Design - PhishGuard

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

Table of contents

Introduction

Commented [MR1]: @Mladinić,Roko R.

The design phase builds on the foundation established during the research and analysis stages. With the requirements, MoSCoW priorities and user stories for employees, students and administrators now clearly defined, this phase focuses on shaping a gamified solution that meets the cybersecurity training needs of our target audience.

The goal is to translate the earlier findings into a structured blueprint that demonstrates how PhishGuard will function, how users will experience the email simulation and how the game logic and data tracking will work together.

In this document, the concepts from the research are transformed into visual layouts, game flow diagrams and architectural plans. These designs will guide the development process, ensuring that the team understands exactly how the inbox mechanics and feedback loops are structured before any coding begins. By the end of this phase, there will be a complete technical and visual roadmap that prepares the project for the realization phase, where the actual implementation takes place.

Design guide (UI/UX Principles)

Commented [MR2]: @Mladinić,Roko R.

This section is about the initial design plan (incorporating layout principles, accessibility, etc.)

- **User-Centric** The design is tailored for employees and students who need a realistic but safe environment to practice cybersecurity skills. The interface mimics familiar email clients to ensure the skills learned in the game translate directly to real-world software usage.
- **Visual Hierarchy** Important game states, such as the current "ROUND" and "Difficulty Level," are highlighted using bold typography and high-contrast badges (e.g., the red "HARD" indicator). Layout and spacing guide the user's eye from the inbox list on the left to the central email content and finally to the action buttons in the bottom-left sidebar.
- **Simplicity and Clarity** The "Welcome Screen" uses clear iconography and bulleted lists to explain game mechanics before a round begins. By showing only one email at a time and using simple "Submit" or "Discard" actions, the interface reduces cognitive load, allowing users to focus entirely on spotting red flags.
- **Appropriate Visualization Selection** Iconography is used to represent abstract concepts, such as "Health" being shown through a row of heart icons. For the IT Administrator dashboard, bar charts and pie charts are planned to help supervisors quickly compare the types of threats users are successfully identifying versus those they miss.

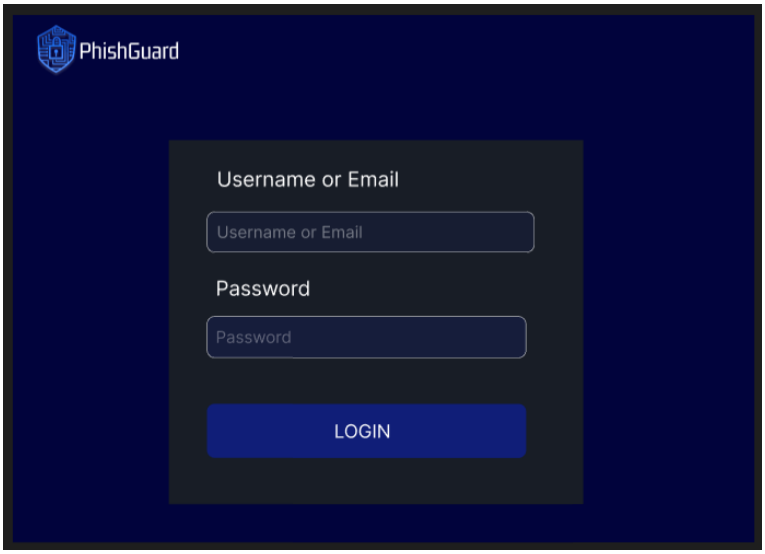
- **Real-Time Data Presentation and Contextual Feedback** Users receive immediate visual feedback; when a discrepancy is selected, it is highlighted with a red circle to confirm the user's action. The "Correct" and "Incorrect" counters update in real-time as the user progresses through the round, ensuring they always know their current standing.

Wireframing

This section describes the structure and purpose of the main pages & screens of the PhishGuard website. Our wireframes focus on functionality, user flow and clarity.

