

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 2



Báo cáo môn học
ĐÁM BẢO CHẤT LƯỢNG PHẦN MỀM

Đề tài:

**TRANG ĐĂNG KÍ THAM GIA CUỘC THI
LẬP TRÌNH THI ĐẤU**

Giảng viên hướng dẫn: ThS. Nguyễn Anh Hào

Sinh viên thực hiện: Nguyễn Nhật Minh – N21DCCN053

Trần Vũ Phương Nam – N21DCCN151

Thành phố Hồ Chí Minh, ngày 15 tháng 06 năm 2025

LỜI CẢM ƠN

Lời đầu tiên, nhóm xin gửi lời cảm ơn chân thành đến thầy Nguyễn Anh Hào, giảng viên Khoa Công nghệ Thông tin 2, đã tận tình giảng dạy, hướng dẫn và tạo điều kiện thuận lợi để nhóm có thể hoàn thành bài báo cáo cuối kỳ này. Sự hỗ trợ quý báu của thầy không chỉ giúp nhóm có thêm nhiều kiến thức bổ ích về chuyên đề Công nghệ Phần mềm, mà còn mang lại những kinh nghiệm thực tế trong quá trình nghiên cứu và phát triển sản phẩm.

Bên cạnh đó, nhóm cũng xin bày tỏ lòng biết ơn đến các thầy cô trong Khoa Công nghệ Thông tin, những người đã truyền đạt những kiến thức nền tảng quan trọng, giúp nhóm có đủ hành trang để thực hiện đề tài "Trang Đăng Ký Tham Gia Cuộc Thi Lập Trình Thi Đấu".

Trong suốt quá trình thực hiện bài báo cáo, nhóm đã cố gắng hết sức để hoàn thiện sản phẩm và nội dung trình bày. Tuy nhiên, do giới hạn về thời gian và kinh nghiệm, nhóm không tránh khỏi những thiếu sót. Vì vậy, nhóm rất mong nhận được sự góp ý và phản hồi từ thầy để có thể chỉnh sửa, hoàn thiện bài báo cáo một cách tốt nhất.

Cuối cùng, nhóm cũng xin gửi lời cảm ơn đến các bạn trong lớp, những người đã luôn đồng hành, hỗ trợ và chia sẻ kiến thức, giúp nhóm có thêm động lực để hoàn thành bài báo cáo này.

Thành phố Hồ Chí Minh, ngày 15 tháng 06 năm 2025

MỤC LỤC

MỤC LỤC	1
DANH SÁCH HÌNH	2
TÓM TẮT.....	4
CHƯƠNG 1. TỔNG QUAN.....	5
1. 1. Giới thiệu đề tài.....	5
1. 2. Vai trò và nhiệm vụ của mỗi thành viên	5
1. 3. Cơ sở lý thuyết	5
CHƯƠNG 2. PHÂN TÍCH & THIẾT KẾ HỆ THỐNG.....	9
2. 1. Yêu cầu từ stake-holders	9
2. 2. Phân tích hệ thống	9
2. 2. 1. Tổng quan hệ thống	9
2. 2. 2. Phân tích yêu cầu.....	10
2. 2. 3. Các usecase chính của hệ thống	11
2. 3. Thiết kế hệ thống	26
2. 3. 1. CRC card của các đối tượng thành phần trong hệ thống.....	26
2. 3. 2. Usecase 1 – Đăng kí tài khoản thí sinh tham gia hệ thống.....	28
2. 3. 3. Usecase 2 – Tạo nhóm tham gia cuộc thi trên hệ thống.....	30
2. 3. 4. Usecase 3 – Đăng kí tham gia cuộc thi trên hệ thống	31
2. 3. 5. Database của phần mềm	32
CHƯƠNG 3. triển khai hệ thống.....	35
3. 1. Phần mềm ứng dụng.....	35
3. 1. 1. Giao diện của ứng dụng phần mềm.....	35
3. 1. 2. Cài đặt hệ thống phần mềm	37
3. 2. Kiểm thử.....	39
3. 2. 1. Đặc tả các testcases cho Usecase 1	39
3. 2. 2. Đặc tả các testcases cho Usecase 2	43
3. 2. 3. Đặc tả các testcases cho Usecase 3	48
3. 2. 4. Hiện thực các testcases.....	52
CHƯƠNG 4. kết luận	66
4. 1. Điểm mạnh và điểm yếu của dự án	66
4. 2. Khả năng cải tiến cách tiến hành đồ án	66

DANH SÁCH HÌNH

Hình 1: Sơ đồ usecase đăng kí tài khoản thí sinh tham gia hệ thống	11
Hình 2: Sơ đồ hoạt động usecase đăng kí tài khoản thí sinh	13
Hình 3: Sơ đồ tuần tự usecase đăng kí tài khoản thí sinh.....	14
Hình 4: Sơ đồ usecase tạo nhóm thi tham gia các cuộc thi	16
Hình 5: Sơ đồ hoạt động usecase tạo nhóm thi tham gia các cuộc thi	17
Hình 6: Sơ đồ tuần tự usecase tạo nhóm thi tham gia các cuộc thi	19
Hình 7: Sơ đồ usecase đăng kí cuộc thi trên hệ thống.....	21
Hình 8: Sơ đồ hoạt động usecase đăng kí tham gia cuộc thi trên hệ thống	22
Hình 9: Sơ đồ tuần tự usecase đăng kí tham gia cuộc thi trên hệ thống.....	24
Hình 10: Hình ảnh form đăng kí tài khoản thí sinh trên hệ thống.....	29
Hình 11: API đăng kí tài khoản thí sinh trên hệ thống	29
Hình 12: Hình ảnh form tạo nhóm thi tham gia các cuộc thi	30
Hình 13: API tạo nhóm thi tham gia các cuộc thi.....	30
Hình 14: Hình ảnh trang chủ hiển thị danh sách các cuộc thi và nút đăng kí thi	31
Hình 15: API đăng kí tham gia cuộc thi	31
Hình 16: Sơ đồ ERD của hệ thống	32
Hình 17: Giao diện trang đăng nhập của ứng dụng.....	35
Hình 18: Giao diện trang đăng kí tài khoản của ứng dụng.....	35
Hình 19: Giao diện trang quên mật khẩu của ứng dụng	36
Hình 20: Giao diện trang chủ của ứng dụng.....	36
Hình 21: Giao diện trang thông tin cá nhân của thí sinh	37
Hình 22: Giao diện form đăng kí tạo nhóm thi.....	37
Hình 23: Cấu hình biến môi trường cho Backend.....	38
Hình 24: Hàm tìm thành phần trên trang đăng kí	53
Hình 25: Hàm điền dữ liệu vào các thành phần nhập liệu trang đăng kí.....	54
Hình 26: Hàm kiểm tra email không đúng định dạng	54
Hình 27: Hàm kiểm tra username đã được sử dụng	55
Hình 28: Hàm kiểm tra mã sinh viên đã sử dụng	55
Hình 29: Hàm kiểm tra email đã sử dụng.....	55
Hình 30: Hàm kiểm tra đăng nhập thành công	56
Hình 31: Hàm tìm các thành phần trang thông tin nhóm	56
Hình 32: Hàm điền thông tin vào các thành phần nhập liệu.....	57
Hình 33: Hàm kiểm tra email không hợp lệ	58

Báo cáo đồ án Trang đăng kí tham gia cuộc thi lập trình thi đấu

Hình 34: Hàm kiểm tra thí sinh không tồn tại	58
Hình 35: Hàm kiểm tra tên nhóm đã tồn tại	59
Hình 36: Hàm tạo nhóm thành công.....	59
Hình 37: Hàm kiểm tra các thông tin điền bị thiếu	60
Hình 38: Hàm kiểm tra đăng kí cuộc thi đơn thành công.....	61
Hình 39: Hàm kiểm tra cuộc thi nhóm thành công.....	62
Hình 40: Hàm kiểm tra ghi danh cuộc thi nhóm nhưng không có nhóm	63
Hình 41: Code đăng nhập để test trong Postman.....	63
Hình 42: API request để kiểm tra thí sinh ghi danh cuộc thi đơn đã ghi danh	64
Hình 43: JS Code kiểm tra kết quả API request khi thí sinh ghi danh 2 lần cho cuộc thi đơn.	64
Hình 44: API request để kiểm tra thí sinh ghi danh cuộc thi nhóm đã ghi danh	64
Hình 45: JS Code kiểm tra kết quả API request khi thí sinh ghi danh 2 lần cho cuộc thi nhóm	64
Hình 46: API request để kiểm tra thí sinh ghi danh cuộc thi không tồn tại.....	65
Hình 47: JS code kiểm tra kết quả API khi thí sinh ghi danh vào cuộc thi không tồn tại	65

