

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**



MÔN HỌC: LẬP TRÌNH MẠNG CĂN BẢN

**BÁO CÁO ĐỒ ÁN GIỮA KỲ**

**Pixel Drift**

**Lớp:** NT106.Q12

**Sinh viên thực hiện:**

Nguyễn Duy Hưng - 24520602

(Nhóm trưởng: email nhóm trưởng: 24520602@gm.uit.edu.vn)

Nguyễn Hoàng Khánh Đăng - 24520254

Ngô Xuân Minh Hoàng-24520549

Lê Tiến Hưng - 24520596

Hoàng Phi Hùng - 24520587

**Link Github:** <https://github.com/HugndUIT/PixelDrift-NT106.Q12>

TP. Hồ Chí Minh, ngày 29 tháng 10 năm 2025

## **MỤC LỤC**

### **I. GIỚI THIỆU ĐỀ TÀI**

- 1. Lý do chọn đề tài**
- 2. Mục tiêu của đề tài**
- 3. Giới thiệu tổng quan về đề tài**
- 4. Các tính năng chính của hệ thống**

### **II. NETWORK STACK**

### **IV. FLOW HOẠT ĐỘNG**

### **III. KIẾN TRÚC HỆ THỐNG**

- 1. Mô hình kiến trúc tổng thể**
- 2. User Story**

### **IV. GIAO DIỆN ỨNG DỤNG**

### **V. CÁC PHẦN KHÁC NHÓM ĐÃ HOÀN THÀNH TÍNH ĐẾN THỜI ĐIỂM BÁO CÁO**

### **VI. PHÂN CÔNG CÔNG VIỆC**

## I. GIỚI THIỆU ĐỀ TÀI

### 1. Lý do chọn đề tài

Nhóm em chọn đề tài trò chơi đua xe vì đây là thể loại vừa mang tính kỹ thuật cao, vừa có tính giải trí hấp dẫn và phổ biến với nhiều người chơi. Việc phát triển một trò chơi đua xe giúp nhóm em có cơ hội vận dụng tổng hợp nhiều kiến thức đã học như lập trình hướng đối tượng, xử lý đồ họa, âm thanh, điều khiển chuyển động và tương tác người dùng. Thông qua quá trình thực hiện, nhóm em có thể rèn luyện tư duy logic, khả năng giải quyết vấn đề và kỹ năng thiết kế giao diện, đồng thời hiểu rõ hơn về quy trình xây dựng một ứng dụng hoàn chỉnh — từ ý tưởng, thiết kế cho đến thử nghiệm và tối ưu. Ngoài ra, thể loại đua xe còn mang ý nghĩa tượng trưng cho tinh thần nỗ lực, không ngừng tiến lên, điều mà tôi muốn thể hiện qua sản phẩm của mình.

### 2. Mục tiêu đề tài

Mục tiêu của đồ án là xây dựng một trò chơi đua xe có khả năng giao tiếp qua mạng, giúp nhiều người chơi có thể tham gia và thi đấu với nhau trong cùng một hệ thống. Thông qua quá trình thực hiện, nhóm em mong muốn vận dụng và củng cố các kiến thức đã học trong môn Lập trình mạng căn bản, đặc biệt là về lập trình socket, mô hình client–server, truyền nhận dữ liệu qua giao thức TCP, cũng như xử lý đồng bộ và đa luồng trong môi trường mạng. Bên cạnh đó, đề tài còn hướng đến việc rèn luyện kỹ năng thiết kế hệ thống, xử lý tình huống thực tế khi lập trình ứng dụng mạng, đồng thời giúp tôi hiểu sâu hơn về cách thức các trò chơi trực tuyến hoạt động và giao tiếp. Cuối cùng, thông qua dự án này, nhóm em muốn nâng cao khả năng phân tích, lập trình và tối ưu ứng dụng có tính tương tác thời gian thực — một kỹ năng quan trọng cho các dự án phần mềm hiện đại.

### 3. Giới thiệu đề tài

Luật chơi:

- + Mỗi người chơi điều khiển một xe trên 2 làn đường riêng biệt.
- + Cả hai player phải điều khiển xe qua trái, phải để tránh các chướng ngại vật. Nếu trúng chướng ngại vật thì xe sẽ bị giảm tốc độ.
- + Tốc độ di chuyển sẽ tăng dần. Trên đường sẽ xuất hiện các buff tăng tốc và giảm tốc.
- + Trò chơi sẽ kết thúc sau 60s, player nào đạt điểm cao hơn sẽ thắng.

#### **4. Các tính năng của đề tài**

- a) Tính năng kết nối mạng (Client – Server)
  - Hệ thống được thiết kế theo mô hình Client–Server.
  - Server chịu trách nhiệm quản lý người chơi, phòng đua và điểm số.
  - Client (người chơi) có thể kết nối vào server thông qua địa chỉ IP và cổng được định nghĩa sẵn trong mạng LAN.
  - Hỗ trợ nhiều client kết nối cùng lúc, xử lý song song bằng đa luồng để đảm bảo hiệu năng.
  - Khi tất cả người chơi đã vào phòng, Server gửi tín hiệu “Start Race”, đồng bộ trạng thái ban đầu (vị trí xe, bắn đỗ, điều kiện đua).
  - Trong quá trình đua, server nhận liên tục trạng thái xe từ client (vị trí, vận tốc) và phát lại dữ liệu cho các client khác, đảm bảo hiển thị thời gian thực.
- b) Tính năng điều khiển và hiển thị trò chơi
  - Giao diện đồ họa mô phỏng xe đua dạng pixel (đơn giản, dễ hiển thị, chạy mượt trên LAN).
  - Người chơi điều khiển bằng bàn phím (phím mũi tên trái, phải hoặc AD).
  - Xe có các thông số vật lý: tốc độ, hướng di chuyển.
  - Hệ thống xử lý va chạm, tăng tốc, giảm tốc và tính điểm theo thời gian thực.
  - Các sự kiện trong game (va chạm, hoàn thành vòng, thắng cuộc) được gửi về server để cập nhật kết quả chung cho tất cả người chơi.
- c) Tính năng truyền – nhận dữ liệu
  - Giao tiếp giữa client và server sử dụng socket TCP.
  - Dữ liệu bao gồm: vị trí, vận tốc, điểm, và trạng thái người chơi.
  - Có cơ chế đồng bộ dữ liệu liên tục để đảm bảo trải nghiệm mượt mà.

- Cơ chế đồng bộ dữ liệu định kỳ giúp đảm bảo mọi người chơi đều thấy cùng trạng thái đua trong LAN.
- d) Tính năng bảng xếp hạng (Leaderboard)
  - Server lưu trữ và cập nhật điểm số của từng người chơi.
  - Hiển thị bảng xếp hạng tổng với số lần thắng và số lần va chạm vật cản của người chơi.
  - Cho phép cập nhật dữ liệu theo phiên chơi.
  - Cho phép người chơi xem lại bảng điểm cũ.
- e) Tính năng lưu và tải dữ liệu
  - Lưu dữ liệu người chơi (tên, điểm, số trận thắng, số lần va chạm).
  - Khi khởi động lại chương trình, hệ thống có thể tải lại dữ liệu cũ.
- f) Tính năng phòng đua
  - Người chơi có thể tạo phòng đua mới hoặc tham gia phòng sẵn có trong mạng LAN.
  - Khi tạo phòng:
    - Các client khác nhập IP và cổng để tham gia.
  - Khi đủ người:
    - Server gửi tín hiệu bắt đầu, đồng bộ vị trí xe và bản đồ.
  - Khi kết thúc:
    - Server gửi thông báo Winner, cập nhật bảng xếp hạng, lưu điểm và thống kê.

