

TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT  
KHOA HỆ THÔNG THÔNG TIN



ĐỒ ÁN CUỐI KỲ MÔN KỸ THUẬT LẬP TRÌNH  
**PHẦN MỀM**  
**QUẢN LÝ VÀ BÁN VÉ BÓNG ĐÁ**

Mã lớp học phần: 242BIE501901

GVHD: Th.S. Nguyễn Quang Phúc

Nhóm 8

TP. Hồ Chí Minh, ngày 25 tháng 03 năm 2025

## DANH SÁCH THÀNH VIÊN

STT	Họ và tên	MSSV	Nhiệm vụ	Mức độ hoàn thành
1	Nguyễn Lê Nhật Khanh	K234060699	- Thiết kế FrontEnd “Add New Products”, “Product Update”  - Slide  - Bảng phân tích Usecase và Flowchart Admin + User	100%
2	Nguyễn Thị Bích Phượng	K234060720	- Xử lý database với MongoDB, các bảng dữ liệu và .env  - Thiết kế UI/UX “Customer’s Tickets”, “Edit match”  - Slide  - Tổng hợp word  - Sơ đồ hoạt động: “Login/Signup”, “Admin”	100%

3	Lê Lê Quyên	K234060721	<ul style="list-style-type: none"> <li>- Phát triển Frontend, Backend phần User :</li> <li>+ “Landing Page”,</li> <li>+ “My Tickets”</li> <li>+ “View Products”</li> <li>+ “Information Customer”</li> <li>+ “Select Section”</li>   <li>Admin:</li> <li>- “Dashboard”</li> <li>- Chính sửa FrontEnd “Add New Product” . “Update product” , “Sold Tickets Information”</li> <li>- File Readme</li> <li>- Format Word</li> <li>- Tổng hợp Source Code</li> </ul>	100%
4	Nguyễn Lê Thi	K234060728	<ul style="list-style-type: none"> <li>- Thiết kế giao diện UI/UX Figma</li> <li>- Slide</li> </ul>	100%
5	Trần Quang Vũ	K234060741	<ul style="list-style-type: none"> <li>- Phát triển Frontend, Backend phần “Login”,</li> </ul>	100%

			<p>“Sign up”, “Customers”, “Invoices”, “Products”</p> <p>- Tạo file “ Install_libs”, “requirements”</p> <p>- Bảng phân tích Usecase và Flowchart Admin</p>	
--	--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

## LỜI CẢM ƠN

Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Quang Phúc, giảng viên môn Kỹ thuật lập trình, người đã tận tình hướng dẫn, chỉ bảo và hỗ trợ chúng em trong suốt quá trình thực hiện đồ án này. Những kiến thức và kinh nghiệm quý báu mà thầy đã truyền đạt không chỉ giúp chúng em hoàn thành phần mềm “Quản lý và bán vé bóng đá Ngoại hạng Anh” mà còn là nền tảng quan trọng cho quá trình học tập và phát triển sau này.

Tất cả thành viên đã cùng nhau cố gắng hết sức, nhưng chắc chắn đồ án vẫn còn nhiều thiếu sót. Chúng em mong nhận được những ý kiến đóng góp, nhận xét từ thầy để có thể hoàn thiện hơn trong tương lai. Và chắc chắn đồ án này đã giúp chúng em học tập và tiếp thu được rất nhiều kiến thức, kinh nghiệm thuận lợi cho những cơ hội trong tương lai.

Một lần nữa, tất cả thành viên Nhóm 8 xin gửi đến thầy lời cảm ơn chân thành nhất.

Nhóm 8

## LỜI CAM KẾT

Tất cả thành viên nhóm 8 xin cam kết rằng đồ án này là kết quả nghiên cứu và làm việc của nhóm. Các nội dung được trình bày trong đồ án đều dựa trên kiến thức trong quá trình học tập, các nguồn tài liệu có trích dẫn rõ ràng và các kết quả do các thành viên tự thực hiện.

Nhóm đảm bảo không sao chép, sử dụng trái phép bất kỳ phần nào của người khác mà không được cho phép hoặc không ghi rõ nguồn gốc. Nếu có bất kỳ sai phạm nào, tất cả thành viên xin chịu hoàn toàn trách nhiệm trước quy định của thầy.

Nhóm 8

## MỤC LỤC

<b>DANH SÁCH THÀNH VIÊN.....</b>	<b>1</b>
<b>LỜI CẢM ƠN .....</b>	<b>4</b>
<b>LỜI CAM KẾT .....</b>	<b>5</b>
<b>MỤC LỤC .....</b>	<b>6</b>
<b>DANH MỤC BẢNG BIỂU .....</b>	<b>8</b>
<b>DANH MỤC HÌNH ẢNH .....</b>	<b>9</b>
<b>CHƯƠNG I: TỔNG QUAN ĐỀ TÀI.....</b>	<b>11</b>
1.1. Lý do chọn đề tài.....	11
1.2. Mục tiêu nghiên cứu.....	11
1.3. Yêu cầu chức năng.....	12
1.3.1. Đôi với Admin. ....	12
1.3.2. Đôi với User. ....	12
1.4. Yêu cầu phi chức năng.....	12
<b>CHƯƠNG II: CƠ SỞ LÝ THUYẾT.....</b>	<b>14</b>
2.1. Cấu trúc dữ liệu trong Python.....	14
2.1.1. Giới thiệu. ....	14
2.1.2. Các kiểu cấu trúc dữ liệu cơ bản.....	14
2.2. Lập trình hướng đối tượng (OOP).....	15
2.2.1. Giới thiệu. ....	15
2.2.2. Khái niệm. ....	15
2.2.3. Thành phần cơ bản. ....	15
2.2.4. Các đặc tính cơ bản của OOP. ....	16
2.3. Xây dựng ứng dụng GUI với Tkinter. ....	17
2.3.1. Mô tả. ....	17
2.3.2. Ưu điểm.....	18
2.3.3. Nhược điểm.....	18
2.4. Thư viện trong Python. ....	19
2.4.1. Pillow. ....	19
2.4.2. Tkinter. ....	19
2.4.3. Pathlib. ....	20
2.4.4. Matplotlib.....	20
2.5. Hệ quản trị CSDL NoSQL MongoDB. ....	21
2.5.1. Mô tả. ....	21
2.5.2. Cấu trúc tổ chức dữ liệu của MongoDB. ....	22
2.5.3. Ưu điểm và nhược điểm của MongoDB. ....	23
<b>CHƯƠNG III: CÔNG CỤ QUẢN LÝ.....</b>	<b>24</b>
3.1. Google Drive.....	24
3.2. Github. ....	25

<b>CHƯƠNG IV: PHÂN TÍCH CHƯƠNG TRÌNH .....</b>	<b>27</b>
<b>4.1. Các tác nhân của hệ thống. ....</b>	<b>27</b>
4.1.1. Admin (Quản trị viên).....	27
4.1.2. User (Người dùng/Khách hàng).....	27
<b>4.2. Các quy trình nghiệp vụ chính. ....</b>	<b>27</b>
<b>4.3. Cấu trúc tổ chức mã nguồn.....</b>	<b>29</b>
<b>CHƯƠNG V: THIẾT KẾ HỆ THỐNG .....</b>	<b>33</b>
<b>5.1. Sơ đồ hoạt động.....</b>	<b>33</b>
<b>5.2. Thiết kế các mô hình xử lý – Use case.....</b>	<b>36</b>
5.2.1. Danh sách Use case.....	37
5.2.2. Mô tả Use case. ....	37
<b>5.3. Các bảng dữ liệu: .....</b>	<b>68</b>
<b>CHƯƠNG VI: THIẾT KẾ GIAO DIỆN.....</b>	<b>71</b>
<b>6.1. Tổng quan về Figma. ....</b>	<b>71</b>
6.1.1. Tính năng chính của Figma.....	71
6.1.2. Ưu điểm khi sử dụng Figma .....	72
<b>6.2. Giao diện Đăng nhập, Đăng ký.....</b>	<b>73</b>
<b>6.3. Giao diện người dùng (User UI). ....</b>	<b>75</b>
<b>6.4. Giao diện quản trị (Admin UI) .....</b>	<b>81</b>
<b>CHƯƠNG VII: KẾT QUẢ CHƯƠNG TRÌNH.....</b>	<b>85</b>
<b>7.1. Đăng nhập vai trò Admin: .....</b>	<b>85</b>
<b>7.2. Đăng nhập với vai trò User .....</b>	<b>95</b>
<b>7.3. Kết quả đạt được.....</b>	<b>104</b>
<b>7.4. Hạn chế .....</b>	<b>104</b>
<b>7.5. Hướng phát triển .....</b>	<b>104</b>
<b>CHƯƠNG VIII: KẾT LUẬN .....</b>	<b>106</b>
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>107</b>

## DANH MỤC BẢNG BIỂU

<i>Bảng 1. Use case Login .....</i>	38
<i>Bảng 2. Use case Sign up .....</i>	40
<i>Bảng 3. Use case Log out .....</i>	43
<i>Bảng 4. Use case Contact.....</i>	45
<i>Bảng 5. Use case Information Customer.....</i>	47
<i>Bảng 6. Use case Purchase History .....</i>	49
<i>Bảng 7. Use case View Product.....</i>	51
<i>Bảng 8. Use case Select Section .....</i>	53
<i>Bảng 9. Use case Payment.....</i>	55
<i>Bảng 10. Use case Dashboard.....</i>	57
<i>Bảng 11. Use case Invoices .....</i>	59
<i>Bảng 12. Use case Products .....</i>	61
<i>Bảng 13. Use case Add New Product .....</i>	64
<i>Bảng 14. Use case Customers .....</i>	66
<i>Bảng 15. Dữ liệu Admin .....</i>	68
<i>Bảng 16. Dữ liệu Users .....</i>	68
<i>Bảng 17. Dữ liệu Teams .....</i>	69

## DANH MỤC HÌNH ẢNH

Hình 1. MongoDB .....	22
Hình 2. Google Drive.....	24
Hình 3. Github .....	25
Hình 4. Cấu trúc tổ chức mã nguồn của phần mềm .....	29
Hình 5. Cấu trúc tổ chức thư mục Data .....	29
Hình 6. Cấu trúc tổ chức thư mục Images .....	30
Hình 7. Cấu trúc tổ chức thư mục Modules .....	30
Hình 8. Sơ đồ hoạt động khi Đăng nhập .....	33
Hình 9. Sơ đồ hoạt động Admin.....	34
Hình 10. Sơ đồ hoạt động khi Mua hàng của User .....	35
Hình 11. Mô hình Use case.....	36
Hình 12. Flowchart cho Use case "Login .....	39
Hình 13. Flowchart cho Use case "Sign up " .....	42
Hình 14. Flowchart cho Use case "Log out.....	44
Hình 15. Flowchart cho Use case "Contact .....	46
Hình 16. Flowchart cho Use case "Information Customer" .....	48
Hình 17. Flowchart cho Use case "Purchase History" .....	50
Hình 18. Flowchart cho Use case "View Product.....	52
Hình 19. Flowchart cho Use case "Select Section.....	54
Hình 20. Flowchart cho Use case "Payment.....	56
Hình 21. Flowchart cho Use case "Dashboard.....	58
Hình 22. Flowchart cho Use case "Invoices.....	60
Hình 23. Flowchart cho Use case "Products .....	64
Hình 24. Flowchart cho Use case "Add New Product .....	65
Hình 25. Flowchart cho Use case "Customers" .....	67
Hình 26. Ứng dụng Figma .....	71
Hình 27. Admin đăng nhập .....	73
Hình 28. User đăng nhập .. ....	74
Hình 29. Trang chủ .....	75
Hình 30. User xem các trận đấu .....	76

<i>Hình 31. User lựa chọn vé .....</i>	77
<i>Hình 32. User thanh toán hóa đơn .....</i>	78
<i>Hình 33. Lịch sử mua vé của User.....</i>	79
<i>Hình 34. User chưa mua vé nào .....</i>	80
<i>Hình 35. Admin xem tổng quan phần mềm .....</i>	81
<i>Hình 36. Admin xem các trận đấu .....</i>	82
<i>Hình 37. Admin xem các hóa đơn.....</i>	83
<i>Hình 38. Admin xem danh sách khách hàng .....</i>	84
<i>Hình 39. Giao diện đăng nhập của Admin .....</i>	85
<i>Hình 40: Giao diện Dashboard .....</i>	86
<i>Hình 41: Giao diện Invoices .....</i>	87
<i>Hình 42: Pop up Invoices .....</i>	88
<i>Hình 43: Giao diện Products .....</i>	89
<i>Hình 44: Pop up Product Update .....</i>	90
<i>Hình 45: Giao diện Add New Products .....</i>	91
<i>Hình 46: Giao diện Customers .....</i>	92
<i>Hình 47: Pop up Customers .....</i>	93
<i>Hình 48: Giao diện Log out.....</i>	94
<i>Hình 49: Giao diện Sign in User .....</i>	95
<i>Hình 50: Giao diện Sign up .....</i>	96
<i>Hình 51: Giao diện Landing page .....</i>	97
<i>Hình 52: Giao diện My Tickets.....</i>	98
<i>Hình 53: Giao diện Information Customer .....</i>	99
<i>Hình 54: Giao diện View Product .....</i>	100
<i>Hình 55: Giao diện Select Section .....</i>	101
<i>Hình 56: Giao diện Payment .....</i>	102
<i>Hình 57: Giao diện khi Log out .....</i>	103

# **CHƯƠNG I: TỔNG QUAN ĐỀ TÀI**

## **ĐỀ TÀI: PHẦN MỀM QUẢN LÝ VÀ BÁN VÉ BÓNG ĐÁ**

### **1.1. Lý do chọn đề tài.**

Bóng đá là một trong những môn thể thao phổ biến và được yêu thích nhất trên thế giới, thu hút hàng triệu người hâm mộ đến các sân vận động để theo dõi các trận đấu. Tuy nhiên, quá trình mua vé xem bóng đá hiện nay vẫn còn tồn tại nhiều bất cập như mất thời gian xếp hàng tại quầy vé, nguy cơ mua trúng vé giả, khó kiểm soát số lượng vé đã bán, vé còn lại cũng như thông tin khán giả. Điều đó sẽ ảnh hưởng đến trải nghiệm của người hâm mộ và cũng như xuất hiện nhiều hạn chế trong công tác của người quản lý.

Với sự phát triển của công nghệ, đặc biệt là các ứng dụng trên nền tảng máy tính (Desktop application), việc số hóa quy trình bán vé là một giải pháp cần thiết để tối ưu hóa trải nghiệm người dùng và giúp các đơn vị tổ chức quản lý hiệu quả hơn. Phần mềm bán vé bóng đá trên desktop sẽ giúp khách hàng dễ dàng tra cứu thông tin trận đấu, chọn chỗ ngồi, thanh toán trực tuyến và nhận vé điện tử mà không cần đến trực tiếp quầy vé. Đồng thời, phần mềm cũng hỗ trợ ban tổ chức trong việc kiểm soát số lượng vé, thống kê doanh thu và quản lý thông tin khách hàng một cách chính xác.

Xuất phát từ những lý do trên, nhóm chúng em quyết định thực hiện đề tài “Phần mềm quản lý và bán vé bóng đá” nhằm xây dựng một hệ thống bán vé hiện đại, thuận tiện và hiệu quả hơn, góp phần nâng cao trải nghiệm người dùng và hỗ trợ các đơn vị tổ chức thể thao trong công tác quản lý vé.

### **1.2. Mục tiêu nghiên cứu.**

“Phần mềm quản lý và bán vé bóng đá” được nhóm thực hiện hướng đến việc hỗ trợ người dùng trong việc mua, bán và quản lý vé bóng đá một cách nhanh chóng, tiện lợi và tối ưu nhất. Ứng dụng sẽ giúp người mua dễ dàng tra cứu thông tin trận đấu, lựa chọn chỗ ngồi và đặt vé một cách thuận tiện ngay trên máy tính mà không cần đến quầy vé truyền thống. Đồng thời, hệ thống cũng hỗ trợ người quản lý kiểm soát, theo dõi các hoạt động mua bán, tối ưu hóa công tác quản lý vé.

Trong tương lai, nhóm sẽ tiếp tục nghiên cứu và hướng đến những tính năng vượt trội hơn như phát triển phiên bản web hoặc mobile, tạo sự thuận tiện nhất cho người dùng. Bên cạnh đó, tận dụng sự phát triển của công nghệ hiện nay, nhóm cũng mong muốn có thể tích hợp thanh toán trực tuyến thông qua các ví điện tử, ngân hàng; quản lý thông tin người dùng bằng CCCD để tăng cường tính bảo mật và đồng bộ dữ liệu; tích hợp xử lý thời gian thực và tự động đổi với tình trạng vé để đảm bảo vé chỉ được bán trong thời gian quy định.

### **1.3. Yêu cầu chức năng.**

#### *1.3.1. Đối với Admin.*

Admin có quyền truy cập và quản lý toàn bộ hệ thống bao gồm các chức năng sau:

- Quản lý doanh thu
- Quản lý sản phẩm
- Quản lý hóa đơn
- Quản lý khách hàng
- Chăm sóc khách hàng

#### *1.3.2. Đối với User.*

User có quyền truy cập và thực hiện các chức năng thuộc phân quyền của mình:

- Xem tài khoản cá nhân
- Xem thông tin trận đấu và mua vé
- Xem đơn hàng đã đặt
- Yêu cầu liên hệ hỗ trợ

### **1.4. Yêu cầu phi chức năng.**

Bên cạnh các yêu cầu chức năng, phần mềm bán vé bóng đá cần đáp ứng các yêu cầu phi chức năng nhằm đảm bảo hiệu suất, bảo mật và trải nghiệm người dùng tốt nhất.

- Ứng dụng phải có tốc độ tải nhanh, phản hồi thao tác người dùng theo thời gian thực.
- Hệ thống phải được thiết kế linh hoạt để có thể dễ dàng mở rộng khi số lượng người dùng và giao dịch tăng lên.

- Các dữ liệu người dùng phải được bảo mật, an toàn.
- Nâng cao hiệu quả hoạt động, quản lý và trải nghiệm người dùng.

## CHƯƠNG II: CƠ SỞ LÝ THUYẾT

### 2.1. Cấu trúc dữ liệu trong Python.

#### 2.1.1. Giới thiệu.

Cấu trúc dữ liệu (Data Structures) là cách tổ chức và lưu trữ dữ liệu để tối ưu hóa việc truy cập và thao tác. Python cung cấp nhiều cấu trúc dữ liệu tích hợp sẵn, đồng thời hỗ trợ triển khai các cấu trúc phức tạp hơn thông qua thư viện hoặc lớp tự định nghĩa.

#### 2.1.2. Các kiểu cấu trúc dữ liệu cơ bản.

##### 2.1.2.1. List.

Khái niệm: là một kiểu dữ liệu tập hợp (collection) dùng để lưu trữ các phần tử thuộc nhiều kiểu dữ liệu khác nhau, có thứ tự và có thể thay đổi (mutable).

Phương thức: Để khai báo một list trong Python thì chúng ta sử dụng cặp dấu [] và bên trong là các giá trị của list: [value 1, value 2,..., value n]

Trong đó: ‘value 1, value 2,..., value n’ là các giá trị của list.

##### 2.1.2.2. Tuple.

Khái niệm: là một kiểu dữ liệu tập hợp giống với List nhưng mà giá trị của các phần tử không được phép thay đổi (immutable).

Phương thức: Để khai báo một tuple thì chúng ta sử dụng cặp dấu () và bên trong là các giá trị với cú pháp sau: (value 1, value 2,..., value n), trong đó: ‘value 1, value 2,..., value n’ là các giá trị của tuple.

##### 2.1.2.3. Set.

Khái niệm: là một kiểu dữ liệu tập hợp lưu trữ các phần tử duy nhất (không trùng lặp), không có thứ tự và có thể thay đổi (mutable). Các phần tử có thể là số, chuỗi, tuple nhưng không thể chứa list.

Phương thức: Để khai báo một set thì chúng ta sử dụng cặp dấu {} và bên trong là các giá trị với cú pháp sau: {value 1, value 2,..., value n}, trong đó: ‘value 1, value 2,..., value n’ là các giá trị của set.

#### 2.1.2.4. Dictionary.

Khái niệm: là một collection không có thứ tự, mỗi phần tử được định nghĩa bởi cặp khóa và giá trị (key:value), với điều kiện key là không trùng nhau.

Phương thức: Để khai báo một dictionary chúng ta sử dụng cặp dấu {} theo cú pháp sau: {key 1: value 1, key 2: value 2,..., key n: value n}. Trong đó: "key 1:value 1, key 2: value 2,..., key n: value n" là các khóa và giá trị của kiểu dữ liệu dictionary.

### 2.2. Lập trình hướng đối tượng (OOP).

#### 2.2.1. Giới thiệu.

Lập trình hướng đối tượng (Object-Oriented Programming - OOP) là một trong những mô hình lập trình mạnh mẽ và phổ biến nhất hiện nay. Mô hình này đưa ra phương pháp để xây dựng ứng dụng bằng cách mô phỏng các đối tượng trong thực tiễn, giúp lập trình viên quản lý code, tái sử dụng và mở rộng dễ dàng hơn.

#### 2.2.2. Khái niệm.

Lập trình hướng đối tượng (OOP) là mô hình lập trình dựa trên khái niệm "đối tượng". Trái với mô hình lập trình thủ tục, OOP tập trung vào việc mô tả các đối tượng qua dữ liệu (thuộc tính) và hành vi (phương thức). Mỗi đối tượng có thể coi là một đơn vị độc lập, nơi dữ liệu và các chức năng xử lý dữ liệu gắn kết chặt chẽ với nhau.

Lợi ích của OOP nhấn mạnh đến:

- Sự rõ ràng: Code được tổ chức thành các đối tượng chuyên biệt, giúp dễ hiểu và bảo trì.
- Tái sử dụng: Các lớp có thể được sử dụng lại trong nhiều chương trình khác nhau.
- Khả năng mở rộng: Cấu trúc phân chia rõ ràng giúp dễ dàng thêm mới hoặc thay đổi chức năng khi cần.

#### 2.2.3. Thành phần cơ bản.

Trong lập trình hướng đối tượng, một số thành phần cơ bản được định nghĩa như sau:

- Lớp (Class): Là khuôn mẫu hay bản thiết kế định nghĩa các đặc tính (thuộc tính) và hành động (phương thức) chung cho những đối tượng cùng loại. Lớp giúp trừu tượng hóa một nhóm đối tượng có tính chất chung.
- Đối tượng (Object): Là thể hiện cụ thể của một lớp. Mỗi đối tượng có trạng thái riêng được biểu thị qua các giá trị của thuộc tính và có thể thực hiện các hành động thông qua phương thức đã định nghĩa trong lớp.
- Thuộc tính (Attributes): Là các biến lưu trữ trạng thái của đối tượng. Thuộc tính giúp mô tả tình trạng hiện tại của một đối tượng, ví dụ như màu sắc, kích thước, tên, v.v.
- Phương thức (Methods): Là các hàm, nhiệm vụ hoặc hành động mà đối tượng có thể thực hiện. Các phương thức định nghĩa cách mà đối tượng tương tác với dữ liệu của nó và với các đối tượng khác.

Ngoài ra, trong một số ngôn ngữ OOP, các thành phần phụ trợ khác có thể được sử dụng như:

- Constructor/Destructor: Các phương thức đặc biệt được gọi khi tạo hoặc hủy đối tượng.
- Interface và Abstract Class: Dùng để định nghĩa các giao diện và khuôn mẫu trừu tượng nhằm đảm bảo tính đồng nhất giữa các lớp kế thừa.

#### *2.2.4. Các đặc tính cơ bản của OOP.*

OOP được xây dựng dựa trên bốn đặc tính cơ bản sau đây:

##### *2.2.4.1. Tính Đóng gói (Encapsulation).*

**Định nghĩa:** Là quá trình nhóm các dữ liệu (thuộc tính) và các phương thức liên quan lại với nhau trong một đối tượng, đồng thời ẩn đi các chi tiết thực thi bên trong.

**Lợi ích:** Giúp bảo vệ dữ liệu khỏi sự truy cập và thay đổi ngoài ý muốn, tạo nên "hộp đen" cho đối tượng.

##### *2.2.4.2. Tính Kế thừa (Inheritance).*

**Định nghĩa:** Cho phép một lớp (lớp con) kế thừa các thuộc tính và phương thức của một lớp khác (lớp cha).

Lợi ích: Giúp tái sử dụng code, giảm thiểu sự trùng lặp, đồng thời tạo mối quan hệ phân cấp rõ ràng giữa các lớp.

#### 2.2.4.3. Tính Đa hình (Polymorphism).

Định nghĩa: Cho phép các đối tượng khác nhau có thể thực hiện cùng một hành động theo cách khác nhau. Điều này thể hiện qua việc một phương thức có thể có nhiều dạng triển khai khác nhau tùy thuộc vào đối tượng thực thi.

Lợi ích: Gia tăng tính linh hoạt và khả năng mở rộng của chương trình, cho phép xử lý đối tượng theo cách tổng quát mà vẫn duy trì sự khác biệt về hành vi khi cần thiết.

#### 2.2.4.4. Tính Trừu tượng (Abstraction).

Định nghĩa: Là khả năng tập trung vào những đặc tính chính của đối tượng, bỏ qua những chi tiết không cần thiết.

Lợi ích: Đơn giản hóa quá trình thiết kế và giao tiếp giữa các phần của chương trình bằng cách chỉ hiển thị những thông tin cần thiết và ẩn đi những chi tiết phức tạp.

#### 2.2.4.5. Ứng Dụng và Tác Động của OOP.

OOP đã được ứng dụng rộng rãi trong nhiều lĩnh vực của công nghệ thông tin như:

- Phát triển phần mềm: Các hệ thống phần mềm lớn được xây dựng theo mô hình OOP giúp dễ dàng mở rộng và bảo trì.
- Phát triển web: Nhiều framework hiện đại (ví dụ: Django, Ruby on Rails) sử dụng các khái niệm OOP để quản lý và triển khai dự án.
- Phát triển game: Người phát triển game sử dụng OOP để mô hình hóa các thực thể trong trò chơi như nhân vật, vật phẩm, môi trường.
- Nhờ khả năng mô hình hóa tự nhiên của đối tượng thực tế, OOP giúp lập trình viên chuyển từ việc xử lý các thủ tục thành xây dựng các mô hình đối tượng sống động, tương tác với nhau theo cách trực quan.

### 2.3. Xây dựng ứng dụng GUI với Tkinter.

#### 2.3.1. Mô tả.

Tkinter là thư viện tiêu chuẩn được tích hợp trong Python, phục vụ cho việc phát triển giao diện đồ họa người dùng (Graphical User Interface - GUI). Thư viện này cung

cấp một nền tảng hiệu quả để xây dựng các ứng dụng có giao diện trực quan, hoạt động dựa trên mô hình lập trình hướng sự kiện. Với Tkinter, người dùng có thể sử dụng các thành phần giao diện (widget) như Button (nút bấm), Label (nhãn), Entry (ô nhập liệu), Menu (thanh menu), Canvas (vùng vẽ) cùng nhiều công cụ khác để tạo ra các ứng dụng GUI đa dạng. Một trong những đặc điểm nổi bật của Tkinter là tính tích hợp sẵn có trong Python, giúp lập trình viên không cần cài đặt thêm phần mềm hỗ trợ. Bên cạnh đó, cú pháp của Tkinter được thiết kế đơn giản, dễ tiếp cận, đồng thời đảm bảo khả năng tương thích trên nhiều hệ điều hành phổ biến như Windows, macOS và Linux. Nhờ những đặc tính này, Tkinter trở thành lựa chọn phổ biến cho các dự án phát triển ứng dụng GUI cơ bản.