TÓM TẮT

Đề tài “Trang web đăng ký tham gia cuộc thi lập trình thi đấu” nhằm xây dựng một nền tảng trực tuyến hỗ trợ sinh viên đăng ký tham gia các cuộc thi lập trình một cách nhanh chóng, thuận tiện và chính xác. Hệ thống cung cấp các chức năng chính như đăng ký tài khoản cá nhân, thành lập nhóm dự thi và đăng ký tham gia cuộc thi. Sinh viên tiến hành đăng ký tài khoản bằng các thông tin cá nhân duy nhất bao gồm mã số sinh viên, email và username; hệ thống đảm bảo tính duy nhất của các thông tin này và sẽ thông báo lỗi nếu có trùng lặp. Đối với việc lập nhóm dự thi, hệ thống cho phép sinh viên tạo các nhóm thi với số lượng thành viên cố định là ba, tuân thủ tiêu chuẩn của cuộc thi lập trình ICPC. Các thành viên phải có tài khoản thí sinh trên hệ thống. Các thành viên trong nhóm đều có quyền tạo hoặc giải tán nhóm, và sau khi giải tán, mỗi thành viên được phép tham gia hoặc lập nhóm mới. Về chức năng tham gia cuộc thi, thí sinh cá nhân chỉ cần đăng ký tài khoản hợp lệ để tham gia cuộc thi cá nhân; trong trường hợp cuộc thi dành cho đội nhóm, bất kỳ thành viên nào của nhóm đều có thể thực hiện thao tác đăng ký và tất cả các thành viên còn lại trong nhóm sẽ được hệ thống tự động đăng ký tham gia.

CHƯƠNG 1. TỔNG QUAN

1. 1. Giới thiệu đề tài

Câu lạc bộ ITMC là câu lạc bộ học thuật dành cho sinh viên có đam mê về công nghệ thông tin và đa phương tiện tại Học viện Công nghệ Bưu chính Viễn thông cơ sở tại TP Hồ Chí Minh. Với mục tiêu nâng cao tư duy thuật toán và kỹ năng lập trình cho các thành viên trong câu lạc bộ cũng như sinh viên trong trường, ITMC thường xuyên tổ chức các cuộc thi lập trình thi đấu cho sinh viên trong và ngoài học viện tham gia.

Hiện tại, câu lạc bộ sử dụng hệ thống DOMJudge – một nền tảng chấm thi theo chuẩn ICPC, giúp đánh giá và chấm điểm bài làm của thí sinh một cách chính xác và tự động. Tuy nhiên, hệ thống này không có chức năng đăng ký thi trực tuyến, dẫn đến việc thí sinh không thể tự đăng ký tham gia mà phải thông qua quản trị viên hệ thống để đăng ký tài khoản tham gia vào hệ thống. Điều này làm giảm tính chủ động của thí sinh trong việc sử dụng hệ thống.

Để giải quyết vấn đề này, nhóm đề xuất phát triển một ứng dụng đăng ký cuộc thi lập trình thi đấu để thí sinh muốn tham gia thi có thể chủ động đăng ký thi mà không cần thông qua quản trị viên để được cung cấp tài khoản thi. Điều này sẽ tạo sự chủ động từ phía thí sinh – chính là đối tượng sử dụng chính của sản phẩm và giảm bớt công việc cho quản trị viên của hệ thống thi hiện tại của CLB.

Mục tiêu của sản phẩm là giải quyết sự phụ thuộc của người sử dụng mới vào quản trị viên của hệ thống hiện tại khi muốn đăng ký tài khoản để tham gia các cuộc thi. Đây cũng chính là chức năng chính của sản phẩm.

1. 2. Vai trò và nhiệm vụ của mỗi thành viên

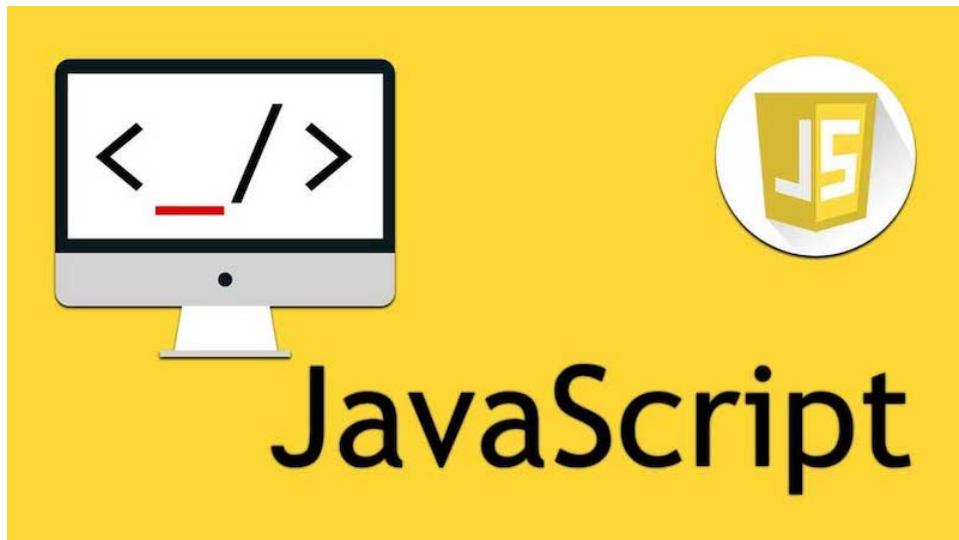
Họ và tên	MSSV	Công việc
Nguyễn Nhật Minh	N21DCCN053	<ul style="list-style-type: none">Viết code phần BackendViết báo cáo phần usecase 1, 3Vẽ diagrams cho usecase 1, 3Viết testcases cho usecase 1, 3
Trần Vũ Phương Nam	N21DCCN151	<ul style="list-style-type: none">Viết code phần FrontendViết báo cáo phần usecase 2Vẽ diagram cho usecase 2Viết testcases cho usecase 2

1. 3. Cơ sở lý thuyết

Ngôn ngữ lập trình Javascript

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

JavaScript là một ngôn ngữ lập trình hướng đối tượng được sử dụng rộng rãi trong phát triển ứng dụng web, đặc biệt là ở phía người dùng (client-side). Ban đầu được phát triển bởi Netscape vào năm 1995, JavaScript ngày nay đã trở thành một phần không thể thiếu trong các ứng dụng web hiện đại. Với khả năng hoạt động trực tiếp trong trình duyệt mà không cần biên dịch, JavaScript giúp tạo ra các giao diện động, tương tác cao và cải thiện trải nghiệm người dùng.



Ngoài vai trò ở phía trình duyệt, JavaScript còn được ứng dụng ở phía máy chủ (server-side) thông qua các nền tảng như Node.js. Một số đặc điểm nổi bật của JavaScript bao gồm khả năng xử lý bất đồng bộ thông qua Promise và async/await, mô hình lập trình hướng sự kiện (event-driven), và khả năng tương tác với DOM (Document Object Model) để cập nhật nội dung trang web theo thời gian thực. JavaScript đóng vai trò then chốt trong việc xây dựng các chức năng như xử lý form đăng ký, gửi yêu cầu API, hiển thị thông báo hoặc xác thực dữ liệu đầu vào trong dự án.

Framework Next.js

Next.js là một framework phát triển dựa trên React.js, được phát triển bởi công ty Vercel, nhằm mục đích hỗ trợ xây dựng các ứng dụng web hiệu suất cao và thân thiện với công cụ tìm kiếm. Với sự kết hợp giữa khả năng kết xuất phía máy chủ (Server Side Rendering – SSR) và khả năng tạo trang tĩnh (Static Site Generation – SSG), Next.js mang đến hiệu quả cao trong việc tối ưu hóa tốc độ tải trang và cải thiện SEO.



Một điểm nổi bật của Next.js là cơ chế định tuyến tự động thông qua cấu trúc thư mục, giúp giảm thiểu cấu hình thủ công cho lập trình viên. Ngoài ra, Next.js còn hỗ trợ tích hợp sẵn các API route, cho phép xây dựng các endpoint phía server trong cùng một dự án mà không cần tách riêng backend. Framework này hỗ trợ tốt cho việc xây dựng giao diện người dùng, xử lý tương tác và quản lý trạng thái ứng dụng.

Trong phạm vi dự án "Trang đăng ký tham gia cuộc thi lập trình thi đấu", Next.js đóng vai trò là nền tảng chính để phát triển giao diện, thực hiện định tuyến các trang như trang đăng ký thí sinh, trang quản lý đội thi, và trang thông tin cuộc thi. Đồng thời, framework này cũng giúp triển khai logic gọi API và xử lý dữ liệu trên cả phía client và server một cách hiệu quả.

Framework Selenium

Selenium là một công cụ mã nguồn mở chuyên dùng để tự động hóa kiểm thử giao diện người dùng trên các ứng dụng web. Được sử dụng rộng rãi trong lĩnh vực kiểm thử phần mềm, Selenium cho phép mô phỏng các hành vi thực tế của người dùng trên trình duyệt, chẳng hạn như nhập dữ liệu, nhấn nút, chuyển trang và kiểm tra phản hồi của hệ thống.