## II. NETWORK STACK

- Client gửi dữ liệu cho Server
  - + Ở phía **Client**, dữ liệu người dùng được đóng gói và truyền đi qua bốn tầng của mô hình mạng như sau:

### Tầng ứng dụng (Application Layer):

Các form như Form\_Dang\_Nhap, Form\_Dang\_Ki, Form\_Game\_Play, Form\_Scoreboard và Form\_Thong\_Tin thu thập dữ liệu khác nhau tùy chức năng. Cụ thể:

- Form\_Dang\_Nhap gửi username và password có cấu trúc action = "login", username = username, password = hashedPassword.
- Form\_Dang\_Ki gửi thông tin tài khoản mới có cấu trúc action = "register", username = username, email = email, password = hashedPassword.

- Form Thong\_Tin gửi hoặc cập nhật hồ sơ người chơi có cấu trúc action = "get\_info", username = currentUsername.

Tất cả dữ liệu này được đóng gói thành thông điệp JSON và mã hóa thành chuỗi byte trước khi truyền đi.

#### **Tầng giao vận (Transport Layer):**

Client sử dụng TcpClient để gửi dữ liệu. TCP sẽ tự động thêm port nguồn, port đích, sequence number và checksum, đảm bảo dữ liệu được truyền đầy đủ, đúng thứ tự và không bị mất gói.

#### **Tầng mạng (Internet Layer):**

Hệ thống thêm địa chỉ IP nguồn và IP đích, tạo thành IP Packet để định tuyến đến đúng Server.

#### **Tầng liên kết dữ liệu (Network Access Layer):**

Gói tin cuối cùng được đóng thành Ethernet Frame có chứa MAC nguồn, MAC đích và mã kiểm tra lỗi CRC, rồi được gửi qua LAN hoặc Wi-Fi đến Server.

- + Ở phía Server, dữ liệu được tiếp nhận và xử lý qua bốn tầng mạng tương tự như phía Client, nhưng theo hướng ngược lại.

#### **Tầng liên kết dữ liệu (Network Access Layer):**

Server nhận các Ethernet Frame từ card mạng thông qua LAN hoặc Wifi. Mỗi frame chứa địa chỉ MAC đích trùng với card mạng của server. Khi nhận được, hệ thống kiểm tra mã lỗi CRC, sau đó bóc phần khung (frame) để lấy ra IP Packet.

#### **Tầng mạng (Internet Layer):**

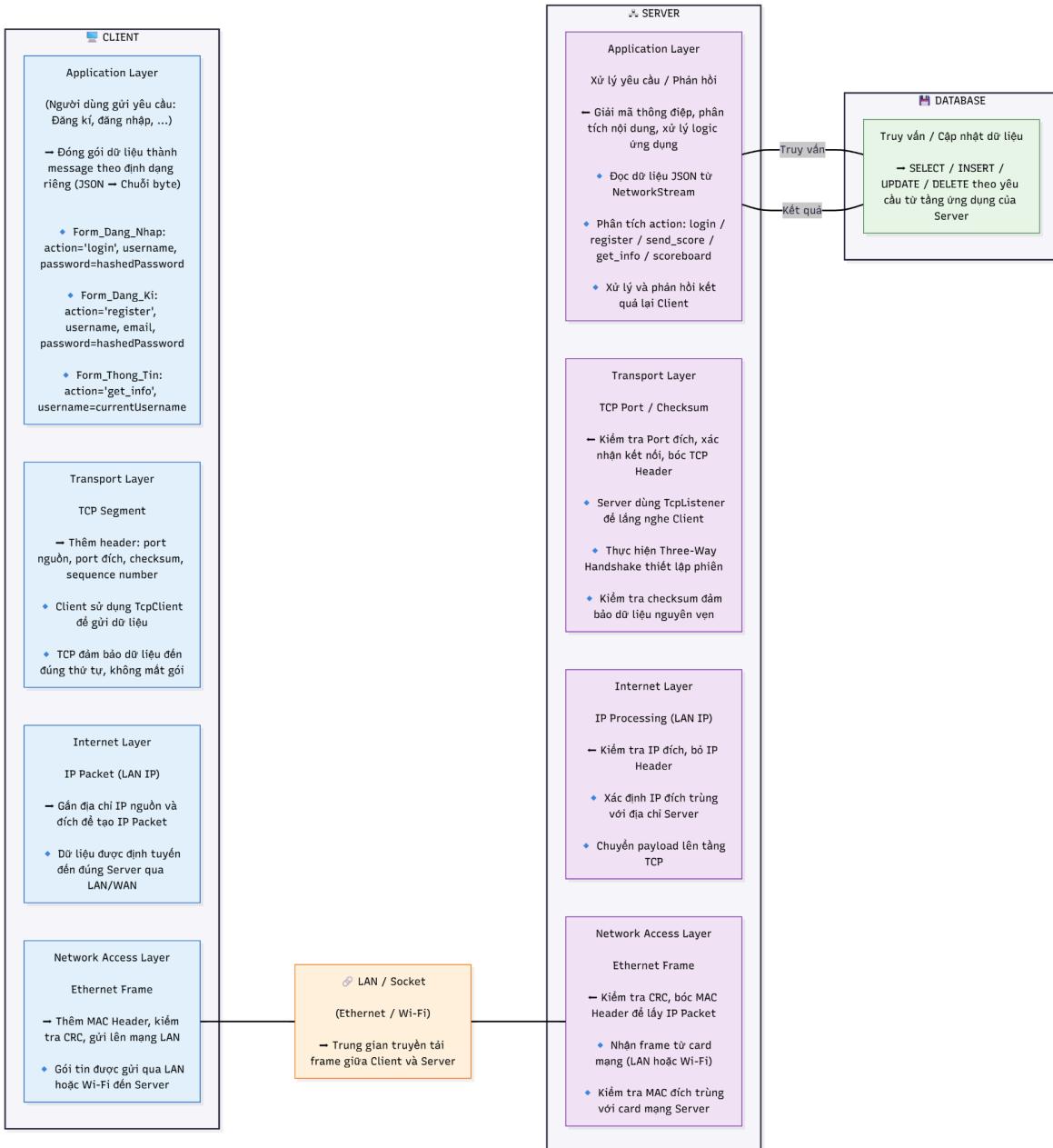
Hệ điều hành trên server sẽ tiếp tục bóc phần IP header, kiểm tra địa chỉ IP đích, và chuyển gói dữ liệu (payload) lên tầng giao vận. Lúc này, gói tin vẫn chứa phần dữ liệu của TCP.

