CEBU INSTITUTE OF TECHNOLOGY UNIVERSITY

COLLEGE OF COMPUTER STUDIES

Software Requirements Specifications

for

CyberKids

Change History

Version	Date	Amendment	Author
1.0	3/14/2025	Initial	Cultura
1.1	3/18/2025	Updated Modules	Baguio Cultura Dedumo Vequiso
1.2	3/21/2025	Updated Modules (relevant to the problem)	Baguio Cultura Dedumo Vequiso

Table of Contents

1. Introduction

1.1. Purpose

CyberKids is an interactive puzzle and strategy game designed to immerse elementary school students in real-world cybersecurity scenarios. It aims to address the increasing vulnerability of children to online threats by providing an engaging and effective learning environment focused on password security, phishing awareness, and online privacy. The intended audience includes grades 5-6 elementary students, computer teachers, system administrators, and developers responsible for maintaining and improving the platform.

1.2. Scope

CyberKids is a web-based educational game that helps students learn about cybersecurity best practices through a series of engaging missions. The game focuses on information sharing awareness, password security, and phishing scam detection.

What the Software Will Do (Core Functionalities**)**

1. **Single-Player Exploration**: Players control a character navigating a virtual world with different digital zones representing online platforms in each level.
2. **Mission-Based Learning:** Players complete three major cybersecurity challenges:
    - Data Leak Investigation – Identifying safe and unsafe personal information to share.
    - Password Fortress Defense – Constructing strong passwords to protect against hacking attempts.
    - Cyber Escape Room – Identifying phishing and scam attempts to escape a

locked digital room.

3. **Teacher Dashboard**:
    - Educators can track student progress and view mission completion rates.
    - Teachers can assign specific missions and review student performance.

What the Software Will NOT Do

- No Real Social Media or Online Interaction: The game does not connect to actual social media platforms or online services. All interactions are within a simulated environment.
- No Multiplayer Features: CyberKids is a single-player game with a leaderboard for competition, but no real-time multiplayer interactions.
- No Real Data Collection: Players will never be required to enter real personal information—only fictional, simulated scenarios are used for learning.

1.3. Definitions, Acronyms and Abbreviations

This section provides definitions of key terms, acronyms, and abbreviations relevant to the implementation of the CyberKids system.

Acronyms:

Acronyms used in this document shall be interpreted as follows:

- API – Application Programming Interface
- CORS – Cross-Origin Resource Sharing
- CRUD – Create, Read, Update, Delete
- CI/CD – Continuous Integration / Continuous Deployment
- JWT – JSON Web Token
- SQL – Non-Relational Database Query Language
- REST API – Representational State Transfer API
- UI – User Interface

Abbreviations:

Abbreviations used in this document shall be interpreted as follows:

- JS – JavaScript
- DB – Database
- UX – User Experience
- HTTP – HyperText Transfer Protocol

- SQL – Structured Query Language
- URL – Uniform Resource Locator

Definitions:

Definitions used in this document shall be interpreted as follows:

Term	Definition
Authentication	The process of officially recognizing an educational institution or program as meeting specific quality standards.
Authorization	The process of granting or denying access to specific resources or functionalities.
Backend	The server-side part of a software application, responsible for data processing, logic, and database interactions
Cybersecurity	The practice of protecting systems, networks, and programs from digital attacks.
Database	An organized collection of structured information, or data, typically stored electronically.
Encryption	The process of converting data into a secure format to prevent unauthorized access.
Front-end	The user-facing part of a web application that handles the visual presentation and interaction.

Git	A distributed version control system for tracking changes in source code.
GitHub	A web-based platform for version control and collaboration using Git
Malware	Malicious software designed to harm, exploit, or disrupt computer systems.
React.js	A JavaScript library for building user interfaces.
Scams	Fraudulent schemes used to deceive people into providing money, personal details, or access to sensitive data.
Spring Boot	A Java-based framework for building microservices and web applications.
Vite	A build tool for modern web development.
Wireframe	A visual guide that represents the skeletal framework of a website or application.

1.4. References

[1] IEEE Computer Society. (1998). IEEE 830-1998: Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998. Source: IEEE Xplore Digital Library.

[2] React. (2019). Glossary. Source: https://legacy.reactjs.org/docs/glossary.html

[3] Axios Contributors. (2024). Axios API Reference. Source: https://axios-http.com

[4] National Institute of Standards and Technology (NIST). (2018). Cybersecurity Framework. Source: https://www.nist.gov/cyberframework

2. Overall Description

2.1. Product perspective

These modules ensure a seamless experience for students while providing teachers with tools to monitor progress.

Modular Decomposition

Module 1: Gamified Online Privacy and Safety

- Players analyze digital case files and sort information into “Safe to Share” vs. “Not Safe to Share” categories.

Module 2: Gamified Password Security

- Players loot treasure chests to collect password fragments and construct a secure password.
- A simulated hacker bot attempts to crack weak passwords, requiring players to strengthen their defenses.

Module 3: Gamified Phishing and Scam Awareness

- Players navigate a virtual escape room where they must identify:
    - Phishing emails with suspicious attachments and links.
    - Fake login pages designed to steal credentials.
    - Scam messages using social engineering tactics.
- Successful identification unlocks security codes needed to escape the room.

Module 4: Teacher Dashboard

- Teachers can view individual and class-wide progress, including mission completion and scores. Allows teachers to filter.
- Enables teachers to export reports in PDF or CSV format for progress tracking.

2.2. User characteristics

CyberKids is designed primarily for elementary and junior high school students, introducing them to cybersecurity concepts in a fun, interactive way. The platform helps students develop safe online habits through gamified learning experiences, while teachers and parents can track their progress and guide them toward better digital awareness

.

1. Student Users

The primary users of CyberKids are young learners who are becoming more engaged with technology and the internet. The platform is tailored to their learning style through an intuitive, interactive, and engaging game-based format.

- **Need for Early Cybersecurity Awareness**

  Many students lack proper guidance on how to stay safe online, making them vulnerable to phishing, scams, and cyberbullying. CyberKids fills this gap by introducing essential cybersecurity knowledge through engaging missions and challenges that teach digital safety without overwhelming young users with technical jargon.

- **Interactive and Gamified Learning**

  Traditional learning materials may not hold the attention of young students. CyberKids ensures engagement by integrating reward systems, level progression, and real-time feedback, making cybersecurity education fun and rewarding. Students can

unlock new levels and earn badges as they progress through different cybersecurity challenges.

- **Age-Appropriate Content**

Unlike general cybersecurity platforms designed for adults, CyberKids presents lessons in a way that is simple, visual, and interactive, making it easier for young learners to grasp concepts like password security, phishing awareness, and safe social media usage.

2. Teachers and Educators

Teachers play a crucial role in guiding students' learning experience and ensuring they understand the importance of cybersecurity. CyberKids supports teachers by offering a dashboard that tracks student progress and performance.

- **Classroom Integration**

Teachers can integrate CyberKids into their curriculum as an interactive learning tool to supplement traditional cybersecurity education. The platform provides structured challenges that align with key digital literacy learning goals.

- **Student Progress Monitoring**

Educators can view student scores, track progress across different cybersecurity topics, and identify areas where students need improvement. This data-driven approach helps teachers offer targeted guidance to students struggling with certain cybersecurity concepts.

- **Facilitating Cyber Safety Discussions**

Teachers can use CyberKids as a starting point for classroom discussions about online safety, responsible internet use, and digital ethics, ensuring students apply what they learn in real-life scenarios.

3. Parents and Guardians

Parents are often concerned about their children's online activities and exposure to digital threats. CyberKids provides them with a way to ensure their children are learning safe online behaviors through an engaging and structured platform.

- **Parental Supervision and Insights**

  Parents can monitor their child's progress through reports on completed missions, scores, and areas needing improvement. This enables parents to have meaningful conversations about cybersecurity at home.

- **Encouraging Safe Internet Habits at Home**

  With CyberKids, parents can actively participate in reinforcing safe online behaviors, helping their children apply cybersecurity lessons beyond the digital classroom.

User Roles and Privileges

1. Students (Players)

- **Description:** Students in Grades 5-6 at CIT-University who participate in the game to learn about cybersecurity concepts.
- **Roles & Privileges:**
  - Access and play all game missions.
  - View their own scores and rankings on the leaderboard.
  - Customize their profile (avatar and display name only).

2. Teachers

- **Description:** Educators who monitor and assess student progress in CyberKids.
- **Roles & Privileges:**

- Access the Teacher Dashboard to track students' progress, scores, and mission completion rates.
- Generate performance reports to assess learning outcomes.
- View leaderboard standings for class rankings.

3. Admin

- **Description:** System administrators responsible for managing and maintaining the CyberKids application.
- **Roles & Privileges:**
  - Maintain system functionality and ensure a smooth gaming experience.
  - Manage technical updates, bug fixes, and security measures.
  - Oversee user management, including teacher and student accounts.
  - Ensure leaderboard accuracy and resolve any scoring discrepancies.

2.4. Constraints

The development of CyberKids is subject to various technical, regulatory, and operational constraints that can impact the system design, operation, and deployment. Such constraints need to be critically analyzed to ensure the system is pragmatic, effective, and compliant with the relevant standards.

Regulatory Policies

CyberKids must comply with data protection law and education code in a bid to protect user information and enable the consistency of career guidance services. The system should be compliant with:

- The Data Privacy Act of 2012 (RA 10173) intends to protect students' information, grades, and career options by guaranteeing confidentiality and security.

Hardware Limitations

The system is designed to run on standard desktop and laptop computers commonly used in CIT-University. However, there are some limitations to consider:

- Support for Low-End Hardware – The system must be optimized to function smoothly on low-end to mid-range hardware, ensuring accessibility for all students, even those with older devices.
- Resource Usage Optimization – To prevent performance issues, the system should minimize excessive CPU and memory usage while still providing a seamless user experience.

Interfaces to Other Applications

The system operates independently and does not integrate with external applications, but this comes with certain constraints:

- No Social Media Integration – Unlike other modern educational tools, the system does not connect with social media platforms for sharing progress or achievements.
- Internal Data Storage Only – All user data, including scores, progress, and mission completion information, is stored within an internal database. This limits external access but ensures security and control over student information.

Parallel Operation

While the game is designed as a single-player experience, multiple students can play simultaneously. However, there are some operational constraints:

- Independent Sessions – Each student's game session runs independently, meaning their progress does not interfere with other users.
- Scalability Considerations – Although the system allows multiple concurrent users, excessive simultaneous logins may require server performance optimization.

Audit Functions

The system includes mechanisms for monitoring student activity and progress, though there are some limitations:

- Teacher Dashboard Tracking – Teachers can monitor student progress through a dashboard, but they can only view the data and cannot modify student scores or mission completion records.
- Session Logs and Data Retention – The system logs game sessions, scores, and mission completion data, but long-term storage may be limited based on server capacity.

Control Functions

Different user roles have varying levels of access and control over the system, but there are restrictions in place:

- Administrative Oversight – Admin users have full control over system maintenance, bug fixes, and security updates. They ensure the system remains functional and up to date.

- Limited Teacher Modifications – While teachers can monitor student progress, they do not have the ability to alter student scores or modify gameplay mechanics.

Reliability Requirements

The system aims to be stable and reliable, though certain factors may impact its performance:

- Crash Prevention – The system must function without unexpected crashes during gameplay to ensure a smooth user experience.
- Data Security and Retrieval – Student progress and leaderboard scores must be securely stored and retrievable in case of system failures.

Criticality of the Application

Although the system is designed primarily as an educational tool, reliability remains a key priority:

- Not a Mission-Critical System – The game serves as a supplementary educational resource rather than a system that directly impacts grades or official records.
- Importance of Seamless Learning – Despite not being mission-critical, the system must remain stable to maintain student engagement and educational value.

Safety and Security Considerations

Ensuring user security and data protection is a fundamental requirement of the system:

- No Collection of Personal Data – The system does not collect real-world personal data, ensuring privacy and compliance with data protection standards.
- Unauthorized Access Prevention – Security measures must be in place to prevent unauthorized access, particularly for student accounts, to protect progress and

sensitive game data.

2.5. Assumptions and dependencies

The development and functionality of CyberKids rely on the following assumptions and dependencies. Any changes to these factors may impact the system requirements and require modifications to the software.

Assumptions

1. **Hardware Availability**
    - The game is assumed to run on standard desktop computers available at CIT-University.
    - Devices used must have basic input peripherals (keyboard and mouse).
2. **Operating System Compatibility**
    - The system assumes that devices will run on Windows (Google Chrome, Firefox, Edge). If a specific OS is not supported, additional development effort may be required.
3. **Internet Connectivity**
    - The leaderboard and teacher dashboard assume a stable internet connection for real-time updates.
4. **User Digital Literacy**
    - Students (Grades 5-6) are assumed to have basic computer skills, such as navigating a game interface, using a mouse/keyboard, and following on-screen instructions.

Dependencies

1. **Database System**
    - The leaderboard, student progress tracking, and user profiles depend on a functional database (e.g., MySQL, Firebase).
2. **Web-Based Frameworks and Technologies**
    - The game relies on web-based technologies (e.g., React, JavaScript, HTML5, CSS) for execution.
3. **Security Policies**
    - The system follows CIT-University's IT security policies to prevent unauthorized access and protect student data.
    - Changes in security regulations may require updates to authentication and access control mechanisms.
4. **Server Availability**
    - The leaderboard and teacher dashboard depend on a centralized server for data storage and retrieval.
5. **Third-Party Libraries and APIs**
    - The system may use external libraries or APIs for password encryption, data storage, and UI enhancements.
    - If these libraries become deprecated, alternative solutions must be integrated.

3. Specific Requirements

3.1. External interface requirements

3.1.1. Hardware interfaces

The system is designed to operate on standard computing hardware used at CIT-University, ensuring compatibility with a wide range of devices. The hardware interface requirements include:

- **Client-side Requirements**
  - **Devices:** Desktop and Laptop with a modern web browser.
  - **Browers:** Chrome, Firefox, Edge, and Safari (latest stable versions).
  - **Internet Connection:** Minimum 5 Mbps for smooth interaction.

- **Supported Devices** – The software will run on desktop and laptop computers with Windows operating systems, ensuring accessibility for students. No support for mobile or tablet devices is currently planned.

- **Minimum Hardware Specifications** – The system must be optimized to function on low-end to mid-range computers, requiring at least:
  - Processor: Intel Core i3 (or equivalent) with a 2.0 GHz clock speed
  - RAM: 4 GB minimum, 8 GB recommended for optimal performance
  - Storage: At least 2 GB of available disk space for installation and data storage
  - Graphics: Integrated GPU support (no dedicated graphics card required)
- **Peripheral Device Support** – The system will support standard input and output devices, including:

- Keyboard and Mouse: Required for user interactions in the system

3.1.2. Software interfaces

The system will interact with various software components to ensure smooth functionality, including the operating system, database management system, and web-based services.

- **Operating System Compatibility** – The software will be compatible with the following operating systems:
  - Windows 10 and later – Officially supported, tested for stability and security
- **Database Management System (DBMS)** – The system will store all user progress, scores, and logs in a relational database. Supported databases include:
  - MySQL 8.0 or later – Primary database for storing structured data
  - PostgreSQL 13 or later – Alternative database option for scalability
- **Frameworks and Development Tools** – The system is built using the following technologies:
  - Game Engine: Roblox as primary game engine which will be deployed in Roblox server.
  - Backend: Java Spring Boot for API handling and business logic processing
  - Frontend: React.js for user interface rendering
  - Data Management: Hibernate ORM for database interactions
- **APIs:**
  - RESTful APIs for communication between frontend and backend.

3.1.3. Communications interfaces

The CyberKids requires connectivity to various network services to enable multiplayer interactions, leaderboard updates, and real-time event handling.

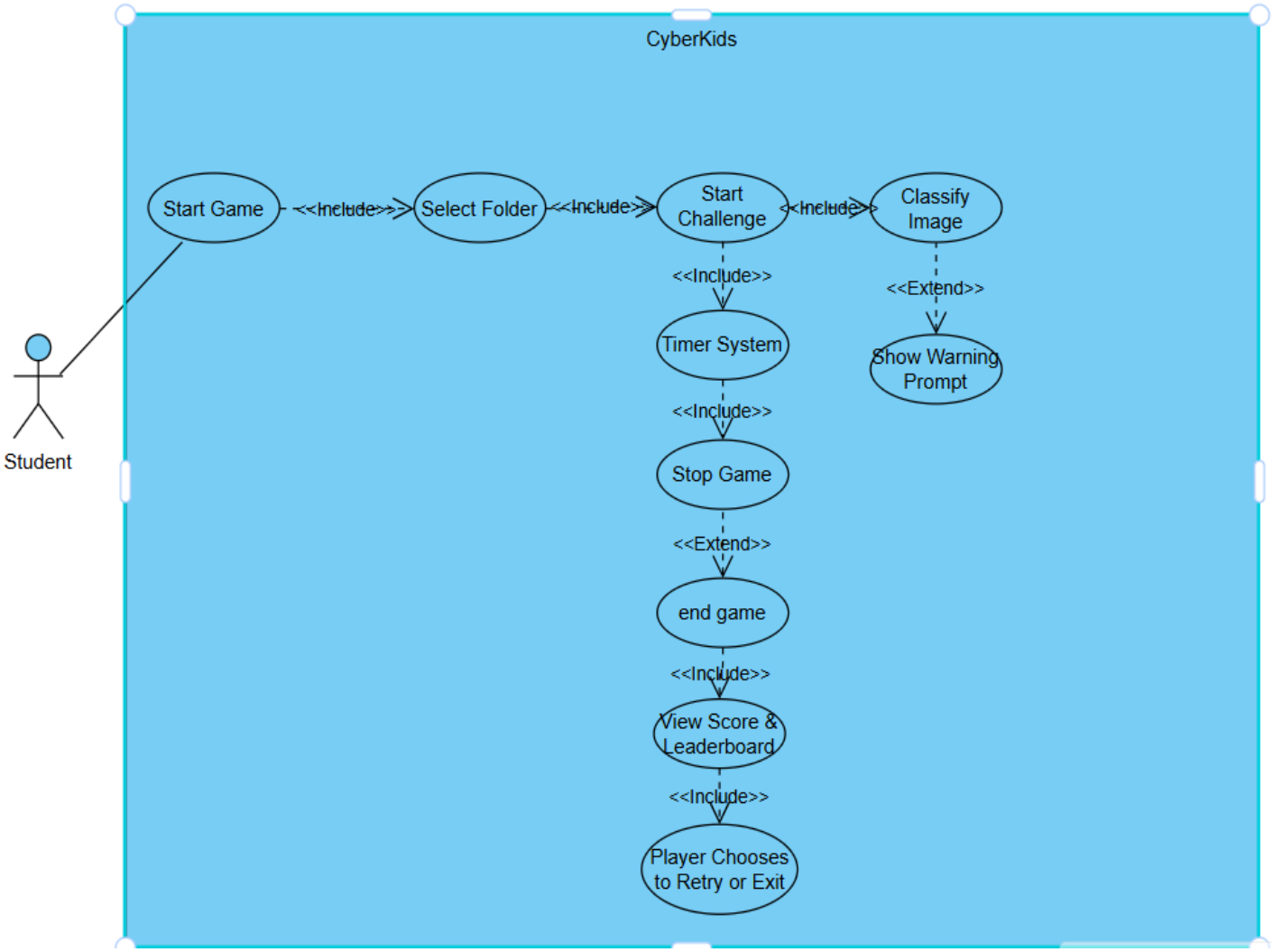
- **Internet Connectivity Requirements:**
    - The game requires an active internet connection for live updates, leaderboard rankings, and cloud-based data storage.
- **Protocols Used:**
    - HTTP/HTTPS: Secure communication between client and game server.
    - WebSocket Protocol: For real-time interactions, such as in-game chat, multiplayer elements, or live updates.
    - RESTful API / GraphQL: Handles game progress, leaderboard updates, and authentication requests.

