist	• Conduct vulnerability assessments. • Execute test cases and refine features.	• Evaluate software systems for security. • Ensure software quality.

Table 4: Team Capability Alignment

5.0 SCHEDULE

Task Name	Duration	Start	Finish	Predecessors
Phishing Detection Browser Extension for Chrome	**54 days**	**Sat 11/23/24**	**Sat 1/25/25**	
1.Project Initialization	**3 days**	**Sat 11/23/24**	**Tue 11/26/24**	
1.1 Requirement Analysis	1 day	Sat 11/23/24	Sun 11/24/24	
1.2 Planning	2 days	Mon 11/25/24	Tue 11/26/24	3
2.Learning Tools and APIs	**5 days**	**Wed 11/27/24**	**Tue 12/3/24**	**4**
2.1 Learn VirusTotal API	2 days	Wed 11/27/24	Thu 11/28/24	
2.2 Understand Chrome Extensions	3 days	Fri 11/29/24	Tue 12/3/24	6
3.Core Feature Development	**7 days**	**Wed 12/4/24**	**Tue 12/10/24**	**7**
3.1 Real-Time URL Monitoring	4 days	Wed 12/4/24	Sat 12/7/24	5
3.2 Phishing URL Detection	3 days	Sun 12/8/24	Tue 12/10/24	
4.Basic Malware Detection	**6 days**	**Wed 12/11/24**	**Tue 12/17/24**	**10**
4.1 API Integration	3 days	Wed 12/11/24	Sat 12/14/24	8
4.2 Functional Testing	3 days	Sun 12/15/24	Tue 12/17/24	
5.Whitelist/Blacklist Management	**6 days**	**Wed 12/18/24**	**Tue 12/24/24**	**13**
5.1 Development	3 days	Wed 12/18/24	Sat 12/21/24	11
5.2 Testing	3 days	Sun 12/22/24	Tue 12/24/24	15
6.Email Phishing Detection	**3 days**	**Wed 12/25/24**	**Fri 12/27/24**	**16**
6.1 Development	2 days	Wed 12/25/24	Thu 12/26/24	14
6.2 Testing	1 day	Fri 12/27/24	Fri 12/27/24	18
7.Browser Notifications Feature	**6 days**	**Sat 12/28/24**	**Fri 1/3/25**	**19**
7.1 Development	5 days	Sat 12/28/24	Thu 1/2/25	17
7.2 Testing	5 days	Mon 12/30/24	Fri 1/3/25	
8.Testing and Debugging	**3 days**	**Sat 1/4/25**	**Tue 1/7/25**	**22**
8.1 Internal Testing	2 days	Sat 1/4/25	Mon 1/6/25	20
8.2 Bug Fixing	1 day	Tue 1/7/25	Tue 1/7/25	24
9.Documentation	**6 days**	**Wed 1/8/25**	**Tue 1/14/25**	**25**
9.1 User Manual	3 days	Wed 1/8/25	Sat 1/11/25	23
9.2 Developer Guide	3 days	Sun 1/12/25	Tue 1/14/25	27
10.Final Presentation	**9 days**	**Wed 1/15/25**	**Sat 1/25/25**	**28**
10.1 Preparation	5 days	Wed 1/15/25	Tue 1/21/25	12
10.2 Delivery	4 days	Wed 1/22/25	Sat 1/25/25	30