### *2.3.2. Ưu điểm.*

Tkinter sở hữu nhiều ưu điểm vượt trội, đặc biệt phù hợp với nhu cầu phát triển ứng dụng GUI ở mức độ đơn giản đến trung bình. Trước hết, tính tích hợp sẵn trong Python giúp loại bỏ nhu cầu cài đặt thư viện bên ngoài, từ đó tiết kiệm thời gian và giảm thiểu sự phụ thuộc vào các yếu tố bên thứ ba. Thứ hai, cú pháp của Tkinter được thiết kế tối giản và dễ hiểu, tạo điều kiện thuận lợi cho người mới bắt đầu làm quen với lập trình giao diện đồ họa. Thứ ba, khả năng hoạt động đa nền tảng cho phép ứng dụng phát triển bằng Tkinter chạy mượt mà trên các hệ điều hành khác nhau mà không cần điều chỉnh nhiều mã nguồn. Ngoài ra, thư viện này cung cấp một tập hợp widget phong phú, đáp ứng tốt nhu cầu thiết kế các giao diện cơ bản. Cuối cùng, Tkinter nổi bật với tính nhẹ và hiệu suất cao, tiêu tốn ít tài nguyên hệ thống, khiến nó trở thành lựa chọn lý tưởng cho các ứng dụng không đòi hỏi quá nhiều sức mạnh xử lý.

### *2.3.3. Nhược điểm.*

Mặc dù có nhiều ưu điểm, Tkinter cũng tồn tại một số hạn chế đáng kể. Trước hết, giao diện mặc định của Tkinter thường mang phong cách đơn giản và có phần lỗi thời, không đáp ứng được nhu cầu thẩm mỹ của các ứng dụng hiện đại so với những thư viện như Qt hay Kivy. Thứ hai, khả năng tùy chỉnh giao diện của Tkinter khá hạn chế, khiến việc phát triển các ứng dụng có thiết kế phức tạp hoặc độc đáo trở nên khó khăn. Thứ ba, về mặt hiệu suất, Tkinter không được tối ưu cho các dự án lớn hoặc những ứng dụng đòi hỏi xử lý giao diện đồ họa nặng, dẫn đến tốc độ xử lý có thể bị giảm trong những

trường hợp này. Cuối cùng, tài liệu hỗ trợ của Tkinter, mặc dù có sẵn từ nguồn chính thức, nhưng chưa thực sự phong phú về ví dụ thực tế và hướng dẫn chi tiết, điều này có thể gây khó khăn cho lập trình viên khi gặp vấn đề phức tạp. Những nhược điểm này khiến Tkinter thường bị giới hạn trong phạm vi các ứng dụng đơn giản hoặc mang tính thử nghiệm.

## 2.4. Thư viện trong Python.

### 2.4.1. Pillow.

#### 2.4.1.1. Khái niệm.

Trong Python, Pillow là một thư viện mạnh mẽ và phổ biến dùng để xử lý hình ảnh. Nhìn chung, Pillow là một phần của PIL- Python Image Library, nhưng sau đó nó được cải tiến để hỗ trợ các phiên bản Python hiện đại (bao gồm Python 3.x) và bổ sung nhiều tính năng mới. Pillow cho phép ta thực hiện các tác vụ như mở, chỉnh sửa, lưu, và xử lý hình ảnh với nhiều định dạng khác nhau một cách dễ dàng.

#### 2.4.1.2. Chức năng.

Chức năng chính của nó bao gồm thay đổi kích thước, cắt, xoay hình ảnh, thêm văn bản hoặc hình vẽ, áp dụng bộ lọc như làm mờ hay tăng độ nét, và điều chỉnh màu sắc. Ngoài ra, Pillow hỗ trợ tạo ảnh động GIF và tích hợp với các thư viện khác như numpy. Với giao diện đơn giản và khả năng hoạt động đa nền tảng, Pillow mang lại tiện ích lớn trong việc xử lý ảnh nhanh chóng, từ tạo thumbnail, thêm watermark, đến phát triển ứng dụng GUI, đáp ứng tốt nhu cầu lập trình liên quan đến hình ảnh.

### 2.4.2. Tkinter.

#### 2.4.2.1. Khái niệm.

Tkinter là thư viện chuẩn của Python dùng để phát triển giao diện đồ họa người dùng (GUI), dựa trên bộ công cụ Tk (Toolkit) được phát triển ban đầu cho ngôn ngữ lập trình Tcl. Tkinter cung cấp một cách đơn giản để tạo ra các ứng dụng có giao diện trực quan, nơi người dùng có thể tương tác thông qua các thành phần như cửa sổ, nút bấm, ô nhập liệu, hoặc nhãn mà không cần viết mã phức tạp. Nó hoạt động bằng cách tạo một vòng lặp sự kiện để quản lý các hành động của người dùng, chẳng hạn như nhấp chuột hoặc nhập dữ liệu, và liên kết chúng với các hàm xử lý trong chương trình. Với

tính chất tích hợp sẵn trong Python, Tkinter không yêu cầu cài đặt bổ sung và được thiết kế để dễ học

#### 2.4.2.2. Chức năng.

Chức năng chính của Tkinter bao gồm tạo các cửa sổ, nút bấm, ô nhập liệu, nhãn, menu, và các thành phần giao diện khác, giúp người dùng tương tác với chương trình thông qua các yếu tố trực quan. Tkinter cho phép tùy chỉnh giao diện như thay đổi kích thước, màu sắc, vị trí của các thành phần, đồng thời hỗ trợ xử lý sự kiện như nhấp chuột hoặc nhập bàn phím.

#### 2.4.3. Pathlib.

##### 2.4.3.1. Khái niệm.

Pathlib là một module trong thư viện chuẩn của Python được sử dụng nhằm đơn giản hóa việc xử lý đường dẫn tệp theo cách hướng đối tượng. Khái niệm cốt lõi của pathlib là thay thế cách thao tác đường dẫn truyền thống bằng chuỗi (qua module os.path) bằng các đối tượng Path, giúp mã nguồn trở nên dễ đọc và ít lỗi hơn.

##### 2.4.3.2. Chức năng.

Chức năng chính của Pathlib là đơn giản hóa xử lý đường dẫn, đảm bảo tính tương thích đa nền tảng (Windows, Linux, macOS), và giảm thiểu lỗi khi làm việc với hệ thống tệp, mang lại sự tiện lợi cho các tác vụ như quản lý dự án, xử lý tệp hàng loạt, hoặc xây dựng ứng dụng cần truy cập tệp. Nó cho phép ta thực hiện các tác vụ như tạo, nối, kiểm tra, hoặc truy cập tệp và thư mục mà không cần lo lắng về sự khác biệt giữa các hệ điều hành (như dấu gạch chéo “/” trên Linux hay “\” trên Windows).

#### 2.4.4. Matplotlib.

##### 2.4.4.1. Khái niệm.

Matplotlib là một thư viện trực quan hóa dữ liệu mã nguồn mở trong Python, được thiết kế để tạo ra các biểu đồ và đồ thị nhằm trình bày thông tin một cách trực quan. Nó hoạt động bằng cách sử dụng các đối tượng như hình vẽ (figure) và trục (axes) để xây dựng giao diện đồ thị và tích hợp chặt chẽ với các thư viện như numpy và pandas, giúp biến dữ liệu thô thành hình ảnh dễ hiểu, phục vụ cho phân tích và nghiên cứu.

#### 2.4.4.2. Chức năng.

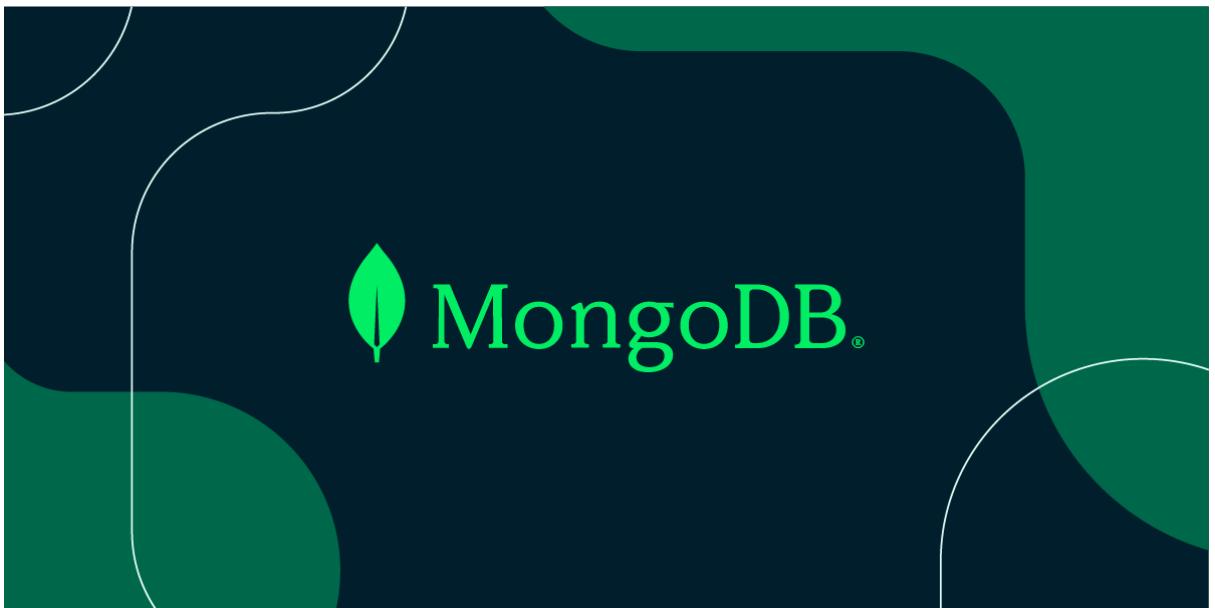
Matplotlib cung cấp nhiều chức năng để tạo và tùy chỉnh các biểu đồ dữ liệu. Nó cho phép vẽ các loại biểu đồ phổ biến như đồ thị đường, biểu đồ phân tán, biểu đồ cột, biểu đồ tròn, và thậm chí đồ thị 3D, với khả năng điều chỉnh màu sắc, kích thước, nhãn trực, tiêu đề, và lối. Người dùng có thể thêm chú thích, điều chỉnh tỷ lệ trực, hoặc áp dụng các kiểu định dạng để tăng tính thẩm mỹ. Thư viện hỗ trợ xuất biểu đồ sang nhiều định dạng như PNG, PDF, SVG để sử dụng trong báo cáo hoặc xuất bản. Ngoài ra, matplotlib cho phép xử lý dữ liệu lớn thông qua tích hợp với numpy, trực quan hóa bảng dữ liệu từ pandas, và tạo animation cho các ứng dụng động.

### 2.5. Hệ quản trị CSDL NoSQL MongoDB.

#### 2.5.1. Mô tả.

NoSQL (Not Only SQL) là một nhóm hệ quản trị cơ sở dữ liệu không sử dụng mô hình bảng như SQL mà sử dụng các mô hình dữ liệu linh hoạt hơn như document, key-value, wide-column và graph. NoSQL giúp lưu trữ dữ liệu linh hoạt, không cần schema cố định và hỗ trợ mở rộng dễ dàng và được ứng dụng rộng rãi trong Big Data, mạng xã hội, thương mại điện tử, IoT và game online.

Bên cạnh đó, MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL hướng document, lưu trữ dữ liệu dưới dạng BSON (Binary JSON) là một dạng mã hóa nhị phân của JSON. MongoDB không yêu cầu schema cố định, hỗ trợ index để tối ưu tìm kiếm, sharding để phân tán dữ liệu và replication để sao lưu dữ liệu. Nhờ những tính năng này, MongoDB được ứng dụng trong hệ thống dữ liệu lớn, website động, thương mại điện tử và các ứng dụng thời gian thực. MongoDB NoSQL giúp lưu trữ dữ liệu linh hoạt và phù hợp với các hệ thống cần mở rộng, thay đổi cấu trúc dữ liệu thường xuyên.



*Hình 1. MongoDB*

#### *2.5.2. Cấu trúc tổ chức dữ liệu của MongoDB.*

MongoDB tổ chức dữ liệu theo mô hình document-oriented, lưu trữ dữ liệu trong các document tương tự như các đối tượng JSON. Cấu trúc của các document trong cùng một collection không cần phải giống nhau, cho phép thêm hoặc xóa các trường dữ liệu một cách dễ dàng mà không cần phải định nghĩa trước schema.

Một MongoDB server có thể có nhiều database, trong mỗi database sẽ có nhiều collection và bên trong collection là các document. Cách xây dựng cấu trúc dữ liệu tương tự như bảng và hàng trong SQL nhưng không yêu cầu cấu trúc cố định, các dữ liệu có thể thường xuyên thay đổi và cập nhật.

Mỗi document là một tập hợp các cặp key-value, được lưu dưới dạng BSON (Binary JSON), giúp truy vấn nhanh hơn JSON thông thường. Các giá trị được định dạng với nhiều thể loại khác nhau, bao gồm như string, number, boolean, array hoặc thậm chí các đối tượng khác.

Trong đó, trường “`_id`” là một thành phần đặc biệt hơn các cặp key-value khác. Vì đây là trường dữ liệu được MongoDB tạo tự động mỗi khi một document được thiết lập. Mỗi document được gán sẵn giá trị cho trường “`_id`” và các giá trị này khác nhau, giúp phân biệt các document, hỗ trợ tìm và truy xuất đúng dữ liệu.

### *2.5.3. Ưu điểm và nhược điểm của MongoDB.*

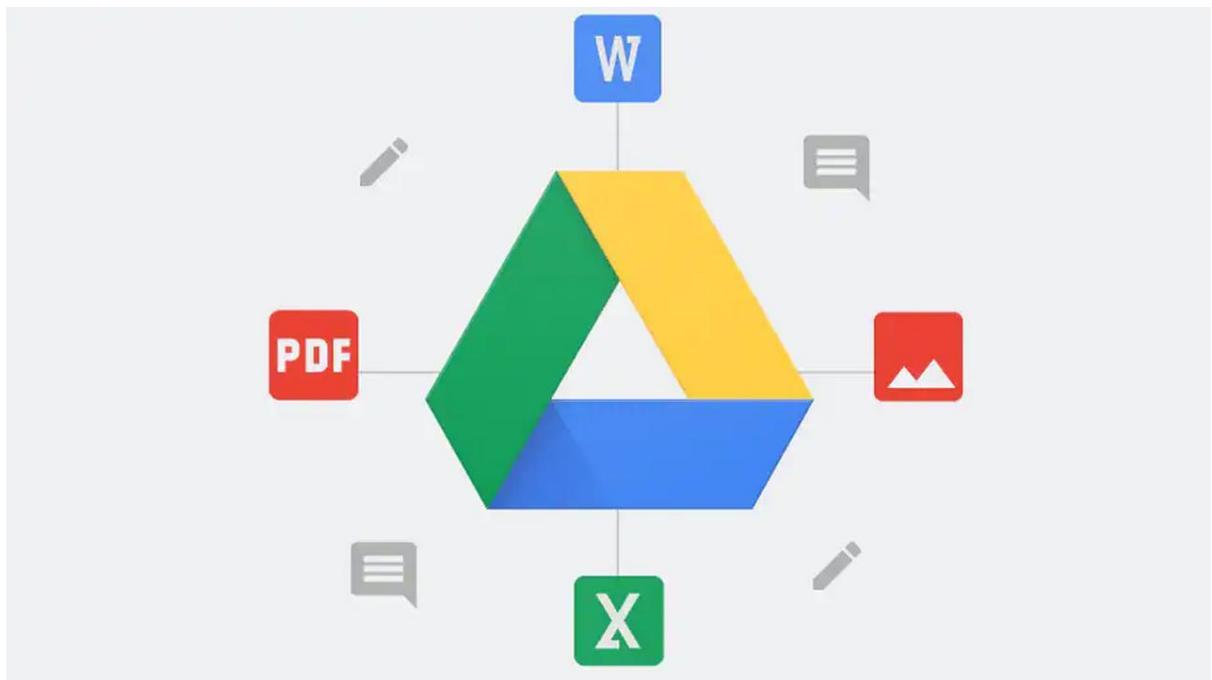
Cơ sở dữ liệu MongoDB có nhiều ưu điểm nổi bật như tính linh hoạt, không yêu cầu schema cố định, cho phép lưu trữ nhiều loại dữ liệu khác nhau. Hệ thống có khả năng mở rộng theo chiều ngang, giúp xử lý lượng truy cập lớn một cách hiệu quả. Nhờ lưu trữ dữ liệu theo dạng document, MongoDB có hiệu suất cao, truy vấn nhanh mà không cần thực hiện các phép join phức tạp như trong SQL. Ngoài ra, MongoDB còn dễ sử dụng, vì các document được tổ chức bao gồm các cặp key-value, có thể ánh xạ trực tiếp với các đối tượng trong nhiều ngôn ngữ lập trình và hỗ trợ đa dạng các ngôn ngữ lập trình phổ biến. MongoDB Compass là một công cụ GUI (giao diện đồ họa) chính thức của MongoDB, giúp người dùng quản lý, trực quan hóa và tương tác với cơ sở dữ liệu MongoDB một cách dễ dàng. Bên cạnh đó, MongoDB Atlats là nền tảng dịch vụ cơ sở dữ liệu đám mây, giúp người dùng truy cập và thao tác online với dữ liệu đám mây, cho phép nhiều người truy cập và dùng chung tài khoản, hỗ trợ làm việc nhóm hiệu quả. Hai hình thức hoạt động này có thể liên kết tài khoản và cập nhật dữ liệu nhanh chóng. Hơn nữa, MongoDB còn hỗ trợ người dùng với các tài liệu hướng dẫn chi tiết và bao quát kiến thức về cơ sở dữ liệu.

Tuy nhiên, MongoDB cũng có một số nhược điểm. Do sử dụng mô hình eventual consistency (tính nhất quán cuối cùng), dữ liệu giữa các node (nút) có thể không đồng bộ ngay lập tức, có nghĩa là có thể có một khoảng thời gian ngắn dữ liệu không nhất quán giữa các node. Dù NoSQL linh hoạt, việc mô hình hóa mối quan hệ có thể khó khăn và không hiệu quả bằng RDBMS, vì không có các phép join truyền thống. Mỗi quan hệ có thể không rõ ràng và cần thiết kế cẩn thận để tránh dư thừa dữ liệu. Bên cạnh đó, mặc dù MongoDB được tối ưu để mở rộng theo chiều ngang, nhưng quản lý một hệ thống phân tán vẫn phức tạp hơn so với SQL. Ngoài ra, người dùng quen với RDBMS có thể mất thời gian để làm quen với cách tổ chức dữ liệu của NoSQL. Hệ sinh thái NoSQL cũng đang phát triển nhưng chưa có mức độ hoàn thiện và hỗ trợ phong phú như RDBMS truyền thống.

## CHƯƠNG III: CÔNG CỤ QUẢN LÝ

Trong quá trình thực hiện đồ án, nhóm đã sử dụng các công cụ quản lý để hỗ trợ lưu trữ, chia sẻ tài liệu và quản lý mã nguồn. Hai công cụ chính được sử dụng là Google Drive và GitHub.

### 3.1. Google Drive.



*Hình 2. Google Drive*

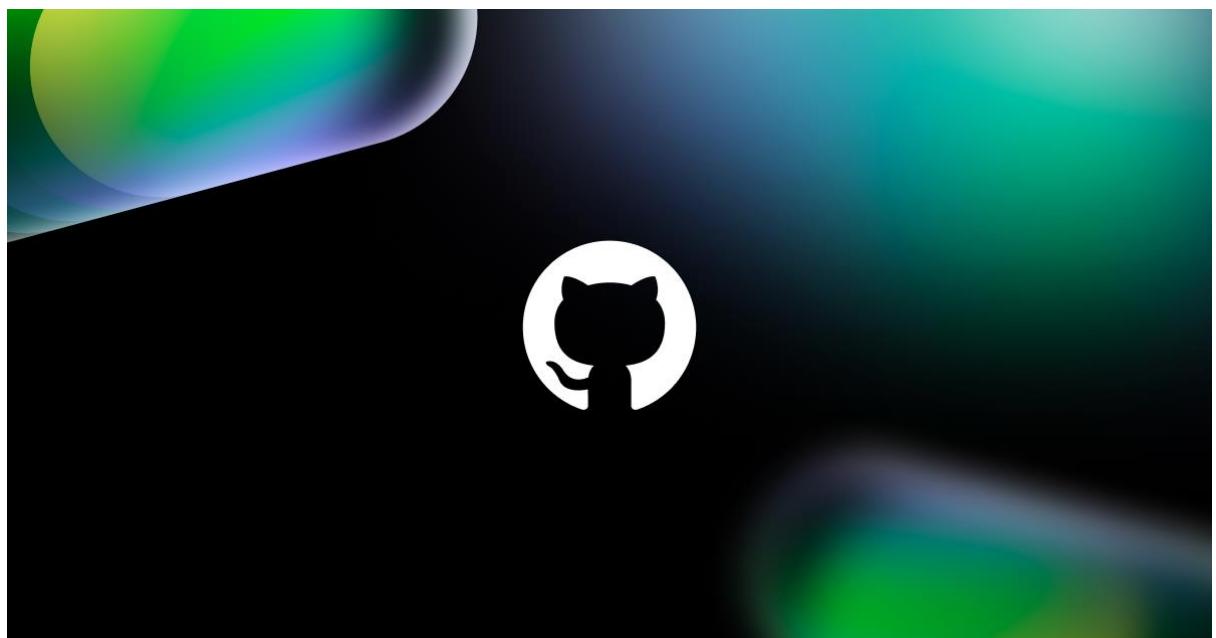
Google Drive là một dịch vụ lưu trữ đám mây do Google phát triển, cho phép người dùng lưu trữ tệp tin trực tuyến, truy cập từ nhiều thiết bị khác nhau và dễ dàng chia sẻ với các thành viên trong nhóm. Công cụ này đóng vai trò quan trọng trong việc quản lý tài liệu, giúp nhóm có thể cộng tác và cập nhật nội dung nhanh chóng mà không gặp rủi ro mất dữ liệu.

Lợi ích của Google Drive:

- Lưu trữ tập trung, truy cập dễ dàng: Các thành viên có thể truy cập tài liệu từ bất kỳ thiết bị nào, miễn là có kết nối internet. Điều này đặc biệt hữu ích khi làm việc từ xa hoặc cần truy cập dữ liệu trên nhiều thiết bị khác nhau.

- Chính sửa theo thời gian thực: Google Drive hỗ trợ các công cụ như Google Docs, Sheets, Slides, giúp nhóm có thể làm việc cùng nhau một cách linh hoạt mà không cần gửi file qua lại.
- Bảo mật và sao lưu tự động: Dữ liệu trên Google Drive được lưu trữ an toàn trên đám mây, hạn chế rủi ro mất dữ liệu do lỗi phần cứng hoặc lỗi cá nhân.
- Tích hợp với các công cụ khác: Google Drive có thể tích hợp với Gmail, Google Calendar và các công cụ quản lý dự án khác, giúp nhóm dễ dàng quản lý lịch trình và công việc.

### **3.2. Github.**



*Hình 3. Github*

GitHub là một nền tảng lưu trữ mã nguồn phổ biến, được xây dựng dựa trên hệ thống kiểm soát phiên bản Git. Công cụ này cho phép lập trình viên theo dõi sự thay đổi trong mã nguồn, cộng tác trong phát triển phần mềm và đảm bảo tính liên tục của dự án.

GitHub không chỉ là nơi lưu trữ mã nguồn, mà còn cung cấp nhiều tính năng hỗ trợ làm việc nhóm, như quản lý phiên bản, theo dõi lỗi, kiểm tra mã nguồn và triển khai phần mềm. Đây là công cụ không thể thiếu đối với các dự án lập trình, đặc biệt là các dự án có sự tham gia của nhiều thành viên.

## Lợi ích của GitHub

- Kiểm soát phiên bản hiệu quả: GitHub cho phép nhóm lưu lại lịch sử thay đổi của mã nguồn, giúp dễ dàng theo dõi và khắc phục lỗi nếu có vấn đề xảy ra.
- Hỗ trợ làm việc nhóm chuyên nghiệp: Các thành viên có thể làm việc trên các phần riêng của dự án mà không ảnh hưởng đến công việc của người khác. GitHub giúp đảm bảo rằng tất cả thay đổi đều được kiểm soát và xem xét kỹ trước khi hợp nhất vào mã nguồn chính.
- Tăng cường bảo mật mã nguồn: GitHub cung cấp các tính năng như phân quyền truy cập, giúp nhóm kiểm soát quyền hạn của từng thành viên và bảo vệ mã nguồn khỏi truy cập trái phép.
- Dễ dàng mở rộng và chia sẻ: Nếu muốn phát triển thêm dự án trong tương lai hoặc chia sẻ mã nguồn với cộng đồng, nhóm có thể dễ dàng sử dụng GitHub làm nền tảng phát triển lâu dài.

## CHƯƠNG IV: PHÂN TÍCH CHƯƠNG TRÌNH

### 4.1. Các tác nhân của hệ thống.

#### 4.1.1. Admin (*Quản trị viên*).

Admin có vai trò quan trọng trong việc quản lý hệ thống. Các chức năng chính của Admin bao gồm quản lý sản phẩm (vé trận đấu) với các thao tác xem, thêm, sửa, xóa vé. Admin cũng quản lý hóa đơn, có thể xem danh sách hóa đơn và kiểm tra trạng thái thanh toán. Bên cạnh đó, Admin còn quản lý khách hàng, xem danh sách khách hàng và liên hệ hỗ trợ qua fanpage Facebook. Để có cái nhìn tổng quan, Admin có thể xem Dashboard để theo dõi doanh số bán vé, số lượng vé còn lại và thống kê khách hàng. Cuối cùng, Admin có thể trực tiếp liên hệ với khách hàng để hỗ trợ thông qua fanpage.

#### 4.1.2. User (*Người dùng/Khách hàng*).

Người dùng hoặc khách hàng có thể tương tác với hệ thống để thực hiện các hành động liên quan đến việc mua vé. Các chức năng chính của User bao gồm xem danh sách đội bóng và trận đấu, chọn vé và số lượng mong muốn, tiến hành thanh toán vé và xem lại các vé đã mua. Ngoài ra, User cũng có thể xem thông tin cá nhân của mình trên hệ thống. Khi cần hỗ trợ, User có thể liên hệ thông qua fanpage.

### 4.2. Các quy trình nghiệp vụ chính.

Khi bắt đầu chương trình, Actor sẽ phải chọn vai trò đăng nhập và tiến hành nhập tài khoản/mật khẩu. Nếu đăng nhập thành công, chuyển hướng đến trang phù hợp với vai trò vừa chọn.

#### a. Admin:

Gồm 5 quy trình chính sau:

- Quy trình xem thống kê doanh thu: Tại trang “Dashboard” Admin có thể xem doanh thu vé theo từng đội bóng. Ngoài ra, doanh thu tổng cộng từ bán vé được hiển thị ở cuối trang.
- Quy trình quản lý sản phẩm: Vào trang “Products”, hệ thống hiển thị danh sách sản phẩm các trận đấu và loại vé, Admin có thể thực hiện các thao tác như xem danh sách các trận đấu, thêm/chỉnh sửa/xóa trận đấu

- Quy trình quản lý hóa đơn: Ở trang “Invoices”, hệ thống hiển thị danh sách hóa đơn đã được tạo, Admin có thể xem chi tiết hóa đơn bằng cách click vào hóa đơn để xem thông tin chi tiết.
- Quy trình quản lý khách hàng: Tại trang “Customers”, hệ thống hiển thị danh sách khách hàng đã mua vé, Admin có thể xem được khách hàng đã mua những trận nào và loại vé nào.
- Quy trình chăm sóc khách hàng: Admin hỗ trợ khách hàng bằng cách trả lời tin nhắn thông qua fanpage.