3.2. Functional requirements

Module 1: Gamified Online Privacy and Safety

1.1 Information Classification Sorting Challenge

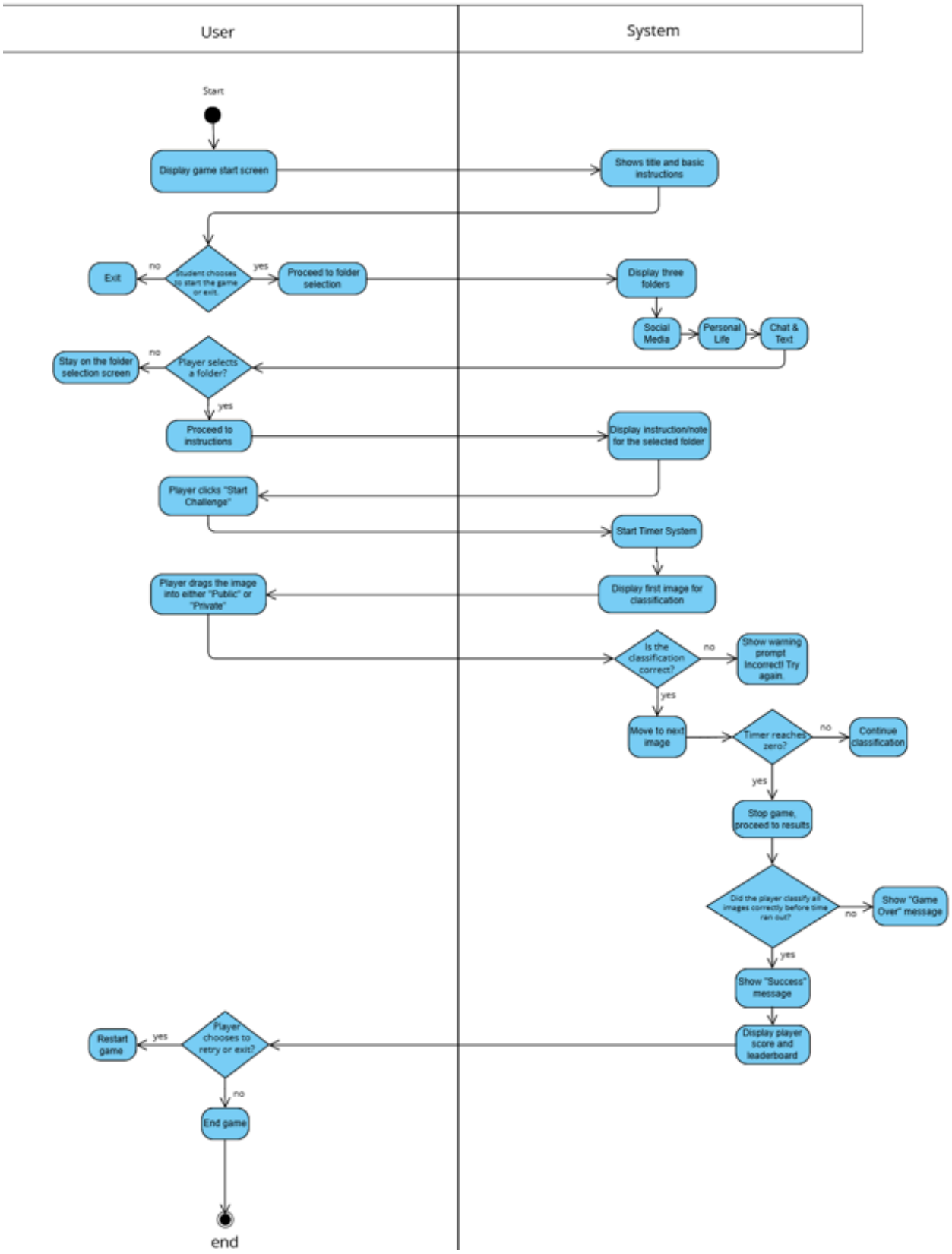
Use Case Diagram *(Information Classification Sorting Challenge)*



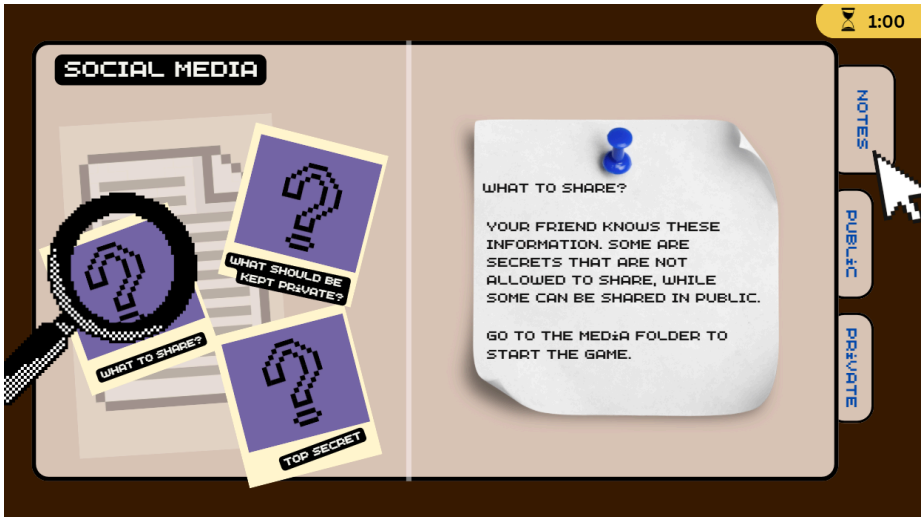
Use Case Description

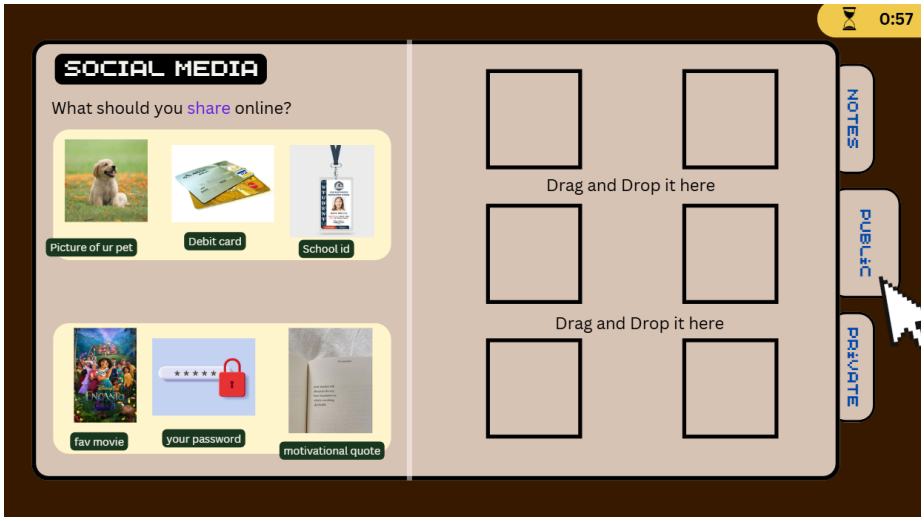
Use Case ID	UC-001
Use Case Name	Information Classification Sorting Challenge
Actor	Student
Description	The Information Classification Sorting Challenge is an interactive game where students categorize pieces of information into Personal (Safe to Share) and Private (Not Safe to Share). This challenge aims to educate students on responsible information sharing and online safety by reinforcing their ability to distinguish between different types of information.
Flow of Events	1. The student starts the challenge, and a set of information items appears on the screen. 2. The student drags and drops each item into either the Personal (Safe to Share) or Private (Not Safe to Share) category. 3. The system validates each selection and provides instant feedback (correct/incorrect). 4. If the answer is incorrect, a brief explanation appears to help the student understand why the information is categorized as such. 5. The challenge continues until all items are sorted or time runs out (if the Timer System is active).
Precondition	The student is logged into the system.
Postcondition	The student's score is recorded and displayed.

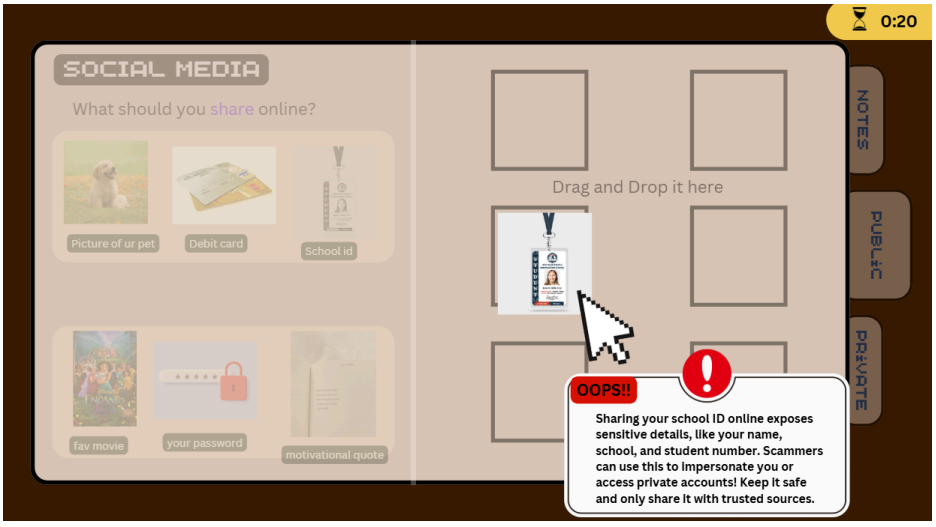
Activity Diagram *(Information Classification Sorting Challenge)*



Wireframe *(Information Classification Sorting Challenge)*

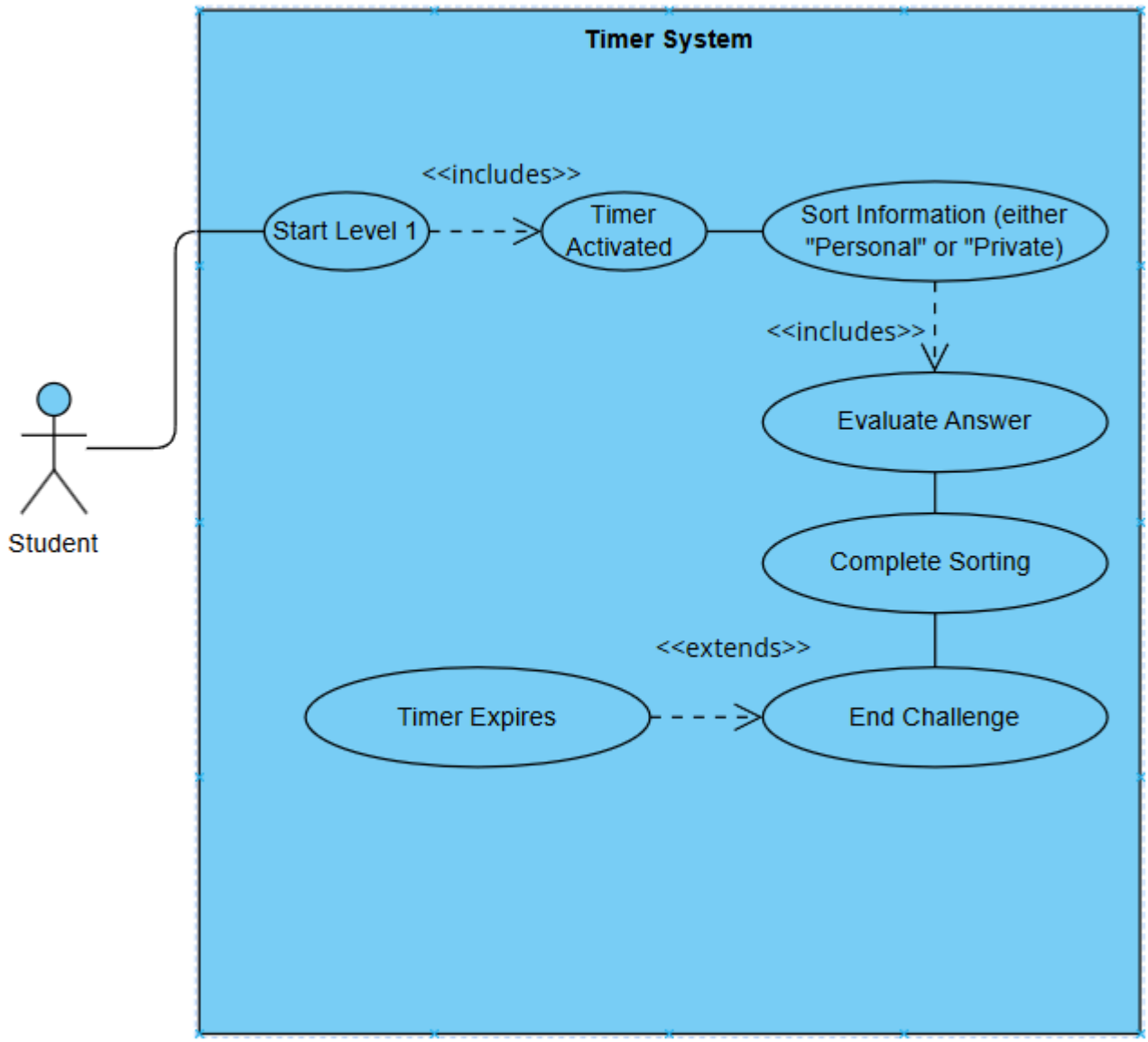






1.2 Timer System

***Use Case Diagram** (Timer System)*

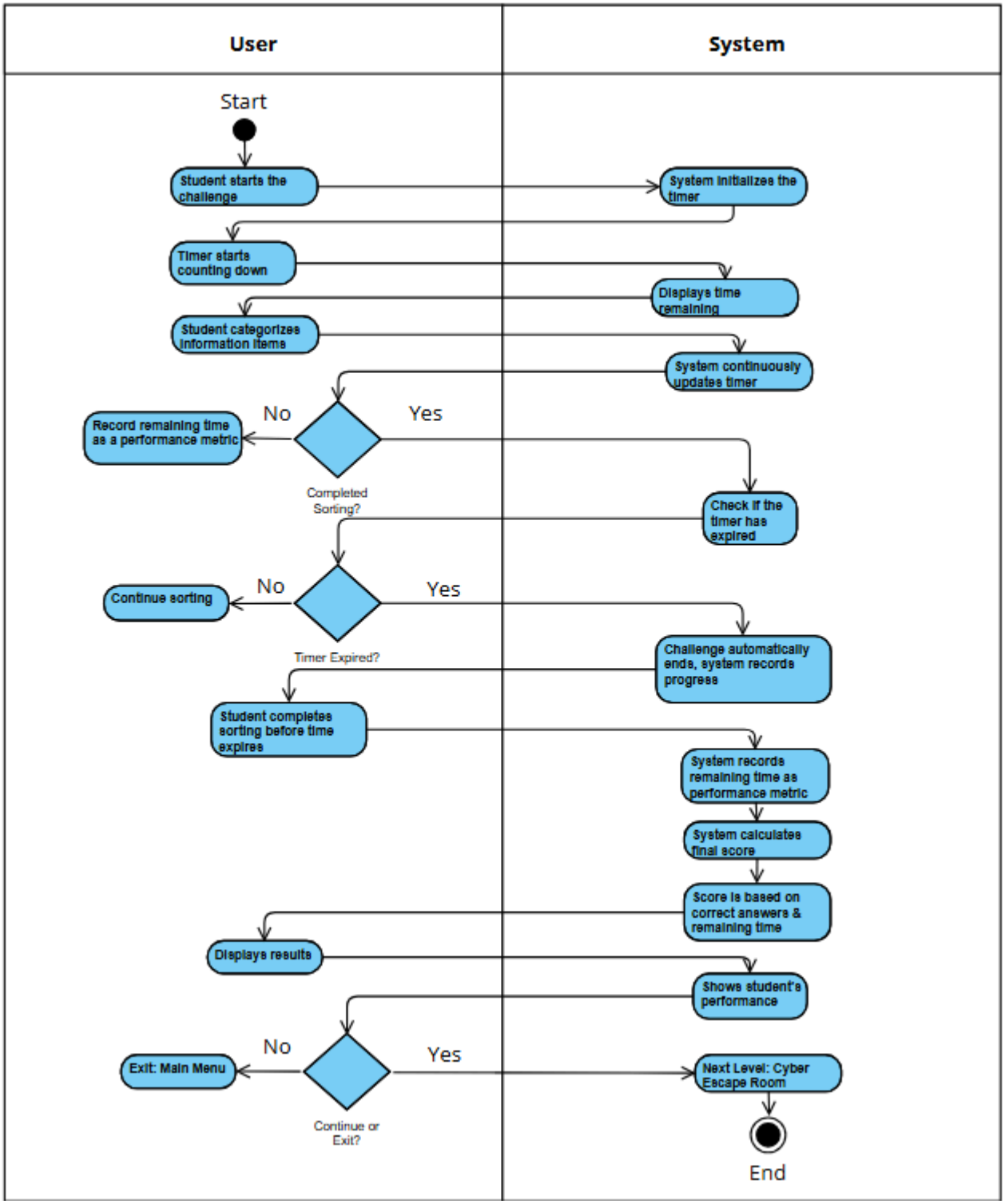


Use Case Description

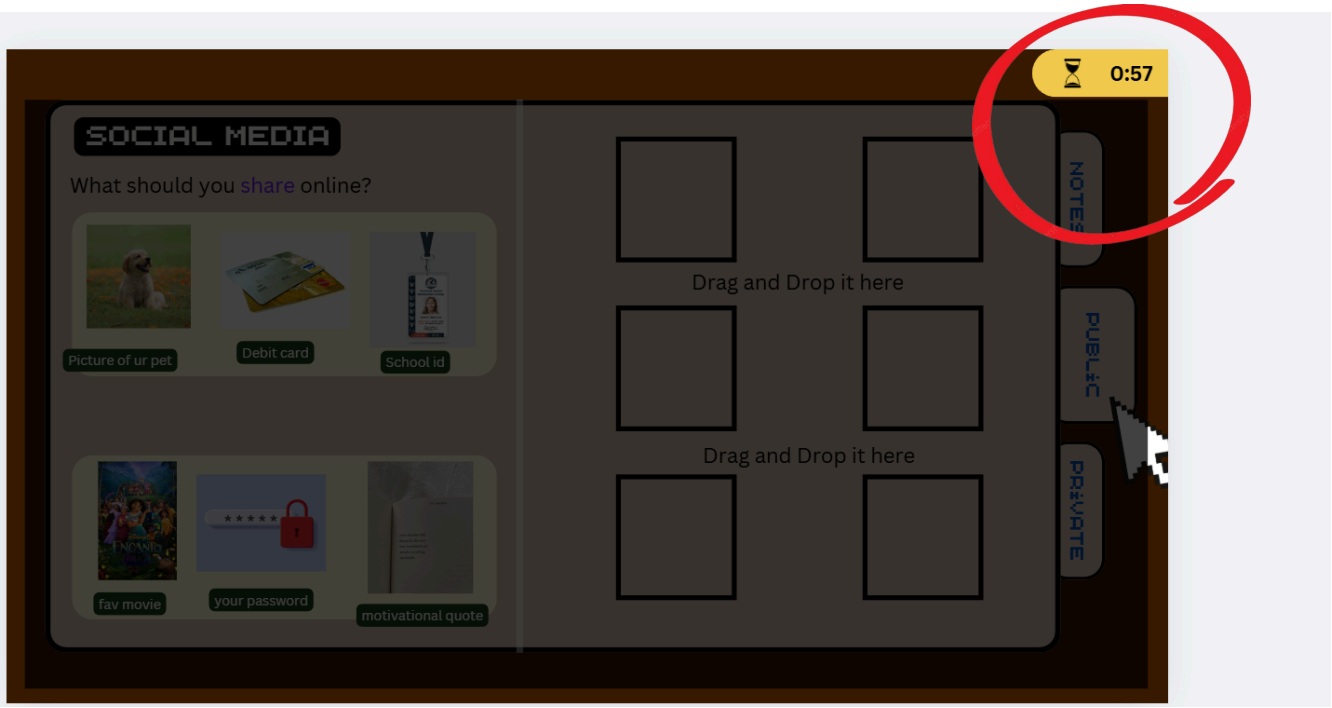
Use Case ID	UC-002
Use Case Name	Timer System
Actor	Student
Description	The Timer System introduces a time-based challenge in the Information Classification Sorting Challenge. It encourages students to quickly and accurately categorize information into Personal (Safe to Share) and Private (Not Safe to Share) before time runs out.
Flow of Events	1. The student starts the challenge, and the timer begins counting down from the predefined time limit. 2. The student categorizes information items into the appropriate categories while the timer continuously updates. 3. If the time runs out before completion, the challenge automatically ends, and the system records the student's progress. 4. If the student completes the sorting before time expires, the remaining time is recorded as a performance metric. 5. The final score is calculated based on correct answers and remaining time.
Precondition	● The Information Classification Sorting Challenge has started. ● The timer is set with a predefined duration.

Postcondition	The system either ends the game due to time expiry or allows the student to complete the challenge within the allotted time.