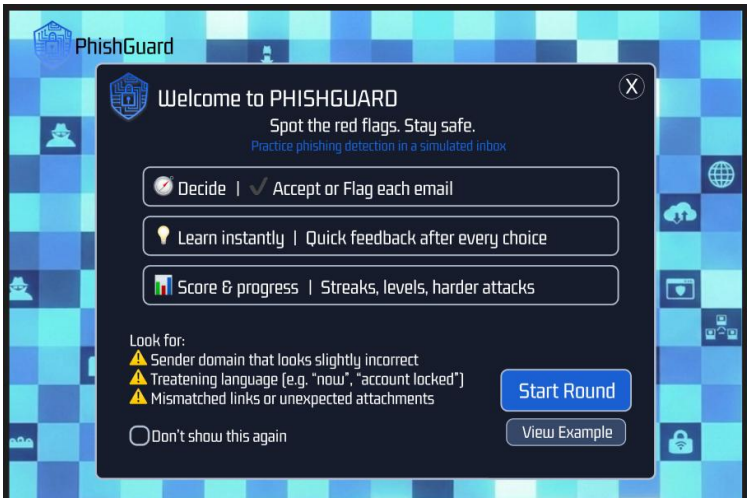
- Login page

The login screen is one of the first pages our users will see when they visit PhishGuard. It'll contain fields for things like usernames, email addresses, passwords and clear buttons. The layout is simple to avoid confusion. The purpose of this page is to give each user a personal learning experience by connecting their results to their account.



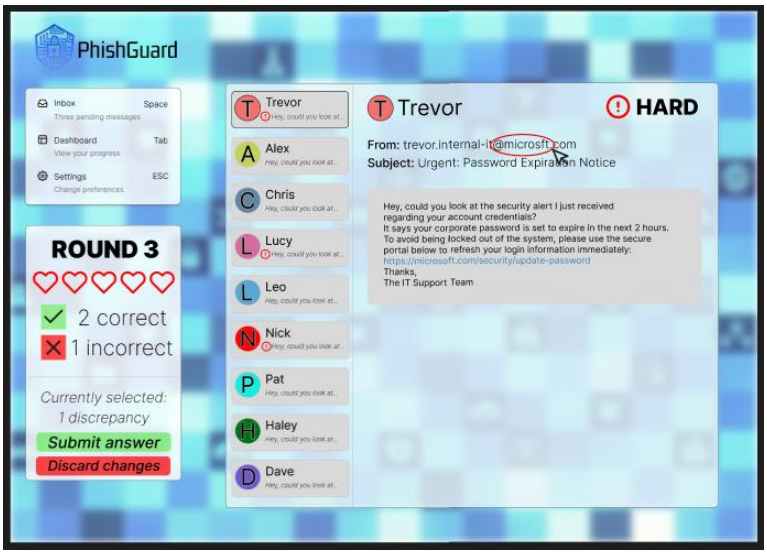
- Welcome screen

After logging in, our users will see our welcome screen. Our welcome screen is meant to explain what PhishGuard is and how the game works. We'll go over some important points of our game and give tips on what users should look for.



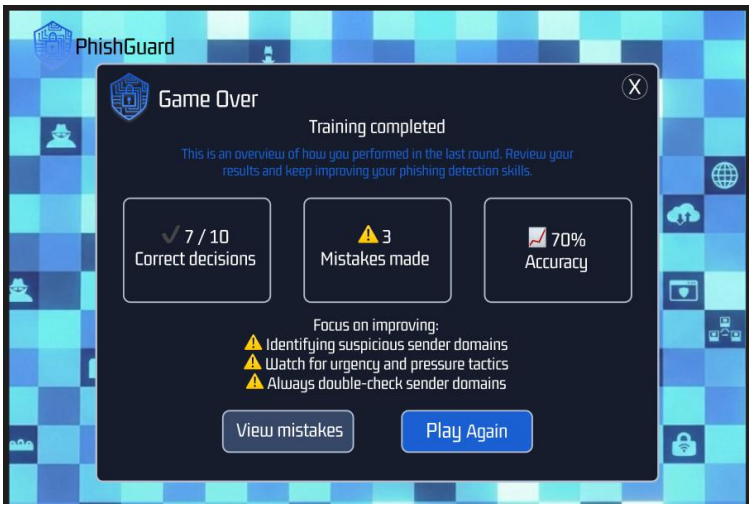
- Game screen

The game screen is the most important part of our website. We'll use our game screen to simulate a realistic email inbox where our users will be shown various emails one by one. Some will and some won't include phishing. This is where our users will be able to spend some time investigating the email and deciding whether they want to flag it as phishing. This screen will show things like remaining lives, the current round and other progress indicators.