Bộ công cụ Selenium bao gồm nhiều thành phần, trong đó phổ biến nhất là Selenium WebDriver – một thư viện lập trình cho phép điều khiển trình duyệt thông qua mã nguồn viết bằng nhiều ngôn ngữ như Java, Python, C# hoặc JavaScript. Selenium hỗ trợ nhiều trình duyệt khác nhau như Chrome, Firefox, Edge và Safari, giúp kiểm thử chéo trình duyệt trở nên đơn giản và hiệu quả.

Trong dự án này, Selenium được sử dụng để xây dựng các tập lệnh kiểm thử tự động nhằm xác nhận tính đúng đắn của các chức năng như đăng ký tài khoản thí sinh, tạo đội thi, xác thực dữ liệu đầu vào và đảm bảo các nút chức năng hoạt động theo đúng yêu cầu. Việc áp dụng Selenium giúp giảm thiểu rủi ro phát sinh lỗi khi triển khai thực tế và nâng cao chất lượng phần mềm.

CHƯƠNG 2. PHÂN TÍCH & THIẾT KẾ HỆ THỐNG

2. 1. Yêu cầu từ stake-holders

Trong quá trình xác định và phân tích yêu cầu của hệ thống “Trang đăng ký tham gia cuộc thi lập trình thi đấu”, nhóm đã sử dụng một số phương pháp thu thập yêu cầu sau:

- **Thiết kế giao diện với Penpot:**

- Nhóm sử dụng ứng dụng Penpot để phác thảo giao diện người dùng (UI mockups, wireframes).
- Từ các bản thiết kế, nhóm thu thập phản hồi từ stake-holders để xác định các chức năng cần thiết, cải tiến trải nghiệm người dùng và đảm bảo thiết kế phù hợp với nhu cầu thực tế.

- **Phỏng vấn trực tiếp:**

- Thực hiện trao đổi, phỏng vấn ngắn với giảng viên hướng dẫn, một số sinh viên, và người có kinh nghiệm tổ chức cuộc thi lập trình.
- Mục tiêu là hiểu rõ quy trình hiện tại, các khó khăn đang gặp phải và mong muốn về một hệ thống hỗ trợ đăng ký hiệu quả.

2. 2. Phân tích hệ thống

2. 2. 1. Tổng quan hệ thống

Mục đích và các chức năng chính của hệ thống:

Hệ thống “Trang đăng ký tham gia cuộc thi lập trình thi đấu” mà nhóm xây dựng có mục đích chính là cho phép thí sinh đăng ký tham gia các cuộc thi lập trình thi đấu trên hệ thống thi và chấm thi của CLB ITMC quản lý một cách chủ động và trực tiếp mà không thông qua quản lý của hệ thống thi và chấm thi hiện có. Các chức năng chính của hệ thống bao gồm:

1. Đăng ký/đăng nhập tài khoản thí sinh trên hệ thống chấm thi.
2. Tạo nhóm các thí sinh theo chuẩn ICPC gồm 3 thành viên.
3. Đăng ký tham gia các cuộc thi được tổ chức trên hệ thống chấm thi.

Đối tượng sử dụng hệ thống (Tác nhân): Đối tượng sử dụng chính của hệ thống là những sinh viên học viện muốn đăng ký tham gia các cuộc thi lập trình thi đấu tại hệ thống do CLB ITMC quản lý và tổ chức.

Ràng buộc đối với hệ thống:

- Hệ thống đảm bảo người sử dụng thực hiện được các chức năng cơ bản đã đề ra.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- Hệ thống có khả năng tương tác ngược lại với hệ thống hiện có của CLB ITMC.
- Hệ thống đảm bảo được các ràng buộc về thông tin cá nhân của người sử dụng là duy nhất. Không có tài khoản nào trong hệ thống có cùng email, username, mã số sinh viên. không bắt buộc phải là MSSV của Học viện. Email chỉ cần đúng định dạng email tiêu chuẩn, không cần email có thật.
- Hệ thống đảm bảo các nhóm (đội) thi trong hệ thống chỉ có đúng 03 thành viên và không có tài khoản nào tham gia vào nhiều hơn 2 nhóm.
- Hệ thống đảm bảo thí sinh có thể ghi danh tham gia cuộc thi nếu đủ điều kiện.
 - Đối với cuộc thi dành cho nhóm thì thí sinh phải lập nhóm với các tài khoản thí sinh hợp lệ và các thành viên trong nhóm đều chưa ghi danh tham gia cuộc thi. Khi đăng ký tham gia thi thì tất cả các thí sinh trong nhóm sẽ được ghi danh vào danh sách thí sinh tham gia cuộc thi.
 - Đối với cuộc thi dành cho cá nhân thì thí sinh phải có tài khoản thí sinh hợp lệ và chưa ghi danh tham gia cuộc thi.
 - Thí sinh đã ghi danh tham gia cuộc thi thì sẽ không thể ghi danh được nữa.

2. 2. 2. Phân tích yêu cầu

Yêu cầu chức năng

Các chức năng chính:

1. Cho phép người dùng đăng ký tài khoản thí sinh trên hệ thống. Sau đó có thể đăng nhập vào hệ thống sử dụng tài khoản đã đăng kí.
2. Cho phép thí sinh tạo nhóm với 02 người dùng có tài khoản thí sinh khác trên hệ thống với điều kiện những thí sinh này chưa có nhóm.
3. Cho phép thí sinh (cá nhân hoặc thành viên của nhóm) có thể đăng ký tham gia vào các cuộc thi lập trình thi đấu nếu hợp lệ. Điều kiện hợp lệ: nếu tham gia thi cuộc thi dành cho đội thì phải có đội hợp lệ, còn nếu tham gia cuộc thi dành cho cá nhân thì phải có tài khoản thí sinh và thí sinh hoặc nhóm thí sinh phải chưa ghi danh vào cuộc thi.

Các chức năng khác:

1. Xem danh sách các cuộc thi có trong hệ thống.
2. Xem và chỉnh sửa thông tin cá nhân của thí sinh trên hệ thống.
3. Xem danh sách thành viên trong đội.
4. Xóa đội để thành lập đội mới.

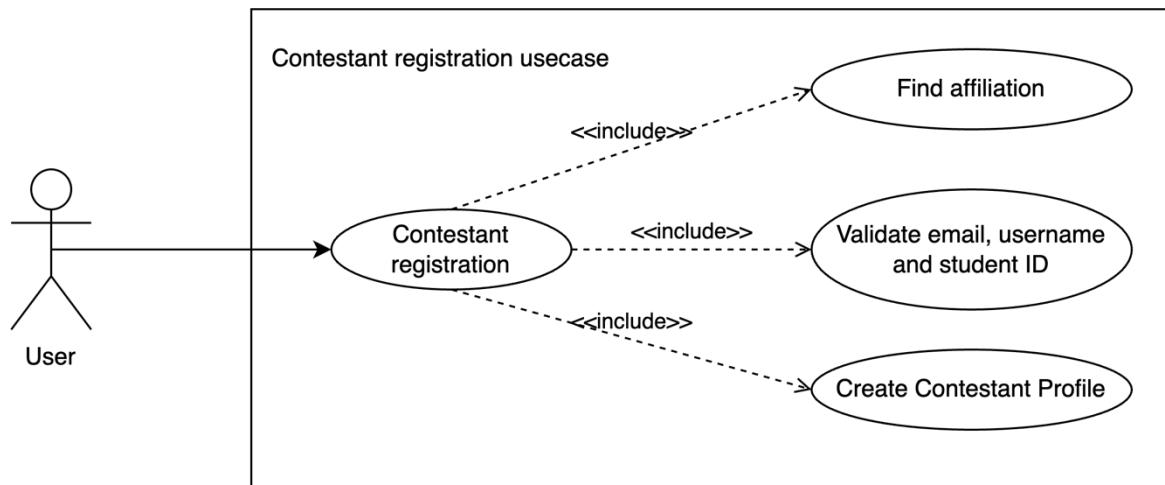
Yêu cầu phi chức năng

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- Hệ thống phải đảm bảo phản hồi các yêu cầu từ người dùng trong tối đa 3 giây.
- Hệ thống có khả năng hoạt động ổn định trong quá trình triển khai.
- Đảm bảo các thông tin cá nhân của người dùng không bị trùng lặp trong hệ thống.
- Hệ thống có khả năng tương tác ngược lại với hệ thống thi và chấm thi hiện có.
- Đảm bảo dữ liệu người dùng không bị sai lệch hoặc mất đi trong quá trình hoạt động và sử dụng của hệ thống.
- Hệ thống phải thông báo rõ ràng và đầy đủ lỗi đến người dùng nếu có bất cứ lỗi nào xảy ra trong quá trình hoạt động.