#### **Tầng giao vận (Transport Layer):**

Server sử dụng TcpListener để lắng nghe các kết nối từ client. Khi có client kết nối, TCP sẽ thực hiện bắt tay ba bước (three-way handshake) để thiết lập phiên làm việc. Tại đây, TCP kiểm tra checksum, đảm bảo dữ liệu nguyên vẹn, rồi chuyển phần nội dung thực (payload) lên tầng ứng dụng.

#### **Tầng ứng dụng (Application Layer):**

Server đọc dữ liệu từ NetworkStream của mỗi client giải mã dữ liệu. Dữ liệu JSON này được phân tích để xác định loại yêu cầu: đăng nhập, đăng ký, gửi điểm, lấy bảng xếp hạng hoặc truy cập thông tin người chơi. Tùy loại yêu cầu, server xử lý tương ứng.



- Server gửi dữ liệu cho client
  - + Ở phía **Server**, sau khi xử lý yêu cầu từ Client, dữ liệu được gửi ngược lại qua bốn tầng mạng như sau:

#### **Tầng ứng dụng (Application Layer):**

Server lấy dữ liệu cần phản hồi (kết quả đăng nhập, điểm số, bảng xếp hạng, thông tin người chơi...) từ cơ sở dữ liệu. Sau đó, dữ liệu được chuyển thành JSON và mã hóa thành chuỗi byte để chuẩn bị gửi về Client.

#### **Tầng giao vận (Transport Layer):**

Dữ liệu JSON được đóng gói thành TCP Segment. TCP thêm vào các thông tin điều khiển như port nguồn, port đích, checksum, và sequence number để đảm bảo quá trình truyền ổn định và không bị mất gói.

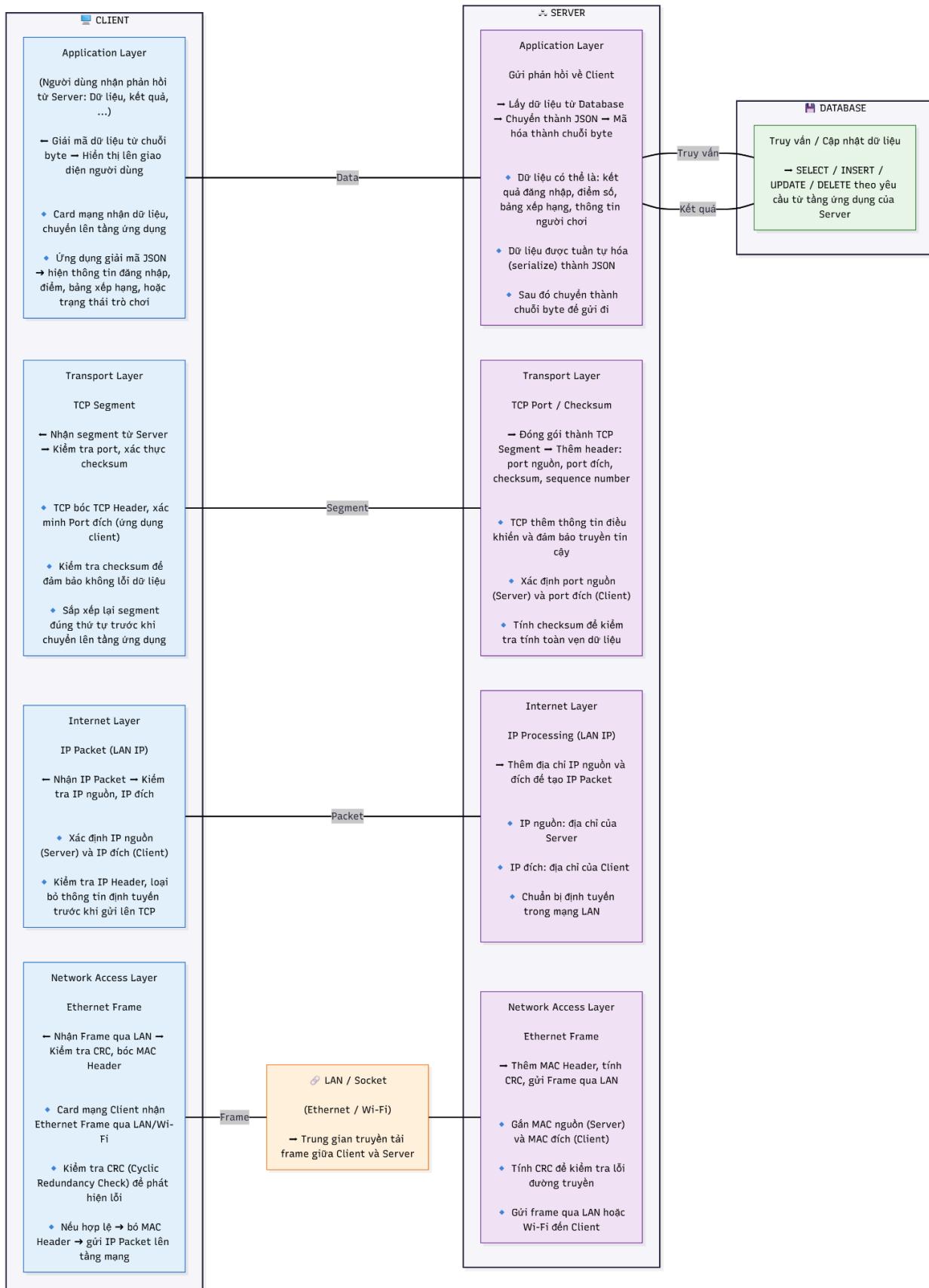
#### **Tầng mạng (Internet Layer):**

Hệ thống gắn địa chỉ IP nguồn (của Server) và địa chỉ IP đích (của Client) để tạo thành IP Packet sẵn sàng định tuyến trong mạng LAN.