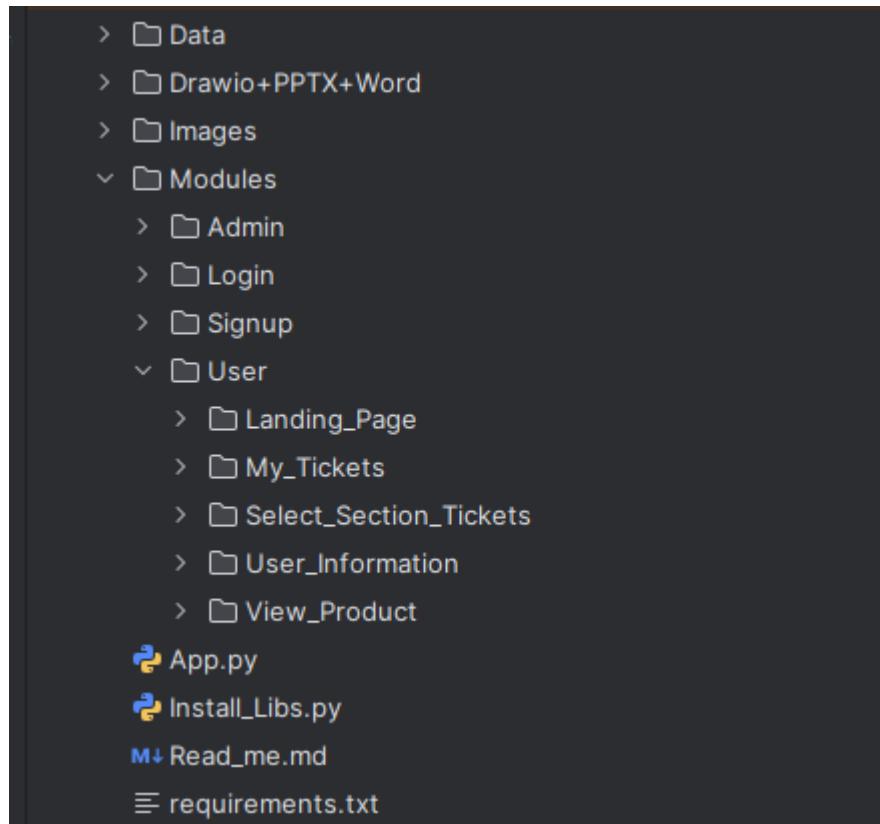
b. User:

Gồm các quy trình sau:

- Quy trình đặt vé: Để đặt vé, User truy cập vào trang “View Products” và chọn đội bóng yêu thích. Hệ thống sẽ hiển thị danh sách các trận đấu của đội bóng đó. Người dùng chọn một trận đấu muốn mua vé tại trang “View Tickets”. Hệ thống hiển thị các loại vé và số lượng vé có sẵn. Người dùng chọn hạng ghế và số lượng vé cần mua, sau đó nhấn vào nút “Payment” để tiến hành thanh toán.
- Quy trình thanh toán vé: Sau khi chọn vé, User vào trang “Confirm Payment”. Hệ thống hiển thị thông tin chi tiết về đơn hàng. User nhấn thanh toán, hệ thống xử lý thanh toán và cập nhật trạng thái vé.
- Quy trình xem vé đã mua: Để xem vé đã mua, user vào trang “My Tickets”. Hệ thống hiển thị danh sách vé đã mua. Người dùng có thể xem chi tiết vé, bao gồm thông tin trận đấu, loại vé, số lượng và tổng tiền.
- Quy trình quản lý thông tin cá nhân: Để xem thông tin cá nhân, user vào trang “Information Customer”. Hệ thống hiển thị thông tin của người dùng như username, email và password.
- Quy trình liên hệ hỗ trợ: Để liên hệ hỗ trợ, user chọn “Contact”. Hệ thống hiển thị link đến fanpage Facebook, user nhấn vào link để mở trang fanpage và nhắn tin.

### 4.3. Cấu trúc tổ chức mã nguồn.

Dưới đây là cấu trúc tổ chức mã nguồn của toàn bộ phần mềm, sử dụng Tkinter để thiết kế giao diện và dùng cơ sở dữ liệu MongoDB để thực hiện xây dựng và lưu trữ dữ liệu cho phần mềm.



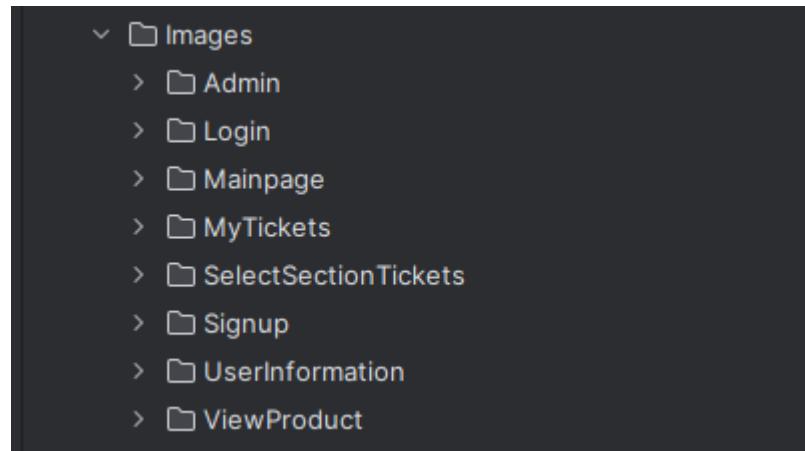
Hình 4. Cấu trúc tổ chức mã nguồn của phần mềm

Thư mục Data (dữ liệu) bao gồm các file môi trường .env, dùng để kết nối dữ liệu từ cơ sở dữ liệu MongoDB xuống phần mềm.



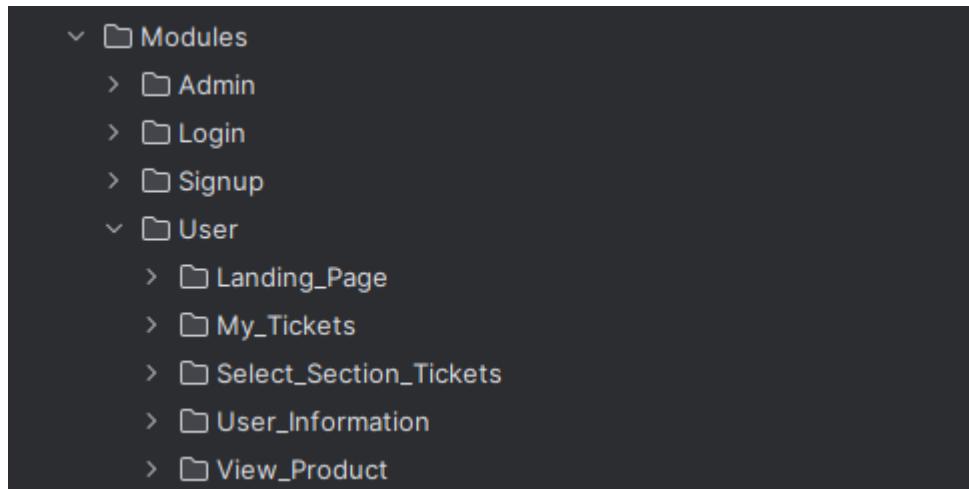
Hình 5. Cấu trúc tổ chức thư mục Data

Thư mục Images (hình ảnh) chứa nhiều thư mục con, mỗi thư mục chứa các file hình ảnh .png tải về từ bản thiết kế trên nền tảng Figma được sử dụng để thực hiện thiết kế giao diện GUI với Tkinter và được tổ chức phân nhóm theo các thư mục con với tên gọi tương ứng theo các trang chức năng của phần mềm.



Hình 6. Cấu trúc tổ chức thư mục Images

Thư mục Modules bao gồm các file code chạy chương trình về cả giao diện người dùng và cả logic xử lý bên trong hệ thống của toàn bộ phần mềm.



Hình 7. Cấu trúc tổ chức thư mục Modules

Ở mỗi thư mục mang tên tương ứng với các vai trò chính sẽ bao gồm 1 file Front-end và 1 file Back-end tương ứng. Cấu trúc tổ chức này giúp việc theo dõi và điều chỉnh dễ dàng hơn, cụ thể như sau:

- Thư mục con Admin bao gồm toàn bộ các chức năng của vai trò Admin được tổ chức trong 2 file như sau :
  - Admin\_Process.py
  - Admin\_View.py
- Thư mục con Login là nơi chứa các xử lý đăng nhập cho cả tài khoản Admin và Users, bao gồm:

- Login\_Process.py
  - Login\_View.py
- Thư mục con Signup chứa file xử lý việc đăng ký tài khoản mới cho User, bao gồm:
  - Signup\_Process.py
  - Signup\_View.py
- Thư mục con User được tổ chức phân nhánh cụ thể hơn với các thư mục con nhỏ hơn tương ứng với các trang chức năng cụ thể trong luồng xử lý của vai trò User, cụ thể như sau:
  - Thư mục con Landing\_Page, đây là nơi User có thể nhìn thấy và lựa chọn các đội bóng chính, bao gồm:
    - + Main\_Page\_Process.py
    - + Main\_Page\_View.py
  - Thư mục con My\_Tickets, là nơi User có thể xem lại các vé và trận đấu đã mua, bao gồm:
    - + My\_Tickets\_Process.py
    - + My\_Tickets\_View.py
  - Thư mục con Select\_Section\_Tickets, hiển thị các hạng vé của trận đấu và xử lý mua vé, bao gồm:
    - + Select\_Section\_Process.py
    - + Select\_Section\_View.py
  - Thư mục con User\_Information, là nơi User có thể xem lại thông tin cá nhân tài khoản của mình, bao gồm:
    - + User\_Information\_Process.py
    - + User\_Information\_View.py

- Thư mục con View\_Product, hiển thị các trận đấu hiện có của đội bóng, thực hiện xử lý chọn trận đấu của User, bao gồm:

- + View\_Product\_Process.py

- + View\_Product\_View.py

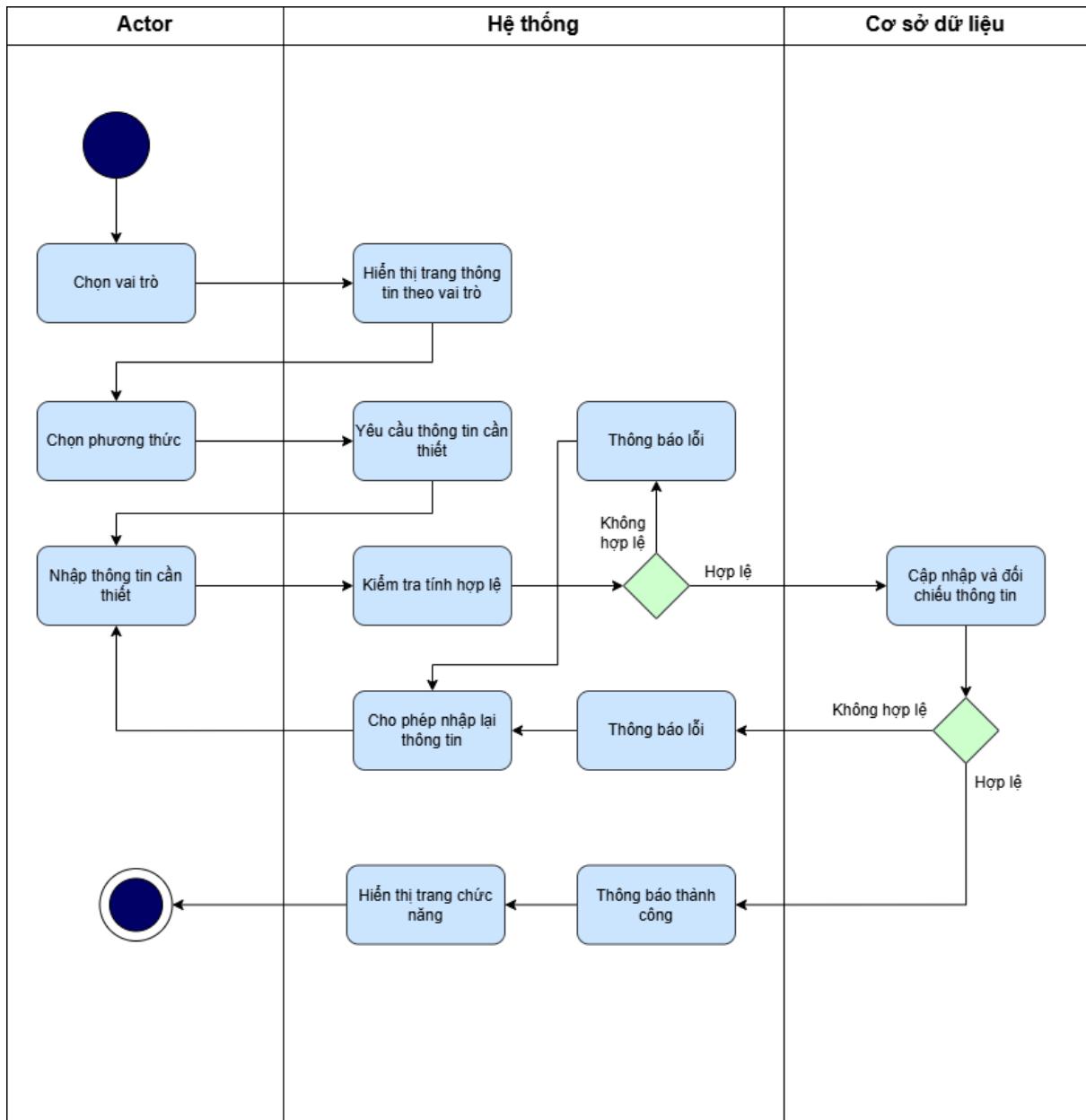
Sau cùng là file App.py chính là file chạy chính của chương trình, thực hiện khởi chạy phần mềm, file Install\_libs.py dùng để tải các thư viện cần thiết.

## CHƯƠNG V: THIẾT KẾ HỆ THỐNG

### 5.1. Sơ đồ hoạt động.

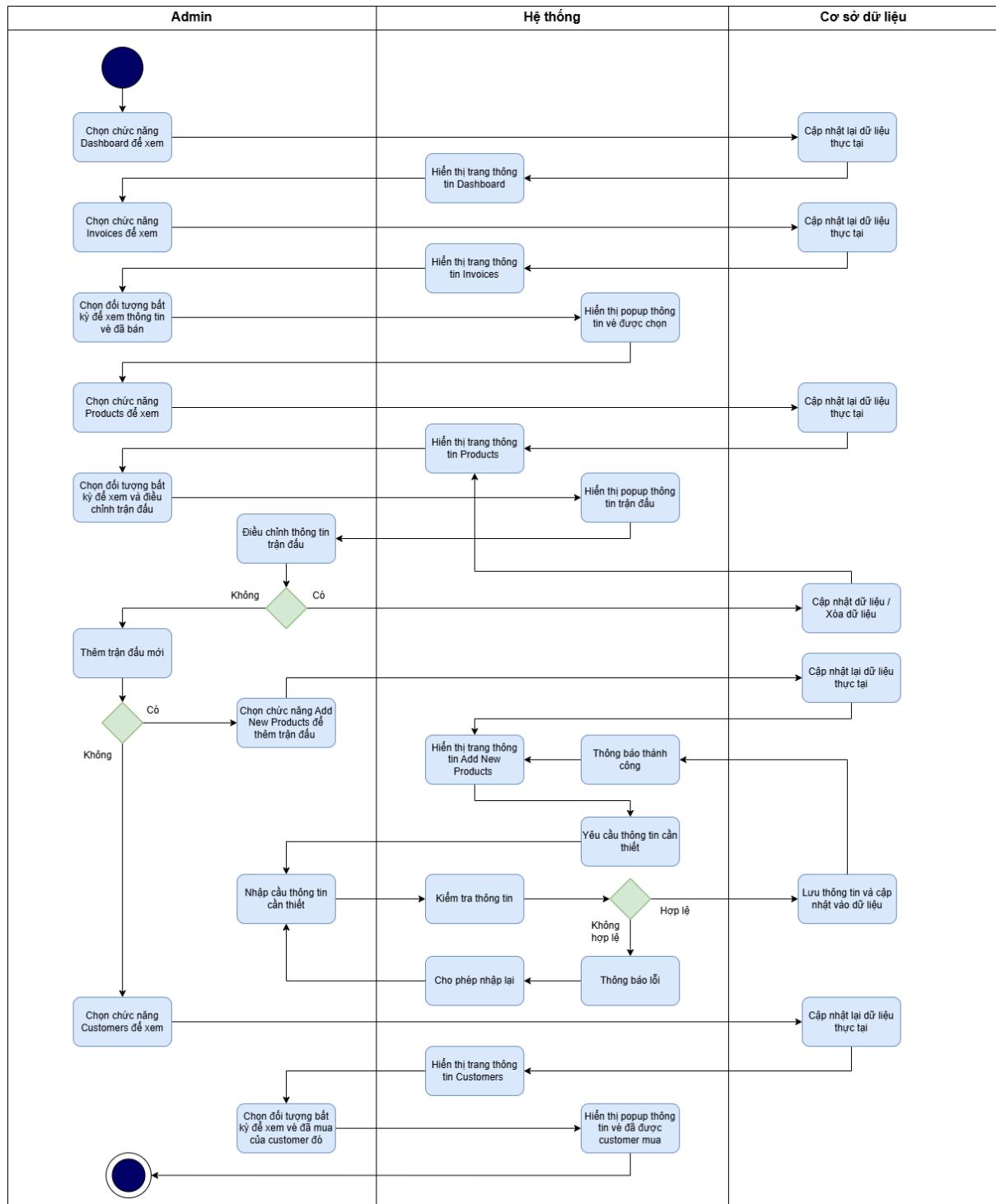
Các quy trình chính trong chương trình bao gồm: quy trình đăng nhập/đăng ký, quy trình hoạt động của Admin và quy trình Mua hàng của User.

Sơ đồ hoạt động khi Đăng nhập:



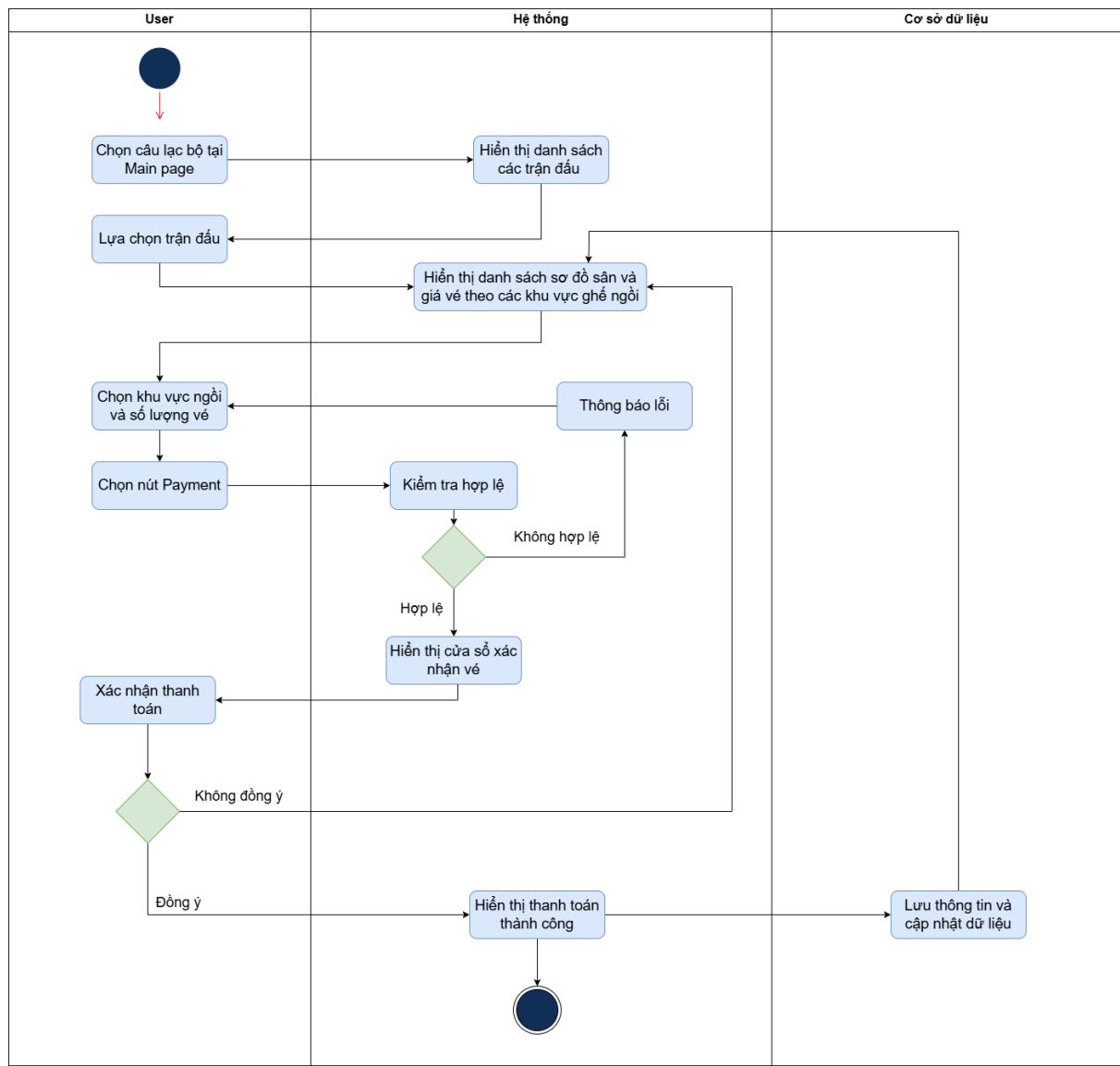
Hình 8. Sơ đồ hoạt động khi Đăng nhập

## Sơ đồ hoạt động Admin:



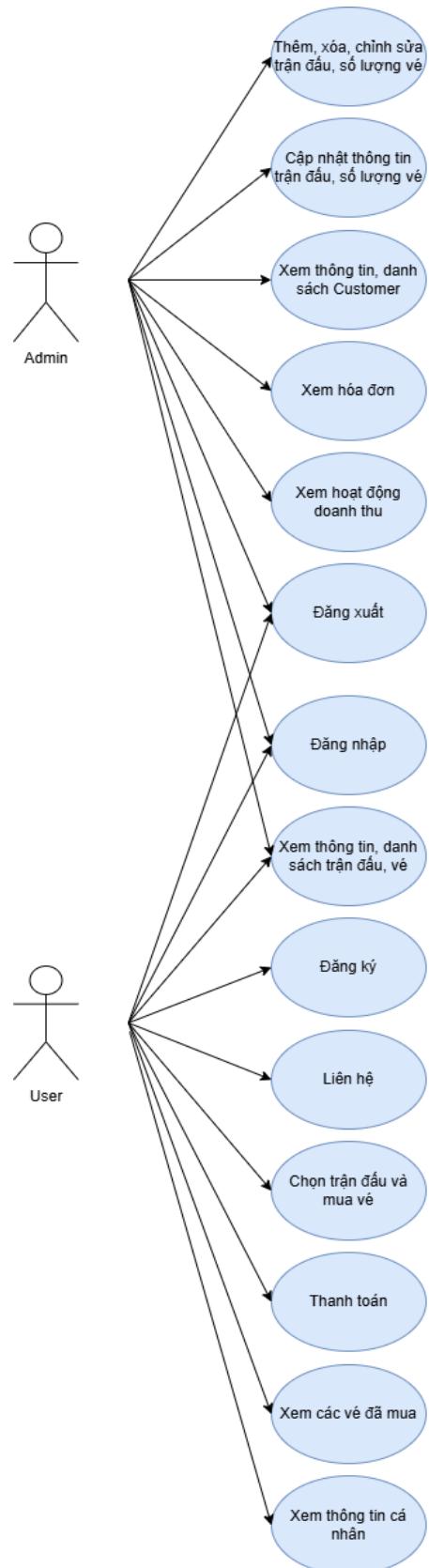
Hình 9. Sơ đồ hoạt động Admin

## Sơ đồ hoạt động khi Mua hàng của User:



Hình 10. Sơ đồ hoạt động khi Mua hàng của User

## 5.2. Thiết kế các mô hình xử lý – Use case.



Hình 11. Mô hình Use case

### *5.2.1. Danh sách Use case.*

- Login
- Sign up
- Log out
- User
  - Contact
  - Information Customer
  - Purchase History
  - View Products
  - Select Section
  - Payment
- Admin
  - Dashboard
  - Invoices
  - Products
  - Add New Product
  - Customers

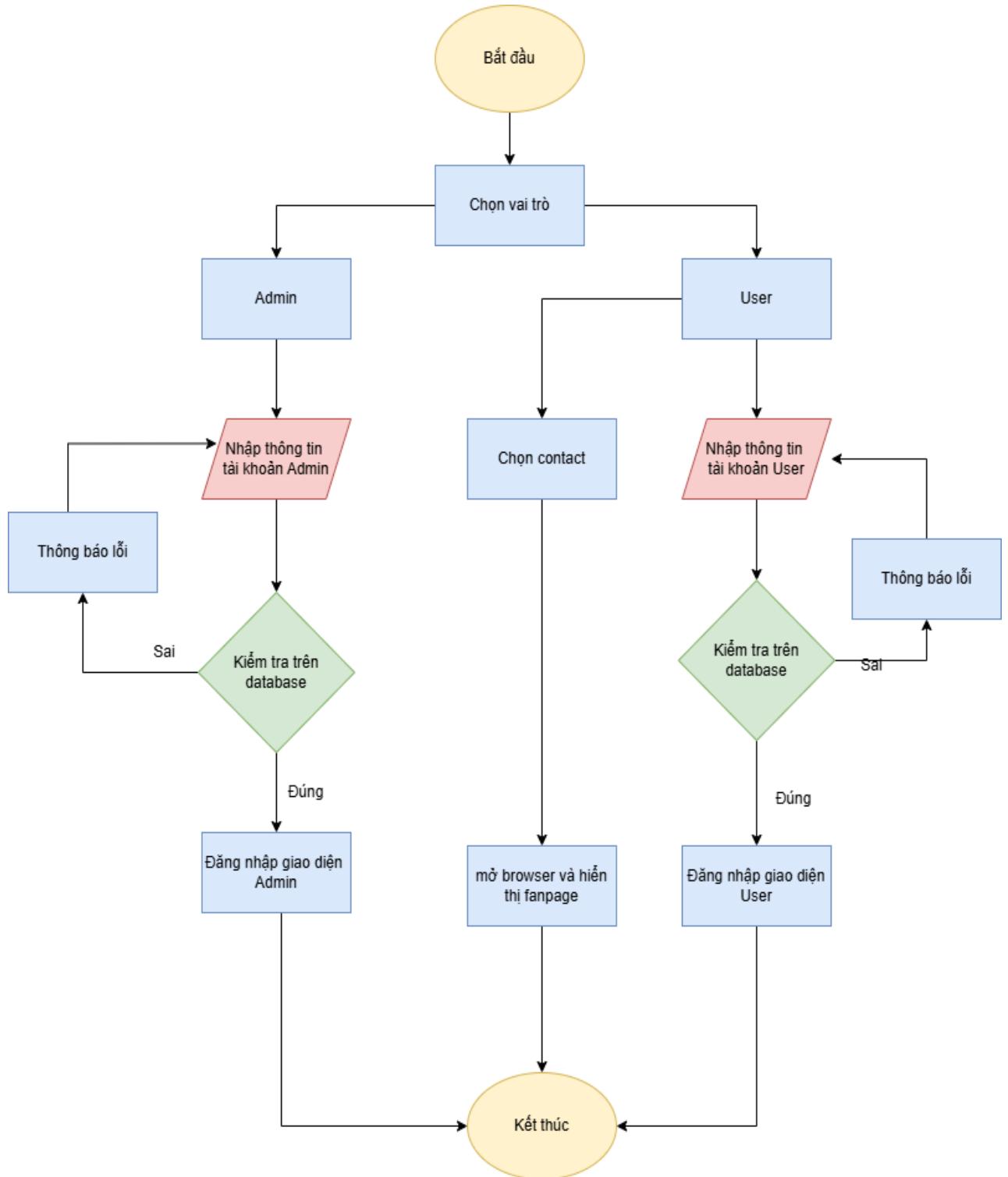
### *5.2.2. Mô tả Use case.*

## Use case Login

Bảng 1. Use case Login

Định nghĩa	Đối tượng	Luồng hoạt động chính	Luồng hoạt động khác	Điều kiện	Kết quả
<ul style="list-style-type: none"> <li>- Mô tả cách người dùng User và Admin đăng nhập vào hệ thống.</li> <li>- Thể hiện quy trình xác thực danh tính để truy cập vào các chức năng hoặc dữ liệu được bảo vệ.</li> </ul>	<ul style="list-style-type: none"> <li>- User</li> <li>- Admin</li> </ul>	<ul style="list-style-type: none"> <li>- <b>Bắt đầu:</b> Người dùng chọn vai trò Admin hoặc User</li> <li>- Người dùng nhập tên đăng nhập, mật khẩu theo yêu cầu hệ thống.</li> <li>- Hệ thống kiểm tra thông tin đăng nhập.</li> <li>- Nếu thông tin người dùng trùng khớp với dữ liệu hệ thống đã lưu, hệ thống cấp quyền truy cập</li> </ul>	<ul style="list-style-type: none"> <li>- Tên đăng nhập hoặc mật khẩu không trùng khớp, hệ thống sẽ thông báo lỗi.</li> <li>- Tên đăng nhập và email đã tồn tại trong database hệ thống sẽ báo lỗi</li> <li>- User không đăng nhập có thể chọn “Contact” để liên hệ hỗ trợ</li> <li>- Use case kết thúc</li> </ul>	<ul style="list-style-type: none"> <li>- Người dùng đã có tài khoản hợp lệ trong hệ thống.</li> <li>- Thiết bị có kết nối internet</li> </ul>	<p>Nếu đăng nhập thành công, hệ thống điều hướng đến trang chính của người dùng và người dùng có thể sử dụng, thao tác ứng với phân quyền của mình</p>

