

**CAPSTONE PROJECT REGISTER**

Class: SE Duration time: From 05/2025 to 08/2025

(\*) Profession: Software Engineer Specialty: SE

(\*) Kinds of person make registers: Lecturer □ Students 🗹

# **Register information for supervisor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Full name | Phone | E-Mail | Title |
| Supervisor 1 | Lâm Hữu Khánh Phương | 0915353001 | [phuonglhk@fpt.edu.vn](mailto:phuonglhk@fpt.edu.vn) | Mr |
| Supervisor 2 | - | - | - | - |

# **Register information for students**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ST | Full name | Student code | Phone | E-mail | Role in Group |
| 1 | Trần Minh Tuấn | QE170030 | 0818035089 | tuantmqe170030@fpt.edu.vn | Leader |
| 2 | Nguyễn Minh Tuấn | QE170254 | 0379206581 | tuannmqe170254@fpt.edu.vn | Member |
| 3 | Huỳnh Gia Bảo | QE170095 | 0934872154 | baohgqe170095@fpt.edu.vn | Member |
| 4 | Võ Nguyễn Kim Ngân | QE170019 | 0935109750 | nganvnkqe170019@fpt.edu.vn | Member |
| 5 | Cao Trung Hoan | QE170187 | 0357859348 | hoanctqe170187@fpt.edu.vn | Member |

# **3. Register Content of Capstone Project**

#### **3.1. Capstone Project Name**

* **English**: ScoreLens – Integrating AI into Cameras for Scoring in Pool and Carom Billiards, with Real-Time Notifications via Web.
* **Vietnamese**: Hệ thống tính điểm ScoreLens – Tích hợp AI vào Camera để nhận diện điểm trong Billiards Pool và Billiard Carom, với thông báo điểm và lỗi trực tuyến qua Web.
* **Abbreviation**: ScoreLens – Every Shot, Every Rule

#### **3.2. Problem Statement**

In competitive Billiards (both Pool and Carom), accurate and real-time score tracking is essential for fair play and efficient game monitoring. Traditional scoring methods rely on manual tracking, which is prone to human errors and delays. There is a significant challenge in implementing an automated system capable of accurately detecting successful shots, fouls, and calculating points during fast-paced games. This results in inconsistencies and missed opportunities for players, referees, and viewers. Therefore, there is a critical need for an AI-based automated scoring system that tracks ball movements, detects shots, calculates scores, and provides real-time feedback through a web interface, improving both the gameplay experience and operational efficiency.

#### **3.3. Context**

The project focuses on creating an AI-powered Billiards scoring system designed for both Billiards Pool and Carom. The system will use cameras placed around the billiards table to capture the movement of the balls and AI to analyze and score the game in real time. This system will address the operational challenges associated with manual score tracking, including human errors, inconsistencies, and delays in decision-making. Additionally, the system will provide real-time score updates and notifications of any fouls or illegal moves directly to a web interface accessible by players, referees, and spectators.

#### **3.4. Proposed Solution**

The proposed solution is an AI-powered Billiards Scoring System that uses advanced computer vision and deep learning techniques to detect and analyze ball movements during gameplay. The cameras installed around the table will continuously monitor the game and track ball positions. The AI will calculate the score based on predefined rules (e.g., successful shots, fouls, legal caroms), and the results will be sent to a web-based interface. This system will provide real-time score updates and notify players and referees of any violations or critical decisions. It will also offer a historical record of the game’s performance for analysis and review. The system will integrate seamlessly with a web dashboard that provides immediate access to scores, game analytics, and decision logs.

#### **3.5. Functional Requirements**

**A. Camera Integration & Shot Detection**

* Use AI-powered cameras to track ball movements and determine the type of shot taken.
* Detect successful shots, fouls, and legal caroms, and update the score accordingly.

**B. Real-Time Score Display**

* Display live scores on a web interface accessible by players, referees, and spectators.
* Automatically update scores as soon as a shot is made or a foul occurs.

**C. Foul & Error Detection**

* Detect common fouls (e.g., illegal shots, wrong ball contact, not hitting the correct object ball).
* Notify the referee and players immediately via the web interface.

**D. Web Interface for Score and Decision Viewing**

* A user-friendly web application will show real-time scores, foul notifications, and game analysis.
* The web interface will include options for both computer and mobile access, ensuring accessibility for players, referees, and spectators.

**E. Game History & Analytics**

* Track and store game data, including shot types, points scored, fouls committed, and other relevant statistics.
* Provide post-match analysis and insights for players and referees.

#### **3.6. System Actors**

**A. Referee (using web interface)**

* Monitor and verify real-time scores.
* Review AI-detected fouls or shot errors when requested.
* Approve or override automated decisions.

**B. Player (using web interface)**

* View live scores and notifications about fouls or valid shots.
* Access historical performance data and game statistics after the match.

**C. Spectator (using web interface)**

* View live scores, shots, fouls, and match progress in real-time.
* Engage with the game via live-streamed scores and updates.

**D. AI and Camera System (backend processing)**

* Use deep learning models to detect ball positions, validate shots, and compute scores.
* Transmit scoring data and fouls to the backend system for display.

#### **3.7. Main Proposal Content (System Requirements and Deliverables)**

**A. Documentation Requirements:**

* Complete software development lifecycle documentation using UML 2.0.
* Required documents:
  + **User Requirements Document (URD)**
  + **Software Requirements Specification (SRS)**
  + **Software Architecture Document (SAD)**
  + **Detailed Design Document (DDD)**
  + **Implementation Documentation**
  + **Test Documentation**
  + **Installation Guides**

**B. Technical Stack:**

* **Backend:** Python with TensorFlow for AI-based image processing and event detection.
* **Database & Caching:** Supabase for game data storage.
* **APIs & Microservices:** RESTful APIs to communicate game data and results in real-time, powered by WebSockets for real-time updates.
* **Client-Side Technologies:** ReactJS for the web interface.
* **Third-Party Services:** Integration with external services for cloud storage, logging, and monitoring.
* **Deployment:** The system will be deployed on AWS EC2 instances and managed via a secure cloud platform.

|  |  |
| --- | --- |
| Supervisor (If have)  (Sign and full name) | On behalf of the Registers |