#### **Tầng liên kết dữ liệu (Network Access Layer):**

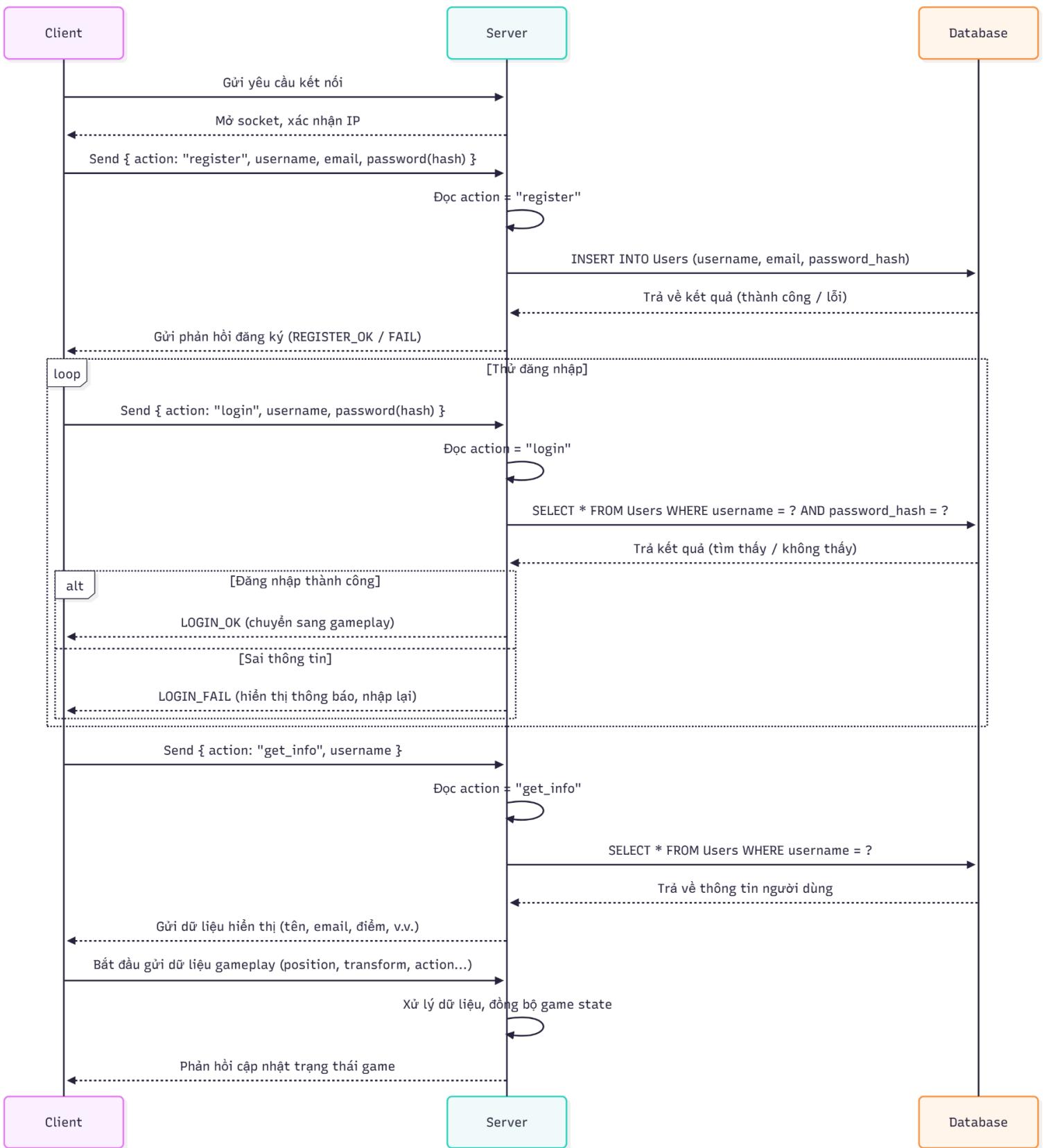
IP Packet được đóng gói thêm MAC Header, tính CRC để kiểm tra lỗi, rồi tạo thành Ethernet Frame. Frame này sau đó được gửi qua LAN hoặc Wifi đến máy Client.

- + Ở phía **Client**, quá trình diễn ra ngược lại: card mạng nhận frame, bóc các lớp header theo thứ tự từ Ethernet → IP → TCP, rồi chuyển dữ liệu JSON lên tầng ứng dụng để giải mã từ chuỗi byte, hiển thị kết quả lên giao diện hoặc cập nhật trạng thái trò chơi



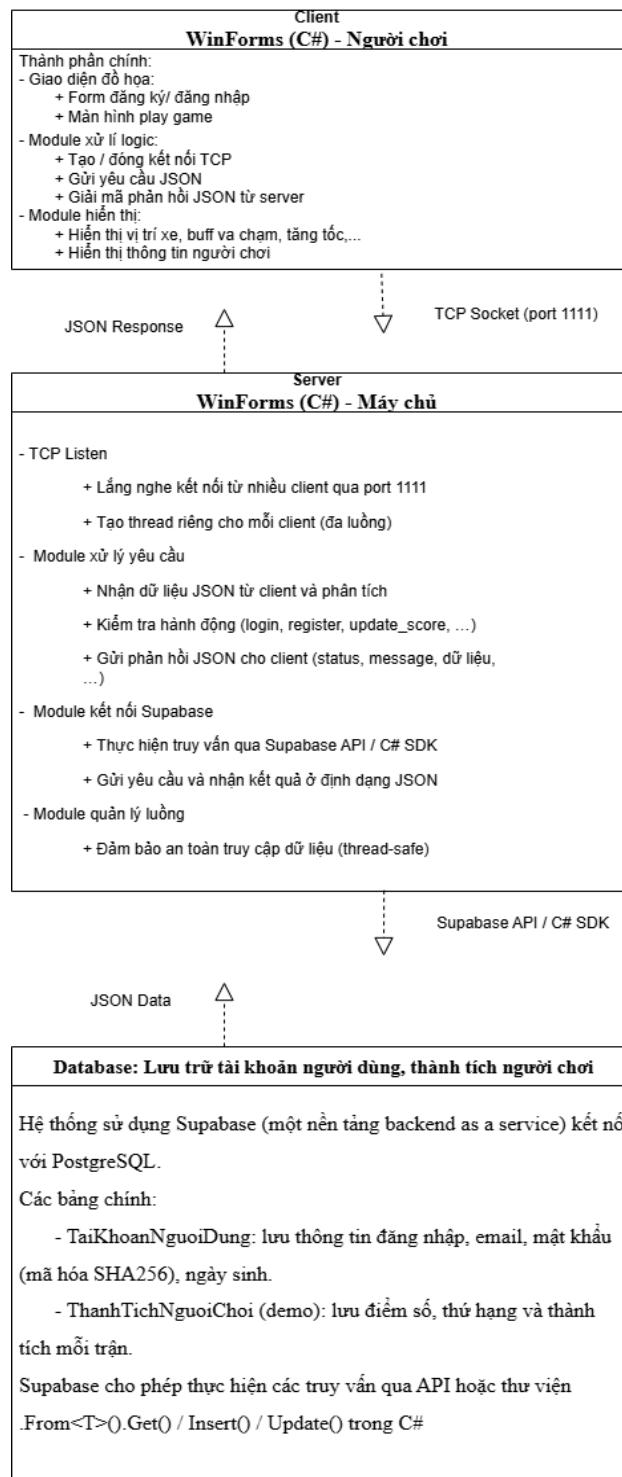
### **III. FLOW HOẠT ĐỘNG**

- Client: Gửi kết nối với server
- Server: Mở socket, xác nhận IP cho phép client kết nối
- Client đăng ký: Gửi đến server data có dạng:
  - ◆ action (Hành động): hành động đăng ký sẽ có giá trị là "register"
  - ◆ username (Tên đăng nhập)
  - ◆ email (email)
  - ◆ password (Mật khẩu đã mã hóa)
- Server: đọc data, lấy ra trường action để nhận diện hành động client muốn thực hiện, server lấy tiếp từ data các dữ liệu còn lại sau đó thực hiện truy vấn SQL để thêm dữ liệu của người dùng vào database hay thực hiện việc đăng ký tài khoản
- Client đăng nhập: Gửi đến server request có dạng:
  - ◆ action (Hành động): hành động đăng nhập sẽ có giá trị là "login"
  - ◆ username (Tên đăng nhập)
  - ◆ password (Mật khẩu đã mã hóa)
- Server: đọc request, lấy ra trường action để nhận diện hành động client muốn thực hiện, server lấy tiếp từ data các dữ liệu còn lại sau đó thực hiện truy vấn SQL để kiểm tra xem tài khoản đã tồn tại hay chưa, nếu rồi thì chuyển vô gameplay, nếu chưa xuất thông báo có sai sót
- Client hiển thị thông tin: Sau khi đăng nhập thành công, client gửi một request có dạng:
  - ◆ action (Hành động): hành động hiển thị thông tin sẽ có giá trị là "get\_info",
  - ◆ username (Tên đăng nhập hiện tại)
- Server: đọc request lấy ra trường action để nhận diện hành động client muốn thực hiện, server lấy tiếp từ data các dữ liệu còn lại sau đó thực hiện truy vấn SQL để lấy ra thông tin người dùng, sau đó gửi dữ liệu cho client để client hiển thị cho người dùng
- Sau khi đăng ký, đăng nhập xong Client sẽ vô màn hình game và bắt đầu chơi, ...



