



# RESEARCH REPORT

Cyberclubs. Enhancing User Experience.

Fontys Hogescholen

<b>Date</b>	<b>:</b>	<b>01.04.2025</b>
<b>Version</b>	<b>:</b>	<b>0.2</b>
<b>Author</b>	<b>:</b>	<b>Author</b>

**Version history**

<b>Version</b>	<b>Date</b>	<b>Author(s)</b>	<b>Changes</b>	<b>State</b>
0.1	09.09	Lisa		
0.2	19.09	Lisa		

**Distribution**

<b>Version</b>	<b>Date</b>	<b>Receivers</b>
1. 1	20/09/2024	

## Contents

1.	Introduction.....	3
1.1	Context.....	3
1.2	Problem statement .....	3
1.3	Objective .....	4
1.4	Strategy .....	4
1.5	Technology used .....	5
1.6	End products .....	5
1.7	Significance of the Research .....	5
2.	Methodology.....	6
2.1	Research Design.....	6
2.2	Data Collection Methods .....	6
2.3	Collecting feedback.....	6
3.	Results .....	7
3.1	Phases of the project .....	7
3.2	Technical analysis.....	7
3.2.1	Architecture Overview .....	7
3.2.2	Key Functional Modules .....	7
3.2.3	New stuff.....	7
4.	Conclusions .....	9
5.	References .....	10

# 1. Introduction

## 1.1 Context

Based on my own experience as a frequent visitor of cyber clubs, I have observed that the user interface and functionality of most cyber club booking systems are far from ideal. Many of these platforms suffer from outdated design, lack of visual clarity, and limited interactivity, making the booking process frustrating and unintuitive for users.

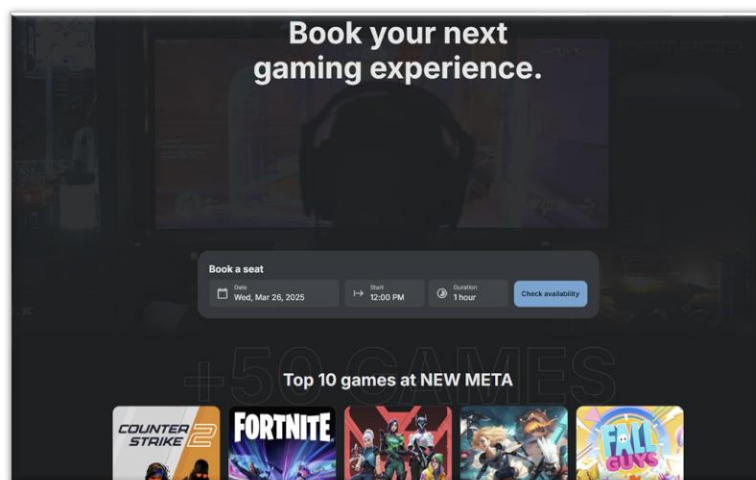
In response to this, I set out to create a solution that offers a smoother, more enjoyable experience for both booking and gaming. The goal was not just to build another system, but to rethink how digital tools can improve the day-to-day operations of a cyber club — from seat selection and session management to customer engagement and service integration.

## 1.2 Problem statement

In major European cities such as **Barcelona, Eindhoven, Berlin, Prague, Warsaw, and Kyiv**, cyber club booking systems exhibit varying degrees of sophistication and user-friendliness. While many of them offer basic features like **zone selection, time slot booking, and session duration**, they often lack modern, intuitive interfaces and advanced functionality — such as real-time **interactive seating maps** or seamless online registration.

This inconsistency highlights a gap in the market for a truly user-centric, all-in-one system that combines visual transparency, convenience, and efficiency, particularly in the context of growing consumer expectations for smooth digital experiences.

Here I provide **examples** from existing cyber club websites that demonstrate a key limitation: **users cannot view seat variations or availability in real time**. Instead, the system automatically assigns them to a randomly available seat within the selected zone and time frame. This lack of transparency significantly reduces user control and flexibility, especially for regular customers who may have preferred seating preferences (e.g., near friends, with better peripherals, or in quieter zones). This approach not only limits personalization but also adds friction to the booking process, as users are unable to make fully informed decisions or plan their sessions around specific seating needs.



*New Meta Arena, Prague<sup>1</sup>*

<sup>1</sup> <https://centers.ggcircuit.com/new-meta>

*GoGaming, Eindhoven<sup>2</sup>*

*El Tempo Esports, Barcelona<sup>3</sup>*

### 1.3 Objective

The objective of this project is to design and develop a modern, intuitive, and efficient desktop application for managing a cyber club. The application aims to provide seamless seat booking, real-time seating visibility via an interactive map, inventory tracking, and staff management — all while avoiding the use of heavy ORM frameworks. The focus is on **user experience**, **administrative control**, and **performance**.