2. 2. 3. Các usecase chính của hệ thống

Use case đăng ký tài khoản thí sinh tham gia hệ thống



Hình 1: Sơ đồ usecase đăng ký tài khoản thí sinh tham gia hệ thống

Ý nghĩa: Người sử dụng chưa có tài khoản thực hiện đăng ký tài khoản thí sinh của hệ thống bằng các thông tin cá nhân của thí sinh bao gồm họ và tên, email, username của tài khoản, mã số sinh viên, niên khóa.

Use case này bao gồm các usecase nhỏ khác:

1. **Find affiliation:** Tìm niên khóa, được thực hiện khi người dùng truy cập vào form đăng ký, hệ thống sẽ tìm các niên khóa hiện có trong cơ sở dữ liệu tạo thành danh sách để người dùng chọn khi điền form.
2. **Validate email, username and student ID:** Xác thực các thông tin email, username và mã số sinh viên được người dùng điền vào form. Hệ thống thực hiện xác thực sau khi người dùng bấm gửi form thông tin đăng ký đến hệ thống. Nếu kết quả xác thực lỗi thì sẽ hiển thị thông báo đến người dùng trên trang web.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

3. **Create contestant profile:** Tạo thông tin thí sinh. Sau khi xác thực thông tin thí sinh thì hệ thống sẽ lưu thông tin này vào cơ sở dữ liệu. Khi lưu trữ thành công sẽ gửi thông báo đến người dùng và thông báo được hiển thị lên trang web.

Tác nhân: User – Người sử dụng truy cập vào trang web, nhưng chưa đăng kí làm thí sinh của hệ thống.

Điều kiện trước:

- Người dùng (user) có thể truy cập vào hệ thống.
- Người dùng có Email, Username và Student ID hợp lệ (Chưa được đăng kí trước đó).

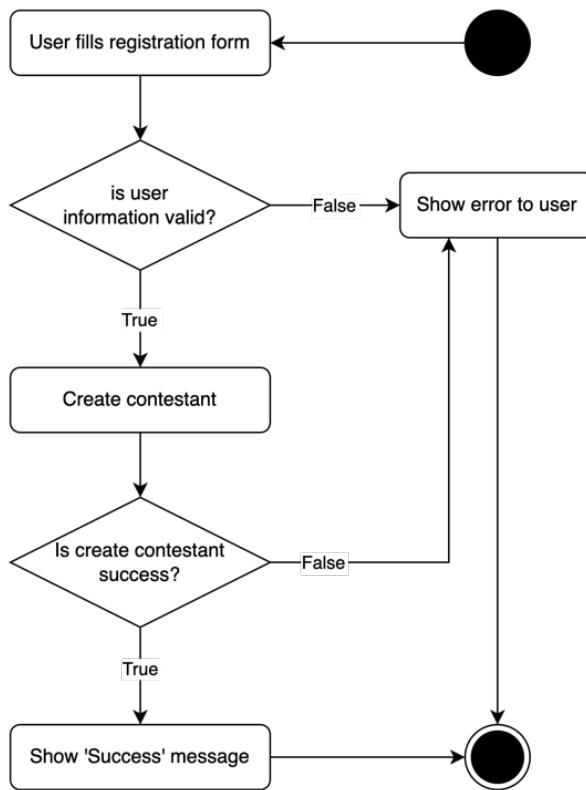
Điều kiện sau:

- Thông tin người dùng cung cấp được xác thực là hợp lệ.
- Người dùng đăng kí thành công với hệ thống và trở thành thí sinh, có thể đăng kí tham gia các cuộc thi lập trình thi đấu do CLB tổ chức trên hệ thống.

Kịch bản tương tác:

1. Người dùng vào trang đăng kí của hệ thống. Form đăng kí thành viên sẽ được xuất hiện trên trang web.
2. Người dùng điền các thông tin: email, username, password, họ và tên, giới tính (gender), mã số sinh viên, và chọn niêm khóa (affiliation) từ danh sách niêm khóa và bấm nút đăng kí (Sign up).
3. Hệ thống nhận thông tin đăng kí từ người dùng và thực hiện xác thực các thông tin: email, username và mã số sinh viên đã hợp lệ, hay đã tồn tại trong cơ sở dữ liệu chưa.
4. Nếu thông tin người dùng cung cấp không hợp lệ hoặc đã tồn tại thì hệ thống thông báo lỗi cho người dùng.
5. Ngược lại, hệ thống lưu thông tin vào cơ sở dữ liệu và người dùng được thông báo đăng kí thành công và trở thành thí sinh của hệ thống.

Lược đồ hoạt động của usecase đăng ký tài khoản thí sinh:



Hình 2: Sơ đồ hoạt động usecase đăng ký tài khoản thí sinh

Lược đồ hoạt động minh họa quá trình xử lý khi người dùng đăng ký tài khoản để trở thành thí sinh của hệ thống thi đấu lập trình. Các bước diễn ra theo trình tự như sau:

Bước 1: Điền form đăng ký

- Quá trình bắt đầu khi người dùng truy cập vào biểu mẫu và nhập thông tin cá nhân cần thiết để đăng ký tài khoản.

Bước 2: Kiểm tra thông tin người dùng

- Hệ thống sẽ tiến hành xác thực các thông tin mà người dùng đã cung cấp (email, username, mã số sinh viên...) để đảm bảo thông tin hợp lệ và chưa từng được sử dụng trên hệ thống.
- Nếu thông tin không hợp lệ (bị trùng lặp hoặc thiếu), hệ thống sẽ hiển thị thông báo lỗi, người dùng phải sửa thông tin để tiếp tục.

Bước 3: Tạo hồ sơ thí sinh

- Nếu thông tin người dùng cung cấp hợp lệ, hệ thống sẽ tiến hành tạo hồ sơ mới cho thí sinh và lưu vào cơ sở dữ liệu.

Bước 4: Kiểm tra việc tạo hồ sơ

- Sau khi tạo hồ sơ, hệ thống xác nhận lại một lần nữa rằng hồ sơ đã được tạo thành công.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

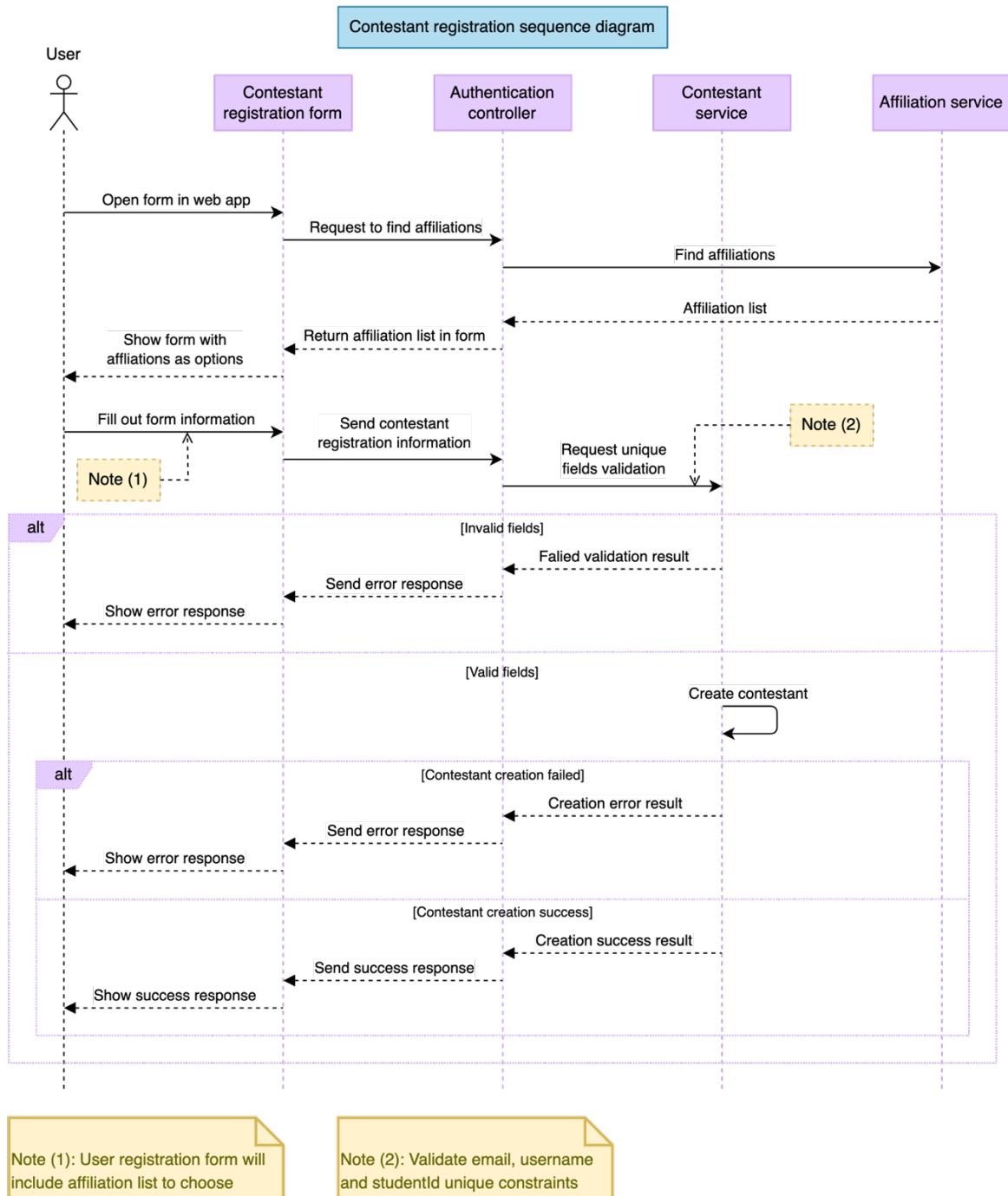
- Nếu có lỗi trong quá trình tạo hồ sơ (ví dụ: lỗi hệ thống hoặc lỗi cơ sở dữ liệu), người dùng sẽ nhận được thông báo lỗi tương ứng.