Flowchart cho Use case “Login”:



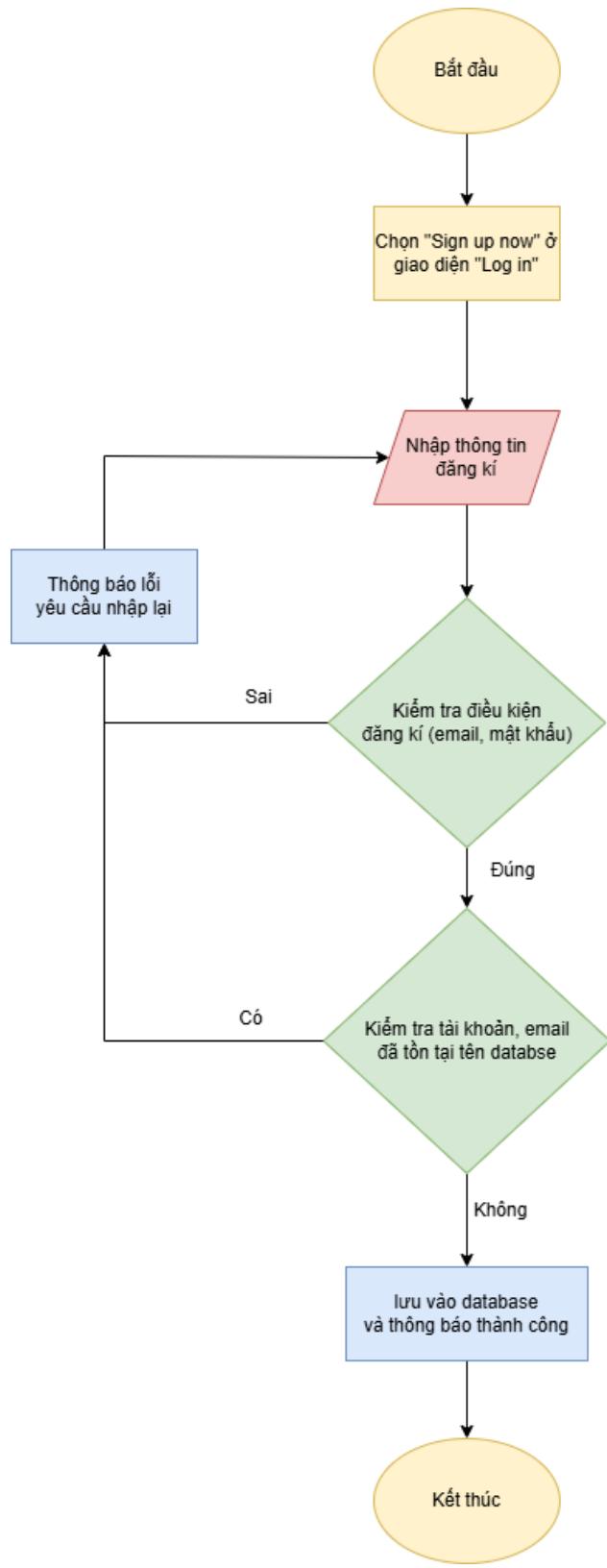
Hình 12. Flowchart cho Use case "Login"

## Use case Sign up

Bảng 2. Use case Sign up

<b>Định nghĩa</b>	<b>Đối tượng</b>	<b>Luồng hoạt động chính</b>	<b>Luồng hoạt động khác</b>	<b>Điều kiện</b>	<b>Kết quả</b>
Mô tả quy trình một người dùng mới tạo tài khoản để sử dụng ứng dụng.	User	<ul style="list-style-type: none"> <li>- Bắt đầu: Người dùng chọn chức năng “Sign up now” trên màn hình Login</li> <li>- Hệ thống hiển thị biểu mẫu đăng ký, yêu cầu nhập thông tin: <ul style="list-style-type: none"> <li>+ Profile name</li> <li>+ Email</li> <li>+Password</li> <li>+Re-enter password</li> </ul> </li> <li>- Người dùng nhập thông tin và nhấn “Sign up”</li> <li>- Hệ thống kiểm tra các điều kiện ở trên, trường hợp các thông tin đã điền đúng yêu cầu (Điền đầy đủ các ô; Email hợp lệ, đã tồn tại trong hệ thống hay chưa; Password đúng định dạng, Password nhập lại trùng khớp) thì hệ thống sẽ thực</li> </ul>	<ul style="list-style-type: none"> <li>- Trường hợp các thông tin chưa đúng yêu cầu (Điền thiếu; Email đăng ký không hợp lệ; Password chưa đúng định dạng, không trùng khớp) thì hệ thống sẽ gửi thông báo lỗi và yêu cầu nhập lại</li> <li>- Trường hợp tài khoản đã tồn tại, hệ thống thông báo Email hoặc Username đã tồn tại. Người dùng chọn “Log in” để đăng nhập hoặc sử dụng</li> <li>Email/Username khác để tạo tài khoản mới</li> <li>- Use case kết thúc</li> </ul>	<ul style="list-style-type: none"> <li>- Người dùng chưa có tài khoản của người</li> <li>- Thiết bị có kết nối mạng</li> </ul>	Nếu đăng ký thành công, tài khoản của người dùng được tạo và sẵn sàng sử dụng, có thể tiến hành thao tác đăng nhập

		hiện tạo tài khoản đăng nhập			
--	--	------------------------------	--	--	--

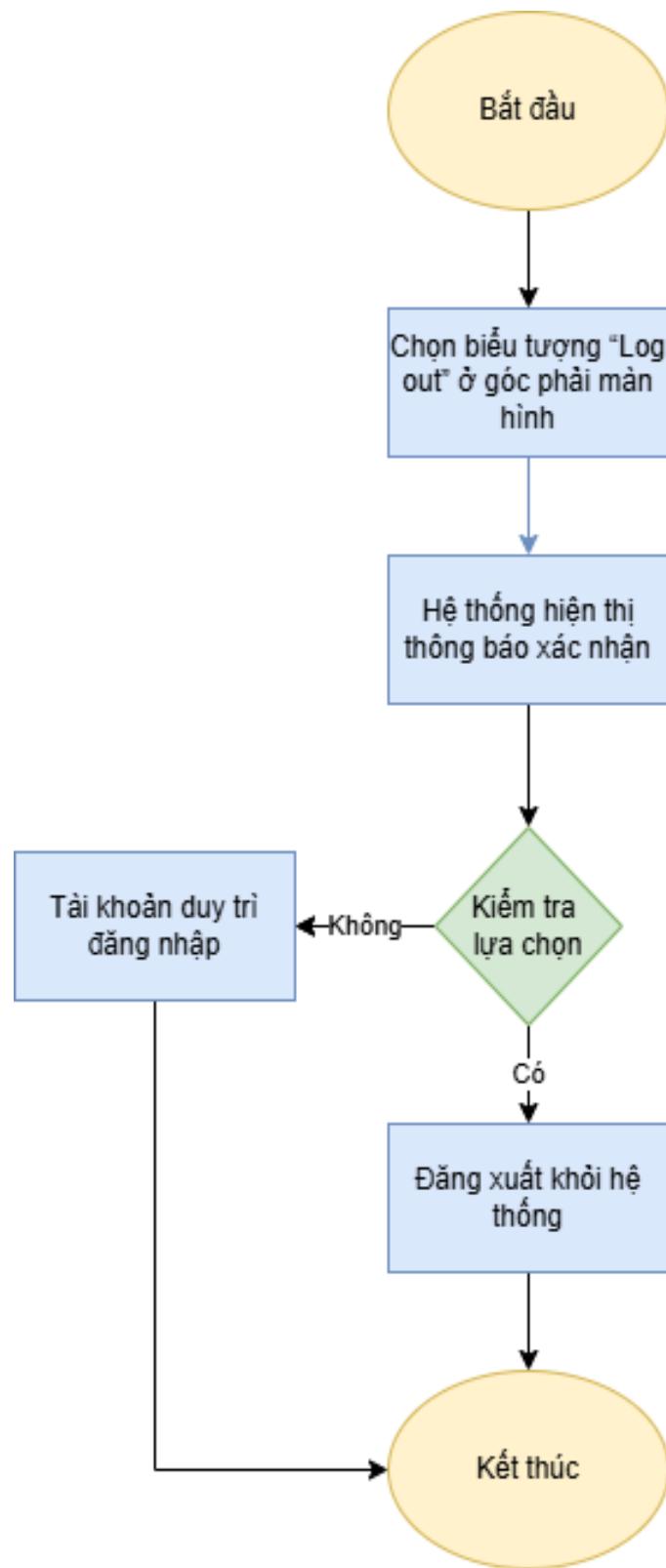


Hình 13. Flowchart cho Use case "Sign up"

*Use case Log out*

*Bảng 3. Use case Log out*

<b>Định nghĩa</b>	<b>Đối tượng</b>	<b>Luồng hoạt động chính</b>	<b>Luồng hoạt động khác</b>	<b>Điều kiện</b>	<b>Kết quả</b>
Mô tả quy trình một người dùng đăng xuất tài khoản khỏi hệ thống	- User - Admin	<ul style="list-style-type: none"> <li>- Bắt đầu: Người dùng chọn biểu tượng “Log out” ở góc phải màn hình           <ul style="list-style-type: none"> <li>- Hệ thống hiện thị thông báo xác nhận:</li> <li>+ Người dùng chọn “Yes”: đăng xuất khỏi hệ thống</li> <li>+ Người dùng chọn “No”: tiếp tục đăng nhập trong hệ thống</li> </ul> </li> </ul>	Không có	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập</li> <li>- Thiết bị có kết nối internet</li> </ul>	Nếu đăng ký thành công, tài khoản của người dùng được tạo, lưu lên database và sẵn sàng sử dụng, có thể tiến hành thao tác đăng nhập

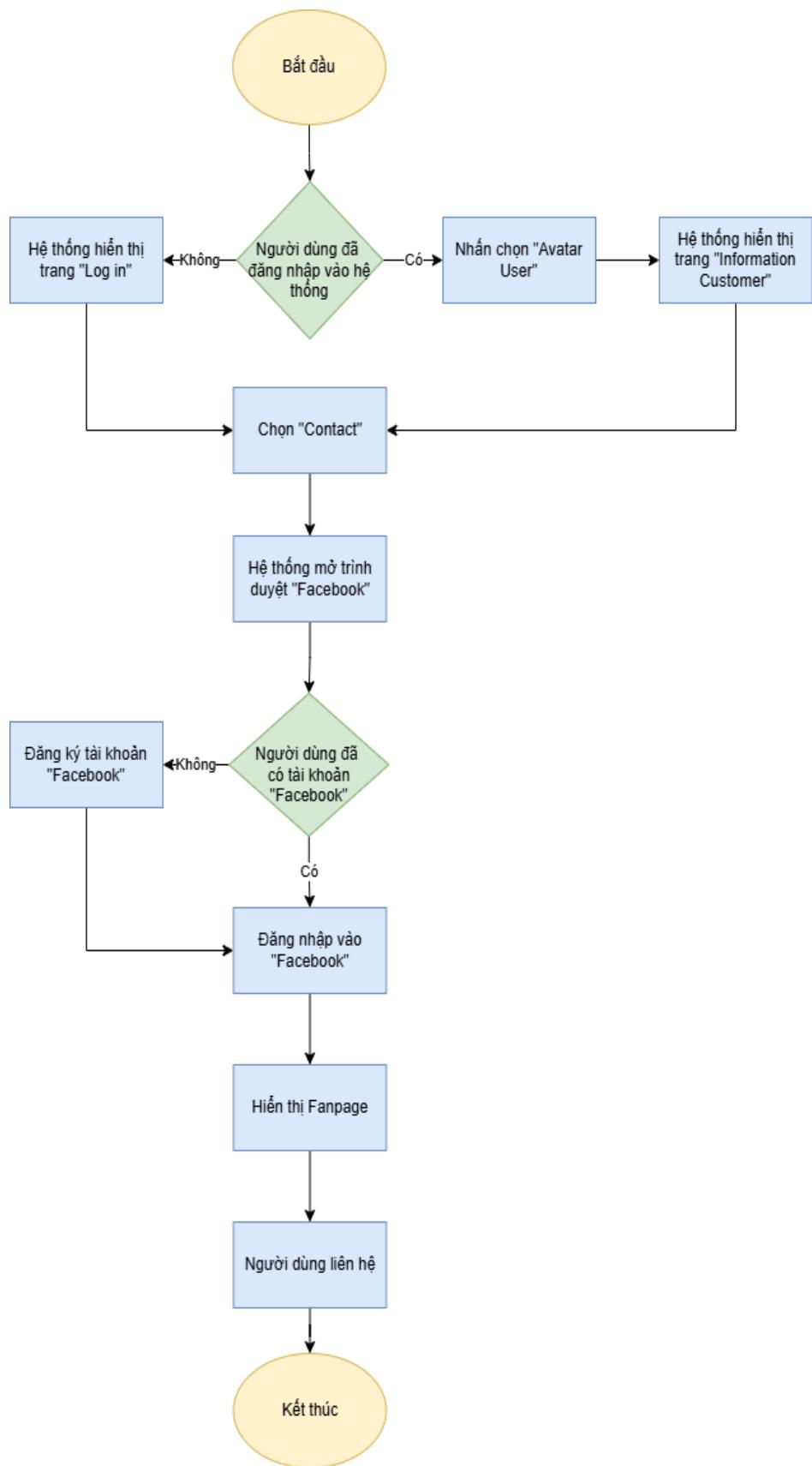


Hình 14. Flowchart cho Use case "Log out"

Use case Contact

Bảng 4. Use case Contact

<b>Định nghĩa</b>	<b>Đối tượng</b>	<b>Luồng hoạt động chính</b>	<b>Luồng hoạt động khác</b>	<b>Điều kiện</b>	<b>Kết quả</b>
Mô tả quy trình người dùng liên hệ bộ phận hỗ trợ khách hàng để đặt câu hỏi, khiếu nại hoặc yêu cầu trợ giúp.	User	<ul style="list-style-type: none"> <li>- Trường hợp người dùng chưa đăng nhập, nhấn chọn “Contact” ở trang “Log in”</li> <li>- Trường hợp người dùng đã đăng nhập, nhấn chọn “Contact” ở trang “Information Customer”</li> <li>- Hệ thống mở trình duyệt và đưa đến trang fanpage</li> <li>- Người dùng đăng nhập và liên hệ hỗ trợ</li> <li>- Use case kết thúc</li> </ul>	Không có	<ul style="list-style-type: none"> <li>- Người dùng có tài khoản Facebook</li> <li>- Thiết bị có kết nối internet</li> </ul>	Liên kết đến fanpage và người dùng có thể liên hệ hỗ trợ

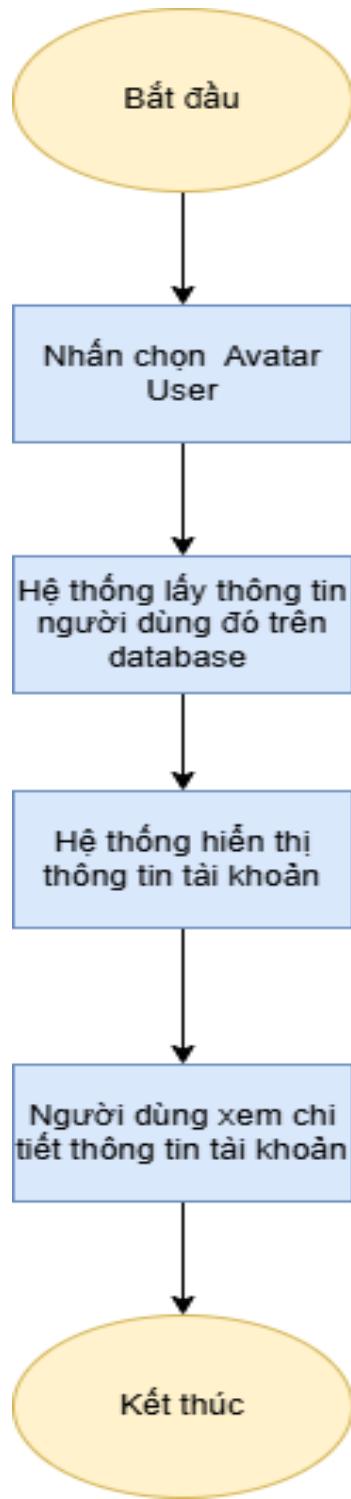


Hình 15. Flowchart cho Use case "Contact"

## Use case Information Customer

*Bảng 5. Use case Information Customer*

Định nghĩa	Đối tượng	Luồng hoạt động chính	Luồng hoạt động khác	Điều kiện	Kết quả
Mô tả quy trình người dùng có thể xem thông tin tài khoản	User	<ul style="list-style-type: none"> <li>- Bắt đầu: Người dùng nhấn chọn “Avatar User” để vào trang “Information Customer”</li> <li>- Hệ thống lấy thông tin người dùng đó trên database về.</li> <li>- Hệ thống hiển thị các thông tin cá nhân bao gồm:</li> <li>+ Username</li> <li>+ Email</li> <li>+ Password</li> </ul>	<ul style="list-style-type: none"> <li>- Người dùng có thể chọn “Hide” để xem mật khẩu tài khoản</li> <li>- Use case kết thúc</li> </ul>	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập</li> <li>- Thiết bị có kết nối internet</li> </ul>	Người dùng xem chi tiết thông tin tài khoản

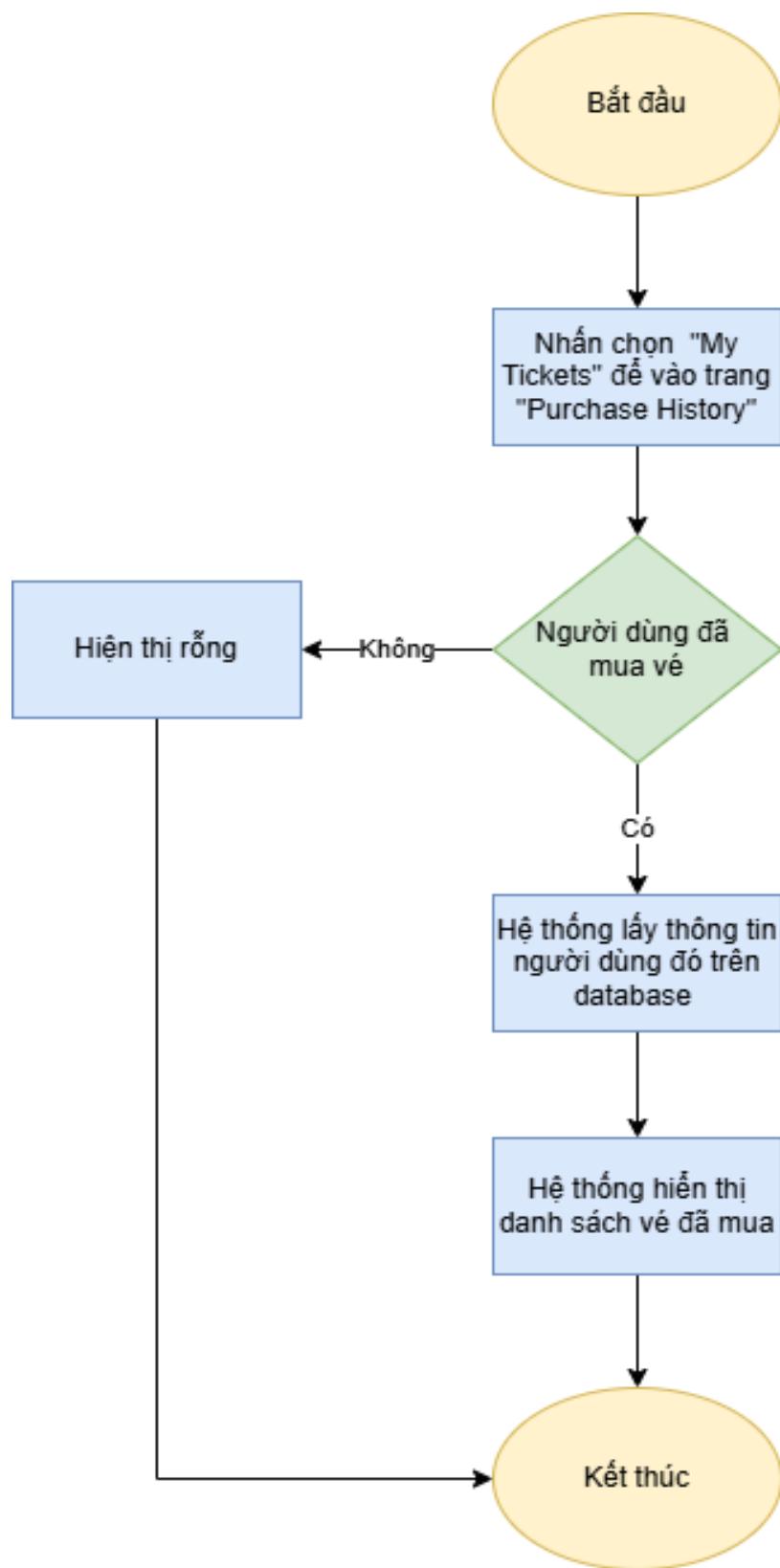


Hình 16. Flowchart cho Use case "Information Customer"

## Use case Purchase History

Bảng 6. Use case Purchase History

Định nghĩa	Đối tượng	Luồng hoạt động chính	Luồng hoạt động khác	Điều kiện	Kết quả
Mô tả quy trình người dùng thao tác để xem danh sách các vé đã mua	User	<ul style="list-style-type: none"> <li>- Bắt đầu: Người dùng vào trang “Purchase History” nằm trong mục “My Tickets”</li> <li>- Hệ thống lấy thông tin người dùng đó trên database vé.</li> <li>- Hệ thống hiển thị danh sách vé đã mua.</li> </ul>	<ul style="list-style-type: none"> <li>- Trường hợp khi người dùng chưa mua vé, hệ thống hiển thị rỗng</li> <li>- Use case kết thúc</li> </ul>	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập vào hệ thống</li> <li>- Thiết bị có kết nối internet</li> </ul>	Người dùng xem được thông tin các vé đã mua

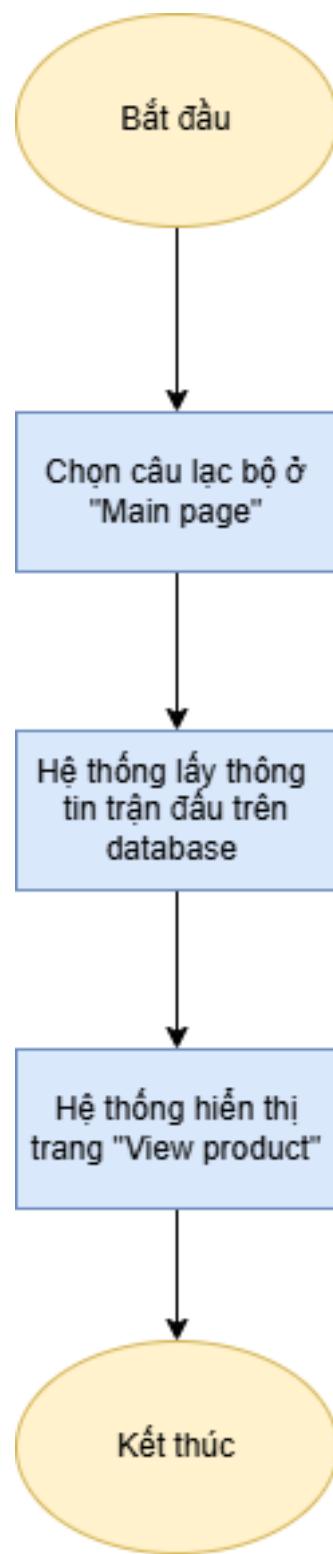


Hình 17. Flowchart cho Use case "Purchase History"

*Use case View Product*

*Bảng 7. Use case View Product*

<b>Định nghĩa</b>	<b>Đối tượng</b>	<b>Luồng hoạt động chính</b>	<b>Luồng hoạt động khác</b>	<b>Điều kiện</b>	<b>Kết quả</b>
Mô tả quy trình người dùng xem danh sách các trận đấu, thông tin chi tiết các trận	User	<ul style="list-style-type: none"> <li>- Bắt đầu: Người dùng chọn Câu lạc bộ mình quan tâm ở “Main page”. Sau khi bấm chọn, hệ thống sẽ chuyển đến trang “View product”</li> <li>- Hệ thống lấy thông tin các trận đấu trên database</li> <li>- Hệ thống hiển thị danh sách các trận đấu với thông tin (đội bóng, ngày giờ, địa điểm).</li> </ul>	Không có	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập vào hệ thống</li> <li>- Thiết bị có kết nối mạng</li> </ul>	<p>Người dùng có thể xem các thông tin về trận đấu và tiếp tục quá trình đặt vé</p>