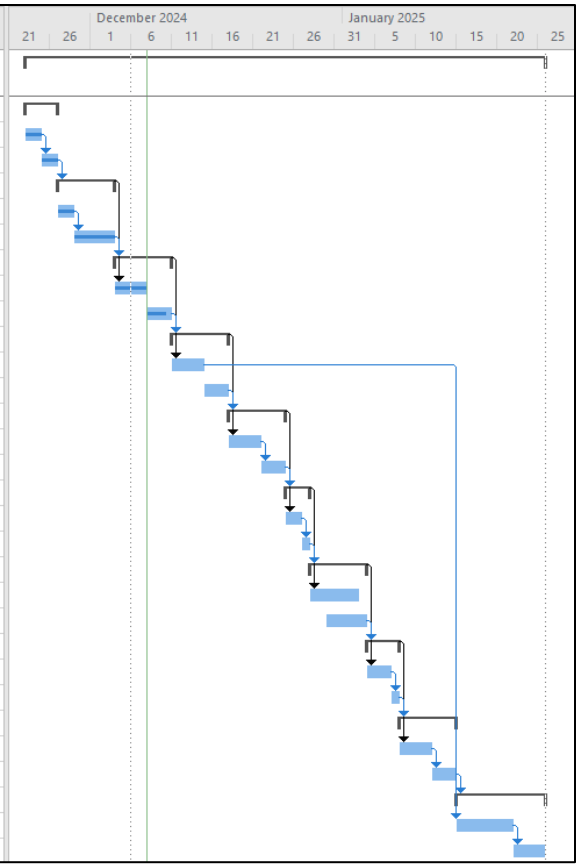


Figure 1: work breakdown structure

6.0 REFERENCES

Akanbi, O. A., Amiri, I. S., & Fazeldehkordi, E. (2015). *A Machine-Learning Approach to Phishing Detection and Defense*. Elsevier. https://doi.org/10.1016/c2014-0-03762-8

APWG. (2024). *Phishing E-mail Reports and Phishing Site Trends 4 Brand-Domain Pairs Measurement 5 Brands & Legitimate Entities Hijacked by E-mail Phishing Attacks 6 Use of Domain Names for Phishing 7-9 Phishing and Identity Theft in Brazil 10-11 Most Targeted Industry Sectors 12 APWG Phishing Trends Report Contributors 13 Unifying the Global Response To Cybercrime PHISHING ACTIVITY TRENDS REPORT*. https://docs.apwg.org/reports/apwg_trends_report_q4_2023.pdf

Bawa, J., Lu, X., Li, J., & Wozniak, A. (2024, March 14). *Real-time, privacy-preserving URL protection*. Google Online Security Blog. https://security.googleblog.com/2024/03/blog-post.html

Bellamkonda, S. (2024). *RANSOMWARE ATTACKS ON CRITICAL INFRASTRUCTURE: A STUDY OF THE COLONIAL PIPELINE INCIDENT*. *7*(2). https://doi.org/10.5281/zenodo.14191113

Do, N. Q., Selamat, A., Krejcar, O., Herrera-Viedma, E., & Fujita, H. (2022). Deep Learning for Phishing Detection: Taxonomy, Current Challenges and Future Directions. *IEEE Access*, *10*, 1–1. https://doi.org/10.1109/access.2022.3151903

Editor, S. (2024, October 10). *Understanding the Risks and Mitigation of Phishing Attacks - SCA Security*. SCA Security - Security Compliance Associates. https://scasecurity.com/blog/understanding-the-risks-and-mitigation-of-phishing-attacks/

FBI. (2023, March 22). *Internet Crime Complaint Center Releases 2022 Statistics*. Federal Bureau of Investigation. https://www.fbi.gov/contact-us/field-offices/springfield/news/internet-crime-complaint-center-releases-2022-statistics

IBM. (2024). *IBM Security X-Force Threat Intelligence Index 2024*. Www.ibm.com. https://www.ibm.com/reports/threat-intelligence

James, N. (2022, December 1). *Phishing Attack Statistics 2023: The Ultimate Insight - Astra Security Blog*. Www.getastra.com. https://www.getastra.com/blog/security-audit/phishing-attack-statistics/

Koselev, N. (2024, January 4). *Understanding the Twitter Hack of 2020: A Deep Dive*. DEV Community. https://dev.to/nikitakoselev/understanding-the-twitter-hack-of-2020-a-deep-dive-2cf4

Kumar, G., Kumar Bhatia, P., & Jambheshwar, G. (2012). Impact of Agile Methodology on Software Development Process. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, *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

Petrosyan, A. (2024, January 10). *Topic: Phishing*. Statista. https://www.statista.com/topics/8385/phishing/#topicOverview