***Activity Diagram** (Timer System)*

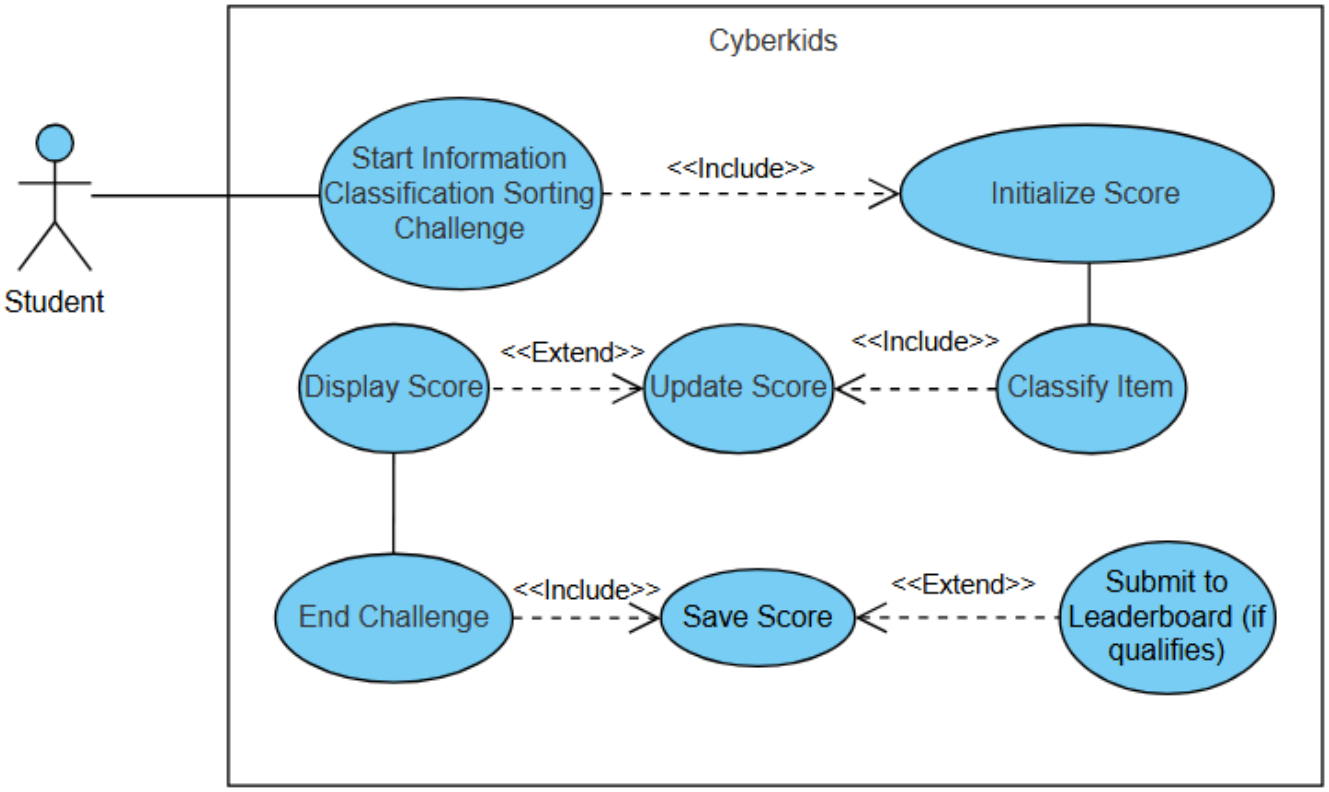


***Wireframe** (Timer System)*



1.3 Point-Based Scoring System

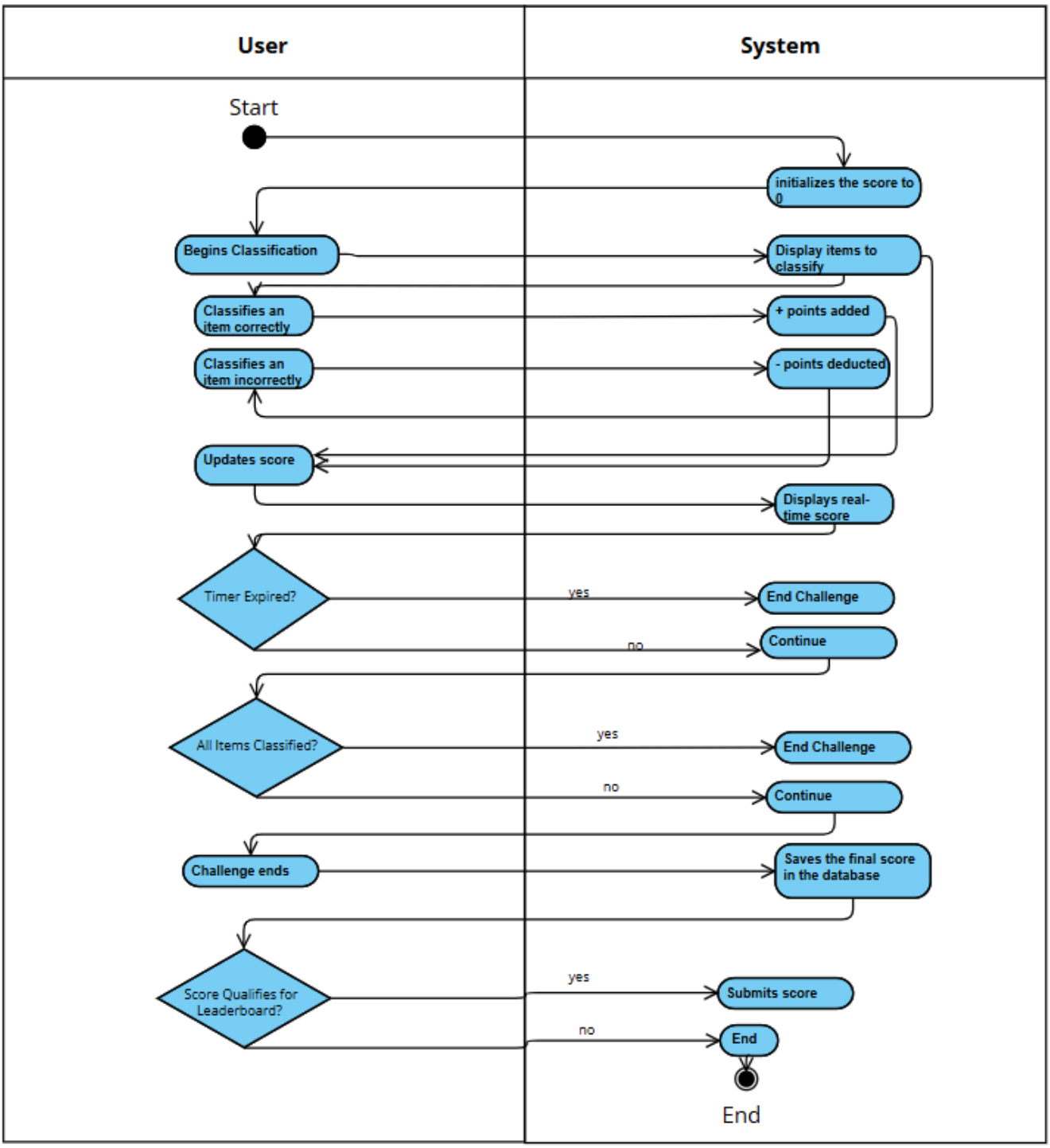
***Use Case Diagram** (Scoring System)*



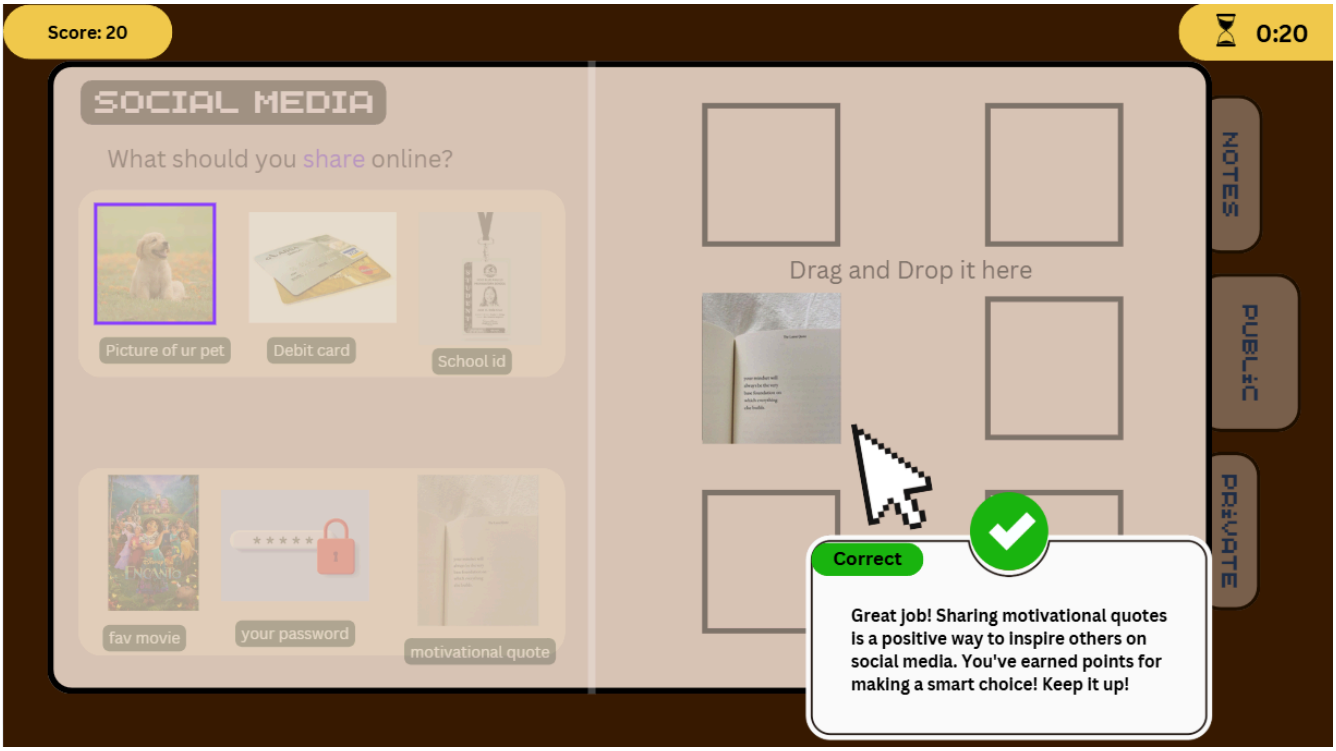
Use Case Description

Use Case ID	UC-003
Use Case Name	Scoring System
Actor	Student
Description	The Point-Based Scoring System assigns points based on student performance in the Information Classification Sorting Challenge.
Flow of Events	1. The student starts the Information Classification Sorting Challenge, and the system initializes the score to 0. 2. As the student classifies each item: ● Correct classification: +10 points 3. Incorrect classification: -5 points 4. The system updates and displays the score in real time. 5. If the student finishes before the timer expires, their remaining time is stored but does not impact the score. 6. When the challenge ends, the system: ● Saves the final score in the database. ● Submits the score to the leaderboard if it qualifies.
Precondition	The student must start the Information Classification Sorting Challenge.
Postcondition	The final score is stored in the database.

***Activity Diagram** (Scoring System)*

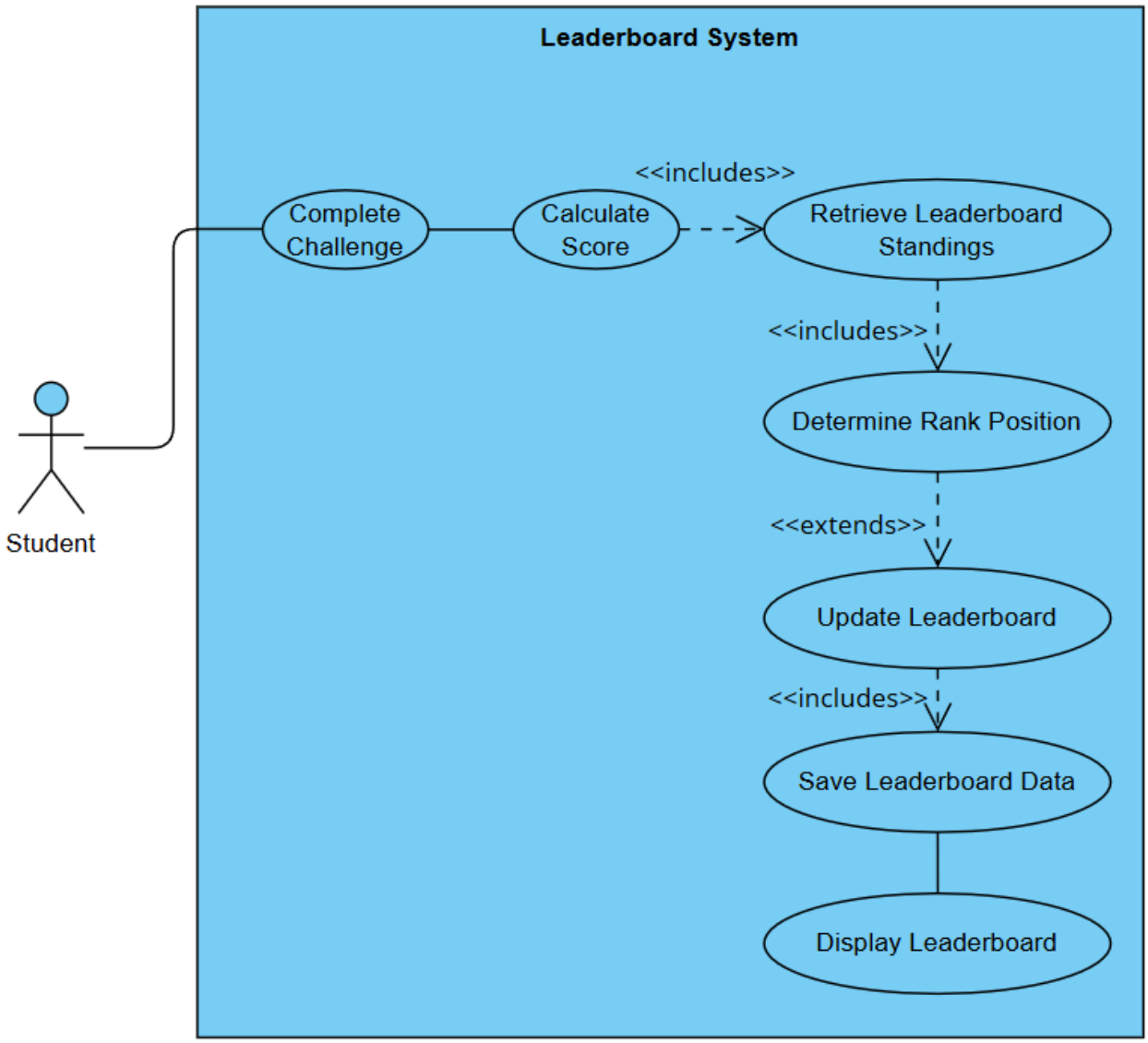


Wireframe *(Scoring System)*



1.4 Leaderboard for the Online Privacy Game

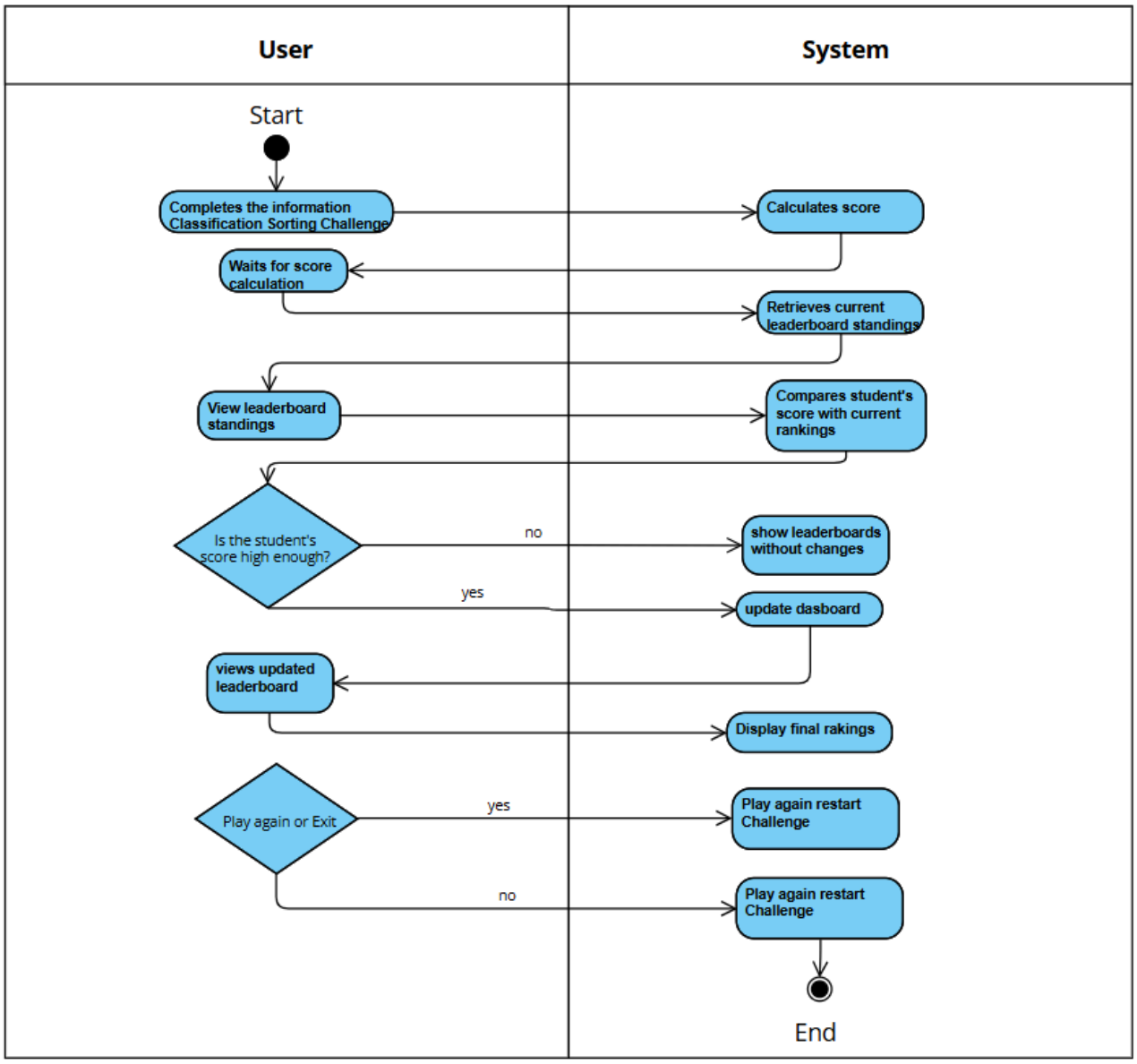
***Use Case Diagram** (Leaderboard)*



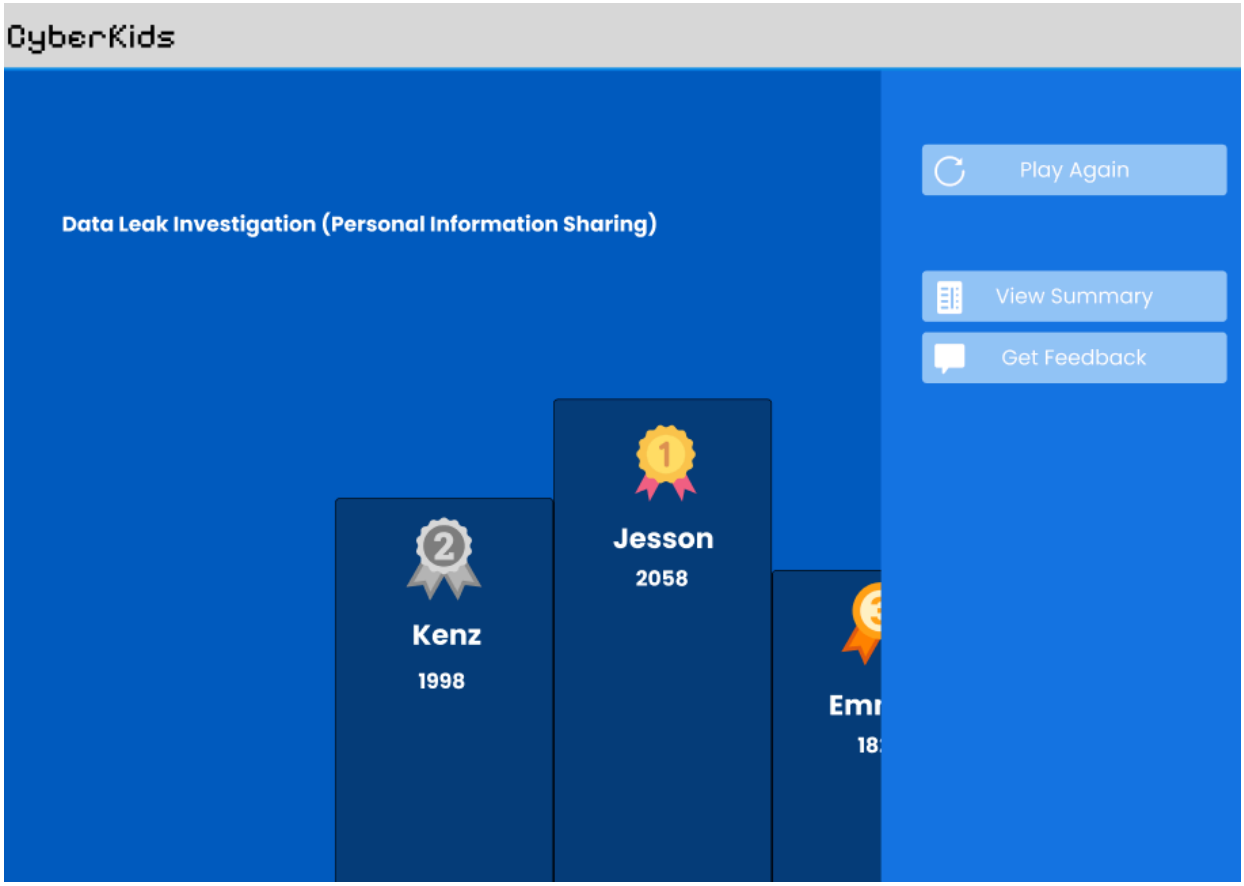
Use Case Description

Use Case ID	UC-004
Use Case Name	Leaderboard for the Online Privacy Game
Actor	Student
Description	The Leaderboard System ranks students based on their performance in the Online Privacy and Safety Game. It promotes engagement by displaying the top players, encouraging students to improve their scores by making accurate and quick decisions when categorizing personal and private information.
Flow of Events	1. After completing the Information Classification Sorting Challenge, the system calculates the student's score based on accuracy and remaining time. 2. The system retrieves the current leaderboard standings and determines if the student qualifies for a ranked position. 3. If the student's score is high enough, the leaderboard is updated dynamically with their ranking. 4. The system displays the leaderboard, showing the top players and their scores.
Precondition	The student has completed the Online Privacy and Safety Game.
Postcondition	The leaderboard is updated if the student achieves a high enough score.