Bước 5: Thông báo thành công

- Nếu hồ sơ được tạo thành công, hệ thống hiển thị thông báo xác nhận rằng quá trình đăng ký tài khoản của người dùng đã hoàn tất thành công.

Quá trình kết thúc tại đây với việc người dùng đã trở thành thí sinh hợp lệ của hệ thống.

Lược đồ tuần tự của usecase đăng ký tài khoản thí sinh:



Hình 3: Sơ đồ tuần tự usecase đăng ký tài khoản thí sinh

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

Lược đồ tuần tự minh họa quá trình tương tác theo thời gian giữa người dùng và các thành phần bên trong hệ thống khi thực hiện chức năng đăng ký tài khoản thí sinh. Cụ thể, các bước tương tác được mô tả như sau:

1. Người dùng mở biểu mẫu đăng ký:

- Người dùng mở biểu mẫu (Contestant registration form) trên ứng dụng web.
- Biểu mẫu gửi yêu cầu đến Authentication controller để lấy danh sách niêm khóa từ Affiliation service.

2. Hệ thống trả về danh sách niêm khóa:

- Authentication controller gọi phương thức từ Affiliation service để truy xuất danh sách niêm khóa.
- Sau khi nhận danh sách niêm khóa, Authentication controller trả về danh sách cho biểu mẫu đăng ký để hiển thị cho người dùng lựa chọn.

3. Người dùng điền thông tin vào biểu mẫu:

- Người dùng nhập các thông tin cần thiết bao gồm email, username, mã số sinh viên và chọn niêm khóa từ danh sách đã hiển thị.

4. Xác thực thông tin người dùng:

- Biểu mẫu gửi thông tin đăng ký đến Authentication controller.
- Authentication controller chuyển yêu cầu xác thực thông tin duy nhất (email, username và mã số sinh viên) tới Contestant service.
- Contestant service thực hiện kiểm tra và trả kết quả xác thực về cho Authentication controller.

Luồng xử lý sau khi xác thực:

5. Nếu thông tin không hợp lệ (invalid fields):

- Authentication controller gửi phản hồi lỗi về biểu mẫu.
- Biểu mẫu hiển thị thông báo lỗi đến người dùng để yêu cầu sửa lại thông tin.

6. Nếu thông tin hợp lệ (valid fields):

- Authentication controller yêu cầu Contestant service tiến hành tạo hồ sơ thí sinh.

7. Nếu tạo hồ sơ thất bại (creation failed):

- Contestant service gửi thông báo lỗi tạo hồ sơ về Authentication controller.
- Authentication controller chuyển thông báo lỗi tới biểu mẫu để hiển thị đến người dùng trên trang web.

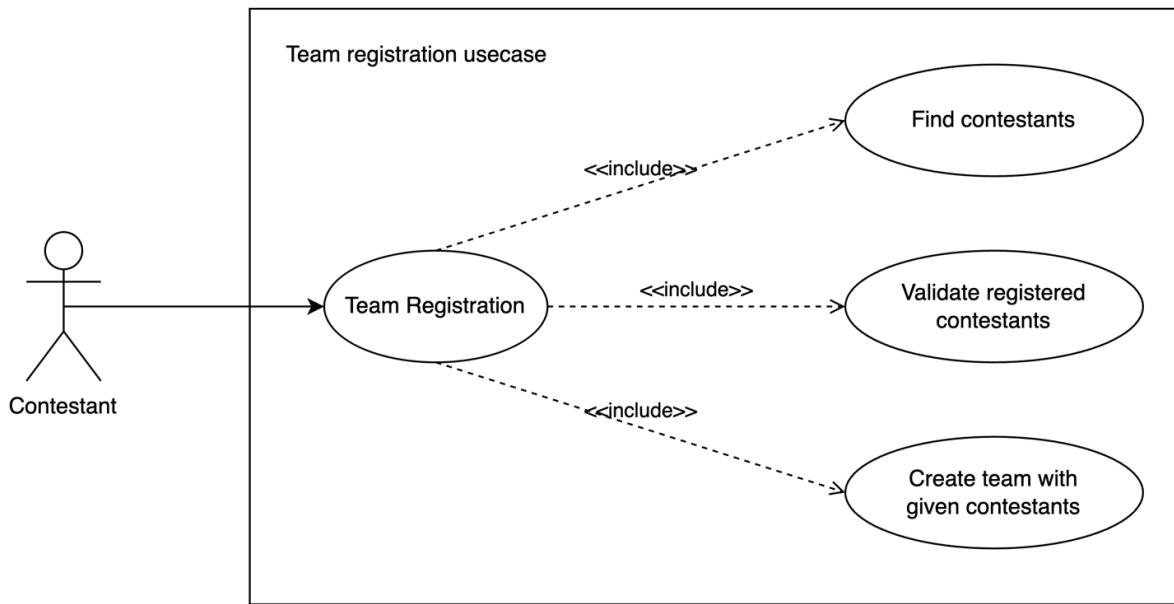
8. Nếu tạo hồ sơ thành công (creation success):

- Contestant service thông báo thành công đến Authentication controller.
- Authentication controller chuyển kết quả thành công về biểu mẫu.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- Biểu mẫu hiển thị thông báo “đăng ký thành công” tới người dùng.

Use case tạo nhóm tham gia các cuộc thi trên hệ thống



Hình 4: Sơ đồ usecase tạo nhóm thi tham gia các cuộc thi

Ý nghĩa: Người sử dụng đã có tài khoản hợp lệ trên hệ thống sẽ tạo nhóm (đội) có tối đa 3 thành viên có tài khoản hợp lệ trên hệ thống để tham gia thi các cuộc thi lập trình thi đấu theo đội gồm 3 thành viên theo chuẩn ICPC.

Use case này bao gồm các usecase nhỏ khác:

- Find contestants:** Tìm danh sách thí sinh khác. Khi thí sinh thực hiện tạo nhóm sẽ điền email của 02 thành viên khác chưa có nhóm vào form thông tin và gửi đến hệ thống. Hệ thống sẽ thực hiện tìm các thí sinh chưa có nhóm trong hệ thống dựa trên email này. Nếu không tồn tại thí sinh với email tương ứng thì sẽ thông báo lỗi đến người dùng.
- Validate registered contestants:** Hệ thống thực hiện kiểm tra các thí sinh đang lập nhóm đã có nhóm hay chưa. Nếu đã có thì sẽ thông báo lỗi thí sinh đã có nhóm đến thí sinh tạo nhóm.
- Create team with given contestants:** Tạo nhóm với danh sách các thí sinh hợp lệ. Hệ thống thực hiện lưu thông tin nhóm bao gồm tên nhóm, mô tả và danh sách thành viên vào cơ sở dữ liệu. Khi tạo nhóm thành công sẽ hiển thị thông báo đến thí sinh tạo nhóm. Ngược lại sẽ thông báo lỗi đến thí sinh tạo nhóm trên trang web.

Tác nhân: Contestant – Người sử dụng đã đăng ký thành công tài khoản hệ thống (thí sinh) nhưng chưa có nhóm (đội).

Điều kiện trước:

Báo cáo đồ án Trang đăng kí tham gia cuộc thi lập trình thi đấu

- Người sử dụng có thể đăng nhập vào hệ thống.
- Tài khoản đăng kí chưa thuộc nhóm (đội) nào.