## IV. KIẾN TRÚC HỆ THỐNG

**1. Kiến trúc hệ thống:** Hệ thống Pixel Drift được xây dựng theo mô hình Client – Server – Database, sử dụng TCP Socket để truyền thông tin giữa người chơi và máy chủ, kết hợp với Supabase để lưu trữ dữ liệu người dùng.

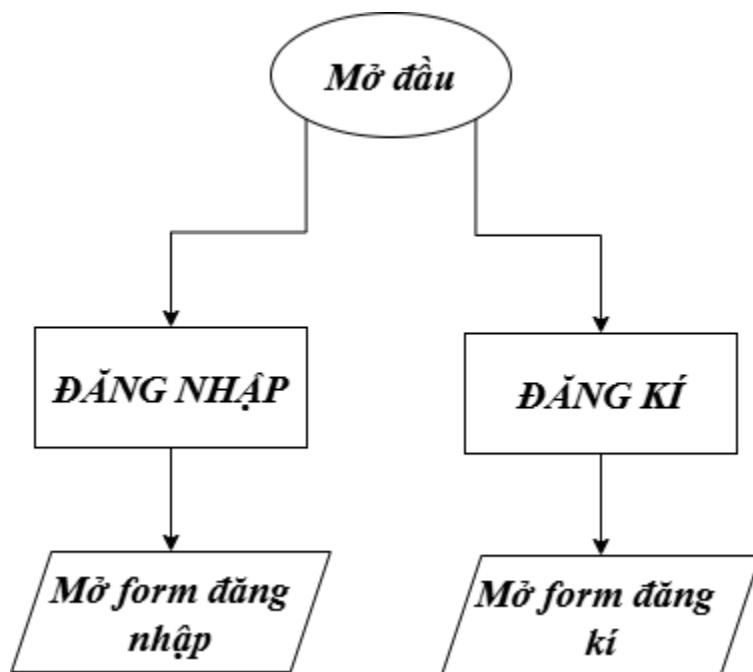


## 2. User story:

### 2.1. Mở đầu

- Với tư cách là một người chơi, tôi muốn được chọn đăng ký để tạo một tài khoản mới.
- Với tư cách là một người chơi, tôi muốn chọn đăng nhập để có thể vào game

Scenario	Given	When	Then
Đăng ký một tài khoản mới	Người chơi đang ở form mở đầu	Người nhấn chọn nút "Đăng ký"	Người dùng được chuyển đến form đăng ký để tạo tài khoản
Đăng nhập vào tài khoản có sẵn	Người chơi đang ở form mở đầu	Người chơi nhấn chọn đăng nhập	Người dùng được chuyển đến form đăng nhập để đăng nhập vô game

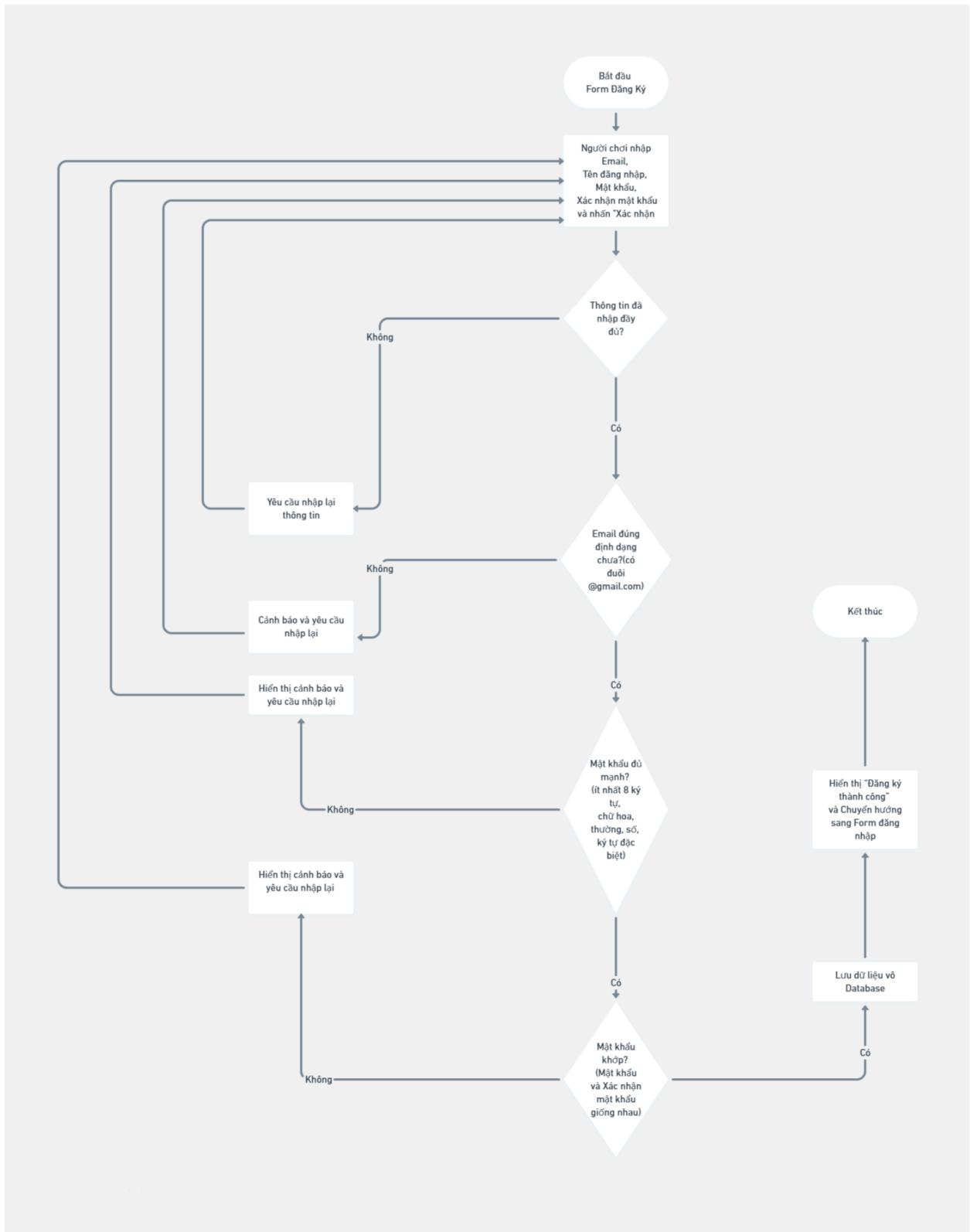


Người dùng ở form mở đầu

## 2.2. Đăng ký

- Với tư cách là người chơi mới, tôi muốn đăng ký một tài khoản mới bằng email, để tôi có thể đăng nhập vô game và chơi game.