Activity Diagram *(Leaderboard)*



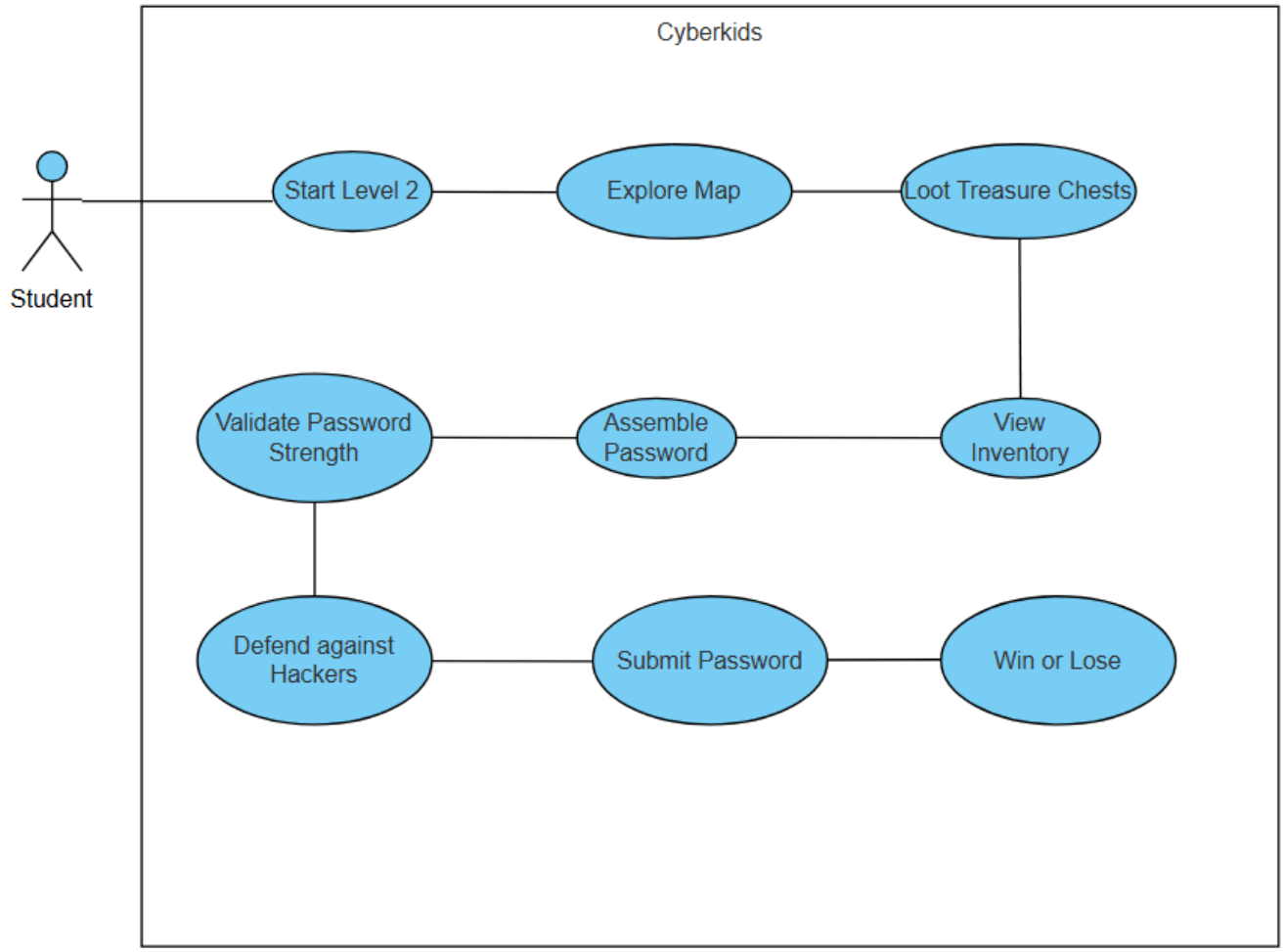
***Wireframe** (Leaderboard)*



Module 2: Gamified Password Security

2.1 Password Security Challenge

***Use Case Diagram** (Password Security Challenge)*

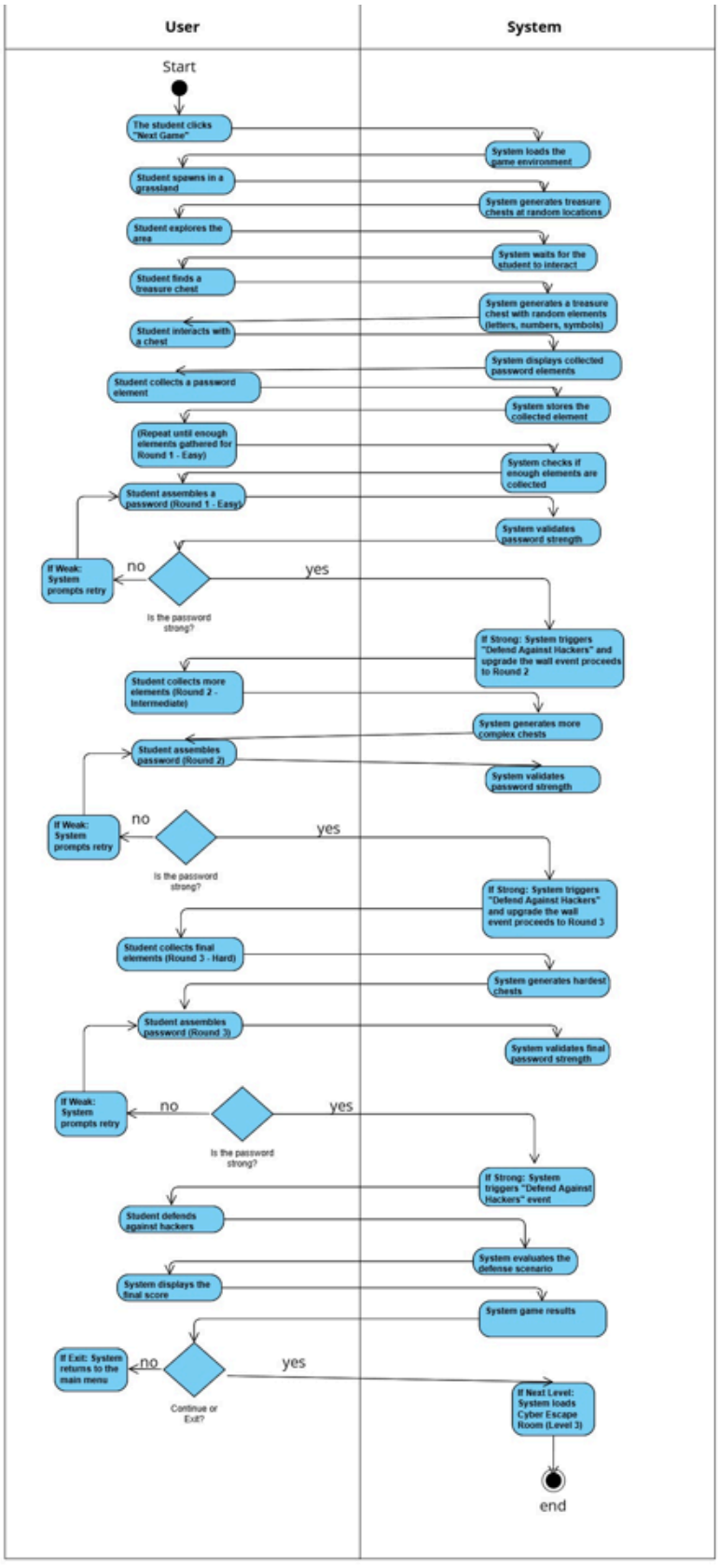


Use Case Description

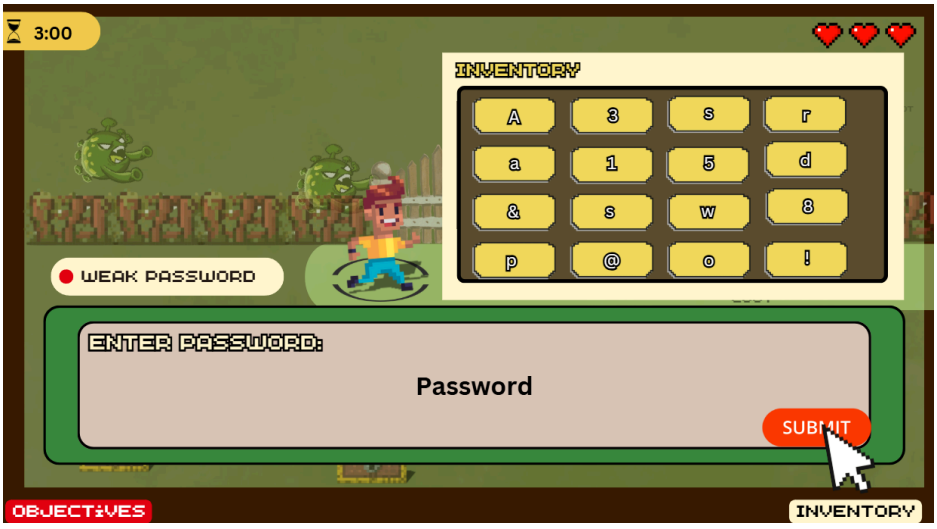
Use Case ID	UC-001
Use Case Name	Password Security Challenge
Actor	Student
Description	The Password Security Game is a gamified learning experience where players must collect password fragments from treasure chests, analyze their strength, and assemble the most secure password before reaching the final gate. If a password is too weak, simulated hacker bots will attack, requiring players to reinforce their defenses before the Cyber HQ is breached.
Flow of Events	1. The player enters the game area and receives an initial weak password. 2. The player explores the environment, searching for treasure chests that contain password fragments (letters, numbers, and symbols). 3. The player collects and reviews the password fragments in their inventory. 4. The player assembles a new password by selecting the strongest fragments. 5. If the password is weak, hacker bots attack, and the player must find stronger fragments. 6. The player proceeds to the security gate and submits the password. 7. If the password is strong, the player successfully secures the gate and wins the challenge. 8. If the password is still weak, the player receives feedback on how to improve it and must try again.

Precondition	The player has logged into the game and started the Password Security mission.
Postcondition	● If the player creates a strong password, the mission is marked as complete. ● If the player fails to create a strong password, they receive feedback and must retry. ● The system records the player's progress and updates their score and leaderboard position.

***Activity Diagram** (Password Security Challenge)*

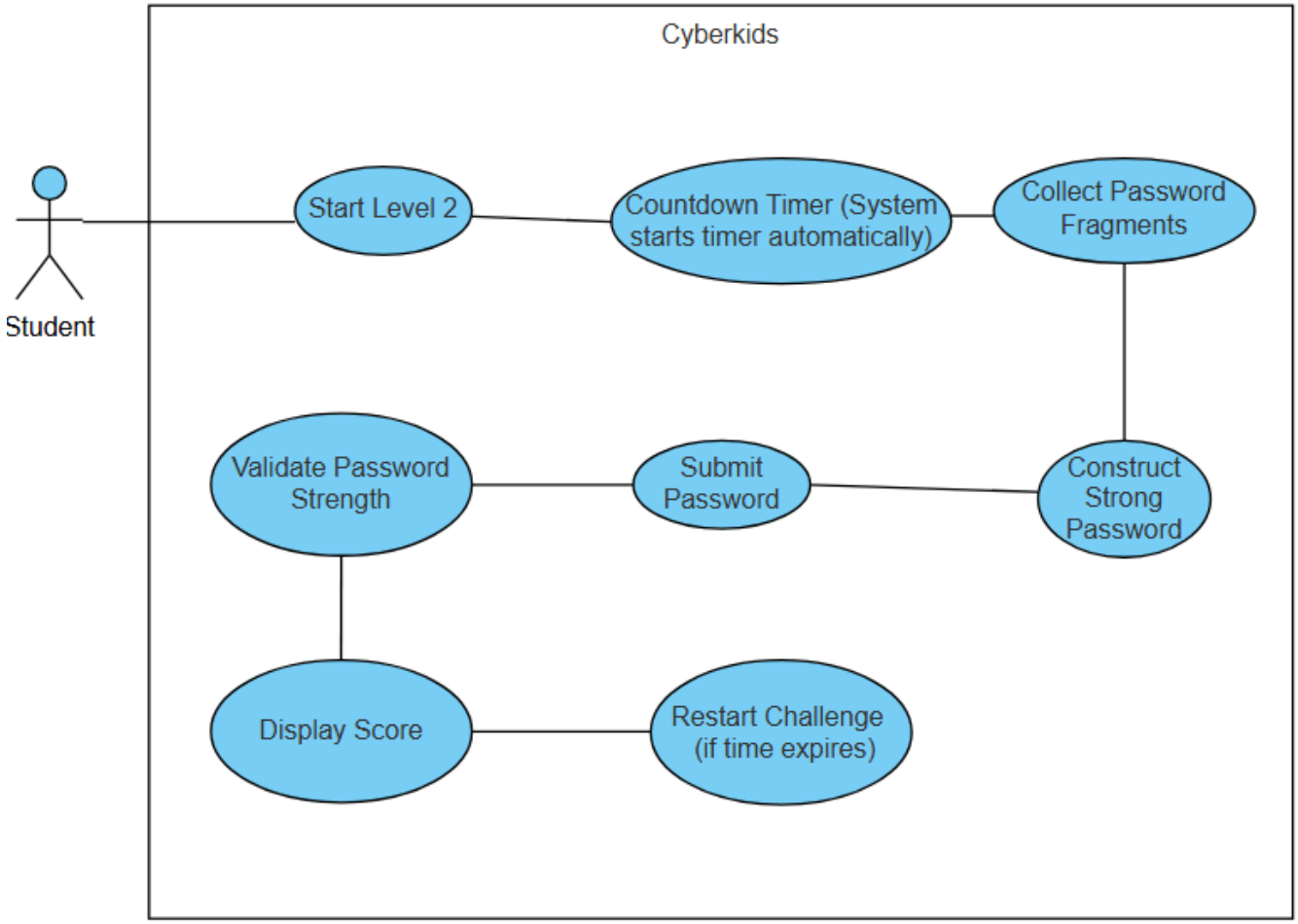


Wireframe *(Password Security Challenge)*



2.2 Timer System

***Use Case Diagram** (Timer System)*

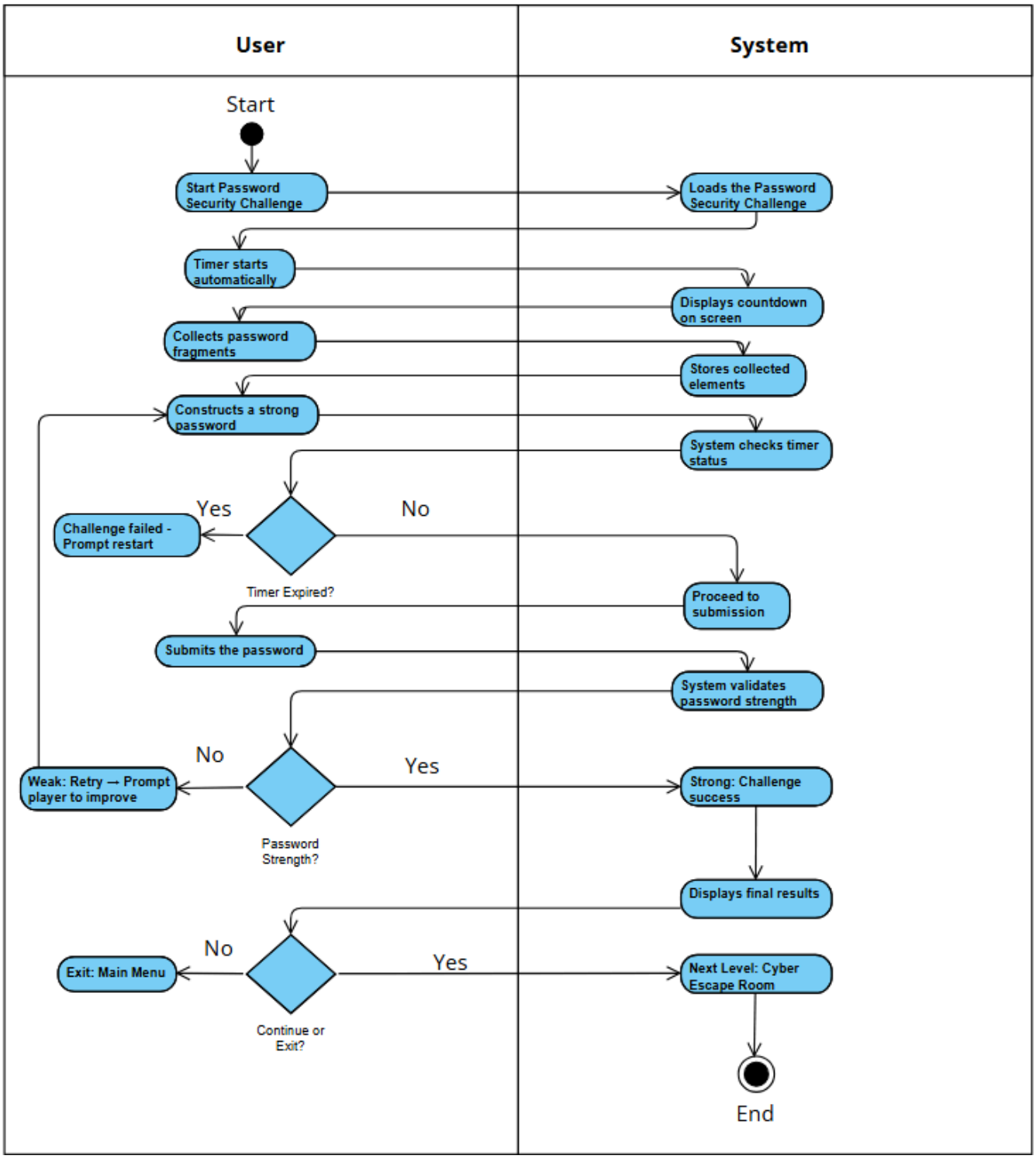


Use Case Description

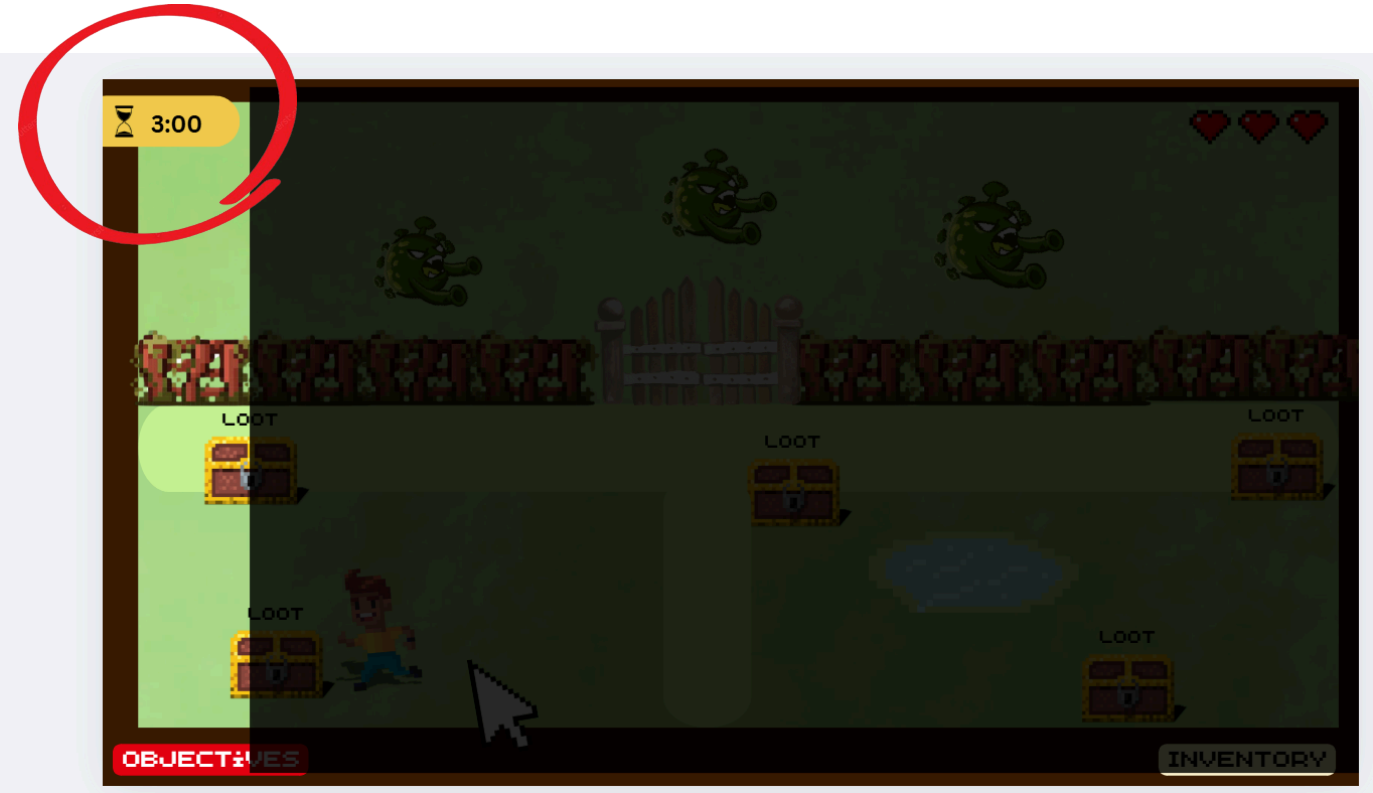
Use Case ID	UC-002
Use Case Name	Timer System
Actor	Student
Description	It tracks the time allocated for players to complete the Password Security Challenge. The timer starts when the challenge begins and stops when the player submits their password. If the timer runs out before submission, the player fails the challenge and must restart.
Flow of Events	1. The player starts the Password Security Challenge, and the timer automatically begins. 2. The countdown is displayed on the screen, showing the remaining time. 3. The player collects password fragments and constructs a strong password while being aware of the timer. 4. The player submits the password: 5. If submitted before time runs out, the system validates the password. 6. If time expires before submission, the challenge is failed, and the player must restart. 7. After submission (whether successful or failed), the final time is recorded for the leaderboard.
Precondition	● The player has entered the Password Security Challenge. ● The timer is initialized and ready to start.
Postcondition	● The timer records the total time taken by the player.

	● If the time runs out before submission, the player fails the challenge and must retry. ● If the player completes the challenge, their time is saved for the leaderboard ranking.