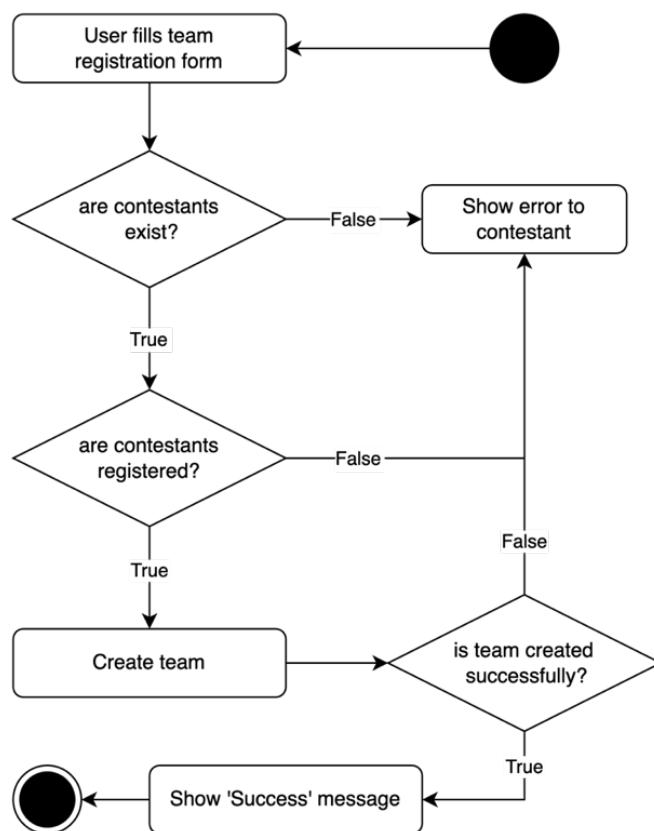
Điều kiện sau:

- Các thành viên được ghép nhóm thành công và trở thành một nhóm.
- Các thành viên chỉ thuộc một nhóm (đội) duy nhất trong hệ thống.

Kịch bản tương tác:

1. Thí sinh đăng nhập vào hệ thống với email và password đã đăng kí.
2. Thí sinh chọn chức năng tạo team (chỉ hiển thị nếu thí sinh chưa có team).
3. Thí sinh điền form tạo nhóm. Gồm email của duy nhất 02 thành viên còn lại.
4. Hệ thống kiểm tra các email được cung cấp (gồm 01 email của thí sinh tạo yêu cầu đăng kí và 02 email của các thành viên còn lại đã thuộc team nào chưa).
5. Nếu quá trình xác thực sai, nghĩa là tồn tại email đã thuộc vào team bất kì. Thì sẽ hiển thị thông báo lỗi “thí sinh đã lập nhóm”.
6. Ngược lại, hệ thống sẽ tạo đội, lưu trữ vào cơ sở dữ liệu và hiển thị thông báo tạo team thành công lên trang web.

Lược đồ hoạt động usecase đăng kí team:



Hình 5: Sơ đồ hoạt động usecase tạo nhóm thi tham gia các cuộc thi

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

Lược đồ hoạt động mô tả chi tiết quá trình xử lý khi thí sinh thực hiện thao tác đăng ký nhóm thi đấu trên hệ thống. Quá trình diễn ra theo các bước cụ thể như sau:

Bước 1: Thí sinh điền biểu mẫu tạo nhóm thi

- Quá trình bắt đầu khi thí sinh đăng nhập vào hệ thống và nhập thông tin cần thiết để tạo nhóm, gồm thông tin các thành viên (through qua email).

Bước 2: Kiểm tra sự tồn tại của các thành viên

- Hệ thống thực hiện kiểm tra các thành viên có tồn tại hay không (đã đăng ký tài khoản trước đó).
- Nếu có thành viên không tồn tại, hệ thống hiển thị thông báo lỗi và yêu cầu thí sinh sửa thông tin.

Bước 3: Kiểm tra trạng thái đăng ký nhóm của các thành viên

- Sau khi xác định các thành viên tồn tại, hệ thống tiếp tục kiểm tra các thành viên này đã tham gia vào nhóm nào chưa.
- Nếu có thành viên đã đăng ký vào nhóm khác, hệ thống trả thông báo lỗi tương ứng.

Bước 4: Tạo nhóm thi

- Nếu tất cả các thành viên đều tồn tại và chưa đăng ký vào bất kỳ nhóm nào, hệ thống tiến hành tạo nhóm thi mới với các thành viên đã cung cấp.

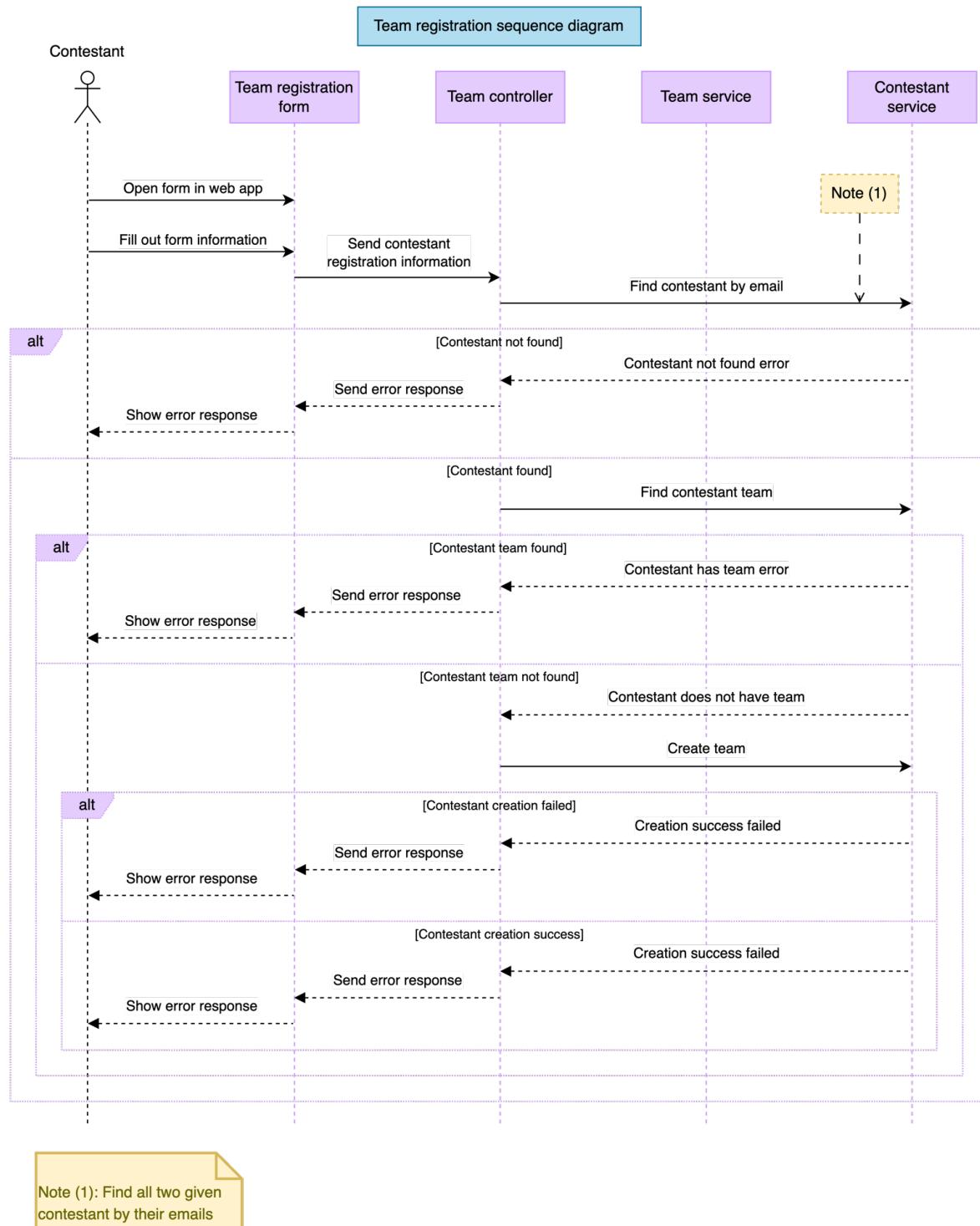
Bước 5: Kiểm tra kết quả tạo nhóm

- Hệ thống kiểm tra xem nhóm có được tạo thành công hay không.
- Nếu xảy ra lỗi trong quá trình tạo nhóm (ví dụ: lỗi lưu dữ liệu), hệ thống sẽ hiển thị thông báo lỗi tới thí sinh.
- Nếu nhóm được tạo thành công, hệ thống hiển thị thông báo xác nhận thành công.

Quá trình kết thúc với việc nhóm được tạo thành công và thông tin nhóm được cập nhật đầy đủ trên hệ thống, giúp các thí sinh có thể tiếp tục thực hiện các thao tác khác như đăng ký tham gia cuộc thi.