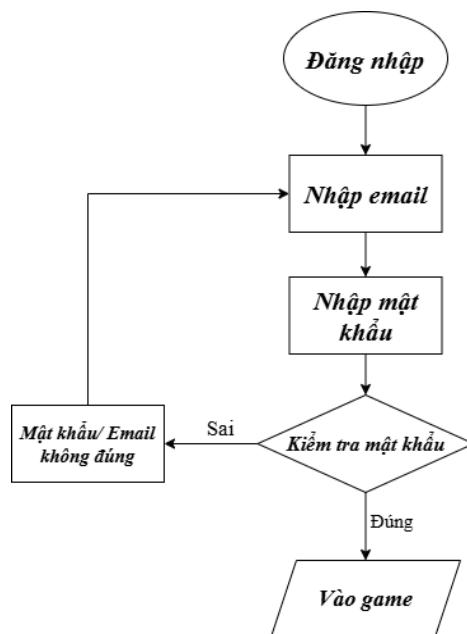
Scenario	Given	When	Then
Đăng ký thành công	Người chơi đang ở "Form Đăng Ký"	Người chơi nhập đầy đủ và đúng định dạng: -Email -Tên đăng nhập -Mật khẩu -Xác nhận mật khẩu Cuối cùng người chơi nhấn nút "Xác nhận"	Hệ thống sẽ kiểm tra, lưu thông tin vào database, hiển thị thông báo thành công và chuyển hướng sang "Form đăng nhập"
Lỗi - Bỏ trống thông tin	Người chơi đang ở "Form Đăng Ký"	Người chơi để trống 1 trong 4 ô "Email", "Tên đăng nhập", "Xác nhận mật khẩu" hoặc "Mật khẩu" và nhấn "Xác nhận"	Hệ thống hiển thị cảnh báo: "Vui lòng nhập đầy đủ thông tin!"
Lỗi - Email/SĐT không hợp lệ	Người chơi đã nhập Tên đăng nhập và Mật khẩu	Người chơi nhập email không đúng định dạng (không có đuôi <a href="mailto:@gmail.com">@gmail.com</a> ) và nhập các thông tin khác, cuối cùng nhấn "Xác nhận"	Hệ thống hiển thị cảnh báo: "Email hoặc số điện thoại không hợp lệ!"
Lỗi - Mật khẩu yếu	Người chơi đã nhập Email và Tên đăng nhập	Người chơi nhập Mật khẩu không thỏa mãn điều kiện và nhập các thông tin khác, cuối cùng nhấn "Xác nhận"	Hệ thống hiển thị cảnh báo: "Mật khẩu phải có ít nhất 8 ký tự, bao gồm chữ hoa, chữ thường, số và ký tự đặc biệt!"
Lỗi - Mật khẩu không khớp	Người chơi đã nhập Email, Tên đăng nhập và Mật khẩu (đúng định dạng)	Người chơi nhập "Xác nhận mật khẩu" không giống "Mật khẩu", cuối cùng nhấn "Xác nhận"	Hệ thống hiển thị cảnh báo: "Mật khẩu không khớp!"



### 2.3. Đăng nhập

- Với tư cách là một người chơi, tôi muốn đăng nhập vào tài khoản của mình để chơi game.

Scenario	Given	When	Then
Đăng nhập thành công	Người chơi đang ở form đăng nhập	Người đã nhập đủ thông tin email và mật khẩu tài khoản. Mật khẩu được nhập đúng	Người dùng được chuyển đến màn hình chơi game
Lỗi - email không hợp lệ	Người chơi đang ở form đăng nhập	Người chơi nhập một email chưa được đăng ký tài khoản	Chương trình hiển thị thông báo lỗi và người chơi phải nhập lại thông tin
Lỗi - mật khẩu không khớp	Người chơi đang ở form đăng nhập	Người chơi nhập một mật khẩu không khớp với mật khẩu tài khoản	Chương trình hiển thị thông báo lỗi và người chơi phải nhập lại thông tin
Lỗi - một trong hai ô bị bỏ trống	Người chơi đang ở form đăng nhập	Người chơi bỏ trống ít nhất một ô thông tin	Chương trình hiển thị thông báo lỗi và người chơi phải nhập lại thông tin



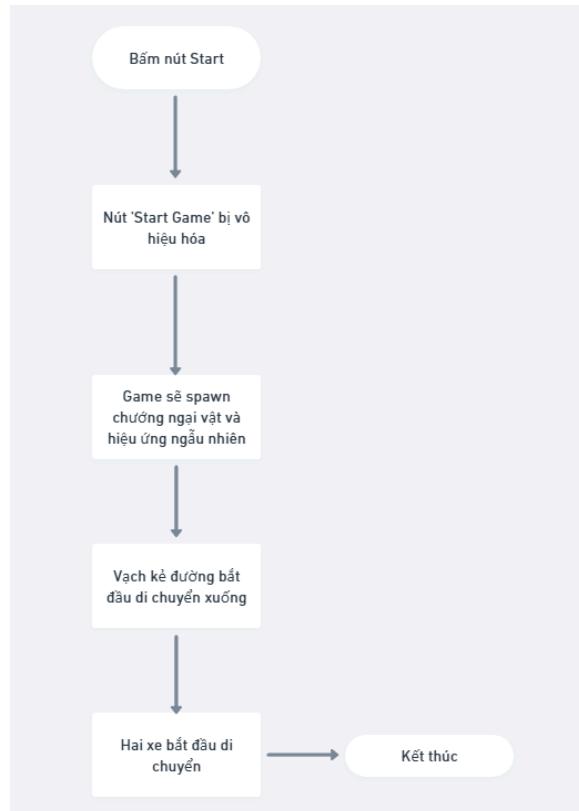
Người dùng ở form đăng nhập

## 2.4. Chơi game

### 2.4.1. Bắt đầu trò chơi

- Với tư cách là người chơi, tôi muốn nhấn nút "Start Game" để bắt đầu cho xe chạy và cái chướng ngại vật, hiệu ứng bắt đầu xuất hiện.