***Activity Diagram** (Timer System)*

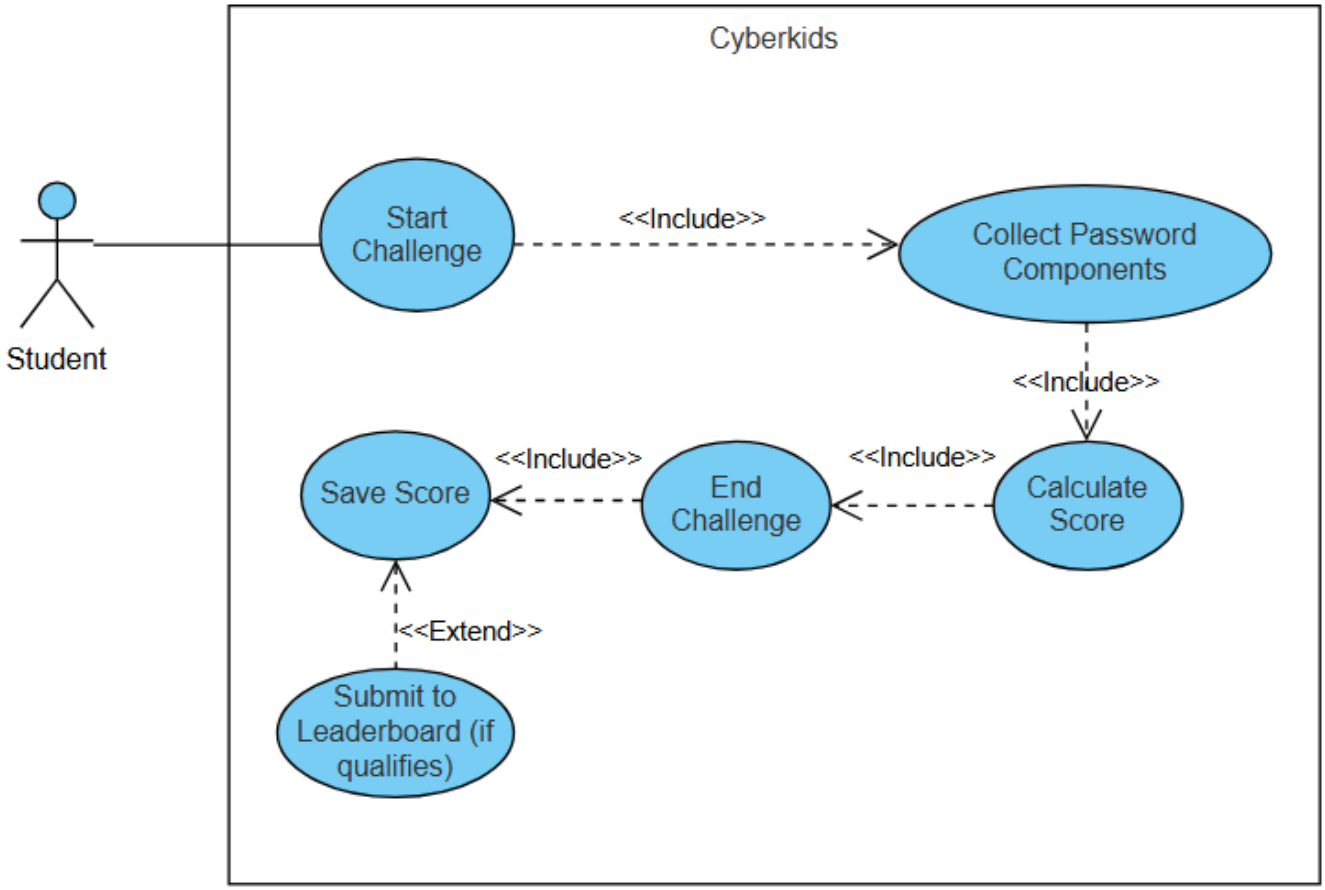


***Wireframe** (Timer System)*



2.3 Point-Based Scoring System

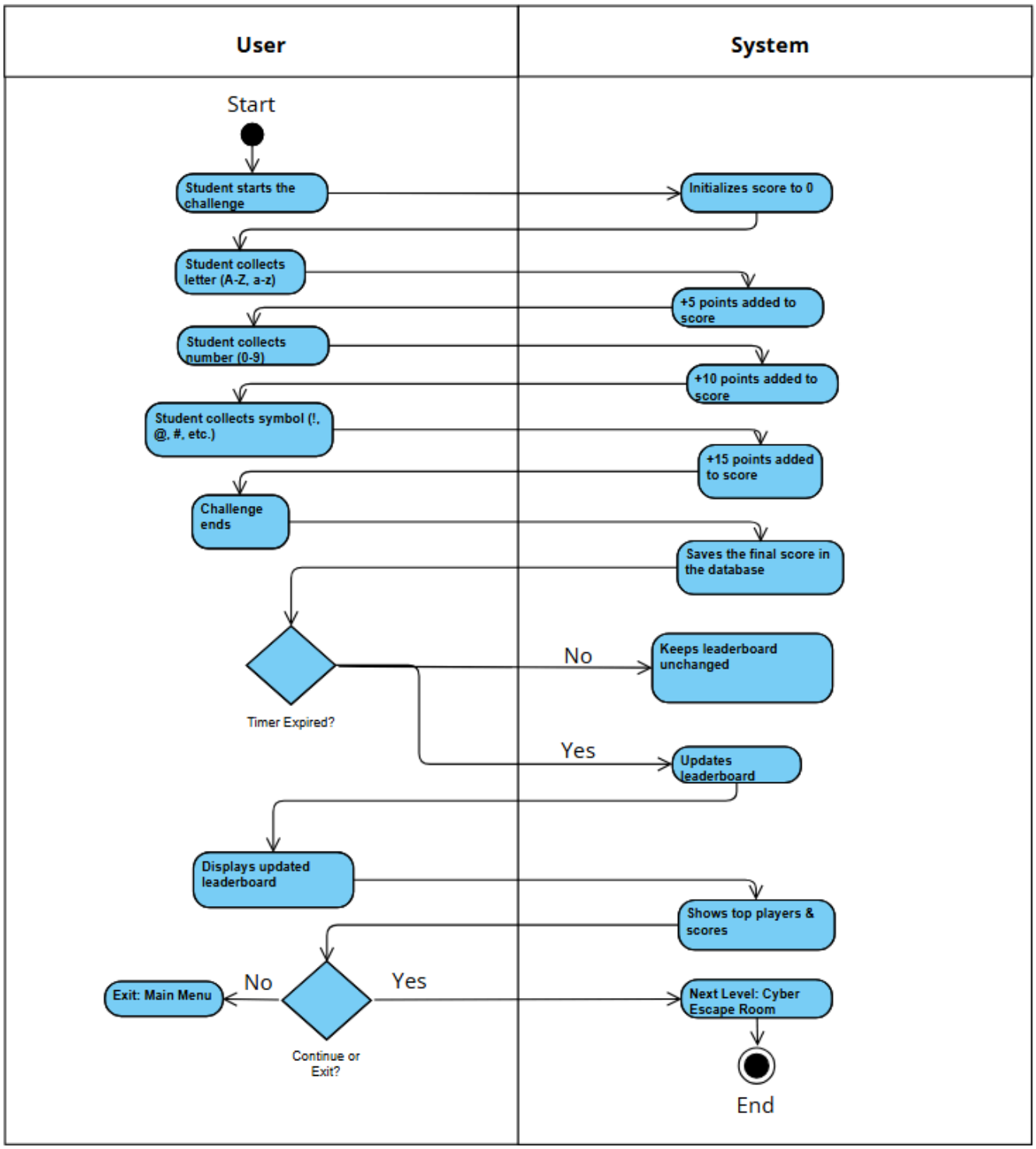
***Use Case Diagram** (Scoring System)*



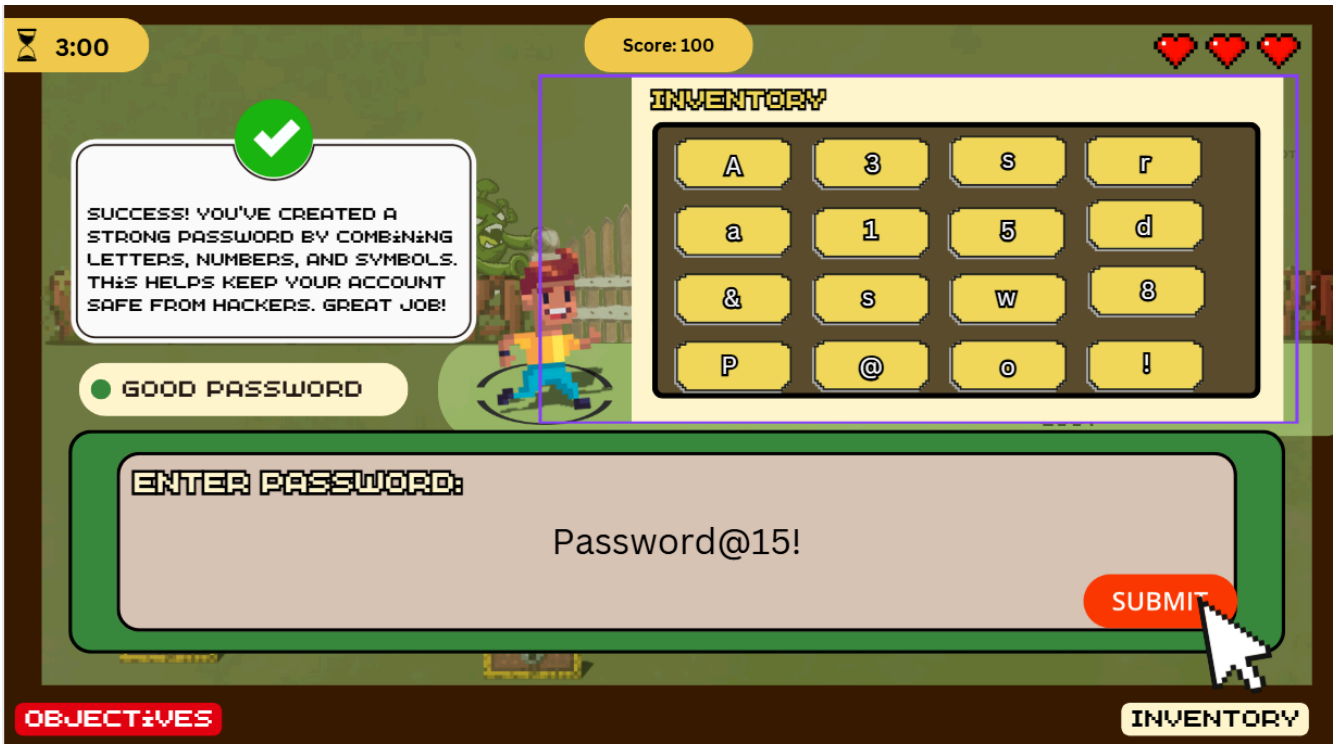
Use Case Description

Use Case ID	UC-003
Use Case Name	Scoring System
Actor	Student
Description	The Point-Based Scoring System assigns points based on how well students collect and assemble strong passwords during the Password Security Challenge. The system rewards students for choosing a diverse mix of letters, numbers, and symbols, promoting the creation of strong passwords
Flow of Events	1. The student starts the Password Security Challenge, and the system initializes the score to 0. 2. The student collects password components (letters, numbers, and symbols) from chests: ● Letter (A-Z, a-z): +5 points ● Number (0-9): +10 points ● Symbol (!, @, #, etc.): +15 points (Encourages password complexity) 3. When the challenge ends, the system: ● Saves the final score in the database. ● Submits the score to the leaderboard if it qualifies.
Precondition	The student must start the Password Security Challenge.
Postcondition	The final score is stored in the database.

***Activity Diagram** (Timer System)*

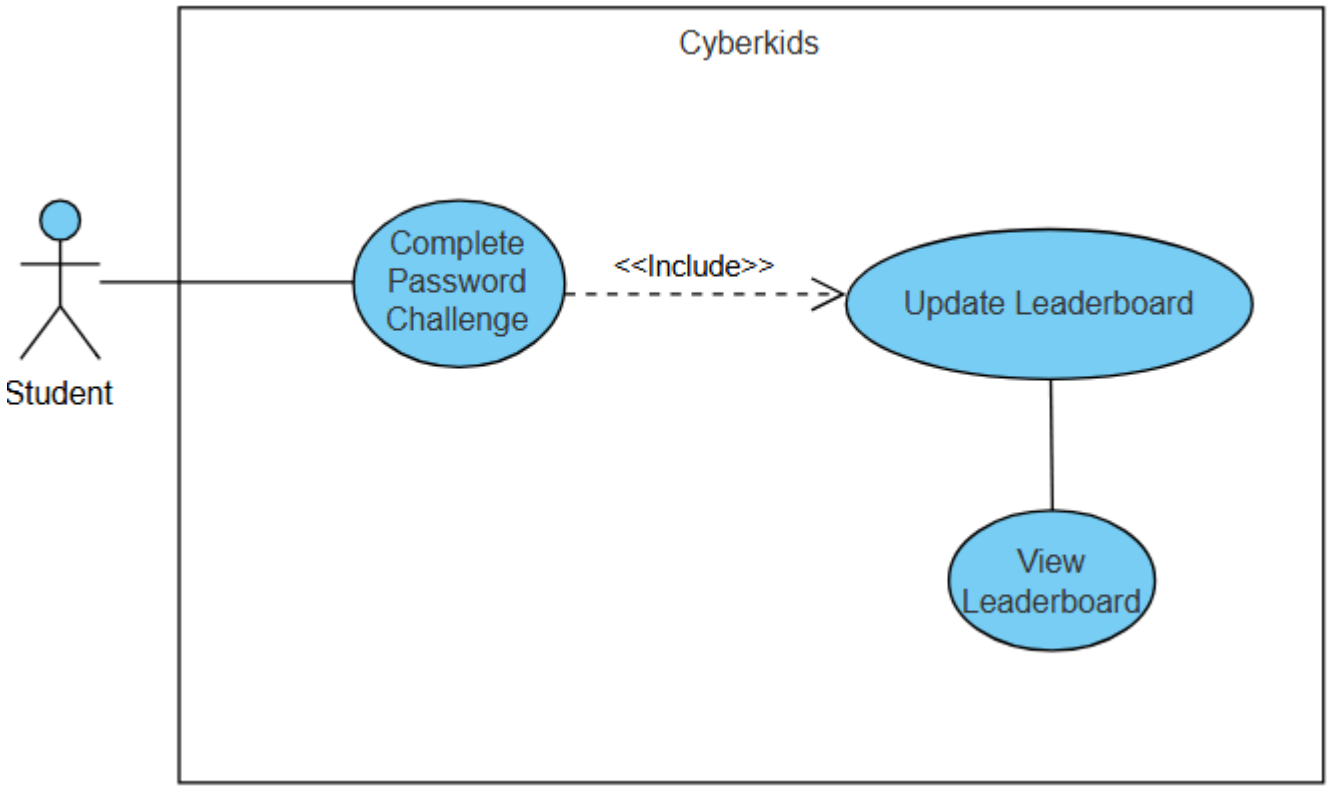


***Wireframe** (Timer System)*



2.4 Leaderboard for the Password Security Game

***Use Case Diagram** (Leaderboard)*

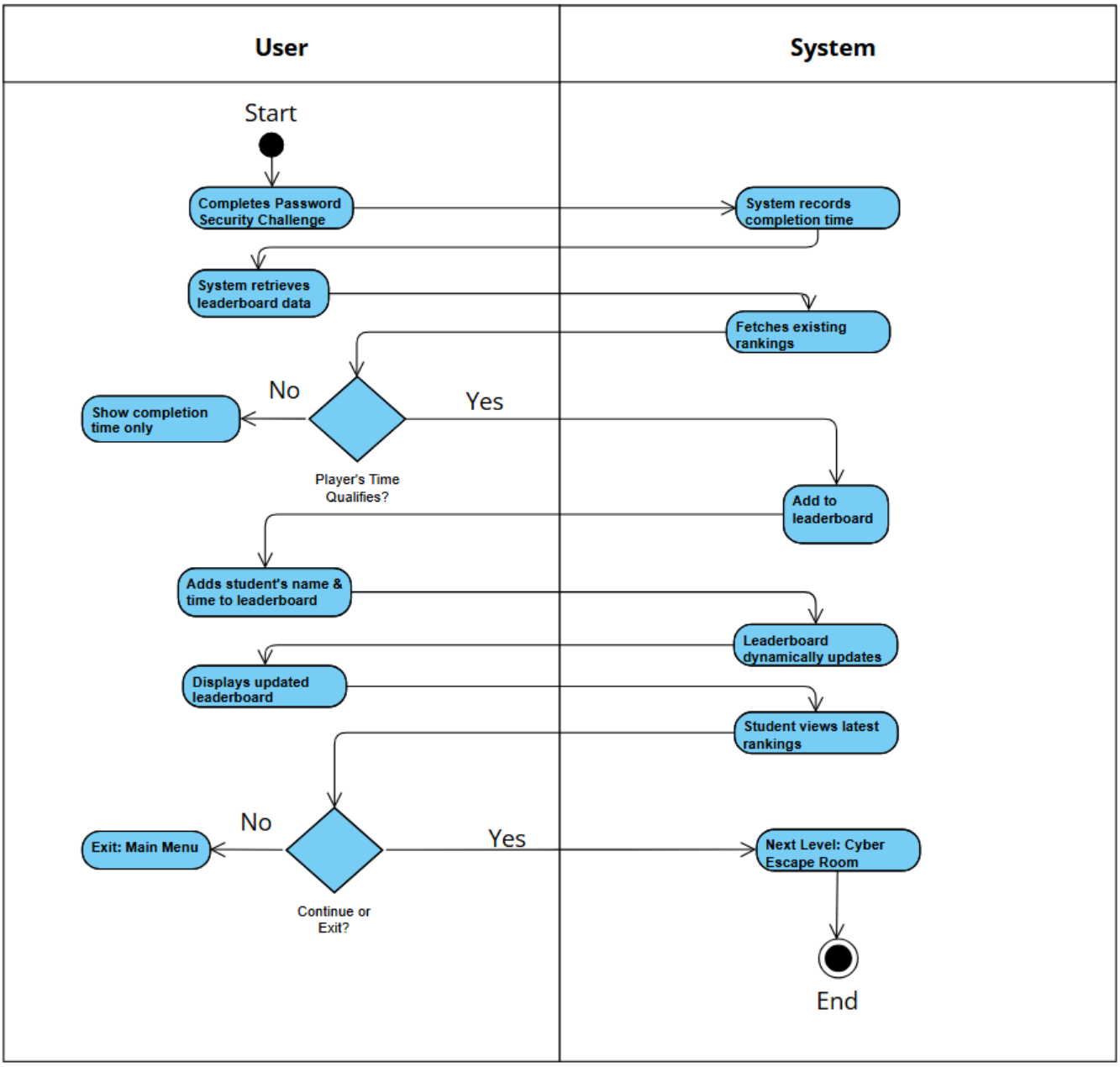


Use Case Description

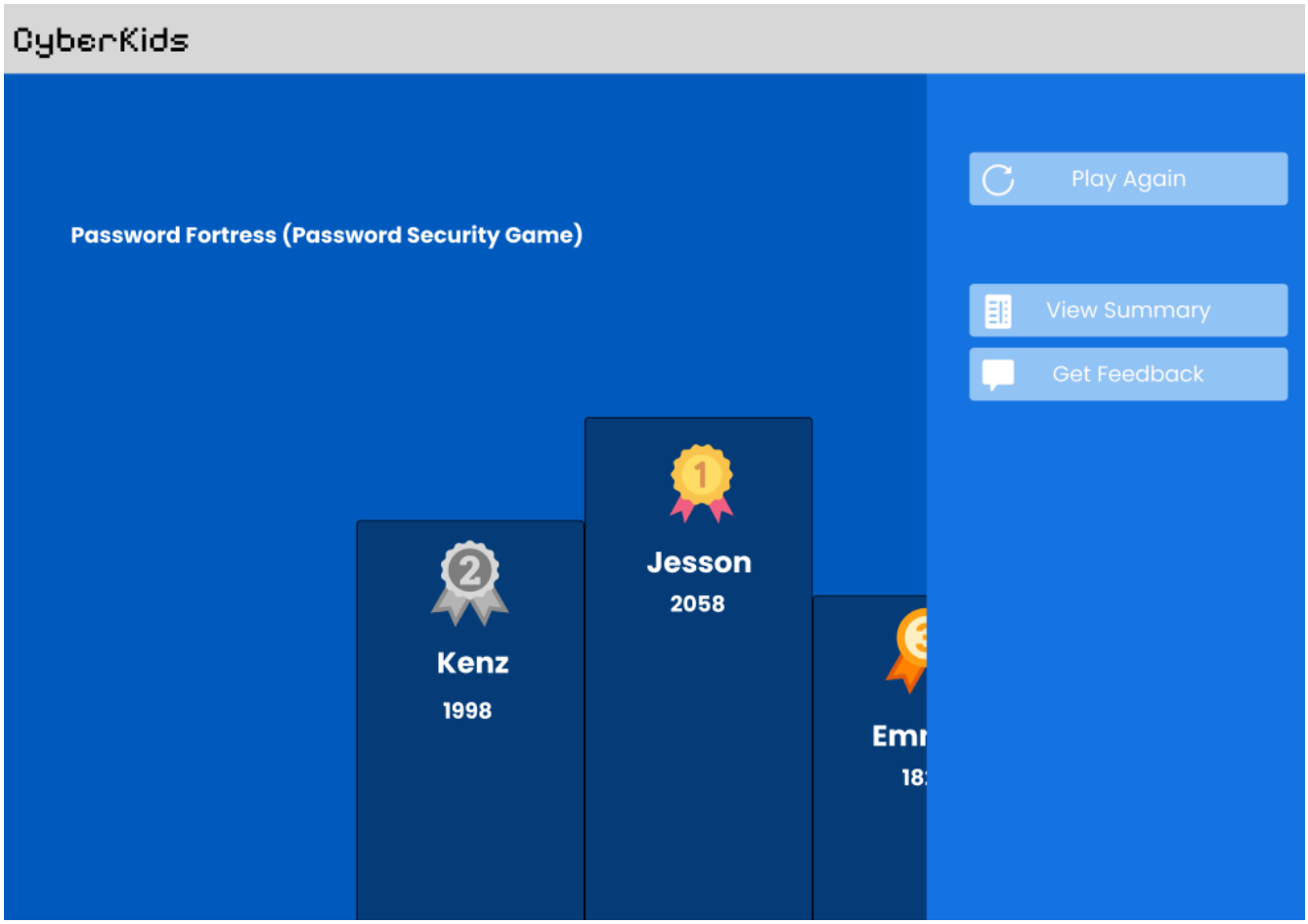
Use Case ID	UC-004
Use Case Name	Leaderboard for the Password Security Game
Actor	Student
Description	The Leaderboard System tracks and displays the top players based on their performance in the Gamified Password Security module. It ranks players based on their score, ensuring competition and motivation. The leaderboard updates dynamically and allows players to compare their scores with others.
Flow of Events	1. The player completes the Password Security Challenge, and their completion time is recorded. 2. The system retrieves existing leaderboard data and checks if the player qualifies for a ranking. 3. If the player's time is among the top scores, their name and time are added to the leaderboard. 4. The leaderboard dynamically updates, reflecting the latest rankings. 5. Players can view the leaderboard at any time to compare scores and track progress.
Precondition	The player has completed the Password Security Challenge successfully.

Postcondition	• The player's score is recorded and updated if eligible. • The leaderboard displays the updated rankings in real-time.