Lược đồ tuần tự của usecase đăng ký team:

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu



Hình 6: Sơ đồ tuần tự usecase tạo nhóm thi tham gia các cuộc thi

Lược đồ tuần tự thể hiện các bước tương tác cụ thể giữa thí sinh và các thành phần của hệ thống khi thực hiện chức năng đăng ký nhóm thi đấu. Quy trình này bao gồm các bước chi tiết như sau:

1. Thí sinh mở và điền biểu mẫu đăng ký nhóm:

- Thí sinh mở biểu mẫu đăng ký nhóm (Team registration form) từ giao diện ứng dụng web.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- Thí sinh điền thông tin gồm email của hai thành viên còn lại trong nhóm.

2. Xác thực sự tồn tại của các thành viên:

- Thông tin từ biểu mẫu được gửi tới Team controller.
- Team controller gửi yêu cầu tìm kiếm thông tin hai thành viên theo email đến Contestant service.
- Contestant service kiểm tra trong hệ thống và trả kết quả về sự tồn tại của từng thành viên.

Trường hợp ngoại lệ (alt):

- Nếu không tìm thấy bất kỳ thành viên nào, hệ thống trả thông báo lỗi (“Contestant not found”) về biểu mẫu, hiển thị cho thí sinh.

3. Kiểm tra trạng thái đăng ký nhóm của các thành viên:

- Nếu các thành viên được tìm thấy, Team controller yêu cầu Team service kiểm tra trạng thái nhóm hiện tại của các thành viên.
- Team service tiến hành xác thực và trả kết quả trạng thái đăng ký nhóm.

Trường hợp ngoại lệ (alt):

- Nếu có thành viên đã tham gia vào nhóm khác, hệ thống sẽ gửi thông báo lỗi (“Contestant has team error”) trả lại biểu mẫu và thông báo tới thí sinh.

4. Tạo nhóm mới:

- Nếu các thành viên chưa đăng ký vào bất kỳ nhóm nào, Team controller gửi yêu cầu tạo nhóm tới Team service.
- Team service thực hiện quá trình tạo nhóm và lưu vào cơ sở dữ liệu.

5. Kiểm tra kết quả tạo nhóm:

- Sau khi tạo nhóm, Team service xác nhận kết quả việc tạo nhóm.

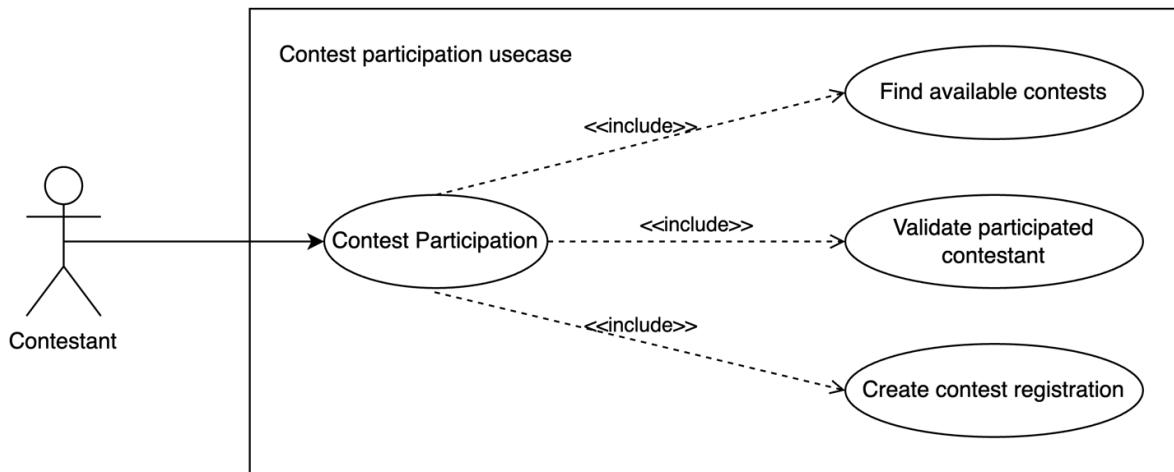
Trường hợp ngoại lệ (alt):

- Nếu tạo nhóm thất bại (ví dụ lỗi cơ sở dữ liệu), thông báo lỗi (“Creation success failed”) sẽ được gửi về biểu mẫu để thông báo tới thí sinh.
- Nếu nhóm được tạo thành công, thông báo “Creation success” sẽ được hiển thị cho thí sinh.

Lược đồ này minh họa rõ ràng các tương tác nội bộ của hệ thống trong quá trình tạo nhóm thi, bao gồm cả các trường hợp xử lý lỗi, đảm bảo người dùng được cung cấp đầy đủ phản hồi cần thiết để tiếp tục thao tác.

Báo cáo đồ án Trang đăng kí tham gia cuộc thi lập trình thi đấu

Use case đăng kí tham gia thi (ghi danh) các cuộc thi trên hệ thống



Hình 7: Sơ đồ usecase đăng kí cuộc thi trên hệ thống

Ý nghĩa: Người sử dụng của hệ thống có tài khoản thí sinh hợp lệ đăng kí tham gia thi 01 cuộc thi lập trình thi đấu trên hệ thống.

Use case này bao gồm các usecase khác:

- Find available contests:** Tìm các cuộc thi sắp diễn ra. Khi thí sinh đăng nhập vào trang web, hệ thống sẽ tìm danh sách các cuộc thi sắp diễn ra từ hệ thống thi DOMJudge và hiển thị trên trang web. Thí sinh muốn đăng kí thi trước tiên cần thực hiện tìm kiếm trên trang web. Khi tìm được cuộc thi muốn tham gia sẽ bấm chọn nút đăng kí thi.
- Validate participated contestant:** Sau khi thí sinh bấm nút đăng kí thi. Hệ thống sẽ thực hiện kiểm tra xem cuộc thi thuộc loại thi theo hình thức cá nhân hay theo nhóm. Tùy thuộc vào hình thức của cuộc thi mà sẽ kiểm tra thí sinh đăng kí hoặc nhóm thí sinh đăng kí đã tham gia vào cuộc thi này hay chưa. Nếu đã tham gia rồi thì sẽ thông báo lỗi đến thí sinh đăng kí.
- Create contest registration:** Sau khi xác thực thí sinh (hoặc nhóm thí sinh) tham gia thi thì hệ thống sẽ thực hiện ghi danh thí sinh (hoặc nhóm thí sinh) vào danh sách tham gia hệ thống và thực hiện cập nhật thông tin ở bên hệ thống DOMJudge.

Tác nhân: Contestant – Thí sinh đã đăng kí thành công hệ thống và chưa đăng kí tham gia cuộc thi đang muốn đăng kí.

Điều kiện trước:

- Thí sinh (contestant) có thể truy cập vào hệ thống.
- Thí sinh chưa tham gia cuộc thi đang muốn đăng kí.

Điều kiện sau:

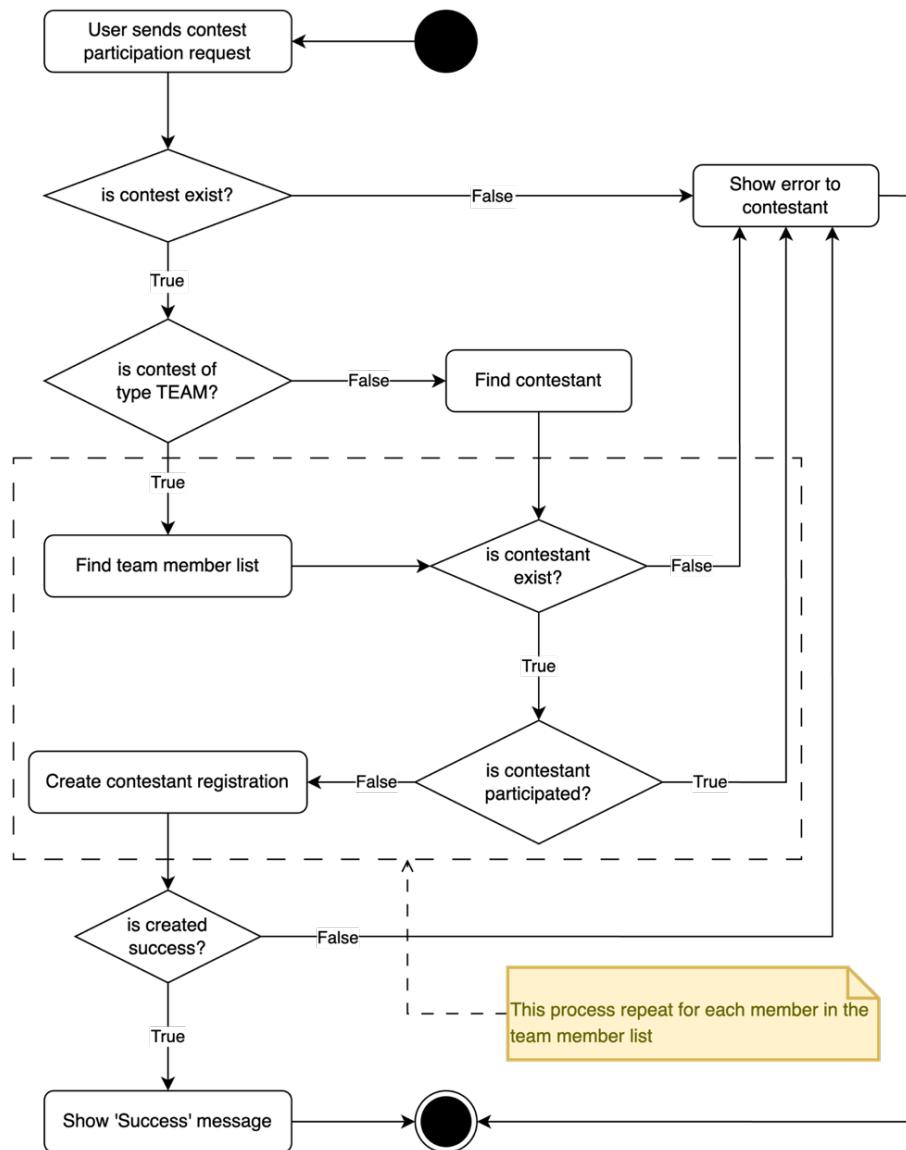
Báo cáo đồ án Trang đăng kí tham gia cuộc thi lập trình thi đấu

