Analysis - PhishGuard

Authors: Roko Mladinić, Daniella Namuli, Simeon Markov & Zed Minabowan

Table of contents

Introduction

In this document, we'll be analyzing what our project will require to function properly. The goal of this analysis is to define the problem, the purpose of the project and the context in which it will be used. We will also think about what our potential users would require from our project for it to be seen as a functional game. This analysis will be our foundation for when we start creating our product during the realization phase.

Our project's problem

A lack of cybersecurity awareness is a big issue for a lot of organizations. Even though phishing attacks are some of the most common online threats, a lot of people still struggle to recognize them. This is mainly because cybersecurity training is usually given through static methods like documents or videos, which are not very engaging. Because users are not actively involved, they usually don't remember what to look for when they receive a suspicious email. Because of that, mistakes like clicking on unknown links or sharing sensitive information still happen very often. That's why we want to create a better solution that teaches people how to recognize phishing in a more interactive and fun way.

Goal of the project

Commented [ND1]: @Namuli,Daniella D. do this

In this project, we want to create an interactive cybersecurity awareness game that helps users recognize and avoid phishing attacks. Our goal is to build:

- A system that stimulates realistic phishing emails so users can learn how cyber-attacks work.
- A feature that allows users to analyse emails and identify suspicious elements such as fake links, sender impersonation and psychological manipulation.
- A progression system with multiple rounds that challenges users and improves their skills over time

With this project, we want to reduce common mistakes like clicking on malicious links or sharing sensitive information. By replacing static training with an interactive and fun learning experience, we aim to create one application that helps users better understand phishing attacks and apply this knowledge in real-life situations.

Target audience

Commented [MR2]: @Mladinić,Roko R.

We are targeting corporate employees, students and IT administrators. For employees and students, the game makes cybersecurity training fun and interactive, allowing them to practice spotting fake emails in a safe, risk-free environment. For IT administrators and teachers, the dashboard provides data to track progress and identify exactly who needs

more help. By serving both the learners and the supervisors, PhishGuard becomes an effective educational tool for schools and businesses alike.

User Stories

Commented [MR3]: @Mladinić,Roko R.

Reference	User Role	User Story	Outcome
EMP-01	**Employee**	As an employee, I want to practice identifying phishing emails in a simulated inbox, so I can learn to spot red flags without risking actual company data.	Increases confidence and reduces the likelihood of clicking on real malicious links.
EMP-02	**Employee**	As an employee, I want phishing emails to increase in difficulty over time, so that I can gradually improve my ability to recognize more advanced attacks.	Helps users build long-term awareness instead of only recognizing obvious threats.
EMP-03	**New Hire**	As a new hire, I want to receive instant feedback when I make a mistake, so I can immediately understand which visual cues I missed.	Accelerates the learning curve for security awareness during the onboarding process.
STU-01	**Student**	As a student, I want to see my score and win-streaks, so I feel motivated to compete with my classmates while learning about digital safety.	Improves engagement and retention of cybersecurity principles through competition.
STU-02	**Student**	As a student, I want a countdown timer on each email, so that the simulation mimics the high-pressure environment of a real workplace.	Adds a layer of difficulty that forces users to rely on quick pattern recognition.
STU-03	**Student**	As a student, I want short explanations after each round that summarize my mistakes, so I can remember what to look out for in future emails.	Reinforces learning through reflection on their mistakes.
ADM-01	**IT Admin**	As an IT administrator, I want to view a dashboard showing which types of attacks users struggle with most, so I can tailor future security briefings.	Provides data-driven insights to strengthen the organization's overall security posture.
ADM-02	**IT Admin**	As an IT administrator, I want to see how long users take to make decisions, so I can identify whether hesitation plays a role in security mistakes.	Provides insight into user behavior and helps improve future training focus.

ADM-03	**IT Admin**	As an IT administrator, I want to see how many users are registered in the system, so I can get an overview of how widely the training tool is being used. Helps measure adoption of the tool and evaluate its reach within the organization or class.	Gives insight into adoption levels and helps evaluate how widely security training is being used.
US-01	**User**	As an user, I want to be able to log-in, in order to track my score and win-streak.	Ensures that the user will have his progress saved wherever he logs in.

Table 1: User stories

MoSCoW:

Must have	**EMP-01, EMP-03, US-01**
Should have	**STU-01, ADM-01, EMP-02, STU-02**
Could have	**ADM-03, STU-03**
Will/Wish have	**ADM-02**

Table 2: MoSCoW requirements prioritization

Tools

Some of the tools we're currently planning to use are

- Design: Figma
- Version control: GIT & GitHub

Requirements

Functional requirements

- **Practicing:** Users could engage with the content (e.g. learning how to spot phishing emails, links).
- **Feedback system:** Users receive feedback on how they performed, what did they get wrong and visual clues are provided.
- **Result:** Users can see their results like overall score, win-streaks.
- **Administration:** The admin could see data analytics like dashboard displaying the most difficult tasks for users, topics that users struggle with.

Non-functional requirements

- **Performance:** Pages load within accepted time; visuals are rendered eagerly, interactions are smooth.
- **Scalability:** The architecture handles future scales, seamless integration of new features.
- **Usability:** The interface should be intuitive, visuals not being overwhelming, users could orient themselves easily.
- **Security:** Validation for common vulnerability attacks, password hashed, CSRF tokenization.

Technical requirements

Commented [MS4]: @Markov,Simeon S.

- **Frontend:**
  - *Frontend frameworks:* Vue.js & Nuxt.
  - *UI/UX:* Tailwind CSS + Shadcn/ui, HeroUI (UI component libraries).

- **Backend:**
  - *Language:* JavaScript.
  - *Backend Server:* Nitro (Nuxt Server) + Node.js.
  - *Architecture:* Client-server communication.
  - *Database:* PostgreSQL.
  - *Deployment:* Railway.
  - *CI/CD pipeline:* GitHub Actions.
  - *Testing:* Playwright.

Conclusion

Our analysis defined the core problem, goals, user needs and requirements of the PhishGuard project. By translating cybersecurity awareness issues into clear user stories, prioritized features and technical requirements, we created a solid foundation for development. This document will guide the design and realization phases and ensure that PhishGuard is built as a secure, user-friendly and effective learning tool.