***Activity Diagram** (Leaderboard)*



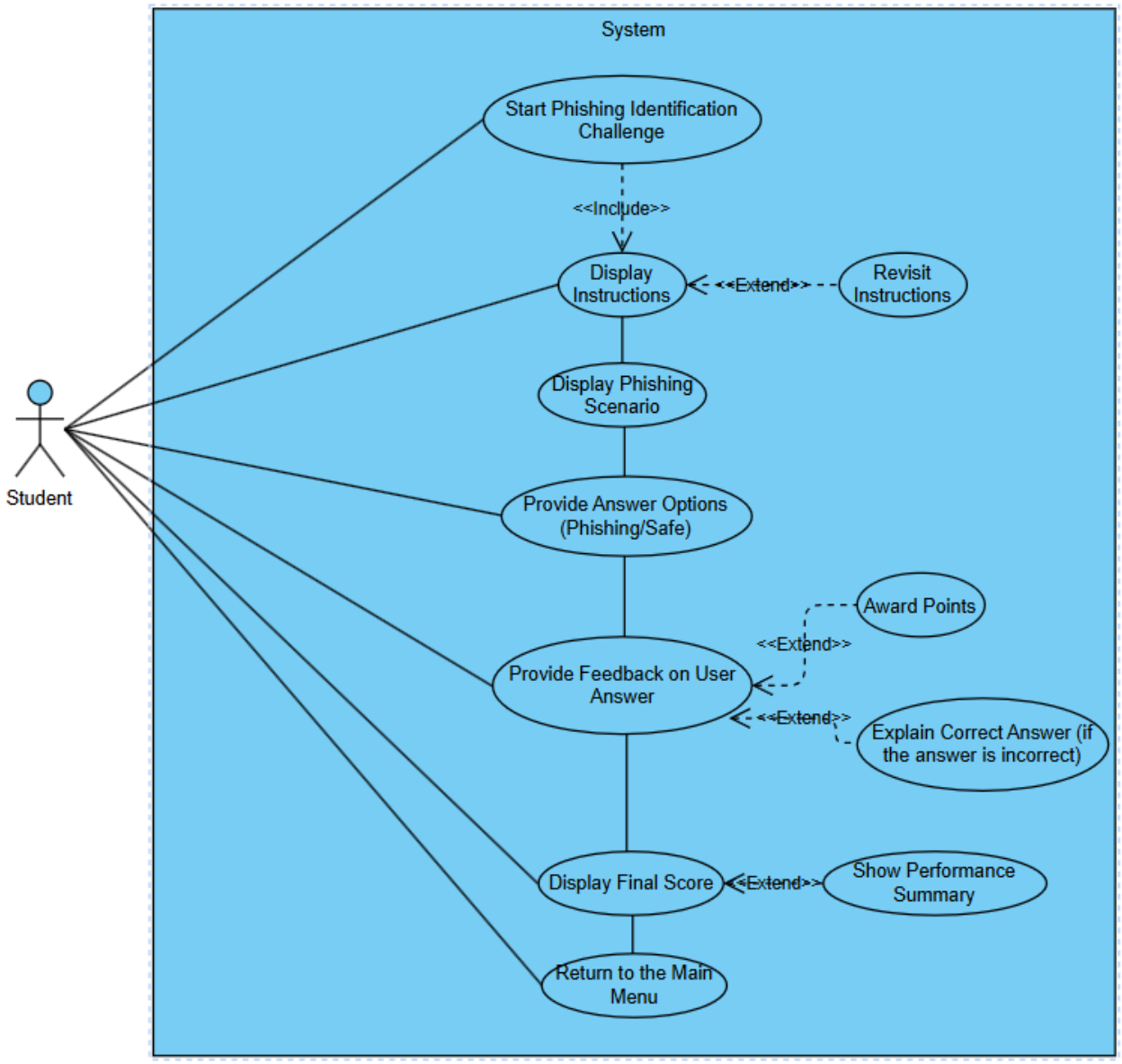
***Wireframe** (Leaderboard)*



Module 3: Gamified Phishing and Scam Awareness

3.1 Phishing Identification Challenge

***Use Case Diagram** (Phishing Identification Challenge)*

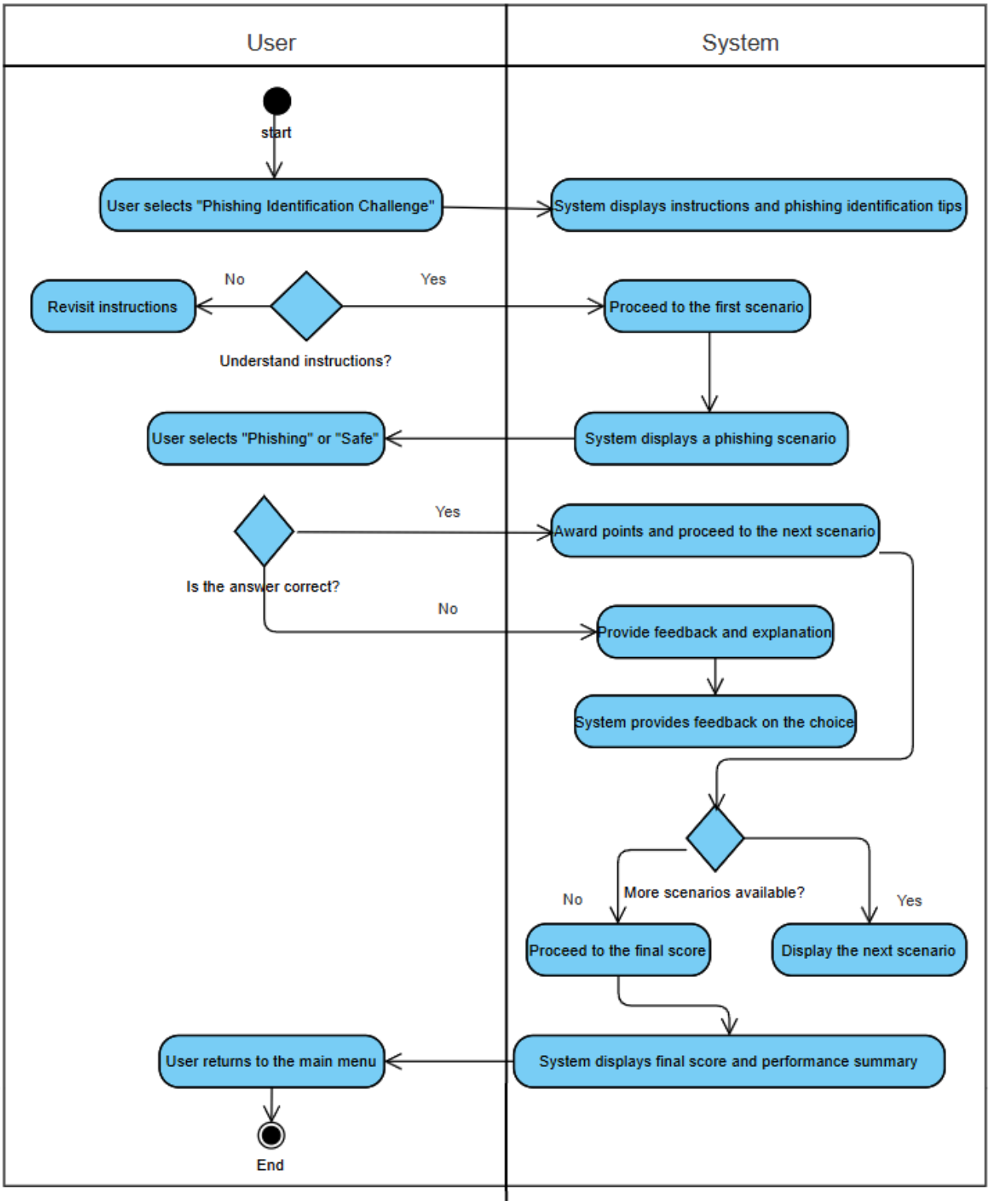


Use Case Description

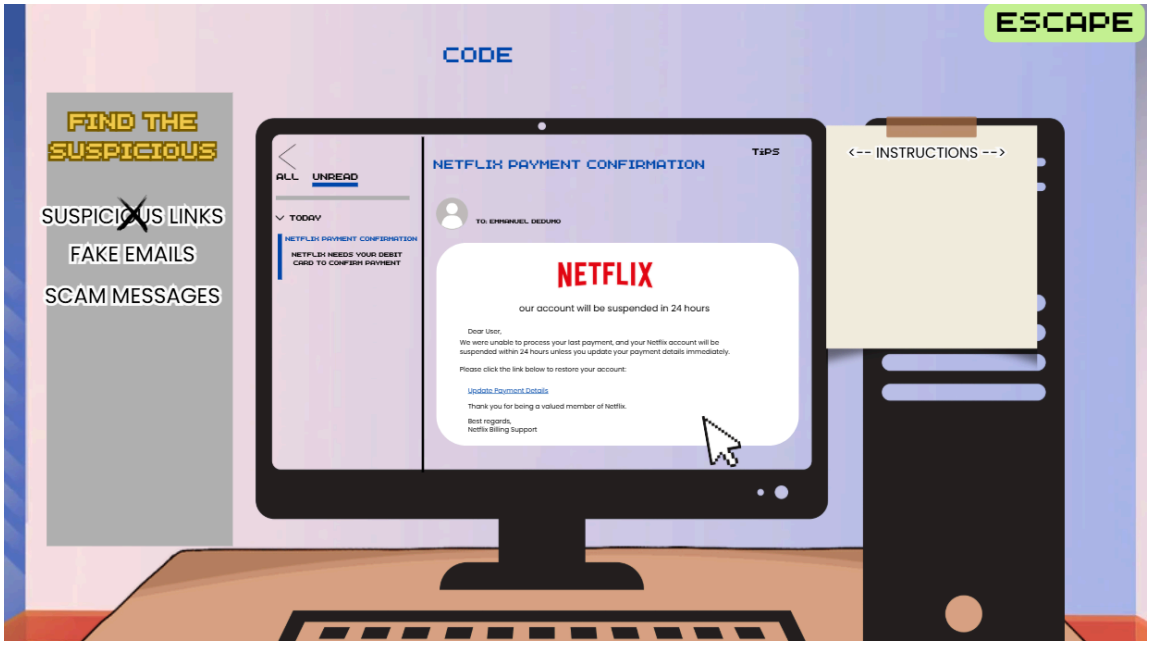
Use Case ID	UC-001
Use Case Name	Phishing Identification Challenge
Actor	Student
Description	The Phishing Identification Challenge is an interactive mystery-solving game where students navigate a simulated PC environment to identify phishing scams, fake links, and malicious messages. Players must analyze emails, websites, and messages to detect deceptive content and successfully unlock their escape from the simulation.
Flow of Events	1. The student starts the challenge and enters the simulated PC environment. 2. The system presents a series of emails, websites, and messages containing both legitimate and phishing elements. 3. The student inspects the content, looking for red flags, such as: ● Suspicious links or URLs. ● Urgent and threatening language. ● Unusual sender addresses. ● Requests for personal or financial information. 4. The student flags phishing attempts by clicking on them or marking them as fraudulent. 5. The system provides immediate feedback on whether the selection was correct or incorrect. 6. The student continues analyzing additional cases until the challenge is completed.

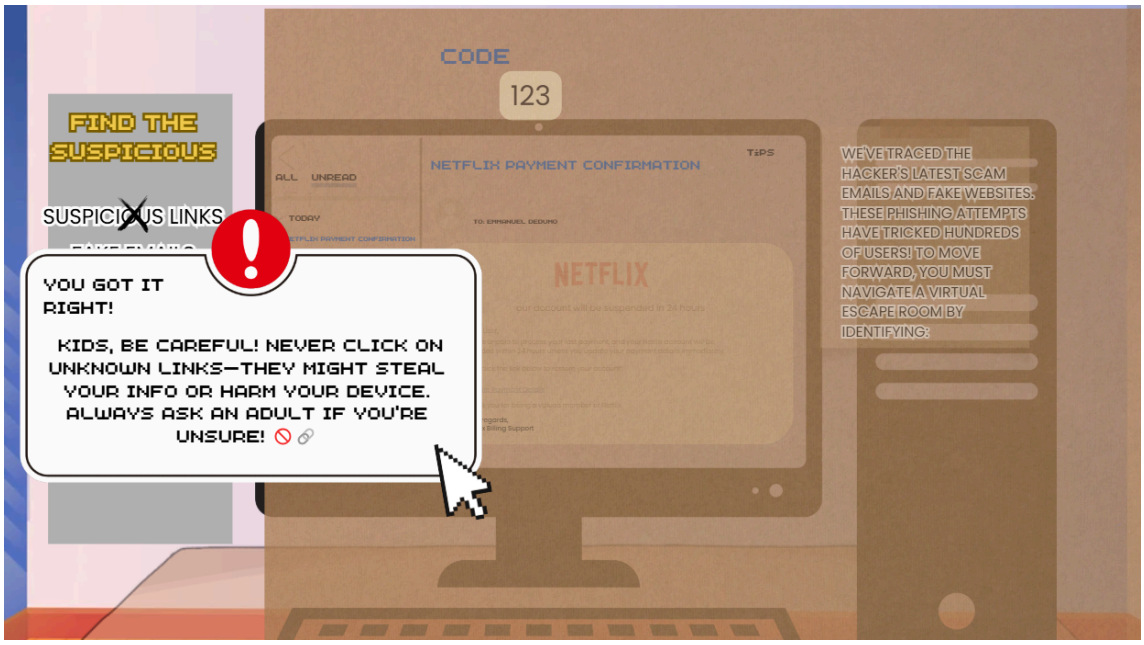
	7. If the student correctly identifies all phishing scams, they successfully "escape" the simulated scenario. 8. The system records the student's performance and updates the leaderboard accordingly.
Precondition	● The student must be logged into the game. ● The Phishing Identification Challenge level must be accessible.
Postcondition	● The student's score and completion status are saved. ● The leaderboard is updated with the student's performance.

***Activity Diagram** (Phishing Identification Challenge)*



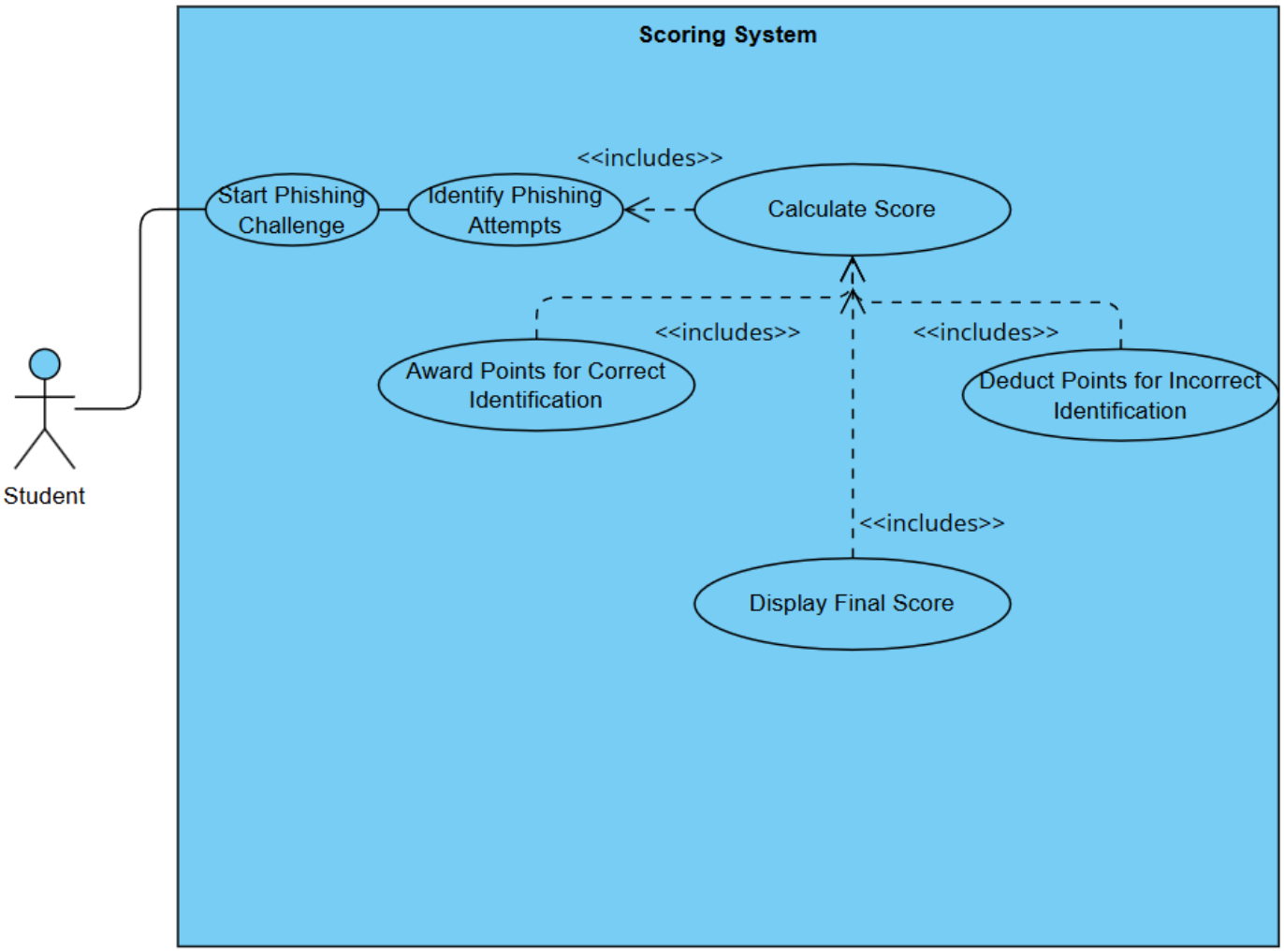
Wireframe *(Phishing Identification Challenge)*





3.2 Point-Based Scoring System

***Use Case Diagram** (Scoring System)*

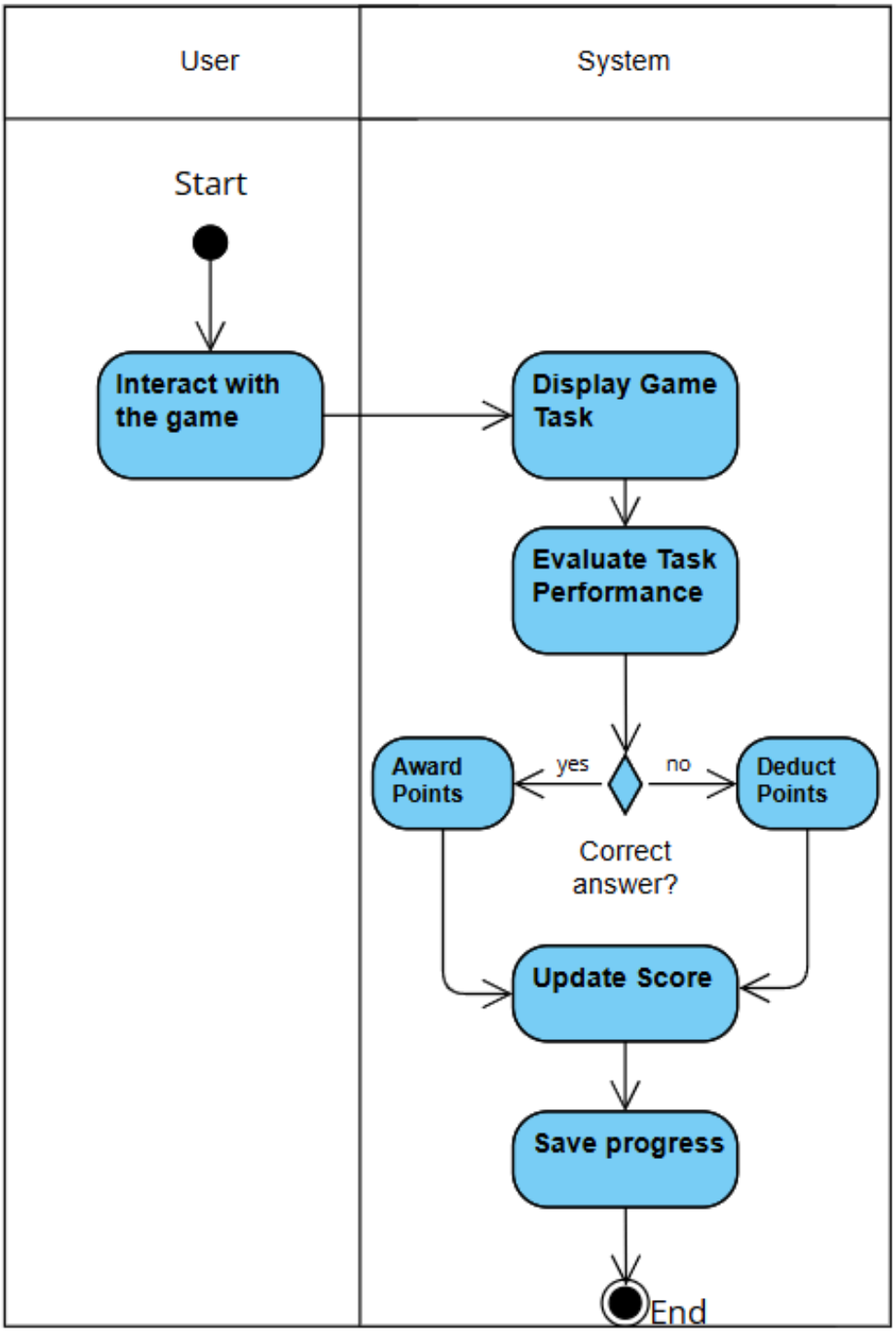


Use Case Description

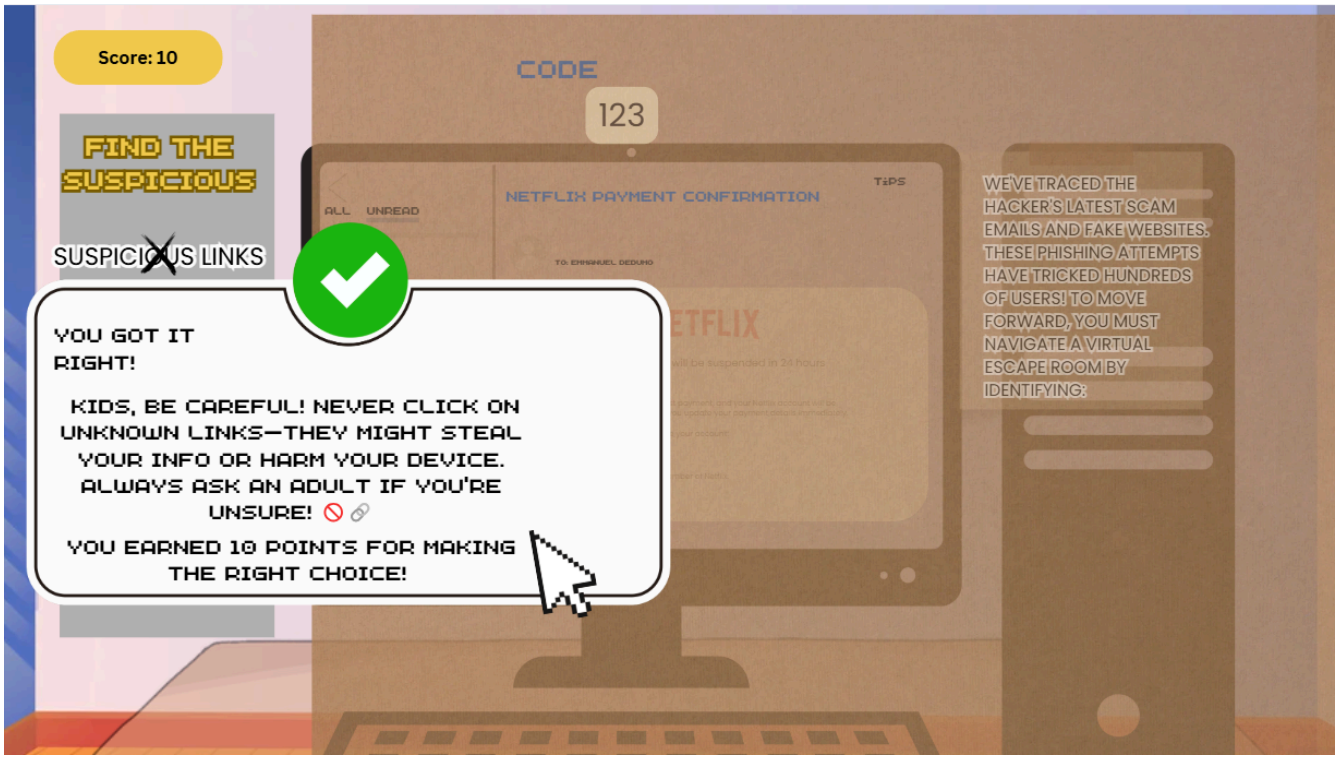
Use Case ID	UC-002
Use Case Name	Point-Based Scoring System
Actor	Student
Description	The Point-Based Scoring System tracks and calculates student performance in the Phishing Identification Challenge. Students earn points by correctly identifying phishing attempts in emails, websites, and messages
Flow of Events	1. Challenge Begins: ● The student enters the phishing simulation challenge. ● The system starts tracking actions and time. 2. Identifying Phishing Attempts: ● The student reviews emails, messages, and websites. ● They select items they believe are phishing scams. 3. Scoring Mechanism: ● Correct Identification: The student earns +10 points for each correctly flagged phishing attempt. 4. Challenge Completion: The system calculates the final score based on accuracy and speed. The score is recorded and displayed to the student.

	The system updates the Leaderboard, ranking students based on their performance.
Precondition	The student has started the Phishing Identification Challenge.
Postcondition	The student sees their final score and ranking on the leaderboard.