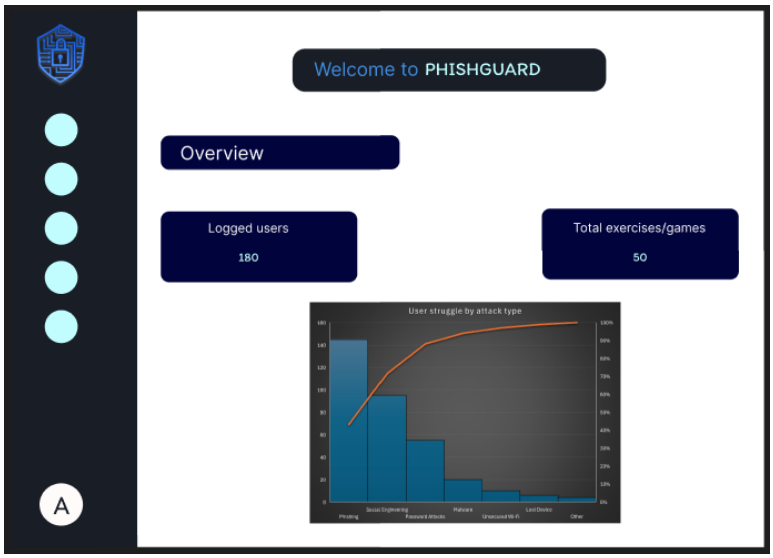
- Game over screen

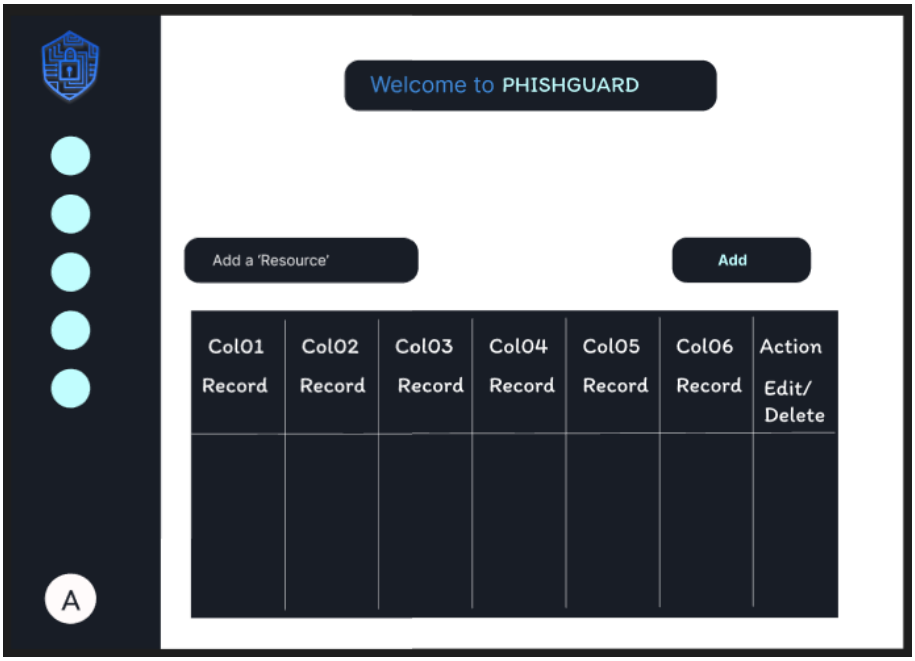
Our game over screen will be shown once a full game has been finished. Once the game has finished, we'll show our users how they performed. Users will get an overview of their correct decisions, mistakes, accuracy and more. We'll also highlight some things the users could focus on in future attempts. The goal of this screen is to properly end the game and the educational message helps users improve on their mistakes.