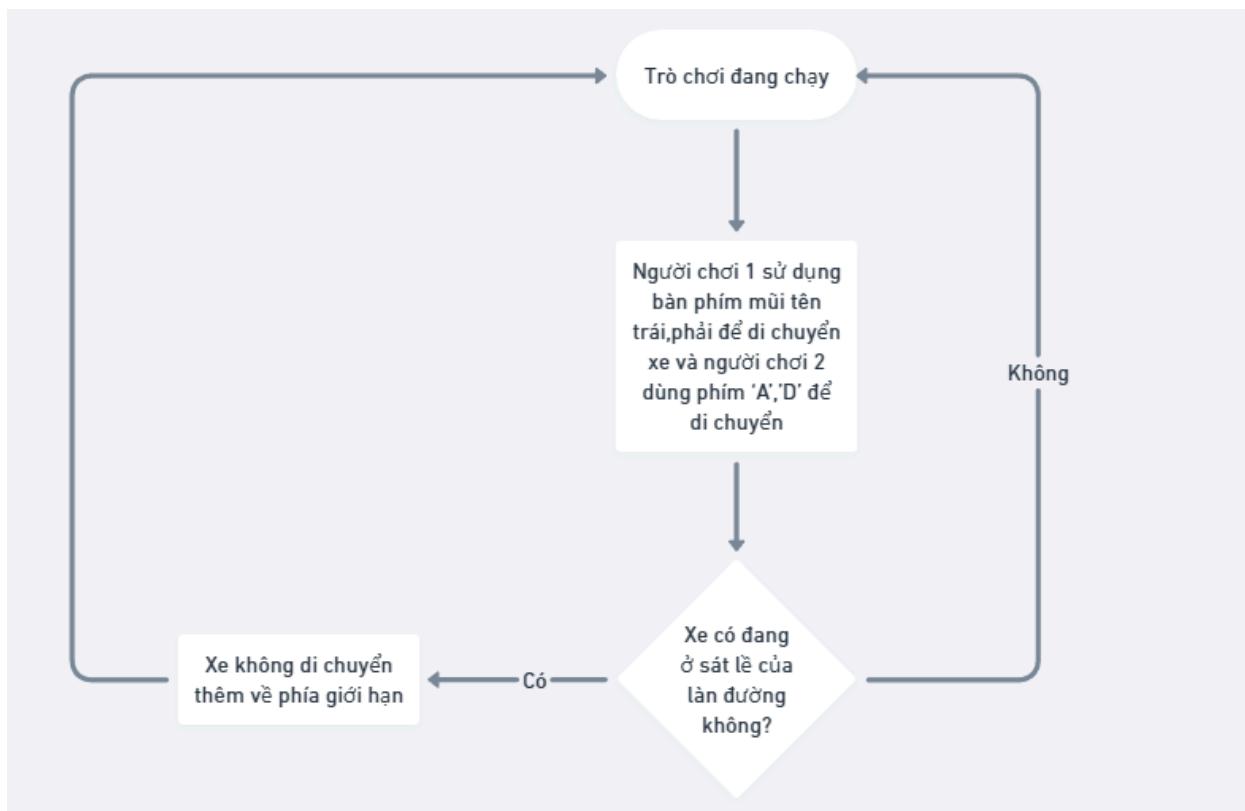
Scenario	Given	When	Then
Vô game	Tôi đang ở màn hình chính và game chưa bắt đầu	Tôi nhấn nút "Start Game"	<ul style="list-style-type: none"> <li>- Hai xe bắt đầu di chuyển</li> <li>- Vạch kẻ đường bắt đầu di chuyển xuống tạo cảm giác xe đang chạy</li> <li>- Game sẽ spawn chướng ngại vật và hiệu ứng ngẫu nhiên trên đường đi</li> <li>- Nút "Start Game" bị vô hiệu hóa</li> </ul>



## 2.4.2. Điều khiển xe

-Với tư cách là người chơi, tôi muốn di chuyển xe của mình sang trái và phải trong làn đường của mình, để né các chướng ngại vật và ăn cái hiệu ứng.

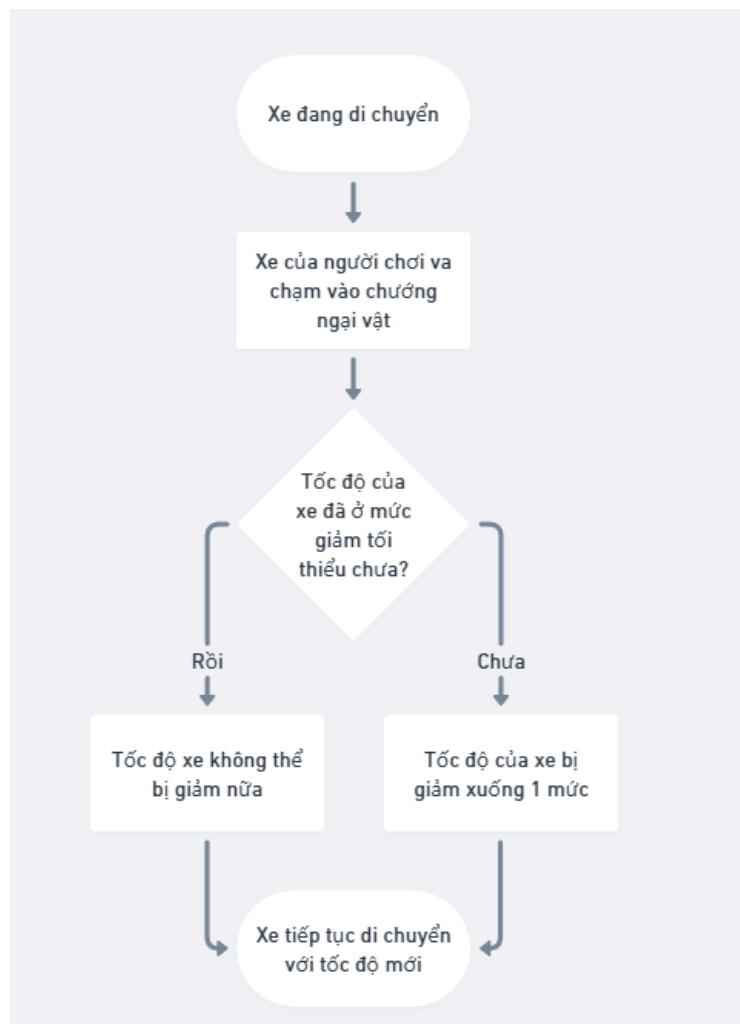
Scenario	Given	When	Then
Player 1 điều khiển	Trò chơi đang chạy	Người chơi 1 (xe đỏ) nhấn phím “Mũi tên trái” hoặc “Mũi tên phải”	Xe đỏ di chuyển trái phải tương ứng
Player 2 điều khiển	Trò chơi đang chạy	Người chơi 2 (xe xanh) nhấn phím “A” hoặc “D”	Xe xanh di chuyển trái phải tương ứng
Giới hạn làn đường	Xe đang ở sát lề của làn đường	Người chơi nhấn phím di chuyển để xe ra khỏi làn đường của mình	Xe không di chuyển thêm



### 2.4.3. Va chạm với chướng ngại vật

- Với tư cách là người chơi, khi xe của tôi va chạm chướng ngại vật, tôi muốn xe bị giảm tốc độ ngay lập tức, để tạo độ khó cho game.