***Activity Diagram** (Scoring System)*

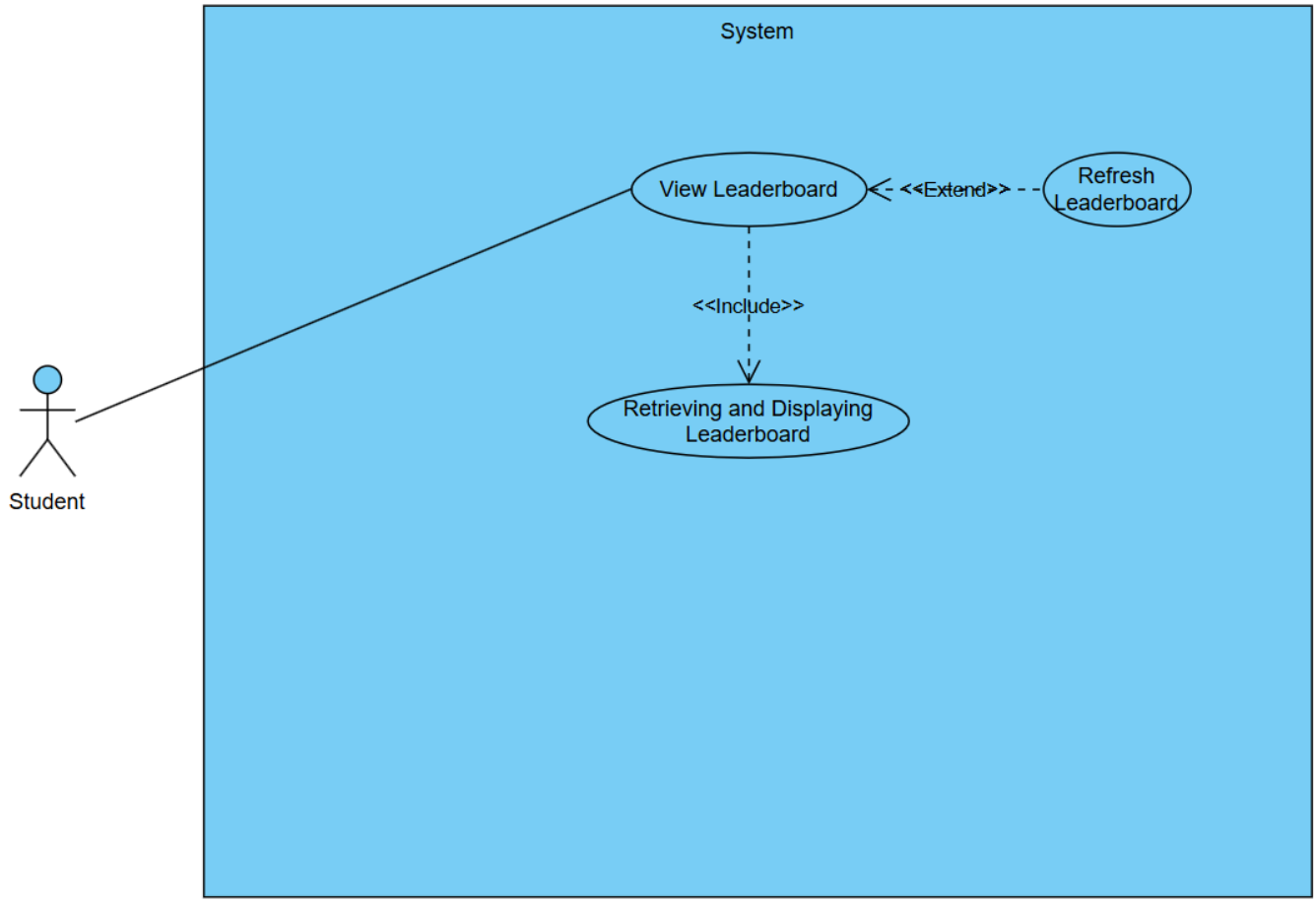


Wireframe *(Scoring System)*



3.3 Leaderboard for the Phishing Awareness Game

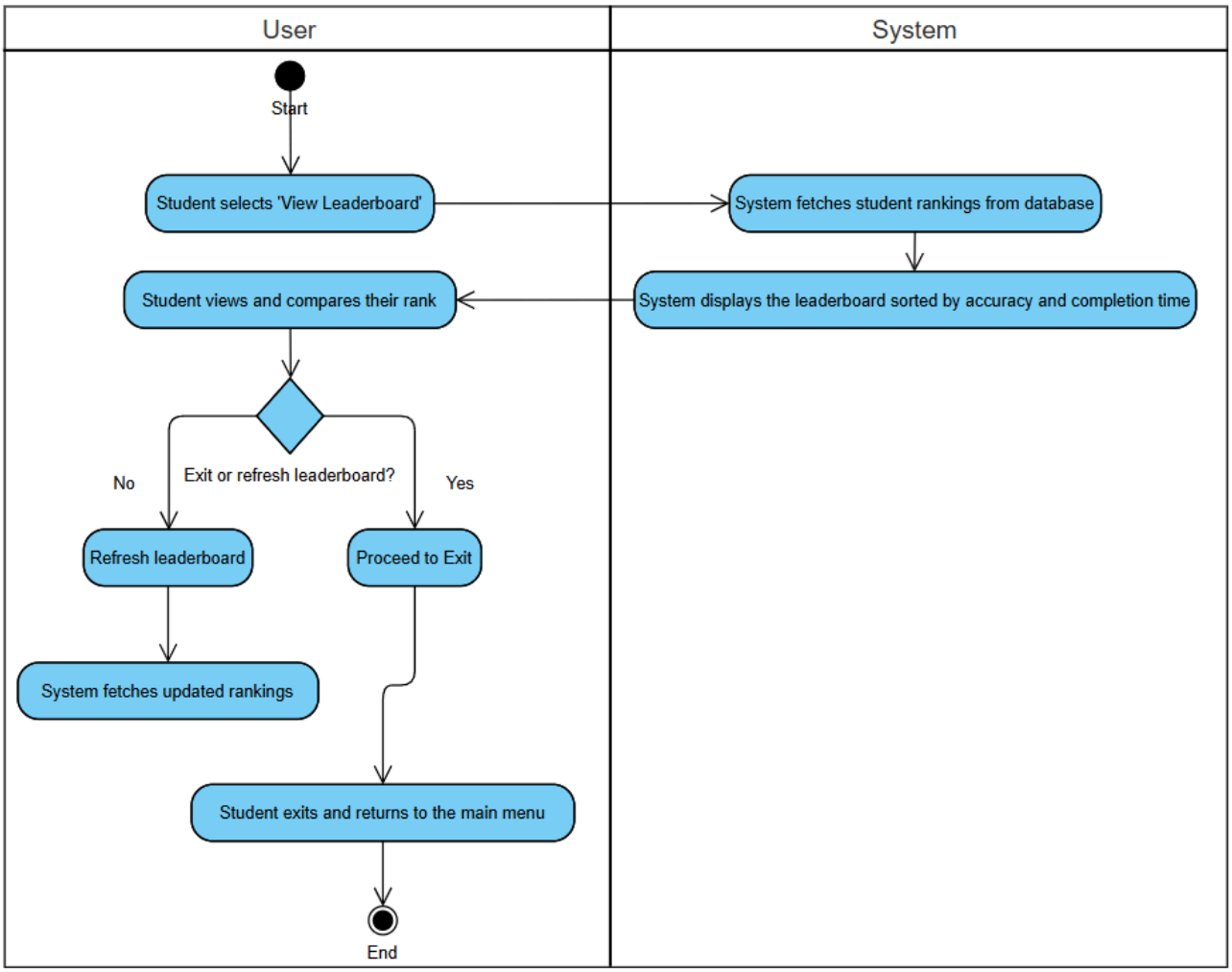
***Use Case Diagram** (Leaderboard)*


System
View Leaderboard
<<Extend>>
Refresh Leaderboard
<<Include>>
Retrieving and Displaying Leaderboard
Student

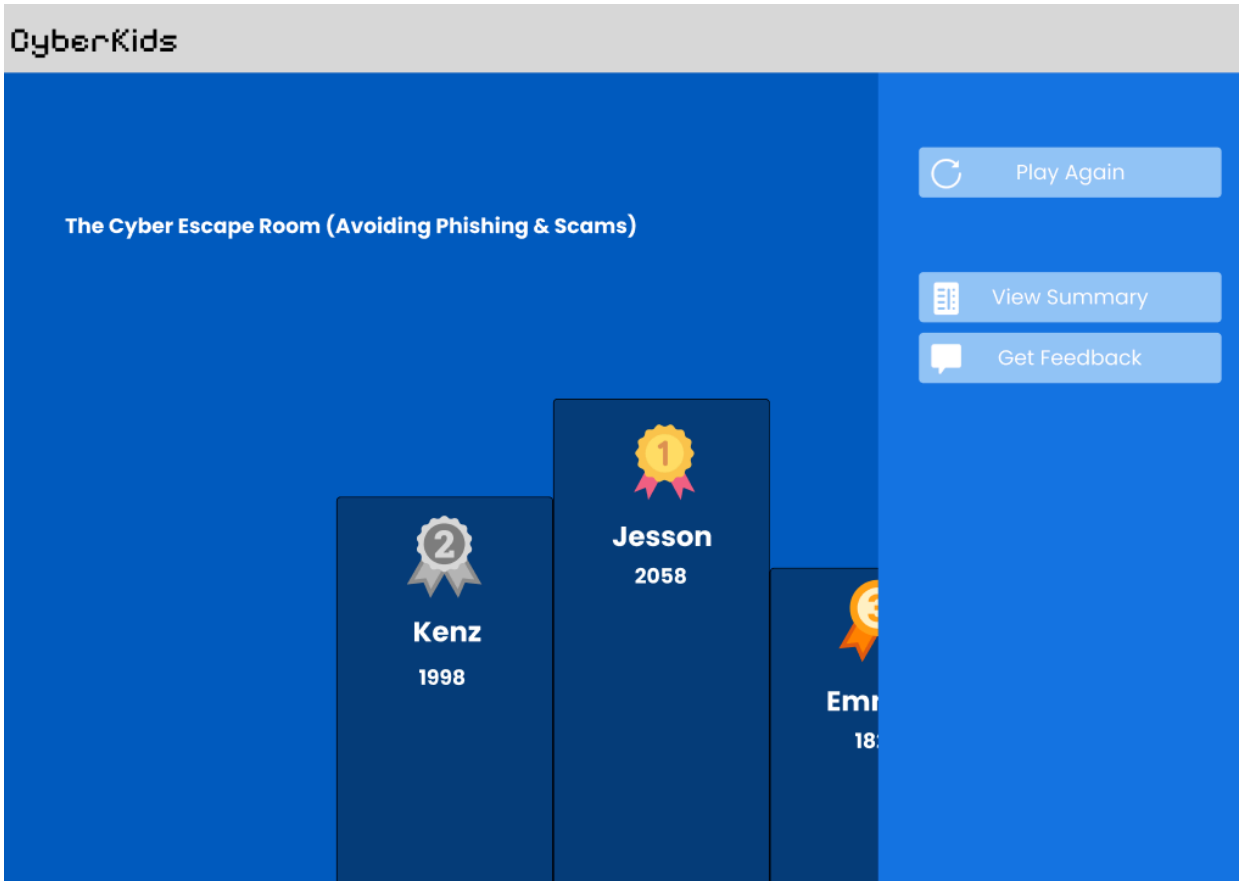
Use Case Description

Use Case ID	UC-003
Use Case Name	Leaderboard for the Phishing Awareness Game
Actor	Student
Description	The Leaderboard for the Phishing Awareness Game ranks students based on their performance in identifying phishing scams.
Flow of Events	1. The system records the student's score after completing the phishing challenge. 2. The system updates the leaderboard based on accuracy and completion time. 3. Students can view their rank and compare their performance.
Precondition	The student has completed at least one Phishing Identification Challenge.
Postcondition	The leaderboard reflects all students' progress in real-time.

Activity Diagram *(Leaderboard)*



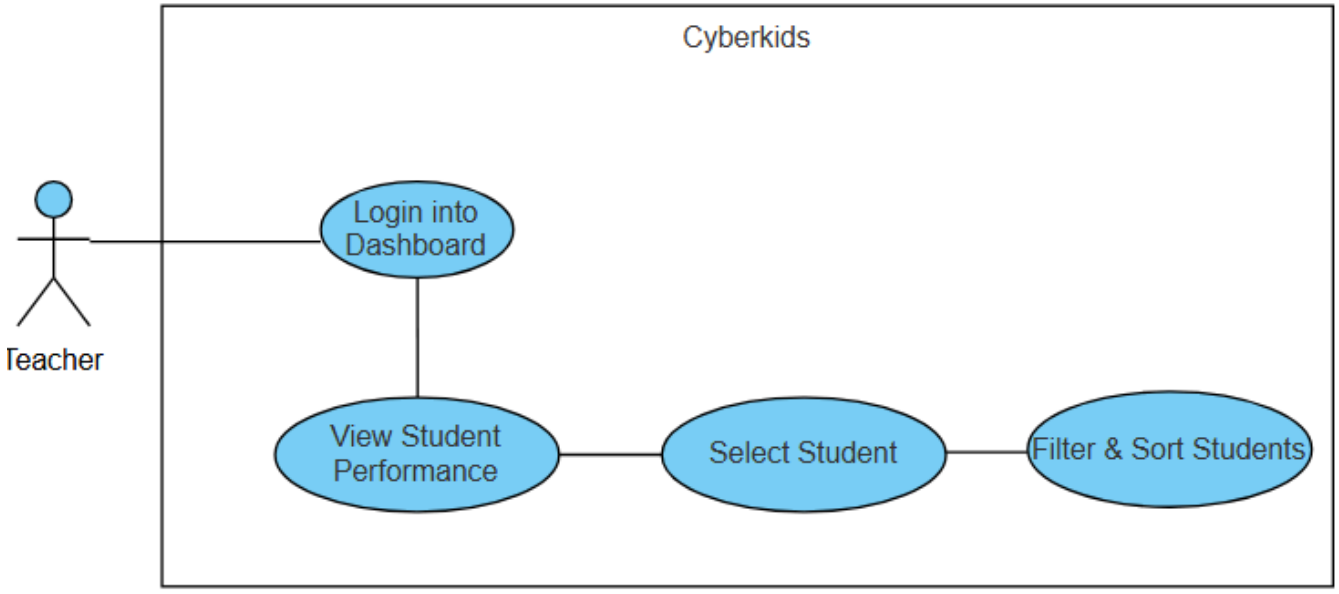
***Wireframe** (Leaderboard)*



Module 4: Teachers Dashboard

4.1 Student Performance Overview

Use Case Diagram (Leaderboard)

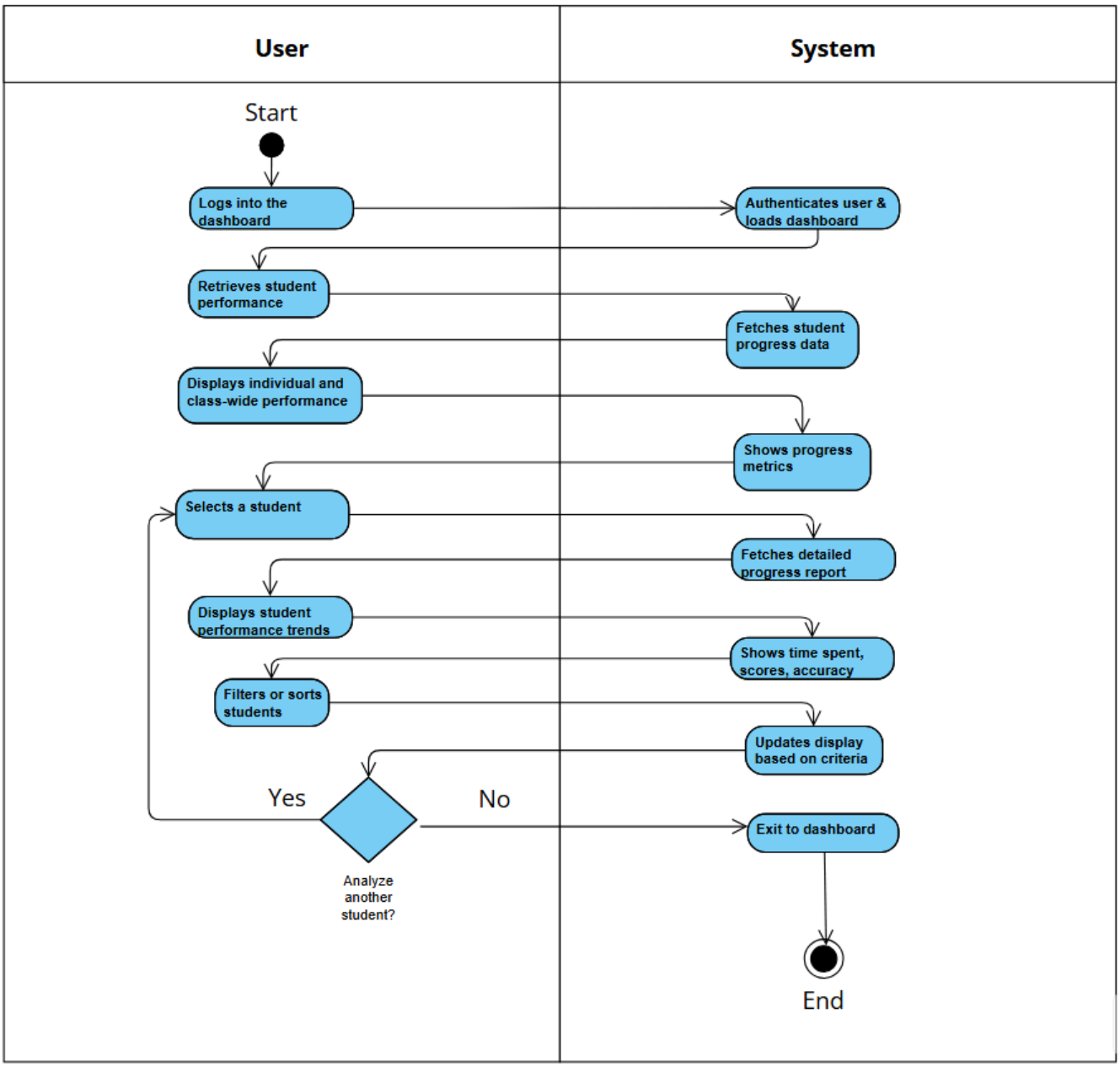


Use Case Description

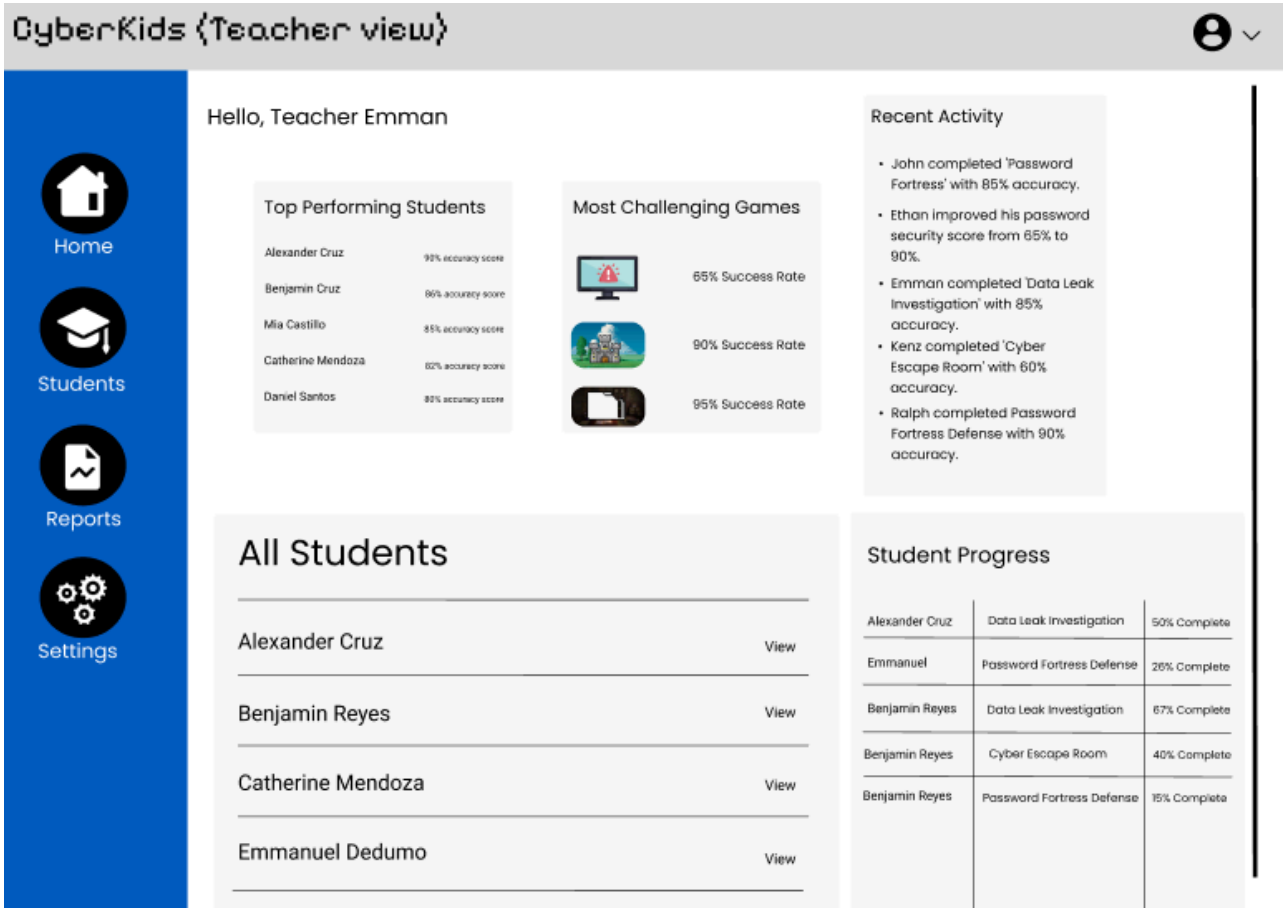
Use Case ID	UC-001
Use Case Name	Student Performance Overview
Actor	Teacher
Description	The Teacher Dashboard provides an overview of student performance in cybersecurity learning activities.
Flow of Events	1. The teacher logs into the dashboard. 2. The system retrieves and displays an overview of student performance, including: 3. Individual student progress (completed challenges, scores, and accuracy). 4. Class-wide progress metrics (average scores, completion rates, and engagement levels). 5. The teacher selects a student to view detailed progress reports, including time spent on challenges and performance trends. 6. The teacher can filter and sort students based on performance, challenge completion, or engagement levels.
Precondition	● The teacher has an active account and is logged into the system. ● Students have engaged in cybersecurity learning activities, and their progress has been recorded.
Postcondition	● The teacher gains insights into student performance. ● The teacher can use this information to adjust

	teaching strategies and provide targeted support.