Pickard, S. (2023, April 7). *6 Best Phishing Protection Tools for 2023*. Comparitech. https://www.comparitech.com/net-admin/phishing-protection-tools/

Spaulding, J., Upadhyaya, S., & Mohaisen, A. (2016). *The Landscape of Domain Name Typosquatting: Techniques and Countermeasures*. https://arxiv.org/pdf/1603.02767

Appendix A

Agile methodology is the most practical and effective choice for our project. It enables iterative development, flexibility, and clear prioritization of tasks, ensuring that the core features of the extension are delivered on time and to a high standard (Kumar et al., 2012).

Alternative Methodologies and Why They May Not Fit as Well	
Methodology	**Why It May Not Fit**
Waterfall	Sequential approach lacks flexibility to adapt to changing requirements or incorporate feedback.
Rational Unified Process (RUP)	Complex and resource-intensive; better suited for large-scale enterprise projects.
Kanban	Focuses on workflow visualization but lacks the structured time-boxing Agile provides for clear progress tracking

Table 5: Why Agile?

Why Agile is the Best Choice

1. **Flexibility:** Agile's iterative process allows for incremental improvements and seamless adaptation to new requirements.
2. **Efficiency:** Agile ensures a clear focus on priority features while maintaining timelines through well-structured sprints.
3. **Collaboration:** It fosters a collaborative environment where team members can contribute effectively and engage with stakeholders for feedback.
4. **Risk Mitigation:** By testing and delivering increments regularly, Agile reduces the risk of major project failures or delays.
5. **Scalability:** Agile supports the integration of future enhancements, aligning with the project's future scope.

Appendix B

The following risk matrix is designed to visually represent the potential risks identified during the development of the phishing detection browser extension project. Risks are categorized based on their **probability of occurrence** and **impact on the project**. This matrix provides a quick overview to prioritize mitigation efforts, focusing on high-probability and high-impact risks that could significantly affect the project outcomes. Each risk is mapped using its unique identifier from the Risk Registry.

Risk Rank	Risk Number	Risk Description	Risk Owner	Impact	Risk Mitigation Plan
1	R1	Exceeding API rate limits or third-party API unavailability	Developer	High	Optimize API calls and implement caching; plan for API key rotation.
2	R2	Integration challenges with VirusTotal or other APIs	Developer	High	Perform integration testing early and document API dependencies.
3	R3	Browser compatibility issues with Chrome versions	Developer	Medium	Test extension on multiple Chrome versions; address browser-specific issues.
4	R4	Potential security vulnerabilities in the extension	Security Analyst	High	Conduct security audits and implement secure coding practices.
5	R5	Performance issues slowing down browser operations	Developer	Medium	Optimize extension code and limit resource-intensive operations.
6	R6	Incomplete implementation of Must-Have features	Project Manager	High	Prioritize Must-Have features in early sprints; monitor progress weekly.
7	R7	Insufficient test coverage leading to undetected bugs	QA Specialist	High	Develop comprehensive test plans and use automated testing tools.
8	R8	Skill gaps in team members for tools or APIs	Project Manager	Medium	Allocate time for learning; provide training resources for team members.
9	R9	Timeline delays due to poor time estimates or task dependencies	Project Manager	High	Use Agile boards to monitor tasks; reallocate resources dynamically if delays occur.
10	R10	Low user adoption due to poor usability or unclear value	UI/UX Designer	Medium	Conduct user testing and iterate on UI/UX design based on feedback.

Table 6: Risk Register

The Risk Matrix categorizes identified risks based on their probability of occurrence and potential impact on the project. This visualization helps prioritize mitigation efforts, focusing on high-probability and high-impact risks that require immediate attention. The matrix ensures the team can allocate resources effectively to minimize disruptions and achieve project goals. Each risk is referenced using its unique identifier from the Risk Registry.

