CyberBlock Game
CyberBlock Team
Nelson Lau
Zach Kelly
Ridhisha Dangol
Xin (Jason) Li

# Software Design Document

**Version: 1** **Date: 10/26/2021**

# Table of Contents

# 1. Introduction

## 1.1 Purpose

The CyberBlock Game Design Document keeps track of and describes the capabilities of the CyberBlock Game, its appearance, and the functions that are present that allow it to ultimately perform. This information will be used to allow software developers to know what is expected to be built as well as understanding how to build the program. The intended audience of the CyberBlock Game Design Document are the project clients from Medtronic.

# 2. General Overview

## 2.1 Design Guidelines/Approach

When designing this program, we followed several important principles to ensure a quality product that meets clients' requirements.

First, throughout the design process we worked to maintain a focus on keeping the application as similar to the paper-based version as possible. We did occasionally consider some extraneous features, but we regularly asked ourselves (as individuals and as a team) “Is this straying too far from the original design?” We used this principle to eliminate unnecessary features while giving the game a modern approach through computer adaptations.

Next, we also needed to consider functional requirements set by the clients and the future implementations of the program. To make sure that we had all requirements prioritized, the team examined the paper-based version thoroughly to ensure that the implementation of a computerized version and the requirements matched. We also considered how the game would be able to change with the addition of new attacks and defenses.

In addition to these two guiding principles, we tried to keep in mind certain design priorities according to information we gathered from the clients. As such, we tried to prioritize a functional program for the first half of the project. Designing and making the program aesthetically pleasing to the users will be implemented in the second half of the project.

# 3. Functional Description

The application is a single page web application. On the landing page, the user can either create a lobby or join a lobby. User who creates a lobby becomes the host of the lobby, and then they will go through a configuration page where they can set up constraints for the game, such as number of rounds, number of teams, time limit, and difficulty. A code is generated once a lobby is created, which can be used by students to join a lobby.

Before the host decides to start the game, the host can move students to different teams. Before the game starts, there is a trivia round where teams answer trivia questions; if they answer correctly, they get extra money during the buying phase. Then teams will be presented with a list of defense cards that can be purchased. Students can use chatroom to discuss their strategy with their teammates, and once they come to an agreement, leaders from each team will purchase defense cards that can be played later.

At the beginning of each round, an attack category is randomly decided by simulating a dice roll, where more common attacks will be given higher probability. The host can then choose a specific attack from the category, but the attack is hidden for now. The students will see what category the attack is from, and team leaders will play two defense cards or a wild card. Once every team has selected their defenses, the specific attack is uncovered, and teams are awarded some points if their defenses are effective against the attack. This process is done by sending a query to the database, which stores information on what defenses are effective against a certain attack. If a wild card is played, the team will need to answer a trivia question; if they answer correctly they can some points as well. Also, the most effective defenses against the chosen attack will be shown for educational purposes, so that the students can reflect on their decision and learn. Team leader is rotated every few rounds so every student gets to have a chance to make decisions.

A scoreboard is displayed as part of the user interface where teams can see how well they are doing.

# 4. Proposed Solution

## 4.1 User Story

A student navigates to the webpage. Since the student is going to join an existing lobby, the student enters the lobby code into the lobby ID box and presses the button named “join a lobby”.

Once the student is in the lobby, the student will randomly be assigned to a team. Once in a team the student will be able to either communicate with their team via an external program or utilizing the built-in game text chat.

During the beginning of the game, the student gets into the trivia phase. By answering questions correctly the student’s team can get additional money to use during the gameplay.

After the trivia section, the team will enter the buy phase where they can use their money to purchase defense cards to use during the game.

Once the buy phase ends, the instructor will roll a random attack category. Once the attack category is selected, the student’s team plays the defense card that they think will counteract the attack the best. Once all of the teams have selected their card, the specific attack from the category is revealed. Teams are awarded points based on their

defense's effectiveness. The student's team gets points and advances to the next round. After each attack round is completed, a leaderboard is shown that can

These rounds repeat for a specified number of times. Once the rounds are over, a final leaderboard is displayed, and the student sees their team's score compared to the other teams.