Activity Diagram *(Leaderboard)*

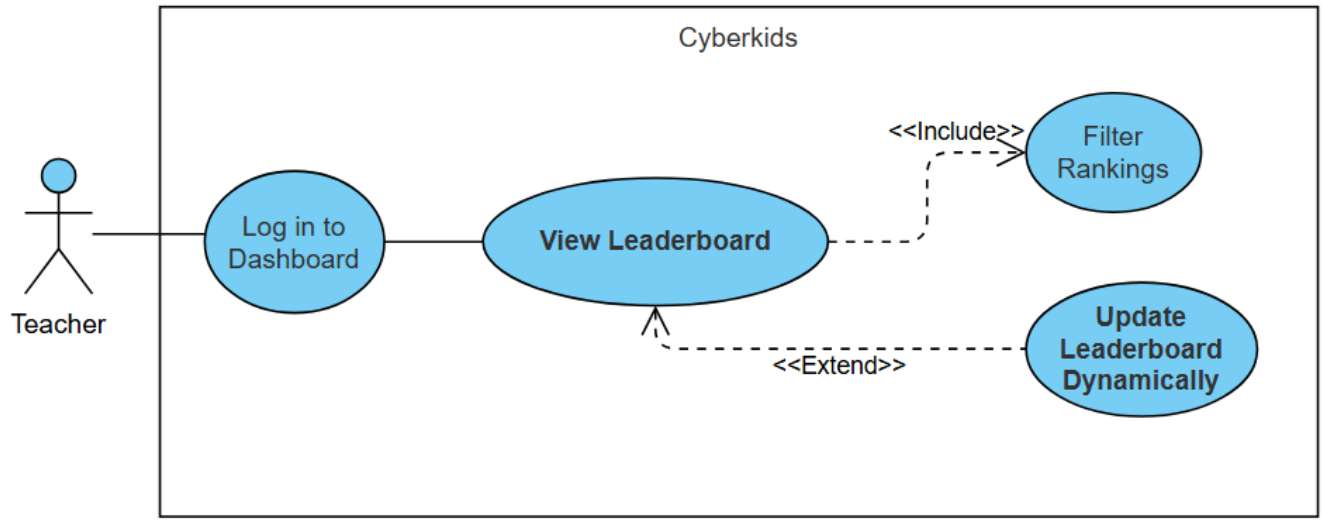


***Wireframe** (Leaderboard)*



4.2 Class Leaderboard (Performance Monitoring)

***Use Case Diagram** (Class Leaderboard)*

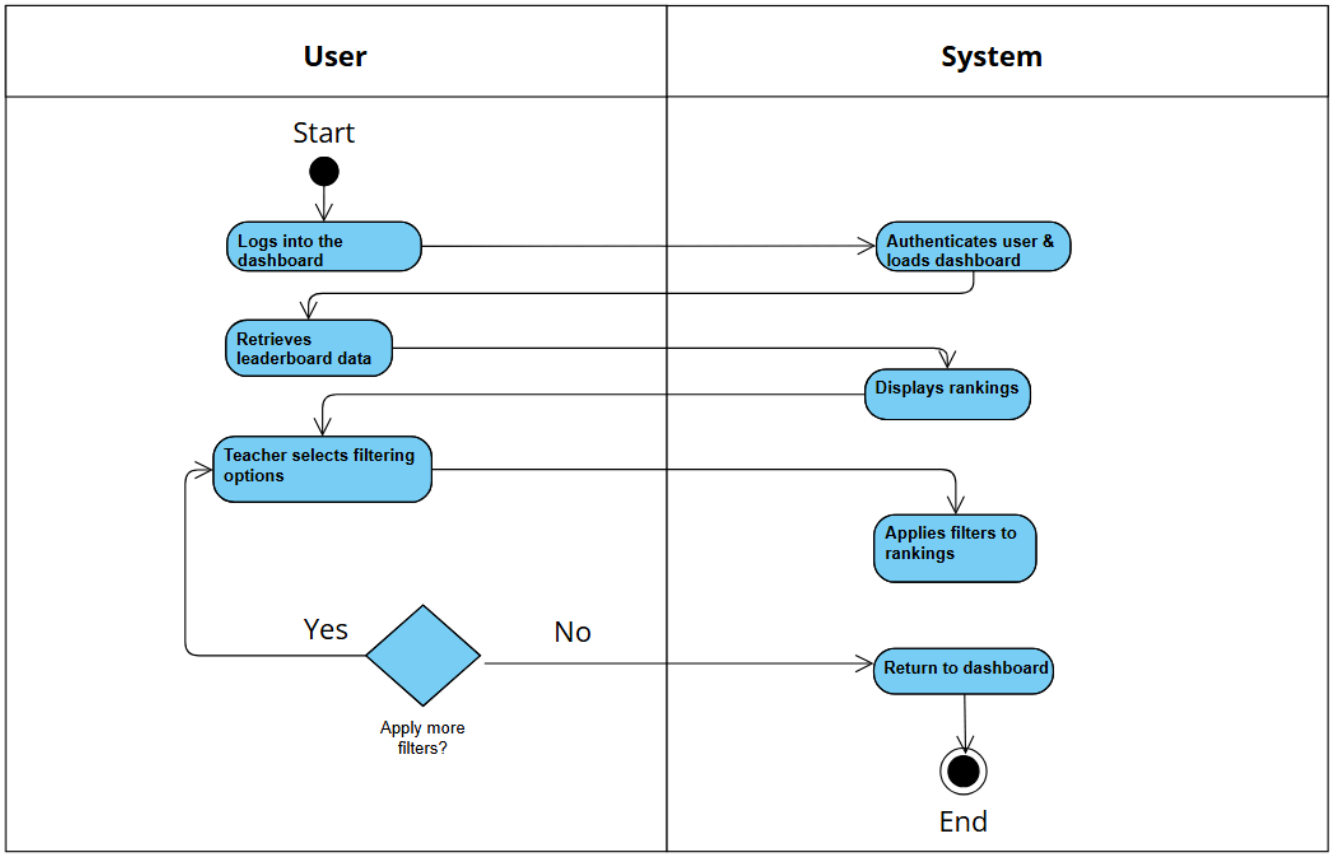


Use Case Description

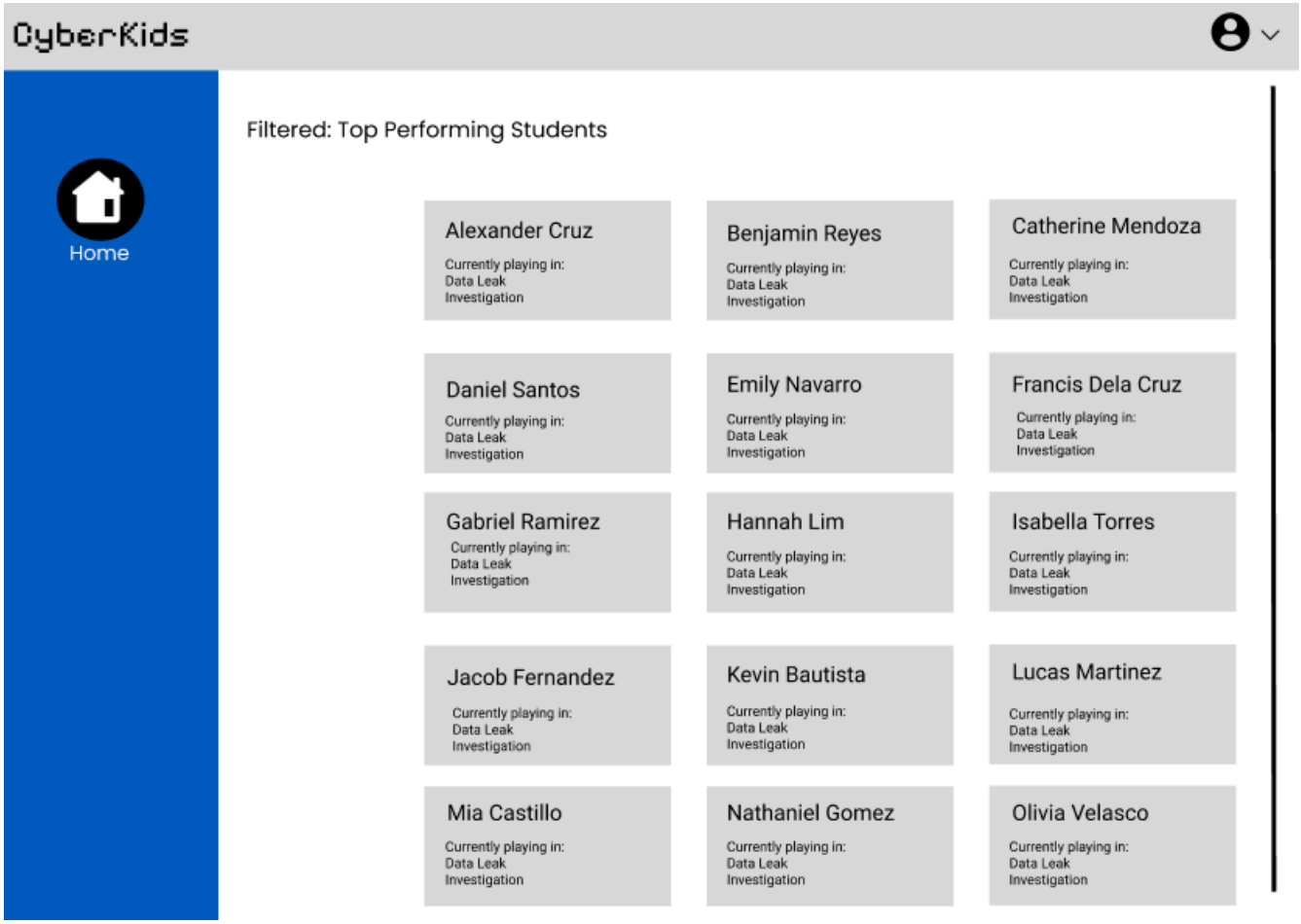
Use Case ID	UC-002
Use Case Name	Class Leaderboard
Actor	Teacher
Description	The Teacher Dashboard includes a class leaderboard that allows teachers to track student performance across different game levels (Online Privacy and Safety, Password Security, and Phishing Awareness). Teachers can filter and sort students based on performance, progress, and ranking.
Flow of Events	1. The teacher logs into the dashboard. 2. The system retrieves leaderboard data and displays rankings for: ● Overall student performance across all game levels. ● Specific game modules (e.g., Password Security, Phishing Awareness). 3. The teacher can filter the leaderboard based on: ● Individual student performance. ● Class rankings by score or completion rate. ● Progress in specific challenges. 4. The system updates rankings dynamically as students complete challenges.
Precondition	● The teacher has an active account and is logged into the system. ● Students have participated in cybersecurity learning activities, and scores are recorded.

Postcondition	● The teacher can analyze student rankings and progress trends. ● Teachers can use leaderboard data to motivate students and encourage competition.

Activity Diagram *(Class Leaderboard)*

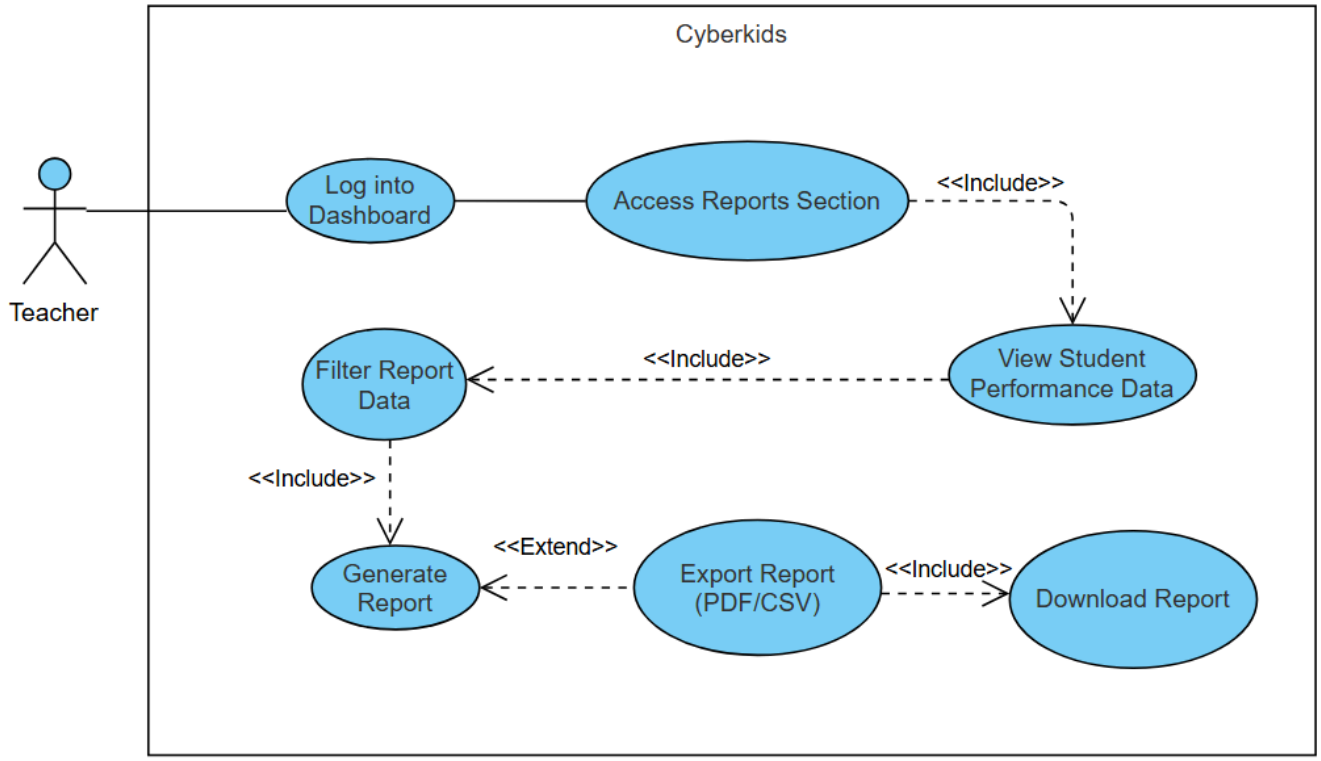


***Wireframe** (Class Leaderboard)*



4.3 Export Student Reports

Use Case Diagram (Student Report)

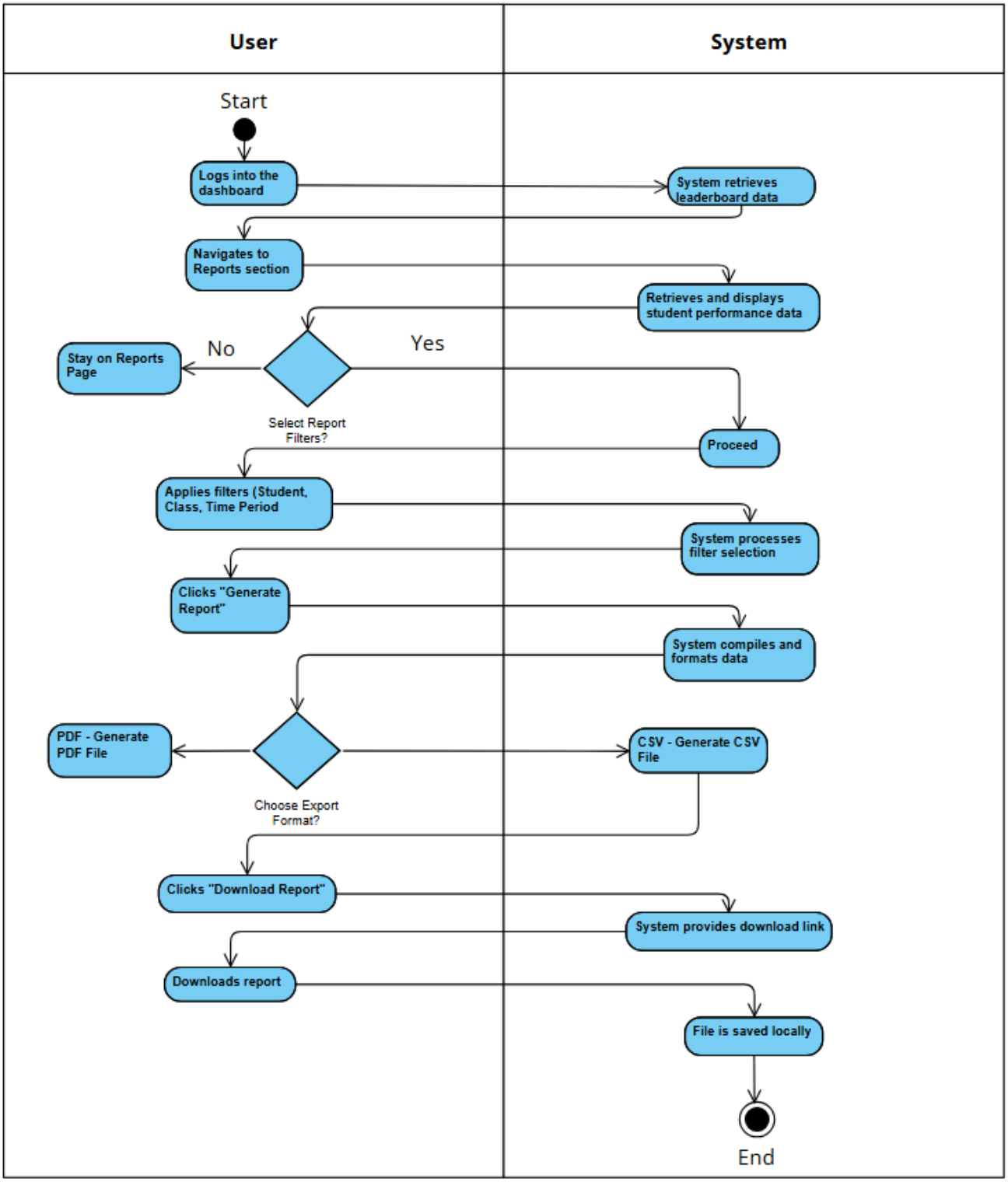


Use Case Description

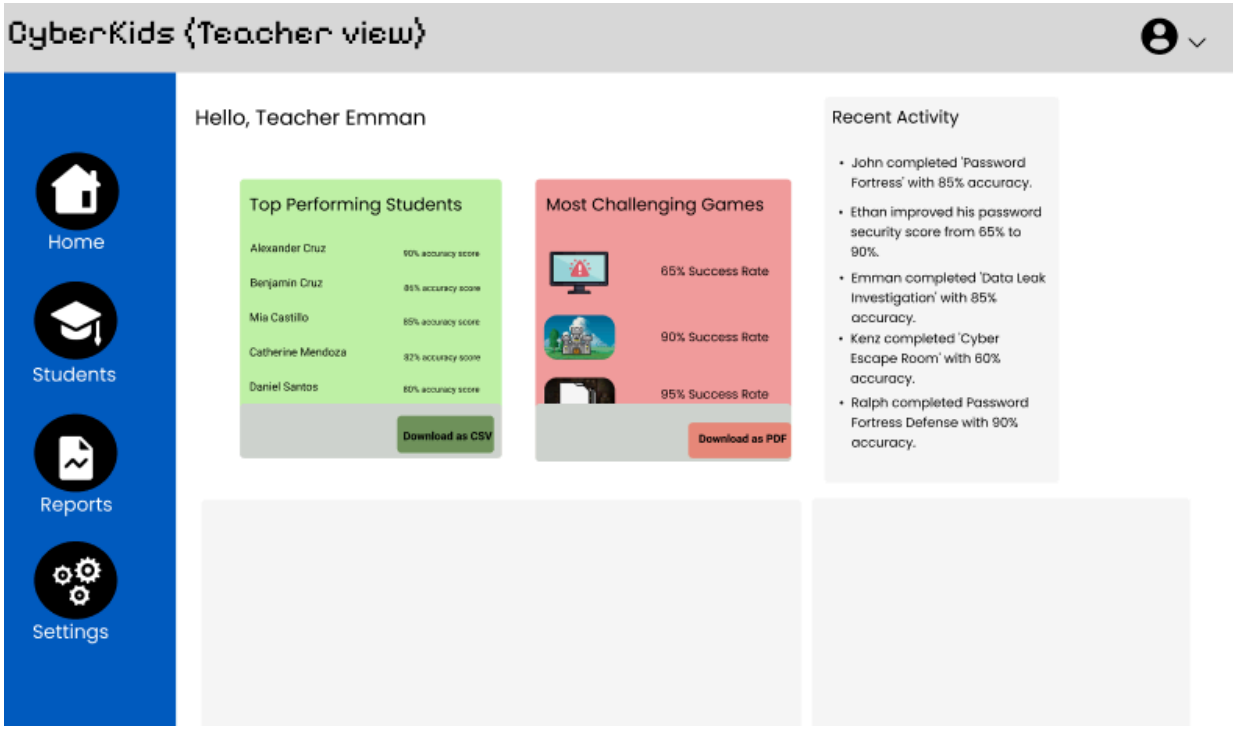
Use Case ID	UC-003
Use Case Name	Export Student Reports
Actor	Teacher
Description	The Teacher Dashboard allows teachers to generate and export student performance reports based on game data. Reports can be created for individual students or entire classes, tracking progress, scores, and completion rates for different cybersecurity challenges. Teachers can export reports in PDF or CSV formats for documentation and further analysis.
Flow of Events	1. The teacher logs into the dashboard. 2. The teacher navigates to the Reports section. 3. The system retrieves and displays student performance data, including: ● Scores and rankings for each cybersecurity challenge. ● Completion rates for individual students and the class. 4. The teacher selects report filters, such as: ● Individual student reports. ● Class-wide reports. ● Specific time periods or game levels. 5. The teacher generates the report, and the system compiles the selected data. 6. The teacher chooses an export format (PDF or CSV) and downloads the report.

Precondition	● The teacher has an active account and is logged into the system. ● Students have engaged in cybersecurity learning activities, and data has been recorded.
Postcondition	The teacher receives a detailed student performance report.

***Activity Diagram** (Student Report)*



***Wireframe** (Student Report)*



3.4 Non-functional requirements

Performance

1. **Game Responsiveness**
    - The system must load the main menu within 3 seconds on standard hardware.
    - Each mission should start within 5 seconds after selection.
2. **Leaderboard Updates**
    - The leaderboard should update in real-time (within 1-2 seconds) when a student completes a mission**.**

Security

1. **User Authentication**
    - All users (students, teachers, and admin) must log in using unique credentials.
    - Teachers and admins should have role-based access control to prevent unauthorized modifications.
2. **Data Protection**
    - No real personal data will be stored—only fictional in-game profiles will be used.
3. **Access Control**
    - Students can only access their own game progress and leaderboard rankings.
    - Teachers can only access data related to their assigned students.
    - The admin has full system control but cannot alter student scores manually.

Reliability

1. **System Availability**
    - The game should be available **99% of the time**, except for scheduled maintenance.
    - If the server goes down, **offline gameplay should still function**, but leaderboard

updates will be delayed.

2. **Error Handling & Recovery**
    - If an error occurs during gameplay, the system should **recover within 5 seconds** without losing progress.
    - If the leaderboard fails to load, users should receive a **retry option** instead of a crash.