Probability \ Impact	Low	Medium	High
Low	-	R3	-
Medium	R10	R5, R8	R9
High	-	-	R1, R2, R4, R6, R7

Table 7: Risk Matrix

Appendix C

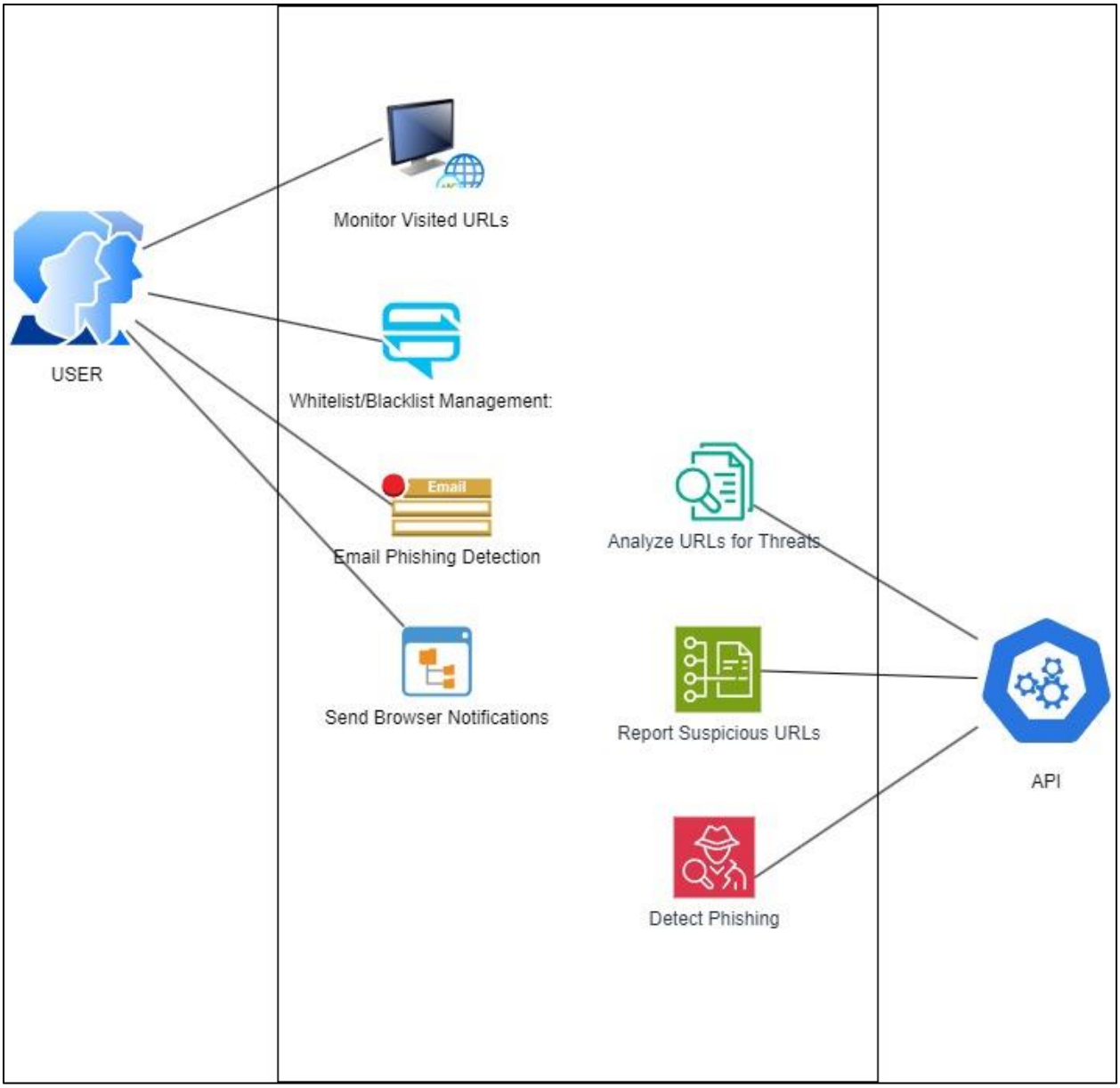


Figure 2: Use case diagram for the phishing detection browser extension for chrome