### 4.2 Host Story

The instructor acts as the host. The instructor navigates to the same website as the student. Since the instructor is the host, the instructor clicks on the "Create a lobby" button. The instructor then specifies different game attributes such as round count and team count and then enters them into the specified text boxes. When pressing "begin game", the instructor is shown a code to join the game lobby. The instructor shares this code with the students and all of the students join and get assigned into teams.

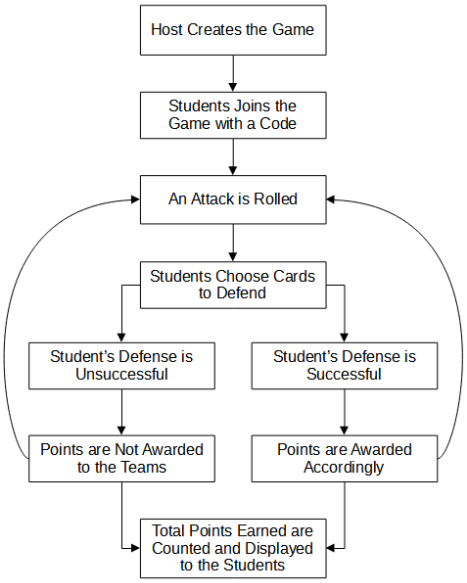
The instructor then waits while the students answer the trivia questions and complete the buy phase. The instructor then has to choose an attack from a randomly chosen attack category. Once the instructor chooses the attack, the students have to play their part. The instructor monitors when all of the teams have chosen their card, and then the game calculates the teams' points.

The instructor repeats this during the gameplay until the maximum rounds is reached. At the end the instructor can see the same final leaderboard as the students.

## 5. Flow Diagrams

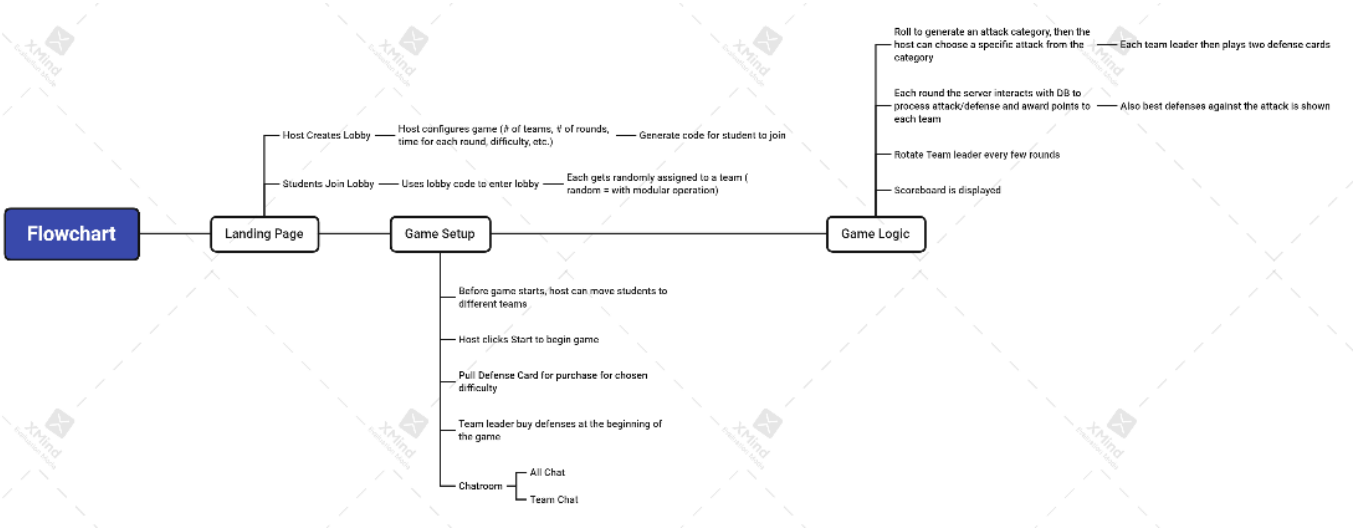
### 5.1 User Experience Flow

Below is a flowchart depicting the overall user experience during a session of a game.



## 5.2 Data Flow

Below is a flowchart depicting the overall dataflow of the project.