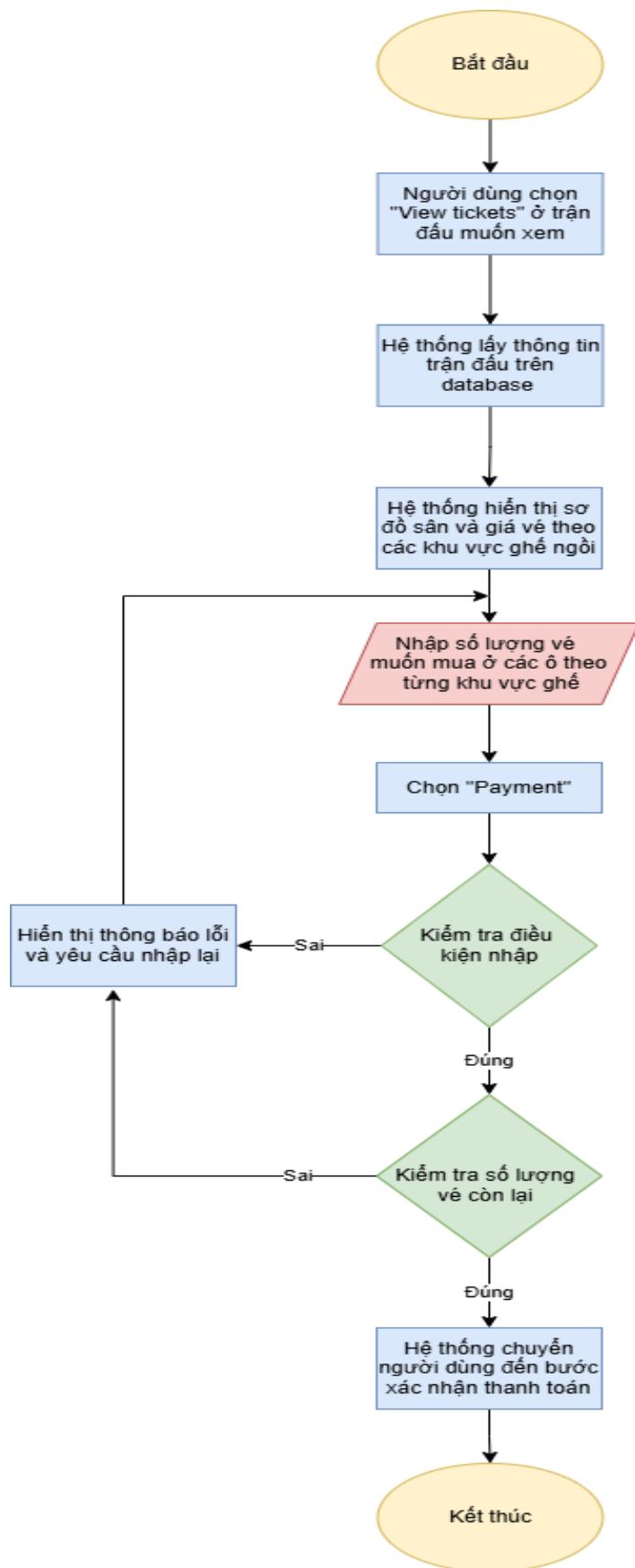


Hình 18. Flowchart cho Use case "View Product"

## Use case Select Section

Bảng 8. Use case Select Section

Định nghĩa	Đối tượng	Luồng hoạt động chính	Luồng hoạt động khác	Điều kiện	Kết quả
Mô tả quy trình người dùng chọn khu vực ghế ngồi trên sơ đồ sân	User	<ul style="list-style-type: none"> <li>- Bắt đầu: Người dùng chọn trận đấu mình muốn xem và nhấn “View tickets”</li> <li>- Hệ thống lấy thông tin trận đấu trên database</li> <li>- Hệ thống hiển thị sơ đồ sân và giá vé theo các khu vực ghế ngồi</li> <li>- Người dùng chọn số lượng vé và chọn khu vực ngồi mong muốn</li> <li>- Sau khi người dùng nhập đủ số lượng mong muốn thì người dùng nhấn vào “Payment” để tiến hành đến bước thanh toán</li> </ul>	<ul style="list-style-type: none"> <li>- Trường hợp người dùng chưa nhập vào số lượng vé nhưng nhấn chọn “Payment”. Hệ thống sẽ hiện thông báo lỗi và yêu cầu nhập số lượng</li> <li>- Trường hợp người dùng nhập không phải là số nguyên dương. Hệ thống sẽ hiện thông báo lỗi và yêu cầu nhập đúng định dạng</li> <li>- Trường hợp khu vực đã hết vé hoặc không đủ số lượng, hệ thống hiển thị thông báo không đủ số lượng yêu cầu và báo số lượng vé còn lại</li> </ul>	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập vào hệ thống</li> <li>- Người dùng đã chọn trận đấu muốn xem</li> <li>- Người dùng đã nhập số lượng vé</li> <li>- Hệ thống có kết nối mạng</li> </ul>	<p>Lưu thông tin vé mua của khách hàng vào các biến tạm thời và hệ thống khởi tạo thông tin xác nhận</p>

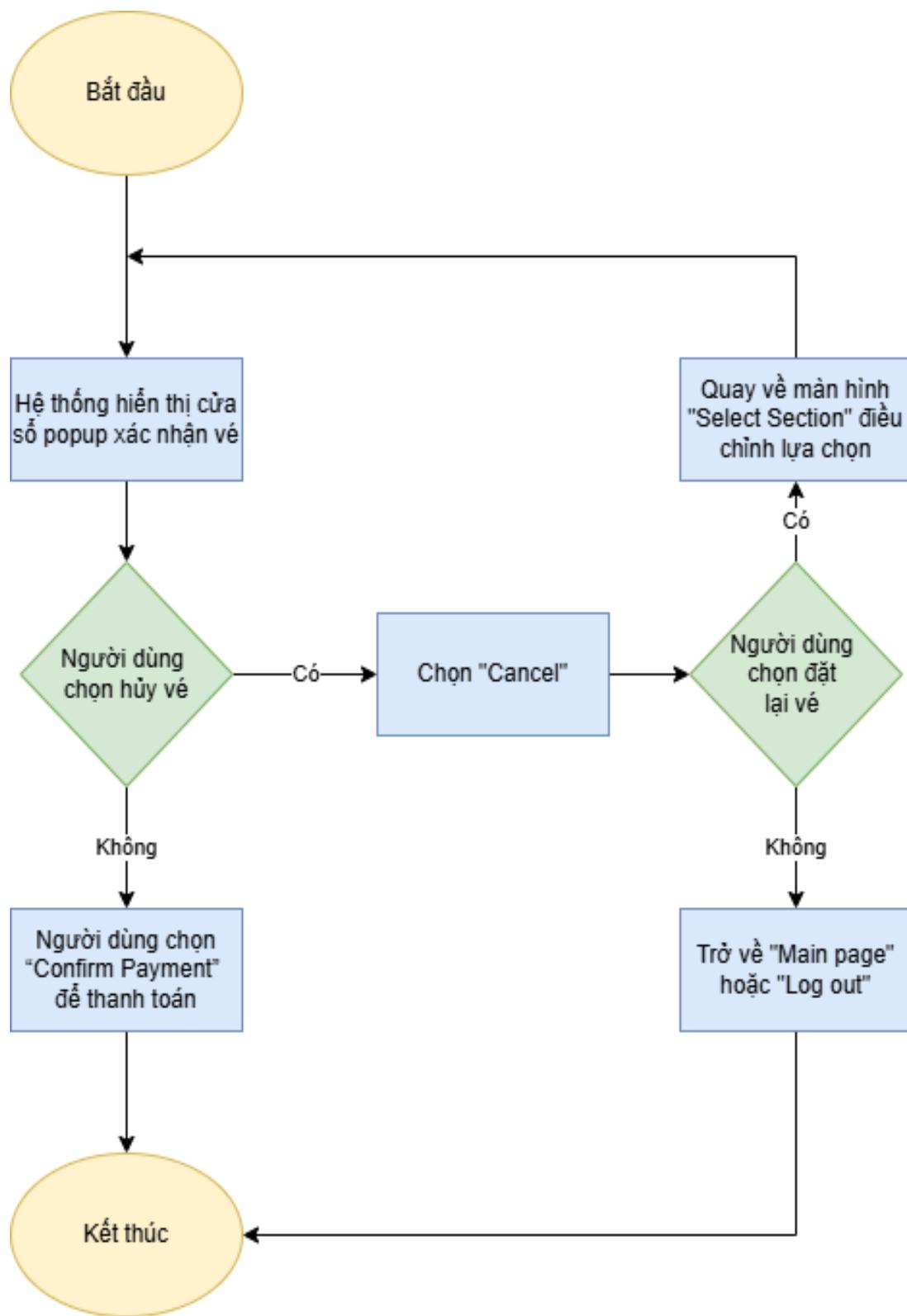


Hình 19. Flowchart cho Use case "Select Section"

## Use case Payment

Bảng 9. Use case Payment

Định nghĩa	Đối tượng	Luồng hoạt động chính	Luồng hoạt động khác	Điều kiện	Kết quả
Mô tả quy trình người dùng xác nhận và thanh toán.	User	<ul style="list-style-type: none"> <li>- Hệ thống hiển thị cửa sổ popup xác nhận vé, bao gồm:           <ul style="list-style-type: none"> <li>+ Tên trận đấu</li> <li>Thời gian, địa điểm diễn ra</li> <li>+ Số lượng, hạng ghế và tổng giá trị đơn hàng</li> <li>- Người dùng kiểm tra lại thông tin đặt vé</li> <li>- Người dùng nhấn “Confirm Payment” để thanh toán</li> <li>- Hệ thống hiện thông báo thanh toán thành công và ghi nhận thông tin đặt vé</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>- Trường hợp người dùng muốn thay đổi số lượng vé thì nhấn vào “Cancel” để thoát cửa sổ popup và quay về “Select section” chọn lại</li> <li>- Trường hợp người dùng muốn hủy đơn hàng thì nhấn vào “Cancel” để thoát cửa sổ popup</li> </ul>	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập vào hệ thống</li> <li>- Người dùng đã chọn số lượng vé và hạng ghế.</li> <li>- Thiết bị có kết nối mạng</li> </ul>	<ul style="list-style-type: none"> <li>- Nếu người dùng xác nhận “Confirm Payment”, thanh toán thành công, vé được lưu trong database khách hàng và chỉnh sửa lại thông tin trận đấu trên database.</li> <li>- Nếu người dùng “Cancel”, họ quay lại trang chọn vé hoặc thoát ra khỏi quá trình đặt vé.</li> </ul>

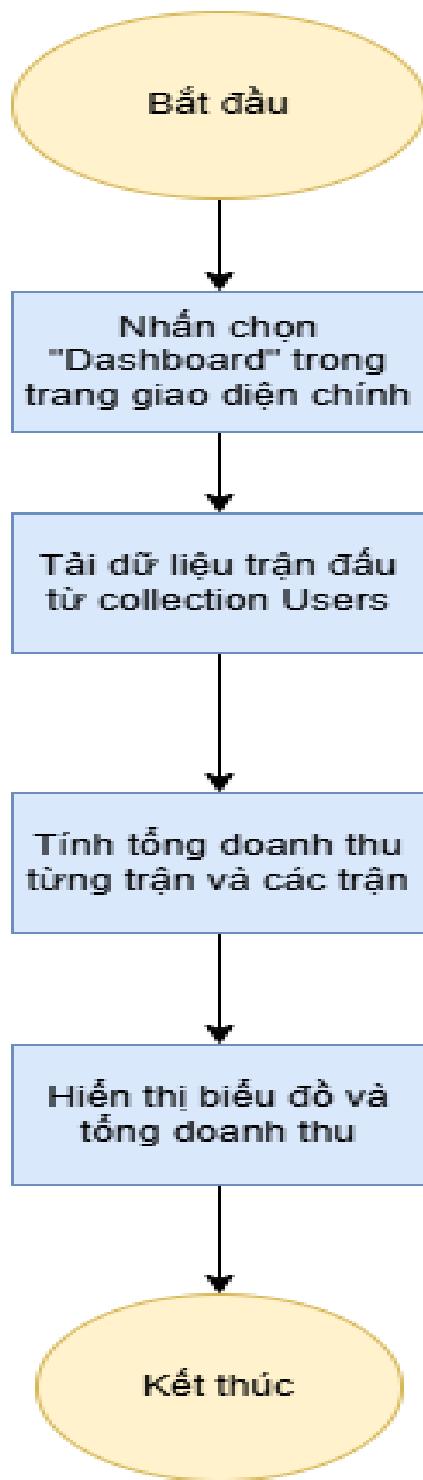


Hình 20. Flowchart cho Use case "Payment"

## Use case Dashboard

Bảng 10. Use case Dashboard

<b>Định nghĩa</b>	<b>Đối tượng</b>	<b>Luồng hoạt động chính</b>	<b>Luồng hoạt động khác</b>	<b>Điều kiện</b>	<b>Kết quả</b>
Mô tả quy trình Admin xem biểu đồ doanh thu từ việc bán vé	Admin	<ul style="list-style-type: none"> <li>- Bắt đầu: người dùng chọn ‘Dashboard’ trong trang giao diện chính (mặc định khi hiện giao diện Admin sẽ hiện giao diện ‘Dashboard’ đầu tiên).</li> <li>- Hệ thống lấy dữ liệu các trận đấu đã được mua từ Users trên database.</li> <li>- Hệ thống tính toán, hiển thị doanh thu theo từng đội bằng biểu đồ và tổng doanh thu của các đội.</li> </ul>	Không có	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập</li> <li>- Thiết bị có kết nối internet</li> </ul>	Người dùng quan sát được biểu đồ doanh thu.

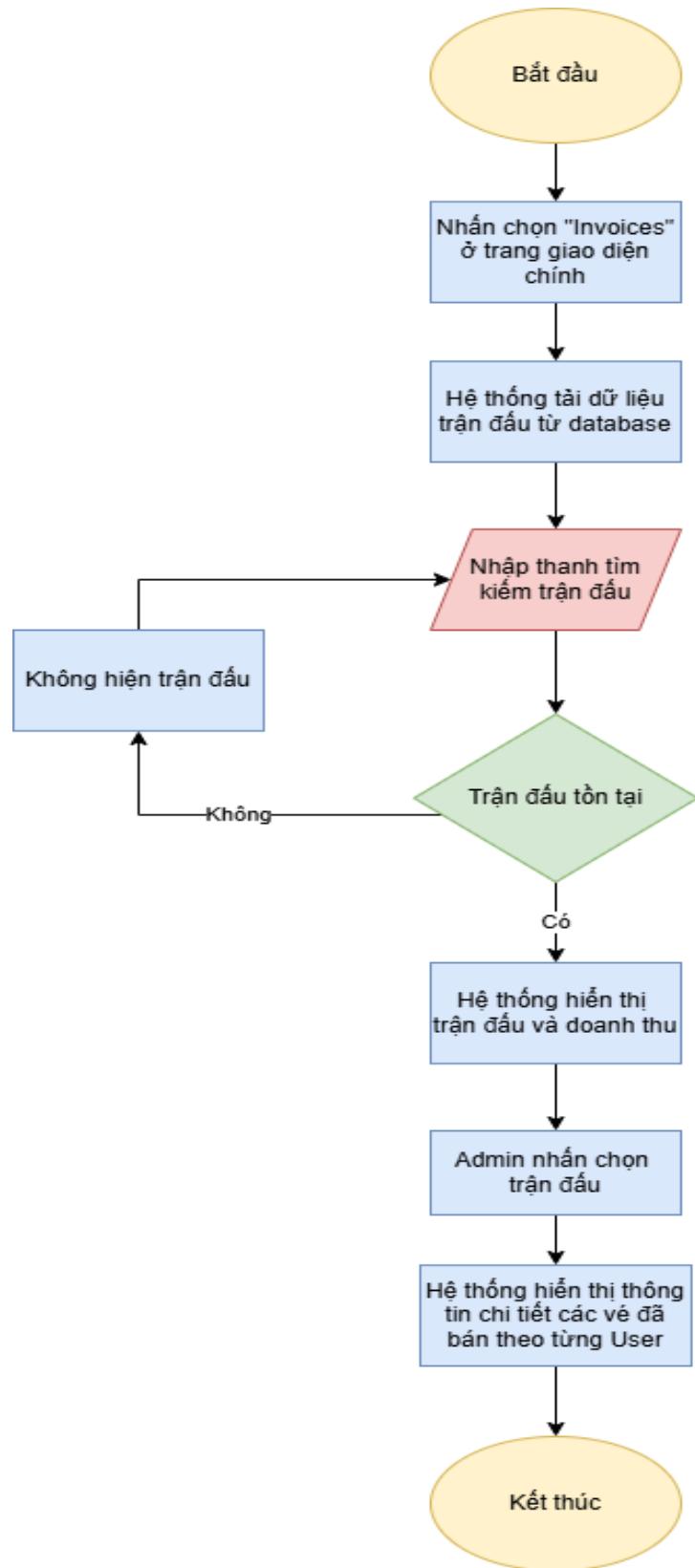


Hình 21. Flowchart cho Use case "Dashboard"

## Use case Invoices

Bảng 11. Use case Invoices

<b>Định nghĩa</b>	<b>Đối tượng</b>	<b>Luồng hoạt động chính</b>	<b>Luồng hoạt động khác</b>	<b>Điều kiện</b>	<b>Kết quả</b>
Mô tả quy trình Admin xem lại các vé đã bán của các trận đấu	Admin	<ul style="list-style-type: none"> <li>- Bắt đầu: người dùng chọn “Invoices” trên giao diện chính.</li> <li>- Hệ thống tải dữ liệu trận đấu từ database để hiển thị các trận đấu đang có.</li> <li>- Người dùng nhấn chọn trực tiếp trận đấu hoặc tìm kiếm ở thanh “Search matches”</li> <li>- Hệ thống hiển thị cửa sổ chứa thông tin chi tiết các vé đã bán theo từng User bao gồm:</li> <li>+ Total</li> <li>+ Section</li> <li>+ Quantity</li> <li>+ Price</li> <li>+ Subtotal</li> <li>- Use case kết thúc</li> </ul>	Không có	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập</li> <li>- Thiết bị có kết nối internet</li> </ul>	<p>Người dùng có thể xem thông tin chi tiết vé đã bán của từng trận.</p>

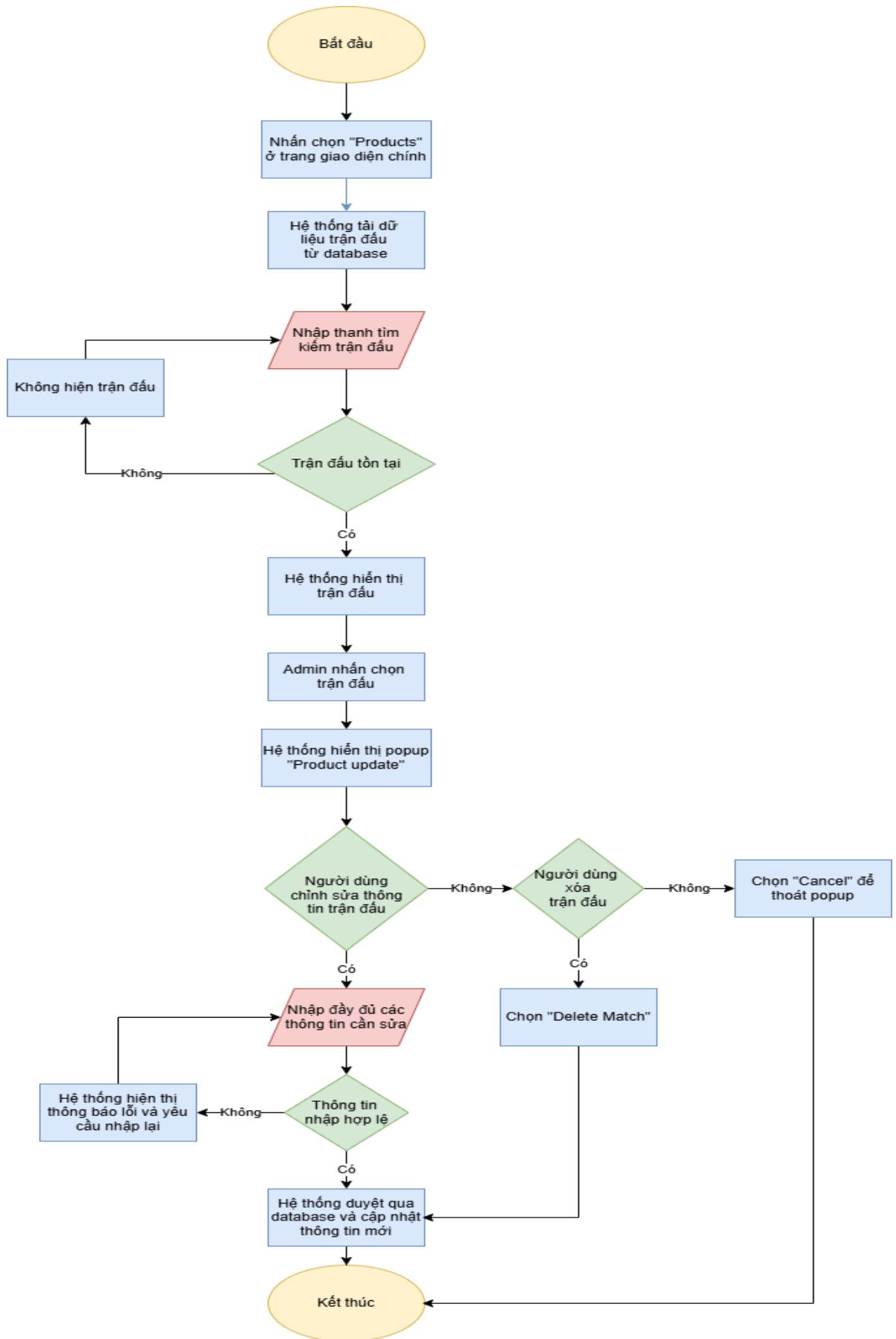


Hình 22. Flowchart cho Use case "Invoices"

Bảng 12. Use case Products

<b>Định nghĩa</b>	<b>Đối tượng</b>	<b>Luồng hoạt động chính</b>	<b>Luồng hoạt động khác</b>	<b>Điều kiện</b>	<b>Kết quả</b>
Mô tả quy trình Admin xem lại các trận đấu đang có và chỉnh sửa hoặc xóa trận	Admin	<ul style="list-style-type: none"> <li>- Bắt đầu: người dùng chọn Products trong trang giao diện chính</li> <li>- Lấy dữ liệu trận đấu từ database để hiển thị các trận đấu đang có</li> <li>- Người dùng nhấn chọn trực tiếp trận đấu hoặc tìm kiếm ở thanh “Search matches”</li> <li>- Hệ thống hiện thị cửa sổ popup “Product Update” cho phép người dùng chỉnh sửa hoặc xóa trận đấu:</li> <li>+ Người dùng chọn sửa trận: Nhập lại thông tin cần sửa của trận đấu và chọn “Save changes”. Hệ thống duyệt qua</li> </ul>	<ul style="list-style-type: none"> <li>Trường hợp người dùng nhập thiếu thông tin hoặc thông tin không hợp lệ khi sửa trận, hiển thị thông báo lỗi và yêu cầu nhập lại</li> </ul>	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập</li> <li>- Trận đấu đã tồn tại trong hệ thống</li> <li>- Thiết bị có kết nối internet.</li> </ul>	<ul style="list-style-type: none"> <li>- Nếu cập nhật thành công, thông tin trận đấu mới sẽ hiển thị trong hệ thống.</li> <li>- Nếu xóa trận đấu, nó sẽ không còn xuất hiện trong danh sách sự kiện.</li> </ul>

	<p>database chọn tất cả trận đấu trùng với trận đấu người dùng chọn và cập nhật thông tin mới lên.</p> <ul style="list-style-type: none"> <li>+ Người dùng chọn xóa trận: Người dùng chọn “Delete match”. Duyệt qua database lấy các trận đấu trùng lặp với trận đấu người dùng chọn và xóa chúng.</li> <li>+ Người dùng không muốn chỉnh sửa hoặc xóa trận: nhấn chọn “Cancel” để đóng cửa sổ</li> </ul>		
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

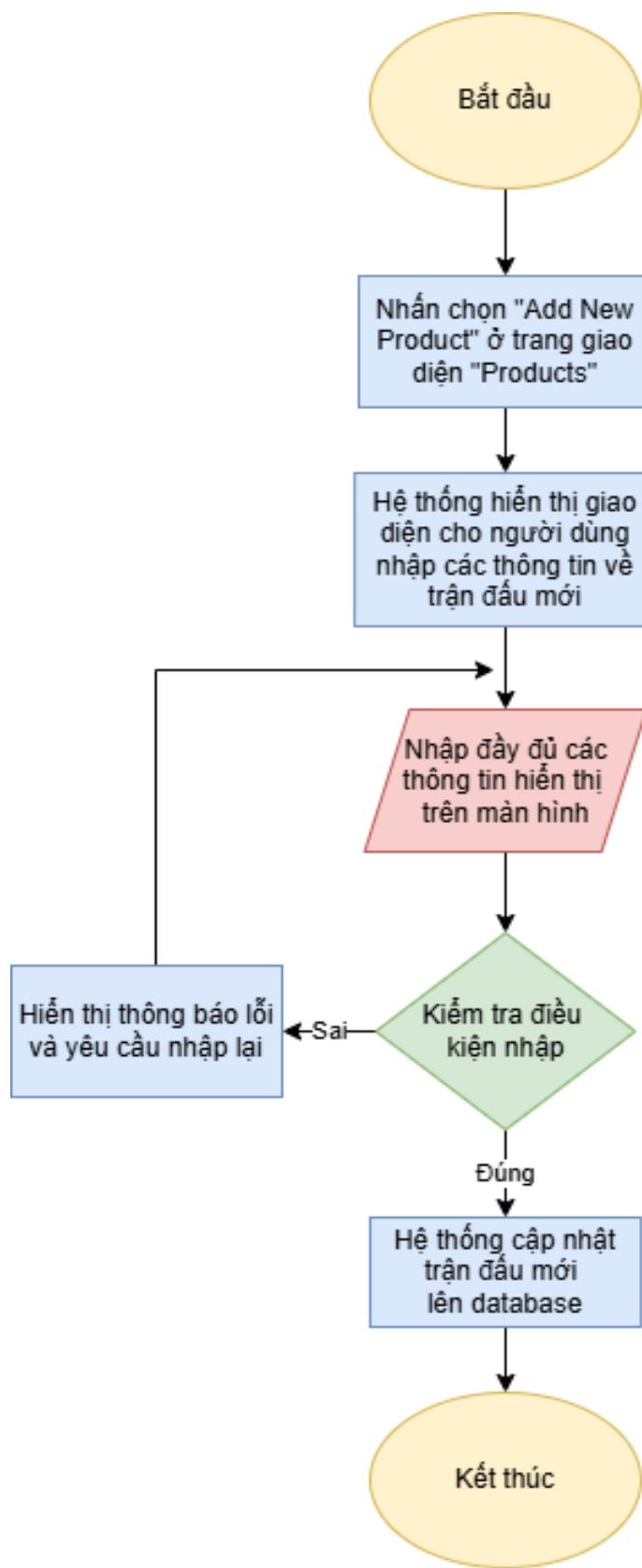


*Hình 23. Flowchart cho Use case "Products"*

Use case Add New Product

*Bảng 13. Use case Add New Product*

<b>Định nghĩa</b>	<b>Đối tượng</b>	<b>Luồng hoạt động chính</b>	<b>Luồng hoạt động khác</b>	<b>Điều kiện</b>	<b>Kết quả</b>
Mô tả quy trình Admin thêm trận đấu mới	Admin	<ul style="list-style-type: none"> <li>- Bắt đầu: người dùng chọn ‘Add New Product’ trong giao diện ‘Products’.</li> <li>- Hiển thị giao diện cho người dùng nhập đầy đủ các thông tin về trận đấu mới.</li> <li>- Người dùng nhấn ‘Save’ để lưu thông tin trận đấu lên database</li> </ul>	<p>Nếu người dùng nhập thiếu, sai định dạng, hệ thống sẽ hiện thông báo lỗi và yêu cầu nhập lại.</p>	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập</li> <li>- Thiết bị có kết nối internet.</li> </ul>	Người dùng thêm một trận đấu mới lên hệ thống

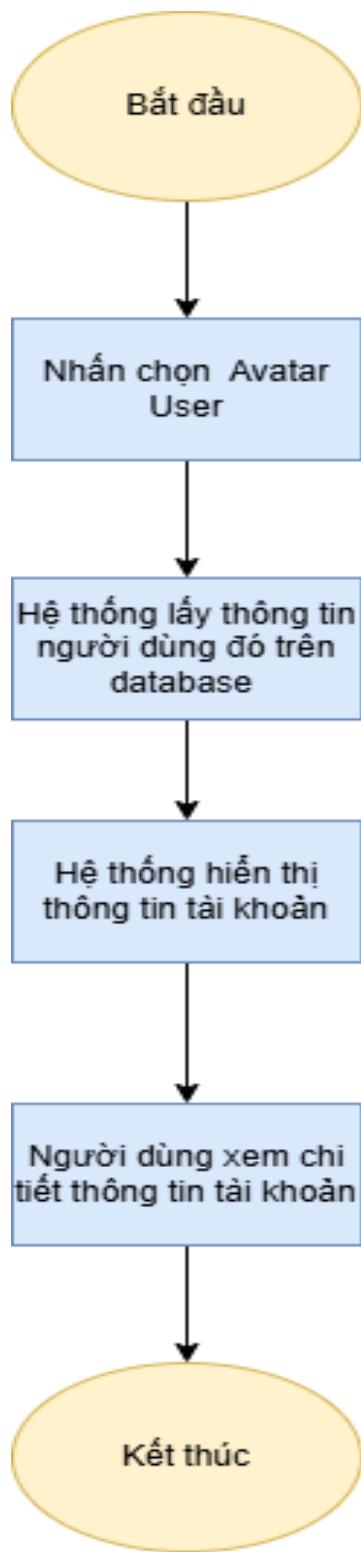


Hình 24. Flowchart cho Use case "Add New Product"

## Use case Customers

Bảng 14. Use case Customers

<b>Định nghĩa</b>	<b>Đối tượng</b>	<b>Luồng hoạt động chính</b>	<b>Luồng hoạt động khác</b>	<b>Điều kiện</b>	<b>Kết quả</b>
Mô tả quy trình Admin xem các khách hàng, và đơn vé của các khách hàng	Admin	<ul style="list-style-type: none"> <li>- Bắt đầu: người dùng chọn “Customers” trong trang giao diện chính.</li> <li>- Hệ thống lọc dữ liệu User trên database và hiển thị về.</li> <li>- Người dùng nhấn chọn trực tiếp hoặc tìm kiếm ở thanh “Search customers”</li> <li>- Hệ thống tìm kiếm dữ liệu User đó trên database.</li> <li>- Hệ thống hiển thị thông tin các đơn vé mà người dùng đó đã mua trên 1 cửa sổ mới.</li> <li>- Use case kết thúc</li> </ul>	Không có	<ul style="list-style-type: none"> <li>- Người dùng đã đăng nhập</li> <li>- Thiết bị có kết nối internet.</li> <li>- Dữ liệu User có trên hệ thống</li> </ul>	Người dùng có thể xem thông tin từng khách hàng.