### 1.4 Strategy

**Modular design** — separating logic into clearly defined components like database handling, UI, and business logic.

**Direct SQL interaction** — using ADO.NET for precise control over database queries and performance.

**Comparative analysis** — reviewing existing platforms in major European cities to identify pain points and improve upon them.

<sup>2</sup> <https://www.gogaming.gg/en/book-gametime/>

<sup>3</sup> <https://eltempoesports.com/en/>

## 1.5 Technology used

**C# (.NET)** — for backend and application logic

**SQL Server** — as the primary relational database

## 1.6 End products

- Interactive seat map with real-time seat availability and visual zones (VIP, Pro, Console).
- Custom booking with time, duration, and seat selection.
- Admin tools for managing users, and availability.
- Custom data access layer using QueryBuilder.cs for direct SQL interaction.
- Authentication system based on session storage and role permissions (extensible).
- Modular codebase designed for future integration (mobile app, payment systems, loyalty systems).

## 1.7 Significance of the Research

The research demonstrates that many existing systems, especially in top-tier European cyber clubs, lack user-friendly, modern solutions for visual booking and full digital control. This project directly addresses:

- The absence of visual seat selection
- Physical-only registration limitations
- Lack of real-time transparency in availability

This solution offers a customizable, and visually intuitive platform for cyber club operations, showing that with proper architectural planning, such tools can be developed quickly and adapted to different business models.

## **2. Methodology**

### **2.1 Research Design**

Key steps:

- Analyzing existing booking systems
- Identifying UX gaps and feature deficiencies
- Developing prototypes with direct user input
- Iterative improvements based on hands-on testing and research

### **2.2 Data Collection Methods**

- Online review of cyber club websites and booking flows
- Screenshots, video recordings, and mockup evaluations
- Analysis of what is visible during the booking process: time pickers, seat visibility, availability logic
- Documentation of best UX practices observed across industries (e.g., cinema, co-working)

### **2.3 Collecting feedback**

- Test users (classmates, teachers, friends) interacted with early versions of the seat map and booking flow
- Feedback centered around:
  - Clarity of seat layout
  - Responsiveness and animation
- Receiving the feedback from teachers in Feedpulse

## 3. Results

### 3.1 Phases of the project

### 3.2 Technical analysis

This section presents an in-depth technical overview of the Cyber Club application, focusing on both implemented and planned modules

#### 3.2.1 Architecture Overview

- **Presentation Layer:** Responsible for user interaction. Implemented using MVS. Contains view models for booking, menu management, and customer views (e.g., \_CustomerLayout.cshtml).
- **Service Layer:** Business logic hub, handling coordination between repositories and data operations (e.g., BookingService.cs).
- **Data Access Layer:** Interacts with the SQL Server database using a custom QueryBuilder class, enabling raw SQL execution with parameterization.

#### 3.2.2 Key Functional Modules

##### 1. Booking System (*Partially Implemented/Testing*)

- Users can select a zone, seat, date/time, and duration.
- BookingService handles seat availability checks and passes data to the BookingRepository for persistence.
- Bookings are inserted with SCOPE\_IDENTITY() to return the ID.
- Real-time seating map integration is planned for visual selection and availability display.
- seats are modeled as distinct entities.

##### 2. Interactive Seating Map (*Planned*)

- Users will visually see all seats and their current status.
- Planned integration with booking system to allow map-based selection.

##### 4. User & Role Management (*Partially implemented*)

- Admin/customer roles.
- Permissions for booking approval, seat assignment override, etc.
- Authentication.

##### 5. Reporting & Analytics (*Planned*)

- Booking history, seat usage heatmap.
- Visual dashboards for admins.

##### 6. Database Management & Query Execution (*Implemented*)

- QueryBuilder allows flexible, secure SQL execution.
- Repository classes use dependency injection to pass queries into the DB layer.
- No ORM used — all SQL is handwritten and optimized for performance.

#### 3.2.3 New stuff

##### 1. Session Management & Custom Authorization

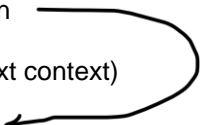
Since I had an experience via using a built-in cookie-based authorization mechanism ([Authorize] with cookie authentication). In this application, user access control is managed through a custom session-



based authorization attribute called SessionAuthorizeAttribute(Because there are no built-in mechanisms to use for sessions).

After the successful login the user email is stored in the session

```
public override void OnActionExecuting(ActionExecutingContext context)
{
    var user = context.HttpContext.Session.GetString("User");
    if (string.IsNullOrEmpty(user))
    {
        context.Result = new RedirectToActionResult("Login", "Auth", null);
    }
}
```



The SessionAuthorizeAttribute class inherits from ActionFilterAttribute and overrides the OnActionExecuting method. It checks whether the session variable User exists.

### Usage Example:

Now I use it to protect my controller for customers:

```
[SessionAuthorize]
public class CustomerController : Controller
```

I plan to extend my custom made attribute to also check roles stored in the session.

## **4. Conclusions**

## 5. References