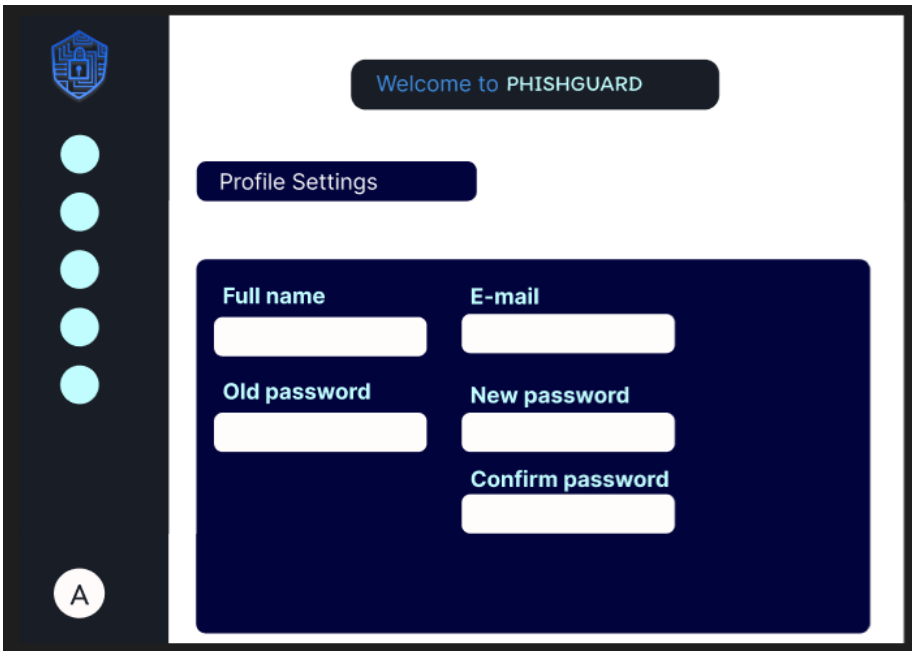


- Admin dashboard overview

The admin dashboard overview is meant for the administrators of the website. This screen will show general statistics like total users, number of completed rounds, overall user performance, etc. This screen will help administrators to decide whether they need to make the games easier or harder.




Welcome to PHISHGUARD
Add a 'Resource'
Add
Col01 Record
Col02 Record
Col03 Record
Col04 Record
Col05 Record
Col06 Record
Action Edit/ Delete
A


Welcome to PHISHGUARD
Profile Settings
Full name
E-mail
Old password
New password
Confirm password
A

Tech stack decisions

Frontend: Vue.js was chosen due to its flexibility and not so steep curve. Moreover, it comes with great community support, which is essential for developing a rapid yet stable application as well as troubleshooting issues on the go.

Backend & Database: As a development environment, VS Code was chosen because of its cross-platform availability and easy set up. For the backend, Nuxt + Nuxt Server was chosen, because it provides a unified codebase based on folder-based architecture, which makes deploying much easier. It comes with a development server for locally testing and single logging of the application. The database is PostgreSQL, which is a relational database with advanced analytics, support for complex queries and native support for deployment.

Commented [MS3]: @Markov,Simeon S.

Deployment: The deployment would be on Railway due to its native support for PostgreSQL and automated SSL certificates distribution.

Database design

Visual Paradigm software was chosen for data modeling (database-first approach), which provides assets for making ERD diagrams faster and tools for exporting in raw SQL.

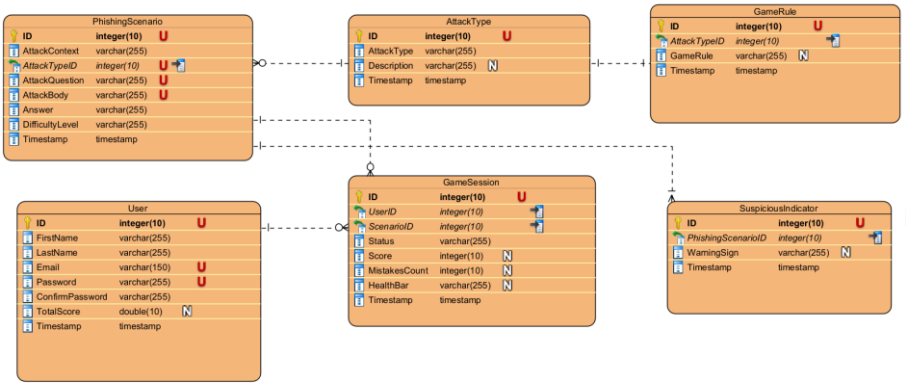


Figure 1: ERD

Database overview

Commented [ND4]: @Namuli,Daniella D.