Hình 25. Flowchart cho Use case "Customers"

### 5.3. Các bảng dữ liệu:

a. Bảng Admin:

*Bảng 15. Dữ liệu Admin*

STT	Tên thuộc tính	Ý nghĩa
1	_id	Trường _id của Admin, được tạo tự động, không trùng nhau
2	username	Tên đăng nhập
3	password	Mật khẩu
4	role	Vai trò

b. Bảng Users:

*Bảng 16. Dữ liệu Users*

STT	Tên thuộc tính	Ý nghĩa
1	_id	Trường _id của User, được tạo tự động, không trùng nhau
2	email	Email cá nhân của mỗi user, có thể trùng lặp
3	username	Tên đăng nhập của mỗi user, có thể trùng lặp
4	password	Mật khẩu, có thể trùng lặp
5	match_info	Thông tin của trận đấu mà user đã mua

6	home_team	Đội nhà
7	away_team	Đội khách
8	date	Ngày diễn ra trận đấu
9	time	Thời gian cụ thể diễn ra trận đấu
10	stadium	Sân đấu
11	total	Tổng số tiền mua vé /tất cả các hạng ghế
12	tickets	Các loại vé đã mua
13	area	Hạng ghế
14	quantity	Số lượng vé/hạng ghế
15	price	Giá vé/hạng ghế
16	subtotal	Tổng số tiền mua vé/ hạng ghế

c. Bảng Teams:

Bảng 17. Dữ liệu Teams

STT	Tên thuộc tính	Ý nghĩa
1	_id	Trường _id của Team, được tạo tự động, không trùng nhau

2	team	Tên đội bóng
3	image	Hình ảnh đội bóng
4	matches	Các trận đấu
5	date	Ngày diễn ra trận đấu
6	time	Thời gian cụ thể diễn ra trận đấu
7	home_team	Đội nhà
8	away_team	Đội khách
9	stadium	Sân đấu
10	tickets	Loại vé chia theo các hạng ghế
11	area	Các hạng ghế
12	price	Giá vé/hạng ghế
13	available_seats	Số lượng ghế sẵn có

## CHƯƠNG VI: THIẾT KẾ GIAO DIỆN

### 6.1. Tổng quan về Figma.



Hình 26. Ứng dụng Figma

Figma là một công cụ thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) trực tuyến, hỗ trợ làm việc nhóm trên nền tảng web. Với Figma, người dùng có thể tạo ra các bản thiết kế, nguyên mẫu (prototype) và cộng tác trong thời gian thực mà không cần cài đặt phần mềm.

#### 6.1.1. Tính năng chính của Figma.

- Thiết kế giao diện: Cung cấp đầy đủ các công cụ vẽ, quản lý bố cục và kiểu dáng.
- Hỗ trợ làm việc nhóm: Cho phép nhiều người cùng chỉnh sửa một file thiết kế đồng thời.
- Tạo prototype: Kết nối các màn hình để mô phỏng trải nghiệm người dùng.
- Quản lý thành phần (Components): Tái sử dụng các phần tử thiết kế giúp tiết kiệm thời gian.
- Tích hợp với plugin: Hỗ trợ nhiều plugin mở rộng để tối ưu hóa quy trình làm việc.

### *6.1.2. Ưu điểm khi sử dụng Figma*

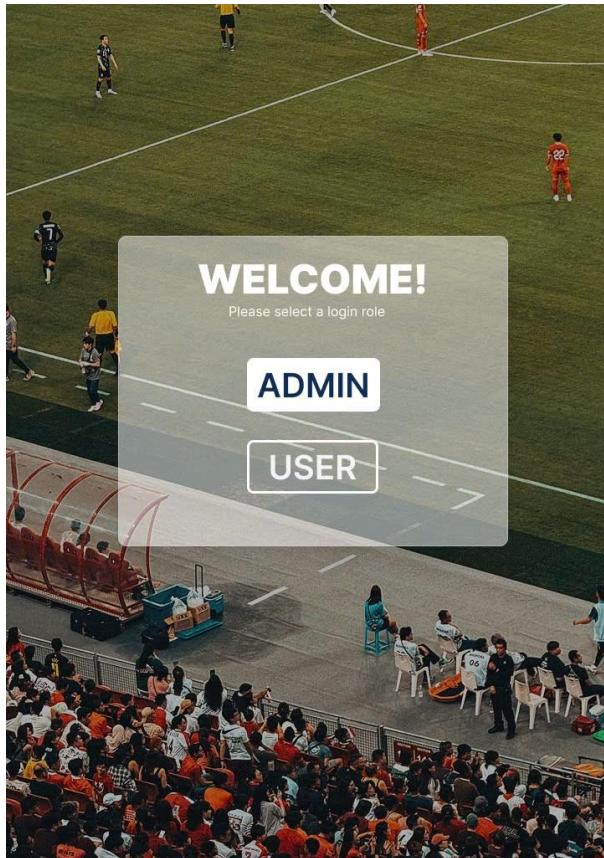
- Làm việc nhóm hiệu quả: Figma cho phép nhiều người cùng chỉnh sửa và xem thiết kế trong thời gian thực, giúp tăng cường sự phối hợp giữa các thành viên.
- Truy cập mọi lúc, mọi nơi: Do hoạt động trên nền tảng web, người dùng có thể làm việc trên bất kỳ thiết bị nào có kết nối internet mà không cần cài đặt phần mềm.
- Giao diện thân thiện, dễ sử dụng: Cung cấp các công cụ thiết kế trực quan, giúp người mới bắt đầu dễ dàng tiếp cận.
- Hỗ trợ tạo prototype nhanh chóng: Cho phép mô phỏng trải nghiệm người dùng mà không cần viết mã, giúp kiểm tra tính khả thi của giao diện.
- Quản lý tài nguyên thiết kế hiệu quả: Hỗ trợ thư viện components giúp tái sử dụng các phần tử giao diện, tiết kiệm thời gian thiết kế.
- Tích hợp plugin đa dạng: Cung cấp nhiều plugin hữu ích để mở rộng tính năng, giúp tối ưu hóa quy trình làm việc.

## 6.2. Thiết kế giao diện Desktop Application bằng Figma: [FIGMA](#)

### 6.2.1. Giao diện Đăng nhập, Đăng ký.

Đăng nhập:

Đăng nhập với vai trò Admin:



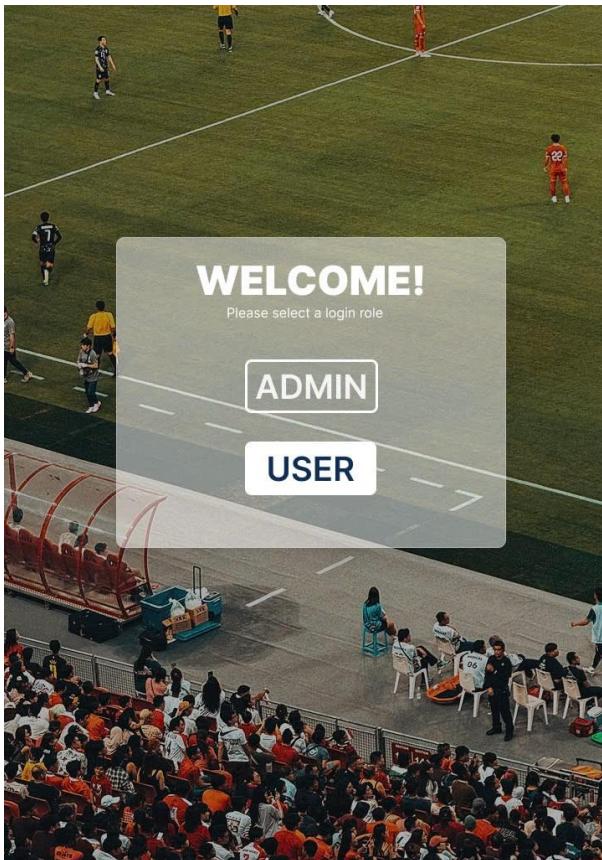
#### Admin Sign in

Username

Password

Hình 27. Admin đăng nhập

Đăng nhập với vai trò User:



### User Sign in

Email

Password

**Sign In**

Contact

OR

Continue with Google

Continue with Apple

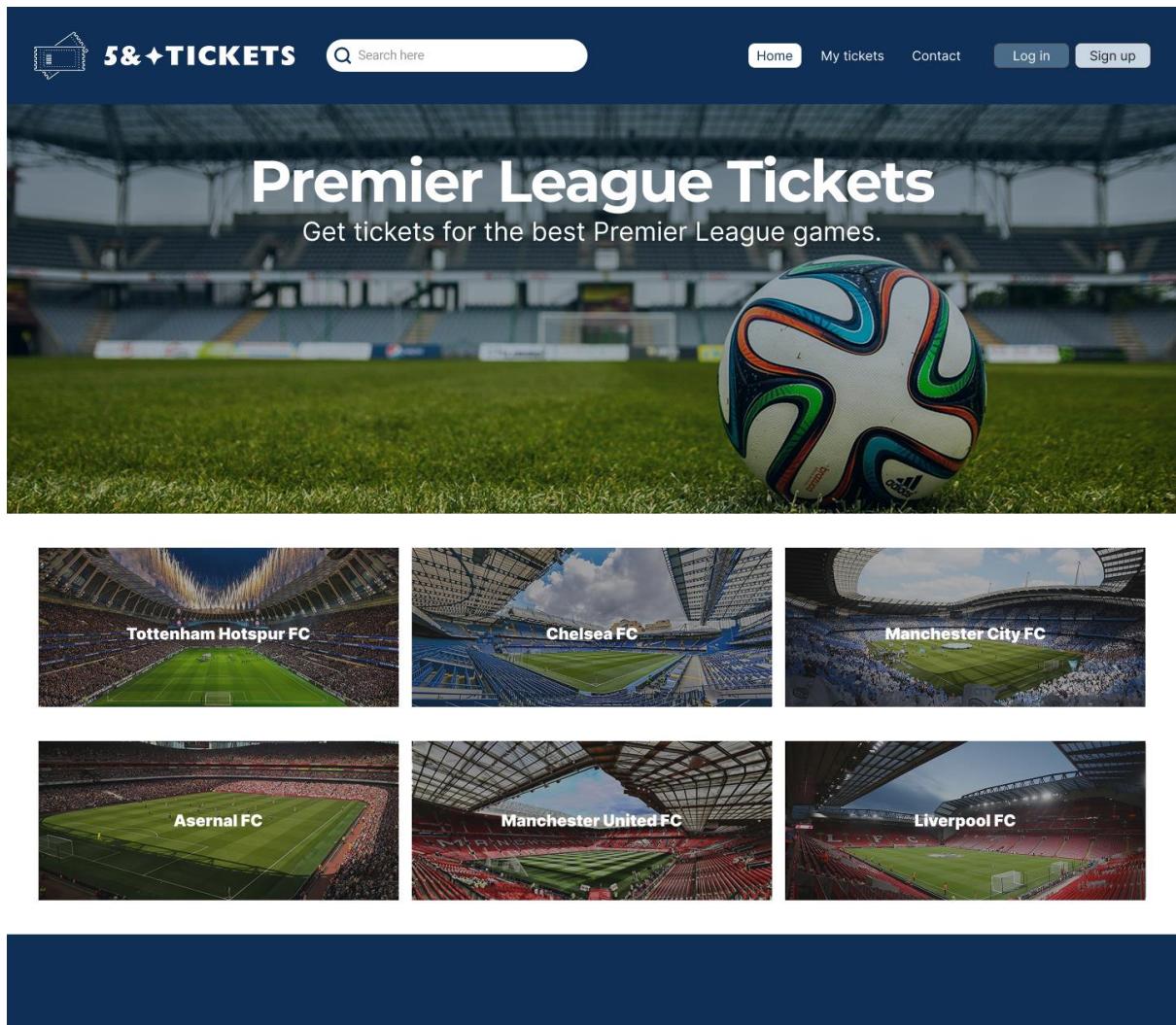
Don't have an account? [Sign up now](#)

S & TICKETS

Hình 28. User đăng nhập

### 6.2.2. Giao diện người dùng (User UI).

Trang chủ (Homepage):



Hình 29. Trang chủ

Trang chi tiết trận đấu:

The screenshot shows the 5&TICKETS website interface. At the top, there is a dark blue header with the 5&TICKETS logo, a search bar, and navigation links for Home, My tickets, Contact, and a user profile icon. Below the header, the main title "Tottenham Hotspur FC Tickets" is displayed, followed by a subtitle "All games 2025 -2026". A large, vibrant photograph of a football stadium at night with fireworks in the background is centered above the game details. Three game listings are shown below the photo:

Date	Match Details	Action
02 Apr Wed 7:45pm	Chelsea FC vs. Tottenham Hotspur FC Stamford Bridge, London	<a href="#">View tickets</a>
05 Apr Sat 3:00pm	Tottenham Hotspur FC vs. Southampton FC Tottenham Hotspur Stadium, London	<a href="#">View tickets</a>
12 Apr Sat 3:00pm	Wolverhampton Wanderers FC vs. Tottenham Hotspur FC Molineux Stadium, Birmingham	<a href="#">View tickets</a>

To the right of the games, there is a "Select your date" section with input fields for "From" and "To" dates, and a "Check availability" button.

Hình 30. User xem các trận đấu

Trang chọn vé:

**Chelsea FC vs. Tottenham Hotspur FC**  
Wed Apr 02 at 7:45pm - Stamford Bridge, London

Ticket quantity  \$244 - \$1,787+

Available tickets

Shortside Upper Tier	\$244 per ticket
Longside Upper Tier	\$258 per ticket
Longside Lower Tier	\$286 per ticket
Shortside Lower Tier	\$348 per ticket
Away Section	\$394 per ticket
VIP Packages	SOLD OUT
VIP Packages The Canoville Suite	SOLD OUT
VIP Packages The Captains Bar	SOLD OUT

**Payment**

**WEST STAND**

**EAST STAND**

**MATTHEW HARDING STAND**

**SHED END**

Legend:  
■ Shortside Upper Tier  
 ■ Longside Upper Tier  
 ■ Longside Lower Tier  
 ■ Shortside Lower Tier  
 ■ Away Section

Hình 31. User lựa chọn vé

Trang thanh toán:

The screenshot shows a ticketing interface for a football match between Chelsea FC and Tottenham Hotspur FC. The main page displays various seating options and their prices. A modal window titled "REVIEW YOUR ORDER" is open, showing the selected tickets for the user.

**Selected Tickets:**

Ticket Type	Quantity	Price
Shortside Upper Tier	1x\$244	\$244
Longside Upper Tier	2x\$258	\$516
Shortside Upper Tier	1x\$244	\$244
Longside Upper Tier	2x\$258	\$516

**Total Payment:** \$ 488

**CONFIRM PAYMENT**      **CANCEL**

**Seating Plan:** The seating plan for the stadium is shown, divided into several sections: WU5, WU6, WU7, WU8, DRAKES, BONETTI, HOLLINS/UTB, WL5, WL6, WL7, WL8, U08, L08, L09, U09, L10, U10, L11, U11, L12, U12, L13, U13, L14, U14, L15, U15, ELN6, OSSES, CHAMPIONS, MANAGERS, EU5, EU7, U16, U17, U18, and U19. The "MATTHEW HARDING STAND" is also labeled on the right side of the plan.

Hình 32. User thanh toán hóa đơn

Trang lịch sử đặt vé:

- Nếu đã đặt:

The screenshot shows a user's purchase history on a website. At the top, there is a header with a logo, a search bar, and navigation links for Home, My tickets, Contact, and a user profile icon. Below the header, the title "PURCHASE HISTORY" is displayed in bold capital letters.

**Chelsea FC vs. Tottenham Hotspur FC**

Date: April 2nd, 2025 Time: 7:45 PM

Address: Stamford Bridge, Chelsea Football Club, London

Section	Quantity	Price (USD)	Subtotal
Shortside Upper Tier	1	\$244	\$244
Shortside Upper Tier	2	\$258	\$516
Shortside Upper Tier	1	\$286	\$286

**Total: \$1,046**

**Liverpool FC vs. Tottenham Hotspur FC**

Date: Feb 7th, 2025 Time: 8:15 PM

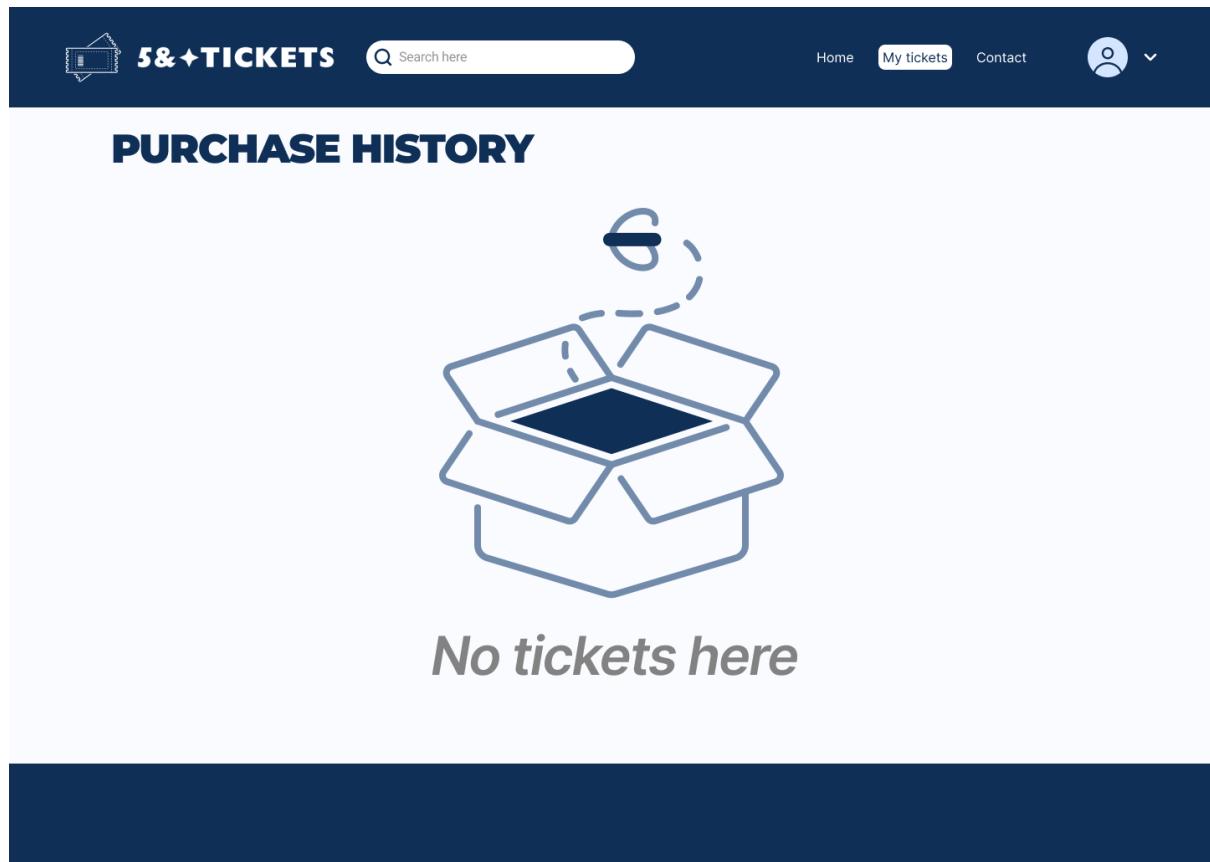
Address: Anfield Stadium, Liverpool Football Club, Liverpool

Section	Quantity	Price (USD)	Subtotal
Shortside Upper Tier	1	\$244	\$244
Shortside Upper Tier	2	\$258	\$516

**Total: \$760**

Hình 33. Lịch sử mua vé của User

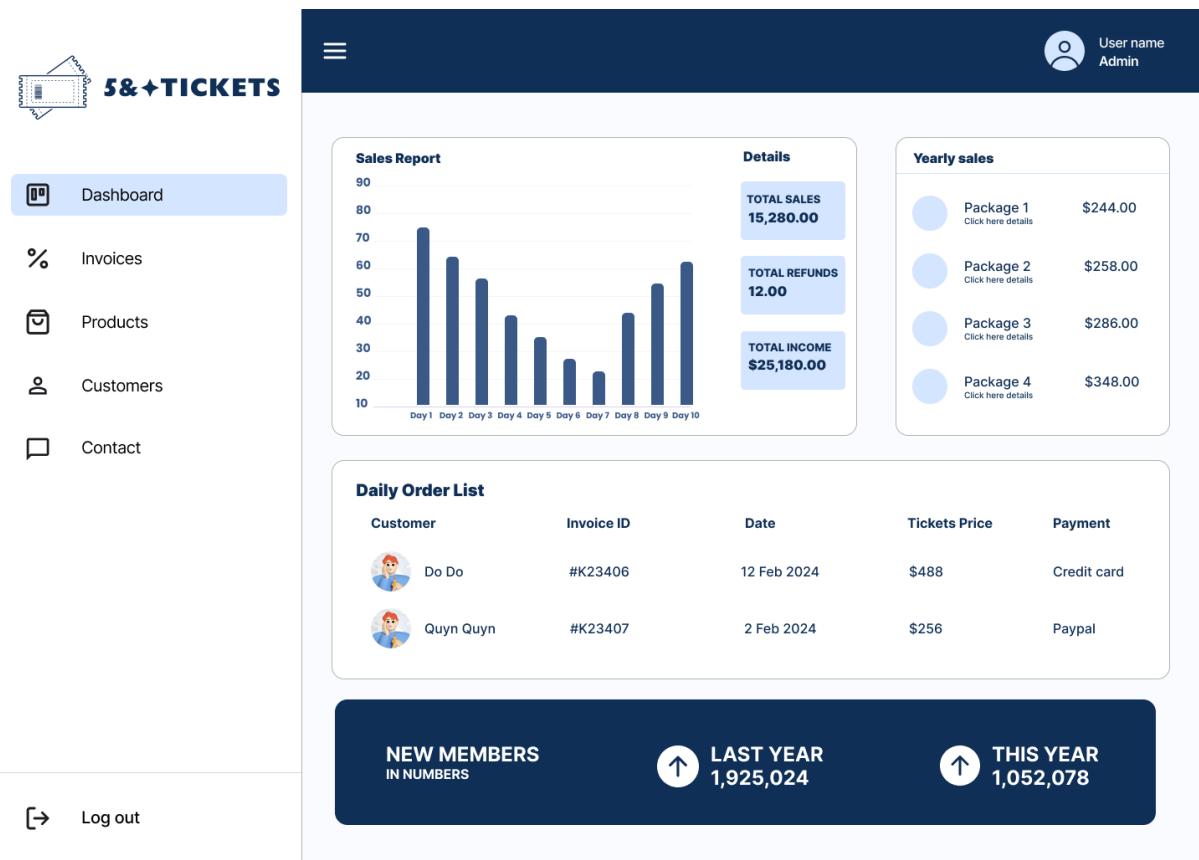
- Nếu chưa đặt:



*Hình 34. User chưa mua vé nào*

### 6.2.3. Giao diện quản trị (Admin UI)

Trang Dashboard:



Hình 35. Admin xem tổng quan phần mềm

Trang sản phẩm:

The screenshot shows the 'Products' section of the application. At the top right, there is a search bar labeled 'Search here', a notification bell icon, and a user profile for 'User name Admin'. Below the header, there are buttons for 'Export', 'Filter', and '+ Add New Product'. The main area displays a table of products with columns: Date, Time, Home team, vs, and Away team. The data in the table is as follows:

Date	Time	Home team	vs	Away team
2025-04-12	3:00 PM	Arsenal FC	vs	Tottenham Hotspur FC
2025-04-12	6:00 PM	Arsenal FC	vs	Manchester City FC
2025-04-12	7:45 PM	Arsenal FC	vs	Chelsea FC
2025-04-12	3:00 PM	Chelsea FC	vs	Liverpool FC
2025-04-12	8:00 PM	Liverpool FC	vs	Manchester City FC
2025-04-12	5:30 PM	Liverpool FC	vs	Arsenal FC
2025-04-12	6:45 PM	Liverpool FC	vs	Tottenham Hotspur FC
2025-04-12	8:30 PM	Manchester City FC	vs	Chelsea FC

On the left sidebar, there are navigation links: Dashboard, Invoices, Products (which is highlighted), Customers, and Contact. A 'Used Space' section shows 'Admin Updated @ 9:00 AM FEB, 22, 2025' and a progress bar indicating '13.5GB / 15GB'. At the bottom left, there is a 'Log out' button.