Scenario	Given	When	Then
Va chạm	Xe đang di chuyển và gặp chướng ngại vật	Xe của người chơi va chạm vào chướng ngại vật	Tốc độ của xe bị giảm xuống
Giới hạn tốc độ giảm	Xe đang di chuyển và gặp chướng ngại vật	Xe của người chơi bị va chạm vào và đã bị giảm tốc trước đó nhiều lần	Tốc độ của xe sẽ bị giảm tới 1 mức nhất định và không thể bị giảm nữa



#### 2.4.4. Tương tác với các hiệu ứng (tăng tốc/ giảm tốc)

##### 2.4.4.1, Tương tác với hiệu ứng tăng tốc

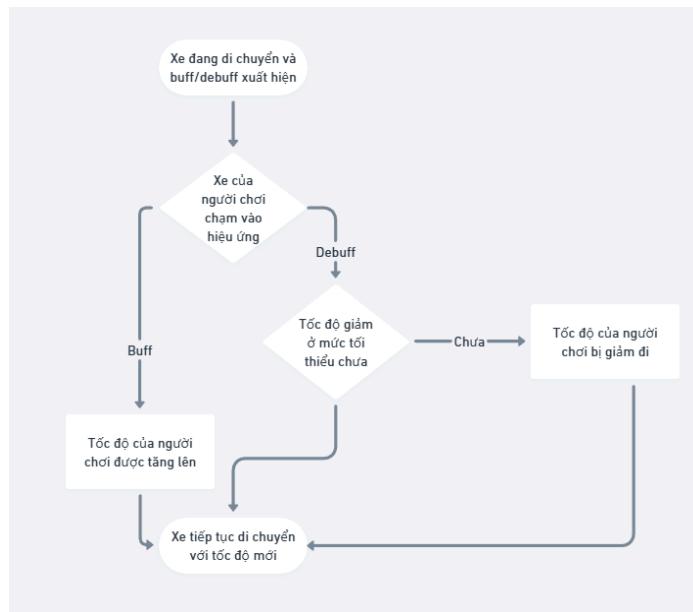
-Với tư cách là người chơi, khi xe của tôi va chạm hiệu ứng tăng tốc , tôi muốn tốc độ xe tăng lên, để có lợi thế vượt lên đối thủ.

Scenario	Given	When	Then
Ăn buff	Xe đang di chuyển và buff/xbuff xuất hiện	Xe của người chơi chạm vào buff	Tốc độ của người chơi sẽ được tăng lên

##### 2.4.4.2, Tương tác với hiệu ứng giảm tốc

-Với tư cách là người chơi, khi xe của tôi va chạm hiệu ứng giảm tốc, tôi muốn tốc độ xe giảm, để tăng độ khó cho game.

Scenario	Given	When	Then
Ăn debuff	Xe đang di chuyển và debuff/xdebuff xuất hiện	Xe của người chơi chạm vào debuff	Tốc độ của người chơi sẽ bị giảm đi



### 2.4.5, Xem bảng điểm

- Với tư cách là người chơi, sau khi game kết thúc, tôi muốn ấn “SCOREBOARD”, để xem lại điểm số và thứ hạng của mình trên toàn server.

Scenario	Given	When	Then
Xem bảng điểm	Trò chơi đã kết thúc và tôi đang ở màn hình game	Tôi nhấn nút "SCOREBOARD"	Một cửa sổ “Score Board” sẽ hiện lên và nó sẽ hiển thị thứ hạng và điểm của tất cả người chơi trên sever



## V. GIAO DIỆN CỦA ỨNG DỤNG

Form mở đầu game:



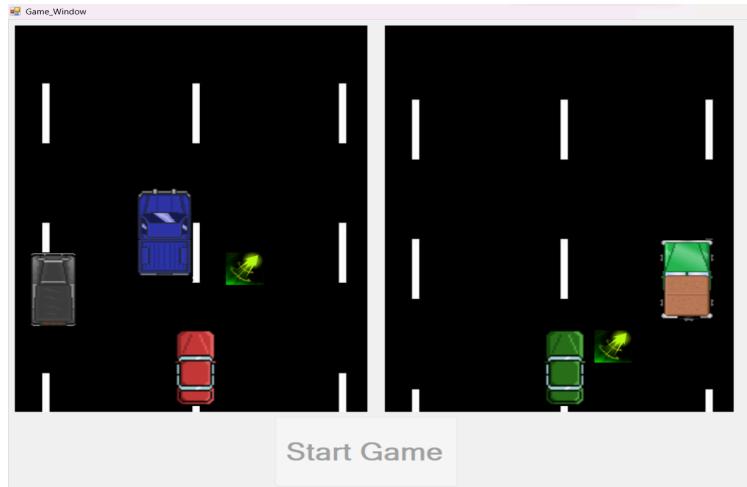
Form đăng kí:



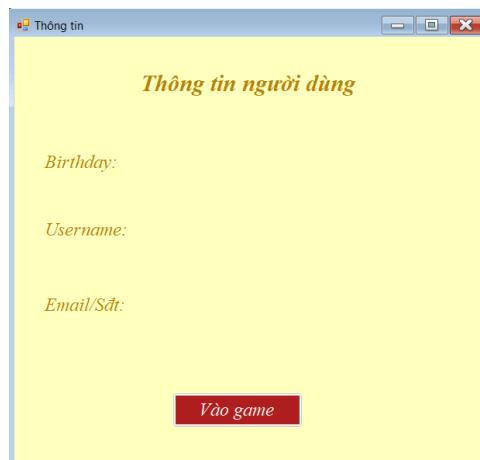
Form đăng nhập:



Form game play:



Form thông tin người dùng:



Form thành tích:

STT	Tên người chơi	Số trận thắng	Số lần va chạm	Tổng điểm
1	Nguyễn Văn A	5	2	1250.4
2	Phạm D	4	3	1150.7
3	Trần Thị B	3	4	980.2
4	Hoàng E	2	5	850.6
5	Lê C	1	6	730.9

Đóng

**Form quên mật khẩu:**



**Form đổi mật khẩu:**



## **VI. NHỮNG PHẦN KHÁC MÀ NHÓM ĐÃ HOÀN THÀNH CHO ĐẾN THỜI ĐIỂM BÁO CÁO**

Các phần khác đã được hoàn thành:

- Form đổi mật khẩu
- Form thành tích
- Form game play
- Form thông tin
- Form mở đầu
- Server TCP
- Form đăng nhập, đăng ký
- Tạo kết nối giữa client, server, database

## **VII. PHÂN CÔNG CÔNG VIỆC**

Tên thành viên	Nhiệm vụ	Đóng góp
Nguyễn Duy Hưng	1. Tổ chức họp nhóm & phân chia nhiệm vụ 2. Phát triển form đăng ký người chơi 3. Xây dựng server TCP cho trò chơi 4. Lập trình chức năng đường đua 5. Lập trình hệ thống buff (hiệu ứng tăng tốc, giảm tốc)	21.5%
Lê Tiến Hưng	1. Code client TCP – phần đăng nhập 2. Phát triển bảng thành tích người chơi 3. Kiểm thử và hoàn thiện ứng dụng	19.5%
Nguyễn Hoàng Khánh Đăng	1. Tạo kết nối đa luồng cho server 2. Code chướng ngại vật trong trò chơi 3. Code form thông tin người dùng 4. Code TCP client – phần thông tin người dùng	19.75%
Ngô Xuân Minh Hoàng	1. Lập trình xe player, logic chơi game 2. Cấu hình truy cập SQL Server qua mạng 3. Mã hóa mật khẩu (SHA-256)	20%
Hoàng Phi Hùng	1. Phát triển form đăng nhập cơ bản 2. Code client TCP – phần đăng ký 3. Thêm tính năng quên mật khẩu, Thêm tính năng khôi phục mật khẩu	19.25%