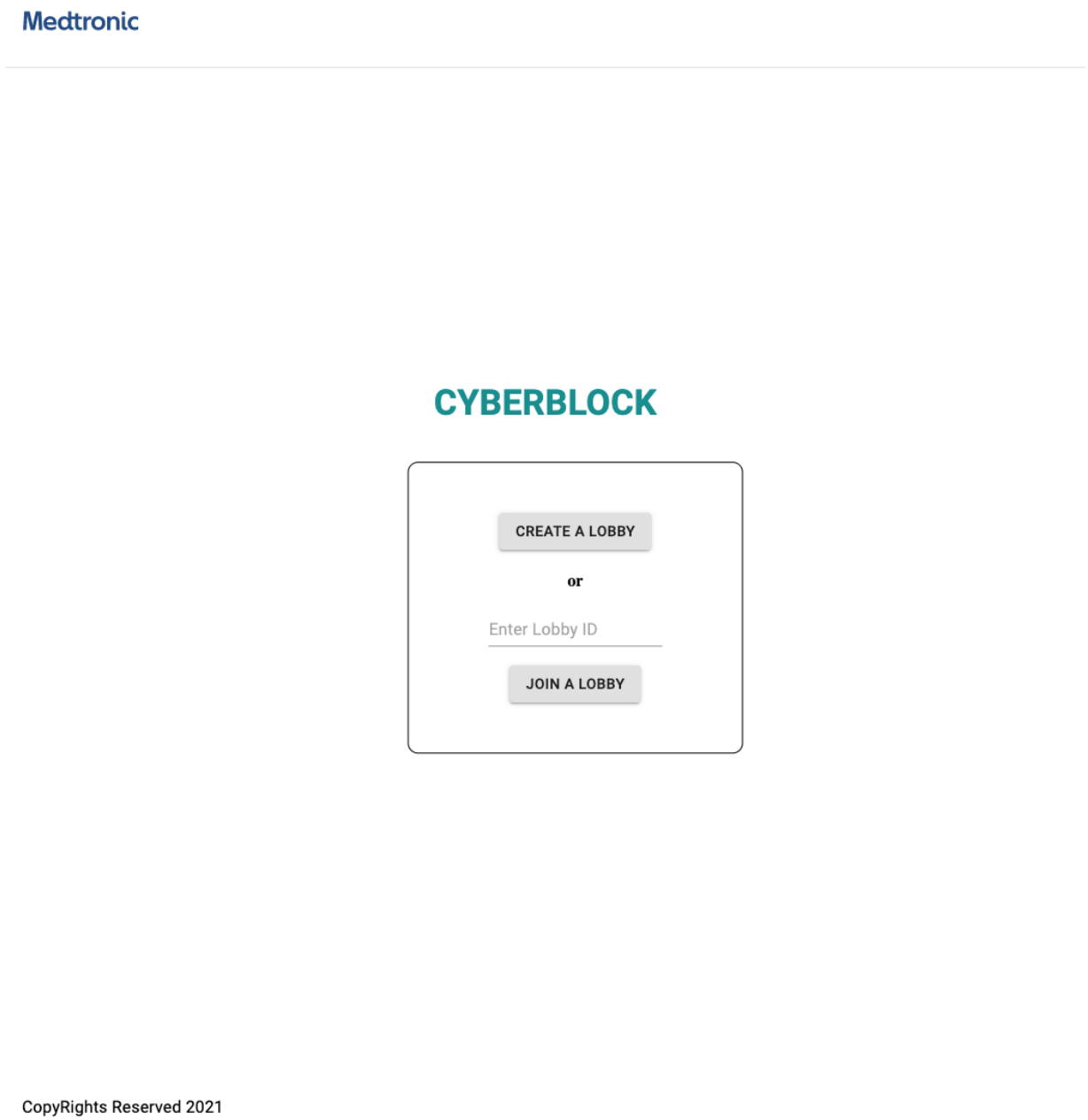

# 6. User interface and User Experience Design

## 6.1 Landing Page

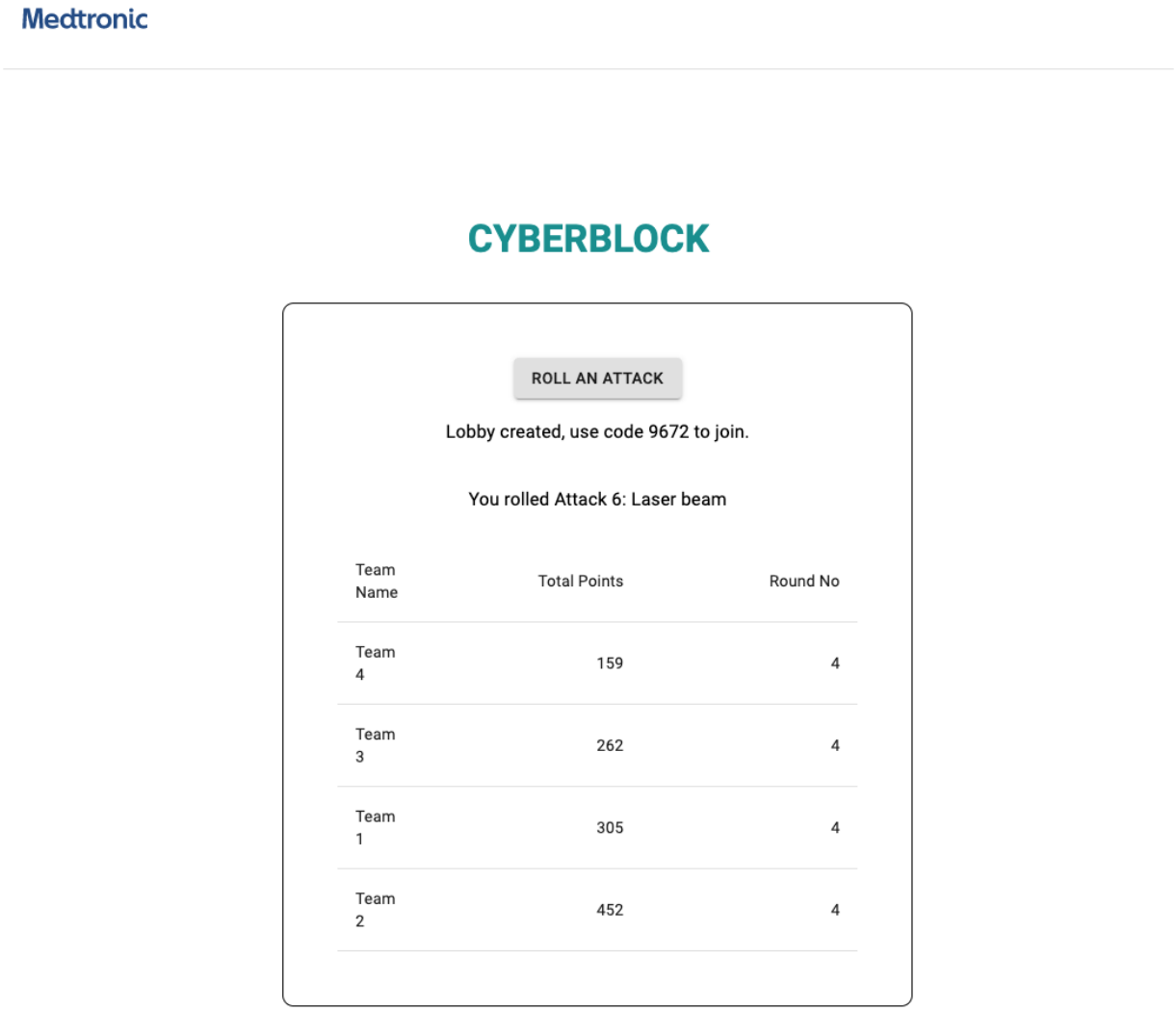
The landing page provides the user with two main functions: starting a game lobby or joining a pre-existing game lobby. To start a new game lobby, the user will select a button to generate a new lobby and will enter the game options such as the team count and number of rounds.

## 7. Interface Wireframe

This is the main page when the user visits the site. It gives the user an option to either host the game or join as a player.



The instructor's interface will always have the leader board up and a button at the beginning to start the game.



The player interface will have multiple options for the user to choose from after an attack is rolled.