Hình 36. Admin xem các trận đấu

## Trang hóa đơn:

The screenshot shows a web-based application interface for managing tickets and invoices. The top navigation bar includes a search bar, a notification bell icon, and a user profile for 'User name Admin'. The left sidebar has a logo '5 & TICKETS' and links for Dashboard, Invoices (which is selected and highlighted in blue), Products, Customers, and Contact. A 'Used Space' section indicates 13.5GB / 15GB used. The main content area is titled 'Invoices' and shows a table of match invoices. The table columns are Match List, Date, Time, and Revenue. The data in the table is as follows:

Match List	Date	Time	Revenue
Tottenham Hotspur FC vs Southampton FC	2025-04-12	3:00 PM	\$1,000,000
Wolverhampton Wanderers FC vs Tottenham Hotspur FC	2025-04-12	6:00 PM	\$500,000
Arsenal FC vs Tottenham Hotspur FC	2025-04-12	7:45 PM	\$700,000
Arsenal FC vs Manchester City FC	2025-04-12	3:00 PM	\$1,000,000
Liverpool FC vs Chelsea FC	2025-04-12	8:00 PM	\$1,000,000
Liverpool FC vs Arsenal FC	2025-04-12	5:30 PM	\$1,000,000
Liverpool FC vs Manchester City	2025-04-12	6:45 PM	\$1,000,000
Manchester City vs Tottenham Hotspur FC	2025-04-12	8:30 PM	\$1,000,000

Pagination at the bottom shows 'Previous' and 'Next' buttons, and page numbers 01, 02, 03, ..., 10, 11.

Hình 37. Admin xem các hóa đơn

Trang khách hàng:

The screenshot shows a web-based application for managing customers. The left sidebar has a logo '5 & TICKETS' and navigation links: Dashboard, Invoices, Products, Customers (highlighted with a blue background), and Contact. A 'Used Space' section shows 'Admin Updated @ 9:00 AM FEB, 22, 2025' and a progress bar at 13.5GB / 15GB. A 'Log out' button is also in the sidebar. The main area is titled 'Customers' with a search bar and a 'New Contact' button. It displays a grid of 16 customer profiles in a 4x4 layout. Each profile card includes a small profile picture, the customer's name, and their contact information (phone number and email). The names of the customers listed are Vu Tran Quang, Thi Nguyen Le, Quyen Le Le, Khanh Nguyen Le Nhat, Phuong Nguyen Thi Bich, Wilhelm Grimm, Jacob Grimm, Charles Dickens, Christian Dior, Louis Vuitton, Coco Chanel, and Gianni Versace.

Name	Contact Information
Vu Tran Quang	+12 345 6789 0 vutq@mail.com
Thi Nguyen Le	+12 345 6789 0 thinl@mail.com
Quyen Le Le	+12 345 6789 0 quyenll@mail.com
Khanh Nguyen Le Nhat	+12 345 6789 0 khanhnln@mail.com
Phuong Nguyen Thi Bich	+12 345 6789 0 phuongntb@mail.com
Wilhelm Grimm	+12 345 6789 0 wilhelmgm@mail.com
Jacob Grimm	+12 345 6789 0 jacobg@mail.com
Charles Dickens	+12 345 6789 0 charlesd@mail.com
Christian Dior	
Louis Vuitton	
Coco Chanel	
Gianni Versace	

Hình 38. Admin xem danh sách khách hàng

## CHƯƠNG VII: KẾT QUẢ CHƯƠNG TRÌNH

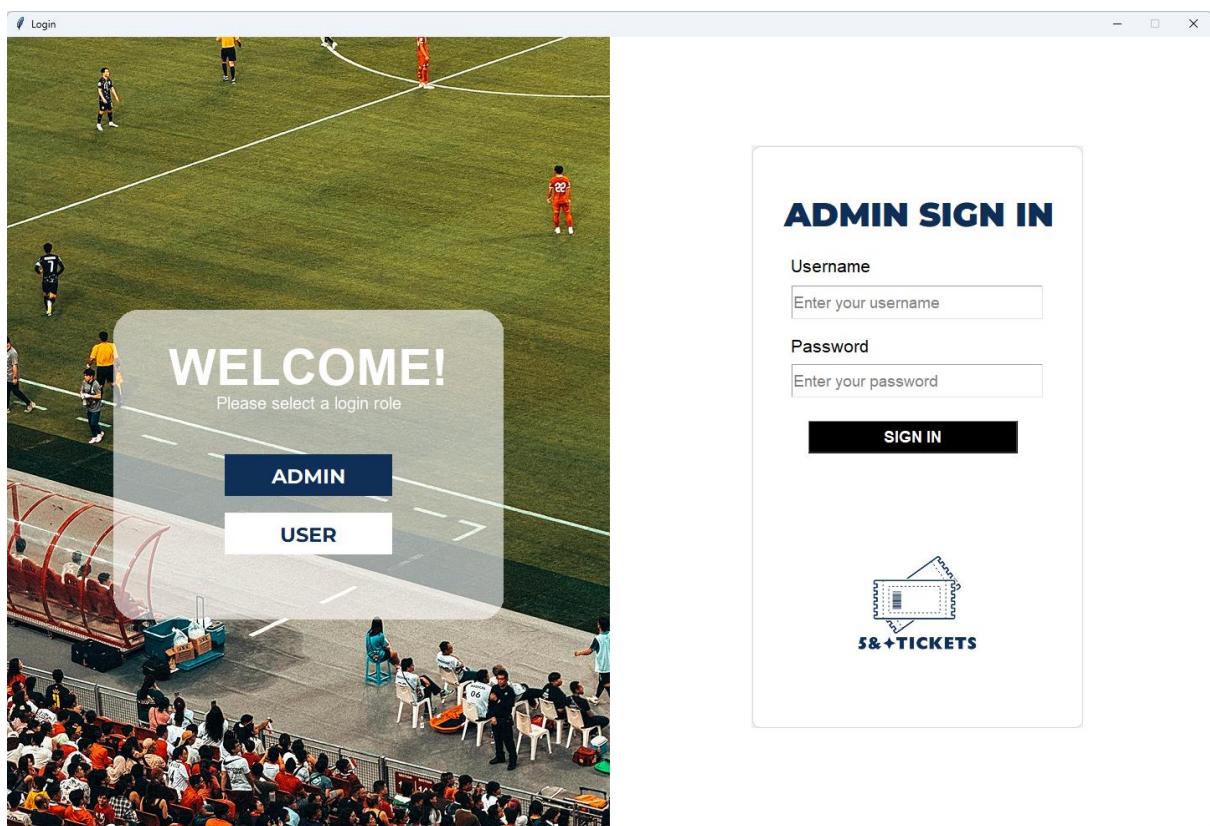
Trước hết ta cần đăng ký, đăng nhập với vai trò User hoặc Admin:

Người dùng chỉ được đăng ký tài khoản User

Người dùng có thể đăng nhập bằng tài khoản User mình đăng ký, hoặc tài khoản Admin sẵn có của hệ thống.

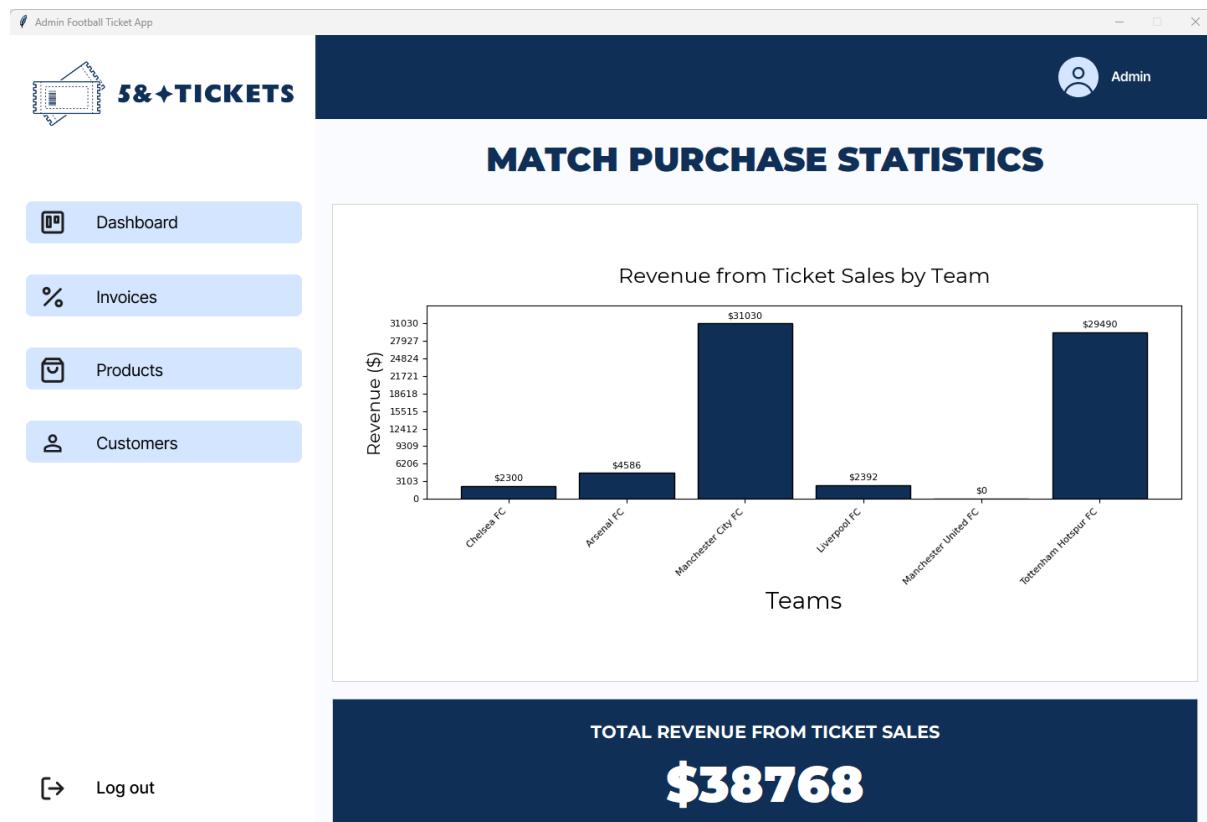
### 7.1. Đăng nhập vai trò Admin:

Đăng nhập: (Giao diện login của Admin)



Hình 39. Giao diện đăng nhập của Admin

Chọn tính năng xem doanh thu: (Giao diện Dashboard)



Hình 40: Giao diện Dashboard

Chọn tính năng xem vé các trận đã bán (Giao diện Invoices)

The screenshot shows a Windows application window titled "Admin Football Ticket App". The main header features a logo with a ticket icon and the text "5&TICKETS". On the right, there is a user profile placeholder and the word "Admin". The left sidebar contains four menu items: "Dashboard", "Invoices" (which is currently selected and highlighted in blue), "Products", and "Customers". At the bottom left is a "Log out" button. The main content area is titled "Invoices" and includes a search bar with the placeholder "Search matches...". Below the search bar is a table listing 14 football matches with their details and revenue. A dropdown menu at the bottom right allows sorting by "Revenue (High to Low)".

Date	Time	Home Team	vs	Away Team	Revenue
2025-05-04	7:30 PM	Manchester City FC	vs	Tottenham Hotspur FC	\$29490
2025-05-17	6:30 PM	Chelsea FC	vs	Manchester City FC	\$1360
2025-05-24	7:45 PM	Manchester City FC	vs	Chelsea FC	\$180
2025-05-03	7:00 PM	Chelsea FC	vs	Arsenal FC	\$0
2025-05-10	8:00 PM	Arsenal FC	vs	Chelsea FC	\$0
2025-05-18	6:30 PM	Arsenal FC	vs	Liverpool FC	\$0
2025-05-25	7:45 PM	Liverpool FC	vs	Arsenal FC	\$0
2025-10-07	02:00 PM	Manchester City FC	vs	Arsenal FC	\$0
2025-08-09	06:00 AM	Tottenham Hotspur FC	vs	Arsenal FC	\$0
2025-05-11	6:00 PM	Tottenham Hotspur FC	vs	Manchester City FC	\$0
2025-05-05	8:00 PM	Liverpool FC	vs	Manchester United FC	\$0
2025-05-12	7:00 PM	Manchester United FC	vs	Liverpool FC	\$0
2025-06-01	03:00 PM	Liverpool FC	vs	Manchester United FC	\$0

Sort by Revenue (High to Low) ▾

Hình 41: Giao diện Invoices

Chọn xem chi tiết vé các trận đã bán (Pop up Invoices)

The screenshot shows the Admin Football Ticket App interface. On the left, there's a sidebar with icons for Dashboard, Invoices (selected), Products, Customers, and Log out. The main area has a dark header with the text "5 & TICKETS" and "Admin". A pop-up window titled "Invoices" is displayed, containing search bar and match details for "Manchester City FC vs Tottenham Hotspur FC" on "2025-05-04, 7:30 PM, Etihad Stadium". It shows a total revenue of "\$29490". Two tables list ticket sales:

Customer: xinchao   Total: \$12710			
Section	Quantity	Price	Subtotal
Shortside Upper Tier	23	\$170	\$3910
Longside Upper Tier	44	\$200	\$8800

Customer: emvu   Total: \$16780			
Section	Quantity	Price	Subtotal
Longside Lower Tier	34	\$280	\$9520
Shortside Lower Tier	22	\$330	\$7260

At the bottom of the pop-up, there's a sorting option: "Sort by Revenue (High to Low)".

Hình 42: Pop up Invoices

Chọn tính năng xem thông tin các trận đấu (Giao diện Products)

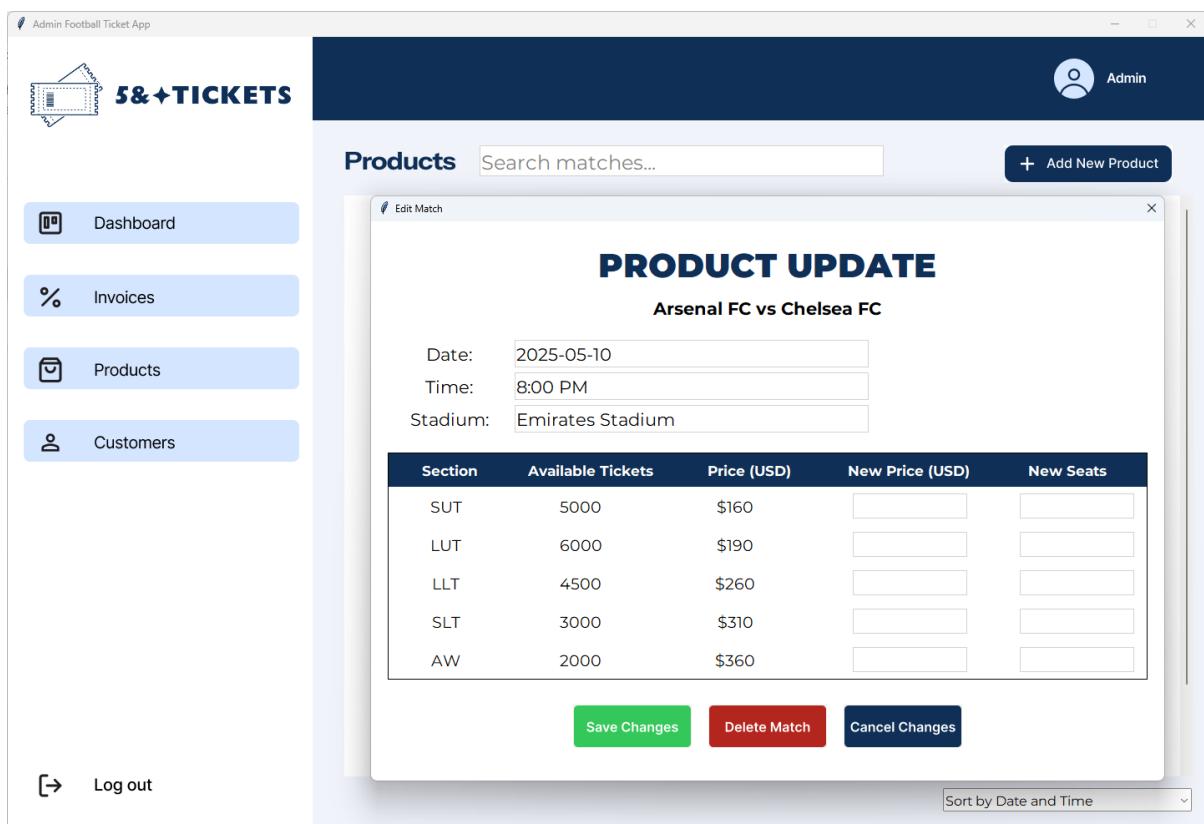
The screenshot shows the Admin Football Ticket App interface. On the left, there's a sidebar with icons for Dashboard, Invoices, Products (which is selected and highlighted in blue), and Customers. At the bottom of the sidebar is a Log out button. The main area is titled "Products" and contains a search bar with placeholder text "Search matches...". A "Add New Product" button is located in the top right corner of the main area. Below the search bar is a table listing football matches:

Date	Time	Home Team	vs	Away Team	Stadium
2025-05-03	7:00 PM	Chelsea FC	vs	Arsenal FC	Stamford Bridge
2025-05-04	7:30 PM	Manchester City FC	vs	Tottenham Hotspur FC	Etihad Stadium
2025-05-05	8:00 PM	Liverpool FC	vs	Manchester United FC	Anfield
2025-05-10	8:00 PM	Arsenal FC	vs	Chelsea FC	Emirates Stadium
2025-05-11	6:00 PM	Tottenham Hotspur FC	vs	Manchester City FC	Tottenham Hotspur Stadium
2025-05-12	7:00 PM	Manchester United FC	vs	Liverpool FC	Old Trafford
2025-05-17	6:30 PM	Chelsea FC	vs	Manchester City FC	Stamford Bridge
2025-05-18	6:30 PM	Arsenal FC	vs	Liverpool FC	Emirates Stadium
2025-05-19	8:00 PM	Manchester United FC	vs	Tottenham Hotspur FC	Old Trafford
2025-05-24	7:45 PM	Manchester City FC	vs	Chelsea FC	Etihad Stadium
2025-05-25	7:45 PM	Liverpool FC	vs	Arsenal FC	Anfield
2025-05-26	6:30 PM	Tottenham Hotspur FC	vs	Manchester United FC	Tottenham Hotspur Stadium
2025-06-01	03:00 PM	Liverpool FC	vs	Manchester United FC	Emirates Stadium

At the bottom right of the main area, there's a "Sort by Date and Time" dropdown menu.

Hình 43: Giao diện Products

Chọn xem và chỉnh sửa trận đấu (Pop up Product Update)



Hình 44: Pop up Product Update

Chọn tính năng thêm trận đấu (Giao diện Add New Products)

The screenshot shows the 'Admin Football Ticket App' interface. At the top right, there is a user icon labeled 'Admin'. On the left, a sidebar menu includes 'Dashboard', 'Invoices', 'Products' (which is highlighted in blue), and 'Customers'. The main content area is titled 'Add New Products' and contains a form for entering basic information about a product. The form includes fields for 'Home Team' (dropdown), 'Visiting team' (dropdown), 'Stadium' (dropdown), 'Date' (YYYY-MM-DD input), 'Time' (HH:MM AM/PM input), and 'Section' (grid with 'Shortside Upper Tier', 'Longside Upper Tier', 'Longside Lower Tier', and 'Shortside Lower Tier' sections, each with 'Seats' and 'Price' inputs). A 'Save' button is at the bottom right of the form.

Hình 45: Giao diện Add New Products

Chọn tính năng xem thông tin khách hàng (Giao diện Customers)

The screenshot shows the 'Customers' section of the application. At the top right, there is a user icon labeled 'Admin'. On the left, a sidebar menu includes 'Dashboard', 'Invoices', 'Products', and 'Customers', with 'Customers' being the active tab. A search bar at the top says 'Search customers...'. Below the search bar, there are two rows of four customer profiles each. Each profile consists of a circular icon with a yellow and blue logo, the customer's name, and their email address. The names and emails listed are:

Name	Email
phuonggg	phuonggg@gmail.com
Khanh	Khanh111@gmail.com
GROUP8	Group8@gmail.com
lethi	lethi@gmail.com
quynh	quynh@gmail.com
xinchao	xinchao@gmail.com
leq	lquyen@gmail.com
emvu	vu@gmail.com.vn

At the bottom left of the main content area, there is a 'Log out' button.

Hình 46: Giao diện Customers

Chọn xem chi tiết thông tin khách hàng (Pop up Customers)

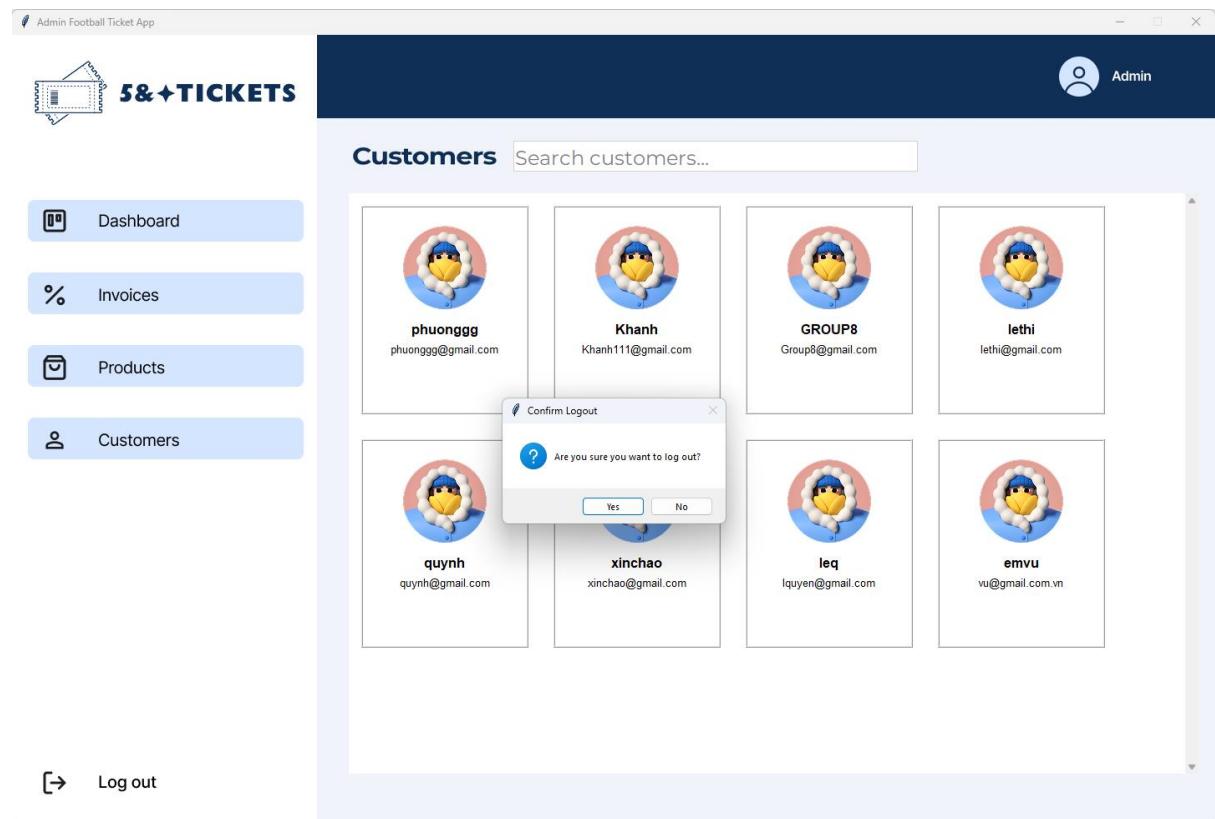
The screenshot shows the Admin Football Ticket App interface. On the left, there's a sidebar with navigation options: Dashboard, Invoices, Products, and Customers (which is currently selected). The main area is titled 'Customers' and includes a search bar. A pop-up window titled 'EMVU'S TICKETS' displays two separate ticket purchases. The first purchase is for 'Manchester City FC vs Chelsea FC' on 2025-05-24 at Etihad Stadium, totaling \$180. The second purchase is for 'Manchester City FC vs Tottenham Hotspur FC' on 2025-05-04 at Etihad Stadium, totaling \$16780. Both purchases include a table with columns for Section, Quantity, Price (USD), and Subtotal.