Entities

- **Users:** They represent all players in the game. This includes basic user information like username, password and email address, as well as their progress in the game, remaining lives and overall performance. This allows players to see their improvement over time.
- **PhishingScenario:** This contains all the phishing examples used in the game. Each scenario is designed to represent a real-world phishing scenario and includes information such as the type of attack and its difficulty level.
- **AttackType:** Containing the category/type of the phishing attack (e.g. social engineering, phishing links).
- **GameRule:** Containing the rules of the game.
- **SuspiciousIndicator:** This represents the warning signs hidden inside phishing emails such as fake links or impersonated senders.
- **GameSession:** This represents the gameplay session each user is currently in.

Relationships

- **One-to-many:** Tables 'PhishingScenario'->'GameSession' (One game could have multiple phishing scenarios), 'User'->'GameSession' (Multiple users may be in the same game, e.g. experience the same phishing scenarios), 'PhishingScenario'->'SuspiciousIndicator' (Each phishing scenario has its own indicator/hint).
- **One-to-one:** Table 'GameRule'->'AttackType' (One attack type has its exact game rule).

Tables description

Commented [MS5]: @Markov,Simeon S.

'Users'

ColumnName	DataType	Constraint	Description
ID	integer	PK	Unique identifier for each user
FirstName	varchar	NOT NULL	User first name
LastName	varchar	NOT NULL	User last name
Email	varchar	UNIQUE	User email address
Password	varchar	UNIQUE	Hashed password for authentication
ConfirmPassword	varchar	NOT NULL	Password confirmation field
TotalScore	double	NOT NULL	Cumulative score across all sessions

Timestamp	timestamp	NOT NULL	Account creation timestamp

'PhishingScenario'

ColumnName	DataType	Constraint	Description
ID	integer	PK	Unique scenario identifier
AttackContext	varchar	NOT NULL	Context or subject line of phishing email
AttackTypeID	integer	FK	References AttackType.ID
AttackQuestion	varchar	UNIQUE	Question to present to the user
AttackBody	varchar	NOT NULL	Email body/content of the phishing scenario
Answer	varchar	NOT NULL	Correct answer/ expected user action
DifficultyLevel	varchar	NOT NULL	Difficulty level (e.g., Easy, Medium, Hard)
Timestamp	timestamp	NOT NULL	Scenario creation timestamp

'AttackType'

ColumnName	DataType	Constraint	Description
ID	integer	PK	Unique identifier for attack type category
AttackType	varchar	NOT NULL	Name of the attack type (e.g., Spear Phishing, CEO Fraud)
Description	varchar	NULL	Detailed description of the attack type

'GameSession'

ColumnName	DataType	Constraint	Description
ID	integer	PK	Unique session identifier
UserID	integer	FK	References User.ID
ScenarioID	integer	FK	References PhishingScenario.ID
Status	varchar	NULL	Current status (In Progress, Completed, Failed)
Score	integer	NULL	Points earned in this session
MistakesCount	integer	NULL	Number of mistakes made during session

HealthBar	varchar	NOT NULL	Health bar value (resets each session)
Timestamp	timestamp	NOT NULL	Session timestamp

'SuspiciousIndicator'

ColumnName	DataType	Constraint	Description
ID	integer	PK	Unique identifier for each warning sign/hint
PhishingScenarioID	integer	FK	References PhishingScenario.ID
WarningSign	varchar	NULL	Description of suspicious indicator (e.g., "Misspelled domain")

'GameRule'

ColumnName	DataType	Constraint	Description
ID	integer	PK	Unique identifier for each rule
AttackTypeID	integer	FK	References AttackType.ID
GameRule	varchar	NULL	Rule specification

Conclusion

Commented [ND6]: @Namuli,Daniella D.

Our design document shows how the PhishGuard game will look and work. We translated our game ideas into clear layouts, user interface choices and system structures that support game logic and our learning goals. The design focuses on creating a secure, engaging and easy to use interface for players while supporting features such as progress and feedback tracking. With this design phase completed, we now have a solid foundation to start developing the actual game in the realization phase.