Section	Quantity	Price (USD)	Subtotal
Shortside Upper Tier	1	\$180	\$180

Section	Quantity	Price (USD)	Subtotal
Longside Lower Tier	34	\$280	\$9520
Shortside Lower Tier	22	\$330	\$7260

Hình 47: Pop up Customers

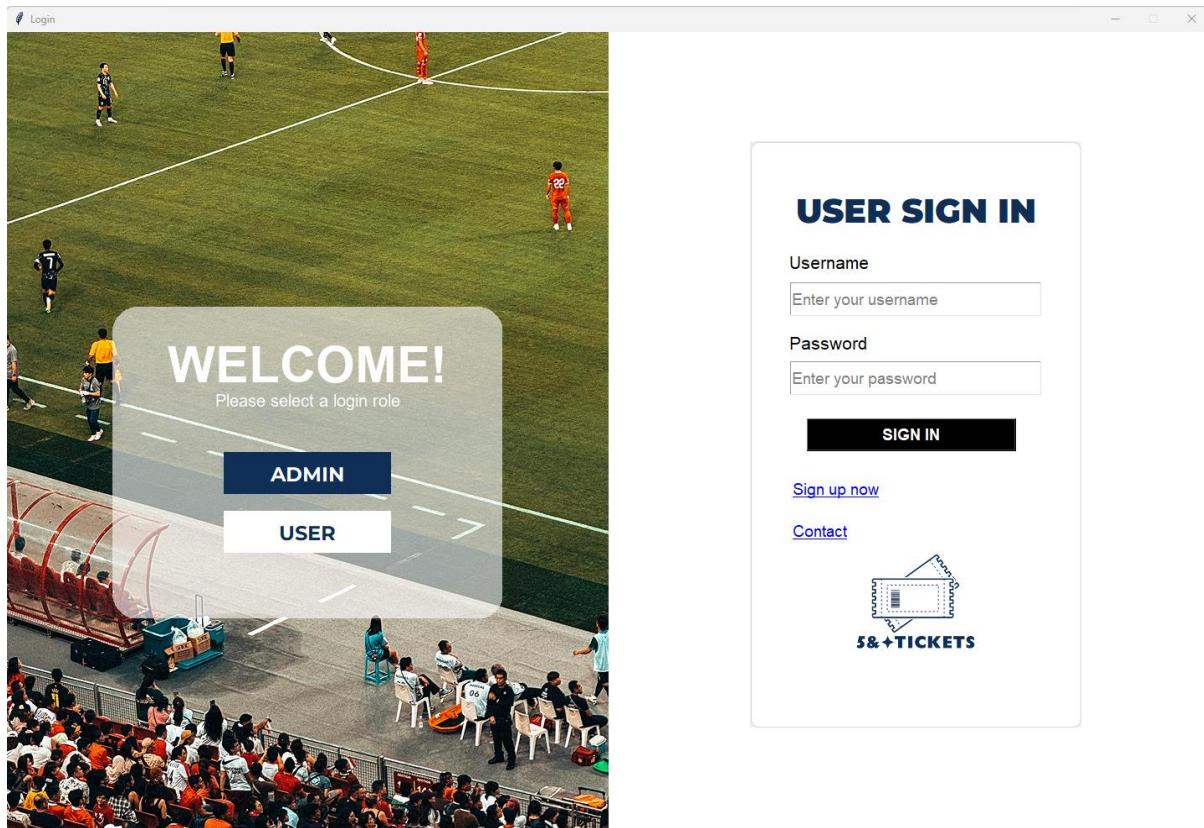
Chọn tính năng đăng xuất (Giao diện Log out)



Hình 48: Giao diện Log out

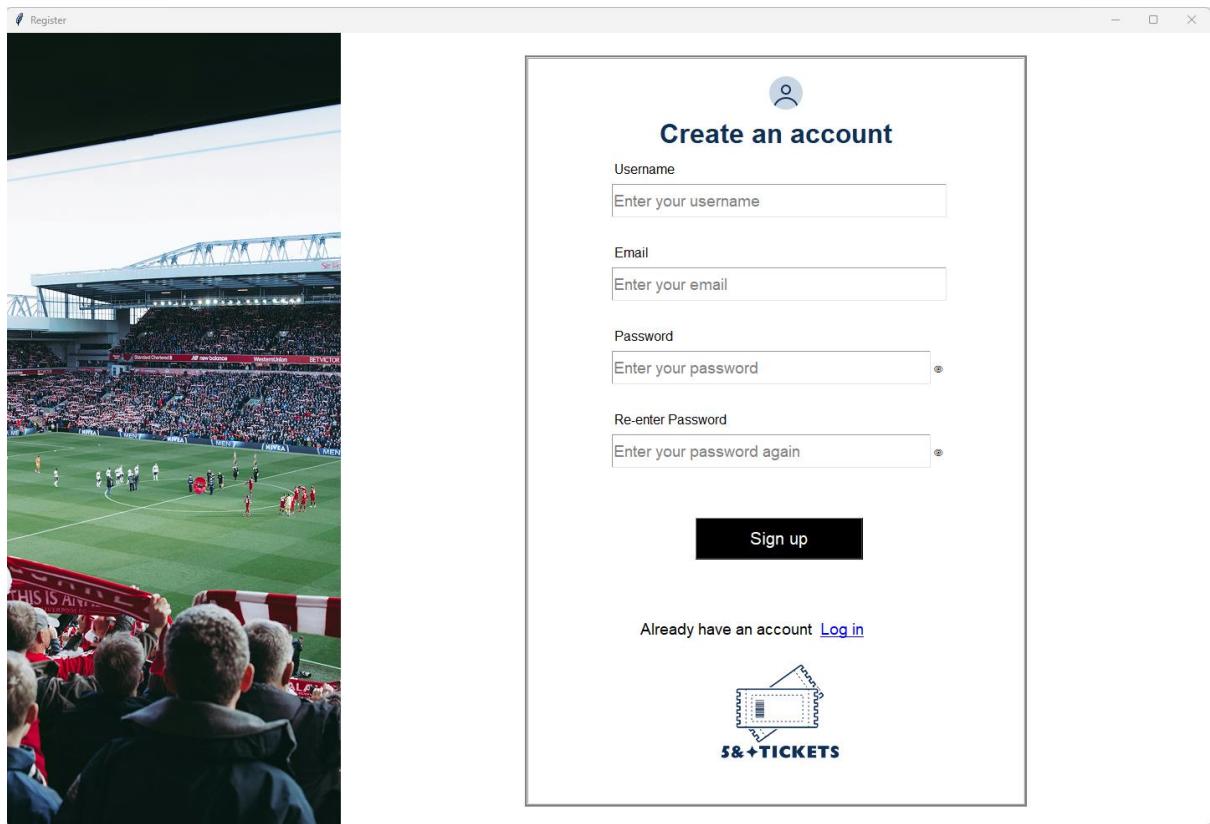
## 7.2. Đăng nhập với vai trò User

Đăng nhập (Giao diện Sign in User)



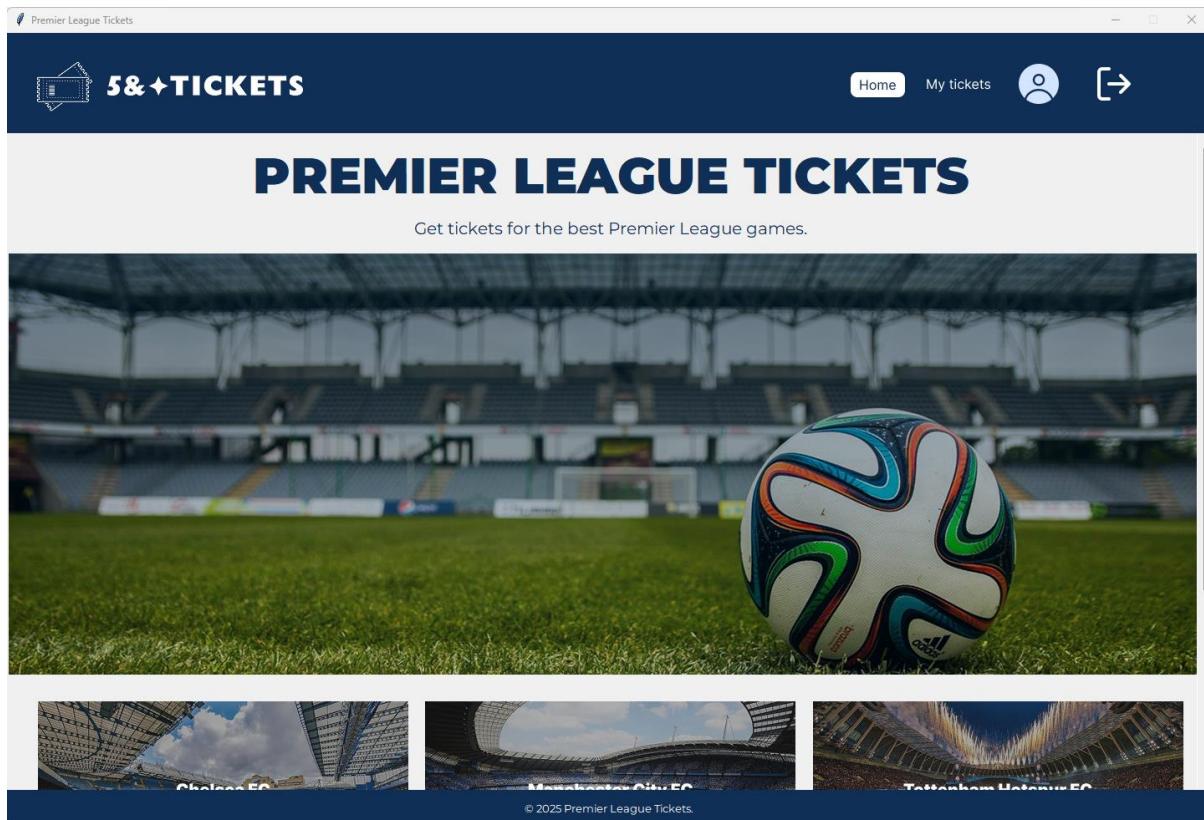
Hình 49: Giao diện Sign in User

## Đăng ký tài khoản (Giao diện Sign up)



Hình 50: Giao diện Sign up

Trang chủ các đội bóng (Giao diện Landing Page)



Hình 51: Giao diện Landing page

Chọn tính năng xem vé (Giao diện My Tickets)

The screenshot shows a web browser window for '5&TICKETS' with the title 'PURCHASE HISTORY'. It displays two purchase records:

**Manchester City FC vs Chelsea FC**  
Date: 2025-05-24  
Time: 7:45 PM  
Address: Etihad Stadium

Section	Quantity	Price (USD)	Subtotal
Shortside Upper Tier	1	\$180	\$180

**Total: \$180**

**Manchester City FC vs Tottenham Hotspur FC**  
Date: 2025-05-04  
Time: 7:30 PM  
Address: Etihad Stadium

Section	Quantity	Price (USD)	Subtotal
Longside Lower Tier	34	\$280	\$9520
Shortside Lower Tier	22	\$330	\$7260

**Total: \$16780**

Hình 52: Giao diện My Tickets

Chọn tính năng xem thông tin (Giao diện Information Customer)

The screenshot shows a web application window titled "Information Customer". At the top, there is a header bar with the "5&TICKETS" logo, navigation links for "Home" and "My tickets", and user icons for profile and logout. Below the header, the main title "INFORMATION CUSTOMER" is displayed in large, bold, dark blue capital letters. To the left of the title is a circular profile placeholder icon. To the right of the placeholder are three lines of text: "Username: emvu", "Email: vu@gmail.com.vn", and "Password: \*\*\*". A "Show" button is positioned next to the password field. Below this section, a dark blue callout box contains the heading "Message to Our Customers" in white. The message text reads: "Thank you for choosing 5& Tickets! We are committed to providing you with the best experience. Your satisfaction is our top priority. Feel free to reach out to us anytime! Have any questions? Don't hesitate to click the button below to contact us!" At the bottom of the callout box is a white button with the word "CONTACT" in black capital letters.

Hình 53: Giao diện Information Customer

Chọn tính năng xem trận đấu của đội bóng (Giao diện View Product)

**Chelsea FC Tickets**

2025-05-03 Chelsea FC vs. Arsenal FC  
7:00 PM Stamford Bridge [View tickets](#)

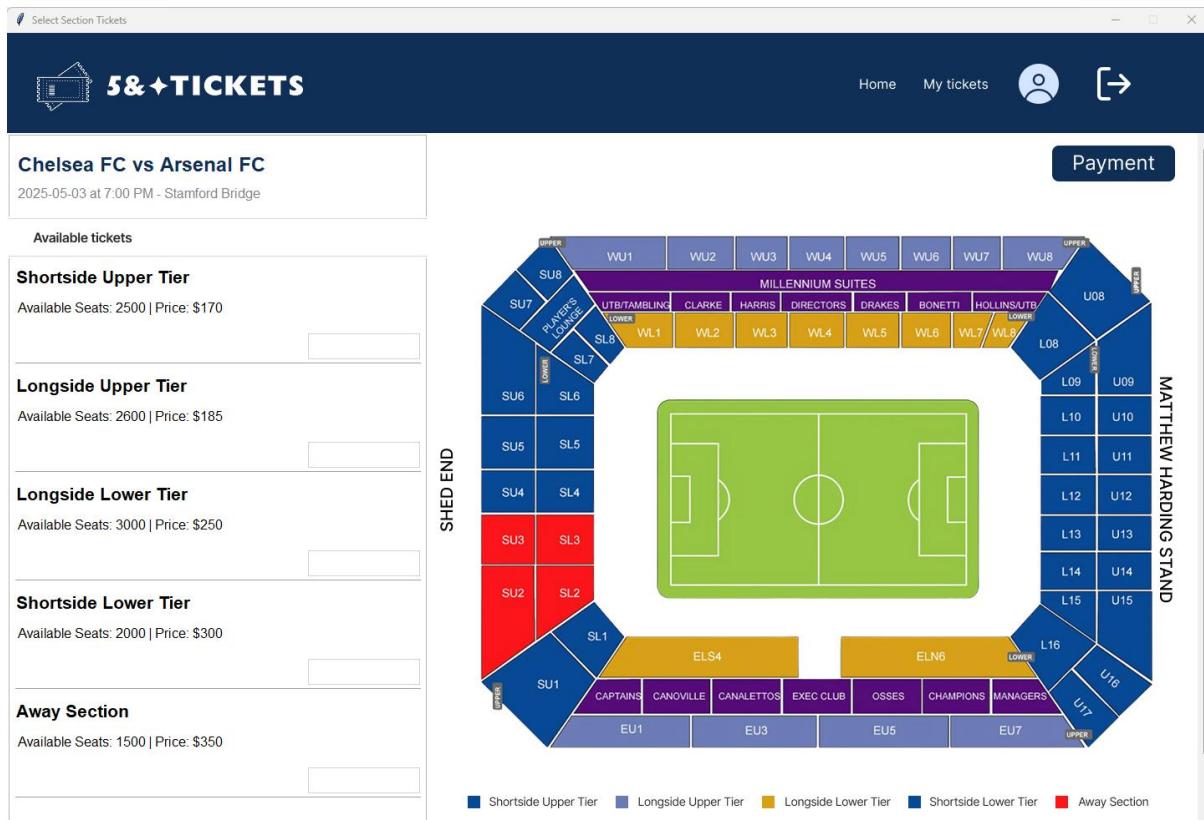
2025-05-10 Arsenal FC vs. Chelsea FC  
8:00 PM Emirates Stadium [View tickets](#)

2025-05-17 Chelsea FC vs. Manchester City FC  
6:30 PM Stamford Bridge [View tickets](#)

2025-05-24 Manchester City FC vs. Chelsea FC  
7:45 PM Etihad Stadium [View tickets](#)

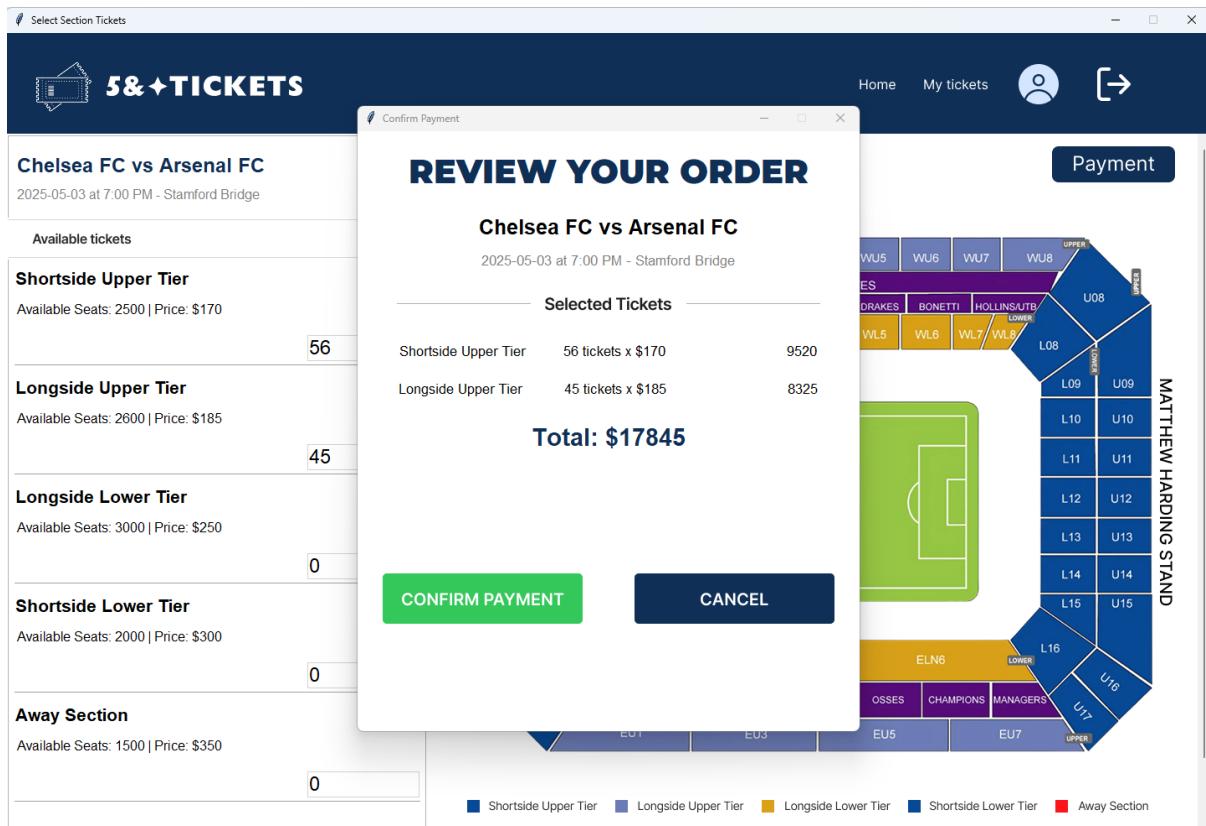
Hình 54: Giao diện View Product

Chọn loại vé và số lượng vé (Giao diện Select Section)



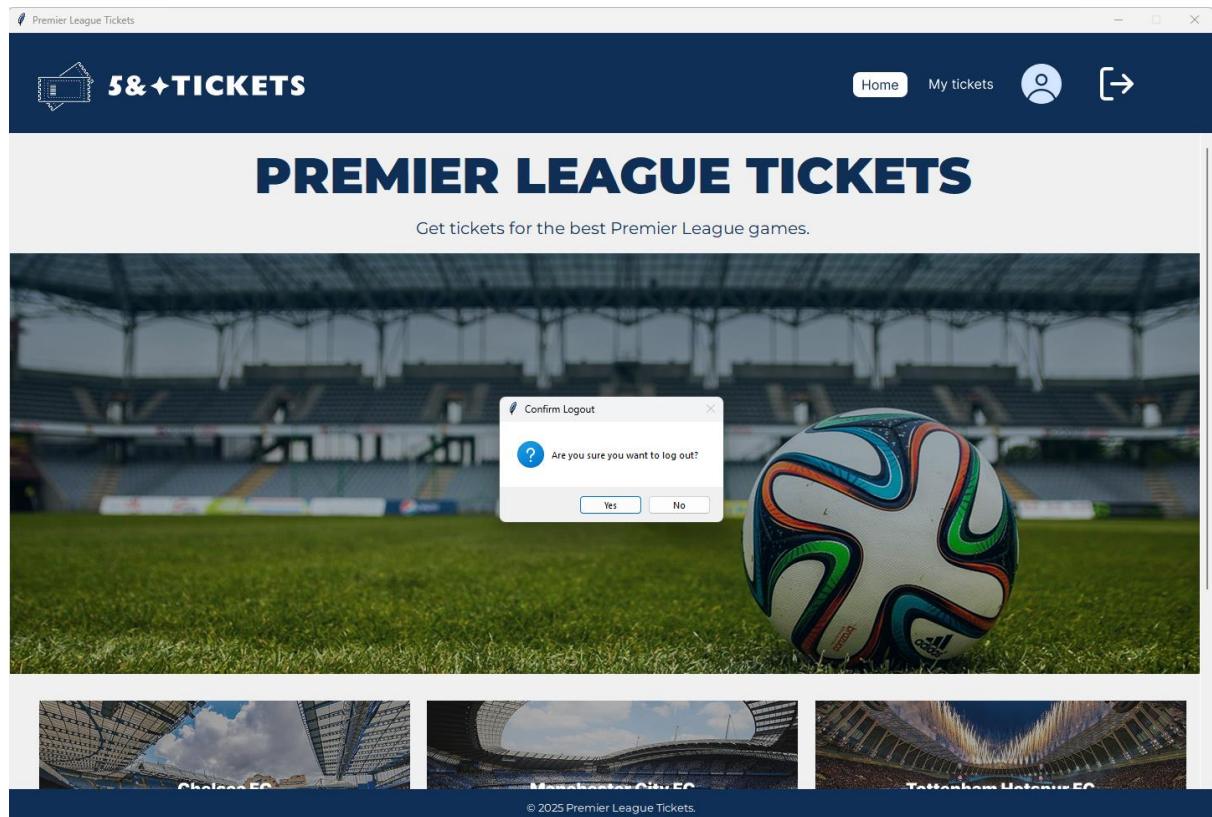
Hình 55: Giao diện Select Section

Chọn chức năng thanh toán (Giao diện Payment)



Hình 56: Giao diện Payment

Chọn chức năng đăng xuất (Giao diện khi Log out)



Hình 57: Giao diện khi Log out

### **7.3. Kết quả đạt được**

Hệ thống quản lý bán vé bóng đá đã được xây dựng với nhiều tính năng quan trọng, trong đó có việc phân chia rõ ràng vai trò giữa User và Admin theo hướng thiết kế hướng đối tượng. Giao diện trực quan, dễ sử dụng giúp người dùng thao tác thuận tiện trên nền tảng. Hệ thống hỗ trợ quản lý dữ liệu hiệu quả, đảm bảo tính chính xác và nhất quán đối với các thông tin quan trọng như các thông tin liên quan đến người dùng, vé và trận đấu. Bên cạnh đó, kiến trúc hệ thống được thiết kế linh hoạt, giúp tách biệt giao diện và luồng xử lý (Front end - Back end), tạo điều kiện thuận lợi cho việc mở rộng và tích hợp thêm các tính năng mới trong tương lai. Hệ thống cũng tập trung xây dựng các tính năng hỗ trợ cho phép người dùng linh động không bị gò bó khi sử dụng ứng dụng, ví dụ: Admin có các chức năng cho phép chỉnh sửa thông tin các trận đấu giúp linh hoạt với các tình huống thực tế.

### **7.4. Hạn chế**

Việc đọc và bảo trì vẫn còn là một vấn đề khó khăn. Ngoài ra, cấu trúc cơ sở dữ liệu chưa tối ưu, có thể dẫn đến việc xử lý chậm khi lượng dữ liệu tăng cao. Do đó, cần có phương án cải thiện hiệu suất trong các phiên bản tiếp theo hướng đến việc xử lý lượng dữ liệu lớn.

### **7.5. Hướng phát triển**

Trong tương lai, hệ thống sẽ được phát triển theo nhiều hướng để nâng cao trải nghiệm người dùng. Một trong những mục tiêu quan trọng là tích hợp thanh toán trực tuyến thông qua VNPay, MoMo, giúp giao dịch nhanh chóng và tiện lợi. Đồng thời, hệ thống có thể quản lý thông tin người dùng bằng CCCD để tăng cường tính bảo mật và đồng bộ dữ liệu (nếu hệ thống được áp dụng ở Việt Nam). Ngoài ra, sẽ có phiên bản website cho phép người dùng truy cập dễ dàng hơn. Hệ thống cũng dự kiến tích hợp trang tin tức bóng đá thông qua ‘crawl’ dữ liệu, cập nhật tình hình các trận đấu theo thời gian thực, giúp tối ưu trải nghiệm người dùng, có thể cập nhật tin tức bóng đá. Bên cạnh đó, trạng thái vé sẽ được hiển thị rõ ràng với các trạng thái như Còn vé, Hết vé, Đóng bán để đảm bảo vé chỉ được bán trong thời gian quy định, tích hợp xử lý thời gian thực

và tự động (ví dụ: khi thời gian bán vé của trận đấu hết, thì vé sẽ tự động đóng bán, hoặc khi gần hết vé sẽ thông báo cho Admin để có các phương án kinh doanh. Cuối cùng, hệ thống sẽ bổ sung trang liên hệ với chức năng nhắn tin trực tiếp, giúp người dùng nhận được hỗ trợ nhanh chóng mà không cần sử dụng nền tảng bên thứ ba.

## CHƯƠNG VIII: KẾT LUẬN

Qua quá trình thực hiện đồ án, có thể nhận thấy rằng hệ thống quản lý bán vé bóng đá đã được xây dựng thành công với những tính năng nổi bật, đáp ứng tốt các yêu cầu cơ bản trong việc quản lý bán vé và hỗ trợ tổ chức các sự kiện bóng đá. Việc triển khai hệ thống không chỉ mang lại một giải pháp thiết thực mà còn là cơ hội để học hỏi và tích lũy nhiều kinh nghiệm quý giá trong việc phát triển ứng dụng thực tiễn.

Hệ thống đã được thiết kế theo hướng đối tượng, phân chia rõ ràng giữa các vai trò User và Admin, giúp quản lý hiệu quả các luồng công việc khác nhau. Giao diện trực quan và dễ sử dụng đã tạo điều kiện thuận lợi cho người dùng trong việc thao tác trên nền tảng, từ việc tìm kiếm thông tin trận đấu đến lựa chọn vé và chỗ ngồi. Bên cạnh đó, hệ thống hỗ trợ quản lý dữ liệu một cách hiệu quả, lưu trữ và xử lý các thông tin quan trọng như thông tin người dùng, thông tin vé và lịch thi đấu, đảm bảo tính chính xác và nhất quán trong suốt quá trình vận hành. Kiến trúc hệ thống cũng được xây dựng với sự linh hoạt, tách biệt giữa giao diện và luồng xử lý, tạo nền tảng vững chắc cho việc vận hành ổn định.

Mặc dù đã đạt được những kết quả đáng ghi nhận, hệ thống vẫn còn tồn tại một số hạn chế cần khắc phục. Mã nguồn của ứng dụng chưa được tối ưu theo tiêu chuẩn clean code, dẫn đến khó khăn trong việc đọc hiểu và bảo trì về lâu dài. Đồng thời, cấu trúc cơ sở dữ liệu chưa thực sự tối ưu, khiến thời gian xử lý có thể bị kéo dài khi lượng dữ liệu tăng cao. Từ những thiếu sót này, nhiều bài học quan trọng đã được rút ra, bao gồm: sự cần thiết của việc áp dụng các tiêu chuẩn lập trình ngay từ đầu để đảm bảo tính bền vững của mã nguồn, tầm quan trọng của việc thiết kế cơ sở dữ liệu hiệu quả để đáp ứng nhu cầu xử lý dữ liệu lớn, và giá trị của việc kiểm tra, đánh giá thường xuyên trong quá trình phát triển để kịp thời phát hiện và khắc phục các vấn đề.

Nhìn chung đồ án này không chỉ mang lại một hệ thống quản lý bán vé bóng đá hữu ích mà còn là cơ hội để học hỏi và trưởng thành hơn trong việc nghiên cứu và ứng dụng công nghệ. Những kết quả đạt được cùng với các bài học kinh nghiệm sẽ là nền tảng vững chắc để tiếp tục cải tiến, góp phần nâng cao hiệu quả quản lý và mang lại trải nghiệm tốt hơn cho người hâm mộ bóng đá.

## TÀI LIỆU THAM KHẢO

[1] Data G. 3. Cấu trúc dữ liệu trong python. Geeks Data.

<https://akatekhanh.github.io/Data-structure-python/>

[2] BeautyOnCode. Cấu trúc dữ liệu trong Python. Viblo. Published March 25, 2025. <https://viblo.asia/p/cau-truc-du-lieu-trong-python-QpmleJ6m5rd>

[3] tkinter — Python interface to Tcl/Tk. Python Documentation.

<https://docs.python.org/3/library/tkinter.html>

[4] Hcm23\_Frf\_Emb\_05\_Minhv. Beginner cần biết: Top 30 thư viện Python tốt nhất (Phần 1). CodeLearn. <https://codelearn.io/sharing/top-30-libraries-packages-4-beginner-p1>

[5] Solomon B. Python plotting with Matplotlib (Guide). Published December 1, 2023. <https://realpython.com/python-matplotlib-guide/>

[6] MongoDB. What is NoSQL? NoSQL databases explained. MongoDB.

<https://www.mongodb.com/resources/basics/databases/nosql-explained>

[7] Content Studio. What is MongoDB? Published March 22, 2023.

<https://www.purestorage.com/uk/knowledge/what-is-mongodb.html>

[8] Tkinter – Giới thiệu về Tkinter. (2016, 1 30). Được truy lục từ Phở Code:

<https://phocode.com/python/tkinter/tkinter-gioi-thieu-ve-tkinter/>

[9] Hướng dẫn lập trình giao diện GUI bằng Tkinter trong ngôn ngữ Python - VniTeach - Giáo viên 4.0. VniTeach - Giáo viên 4.0 - Let's share to be shared. Published February 5, 2022. <https://www.vniteach.com/2022/02/05/huong-dan-lap-trinh-giao-dien-gui-bang-tkinter-trong-ngon-ngu-python/>