- Thí sinh đăng kí tham gia thành công đối với cuộc thi đơn và mọi thành viên trong team đăng kí tham gia thành công đối với cuộc thi nhóm.
- Danh sách thí sinh được tạo mới hoặc cập nhật khi có thí sinh tham gia thi.

Kịch bản tương tác:

- Thí sinh đăng nhập vào hệ thống.
- Thí sinh tìm trong danh sách cuộc thi và chọn cuộc thi muốn tham gia.
- Thí sinh bấm nút đăng kí tham gia.
- Hệ thống kiểm tra thí sinh có hợp lệ hay không.
- Nếu thí sinh không hợp lệ sẽ thông báo lỗi đăng kí tham gia không thành công.
- Ngược lại, hệ thống sẽ thông báo đăng kí tham gia cuộc thi thành công.

Lược đồ hoạt động usecase đăng kí tham gia cuộc thi:



Hình 8: Sơ đồ hoạt động usecase đăng kí tham gia cuộc thi trên hệ thống

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

Lược đồ hoạt động này mô tả chi tiết các bước xử lý khi thí sinh thực hiện yêu cầu đăng ký tham gia một cuộc thi trên hệ thống. Các bước cụ thể được diễn giải như sau:

Bước 1: Gửi yêu cầu tham gia cuộc thi

- Quá trình bắt đầu khi thí sinh gửi yêu cầu đăng ký tham gia một cuộc thi cụ thể.

Bước 2: Kiểm tra sự tồn tại của cuộc thi

- Hệ thống tiến hành xác thực cuộc thi được chọn có tồn tại trong hệ thống hay không.
- Nếu không tìm thấy cuộc thi, hệ thống hiển thị thông báo lỗi và kết thúc quá trình.

Bước 3: Kiểm tra loại cuộc thi (cá nhân hay theo đội)

- Hệ thống kiểm tra xem cuộc thi là dành cho cá nhân hay theo đội.

Trường hợp cuộc thi dành cho cá nhân:

- Hệ thống tiến hành tìm kiếm và xác thực sự tồn tại của thí sinh đăng ký.
 - Nếu thí sinh không tồn tại, hệ thống hiển thị lỗi và kết thúc quá trình.
- Kiểm tra xem thí sinh đã từng đăng ký cuộc thi này chưa.
 - Nếu đã tham gia rồi, thông báo lỗi sẽ được hiển thị và quá trình kết thúc.
- Nếu thí sinh chưa đăng ký, hệ thống thực hiện ghi danh thí sinh vào danh sách tham gia cuộc thi.

Trường hợp cuộc thi dành cho nhóm (đội):

- Hệ thống tìm kiếm và lấy danh sách các thành viên trong nhóm của thí sinh đăng ký.
- Quá trình sau được lặp lại đối với từng thành viên trong nhóm:
 - Kiểm tra từng thành viên có tồn tại trên hệ thống hay không.
 - Nếu không tồn tại, hiển thị lỗi và kết thúc quá trình.
 - Kiểm tra trạng thái đăng ký tham gia cuộc thi của thành viên đó.
 - Nếu thành viên đã đăng ký trước đó, hiển thị thông báo lỗi và kết thúc quá trình.
- Nếu thành viên chưa tham gia, hệ thống tiến hành ghi danh thành viên vào danh sách tham gia cuộc thi.

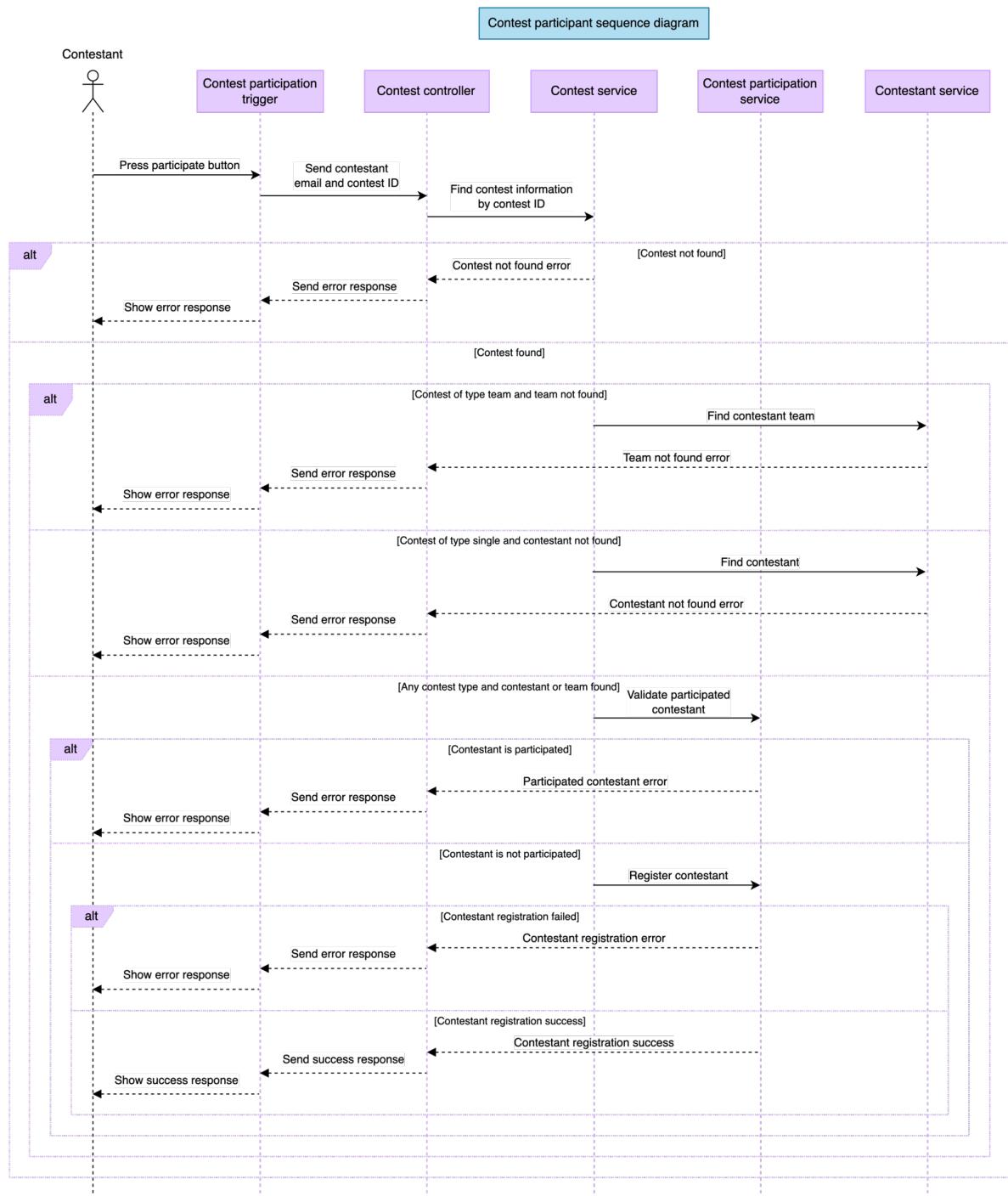
Bước 4: Kiểm tra kết quả ghi danh

- Hệ thống xác nhận việc ghi danh có thành công hay không.
 - Nếu thất bại (do lỗi xử lý hoặc lỗi dữ liệu), hệ thống hiển thị thông báo lỗi.
 - Nếu ghi danh thành công, hệ thống hiển thị thông báo xác nhận thành công cho thí sinh.

Báo cáo đồ án Trang đăng kí tham gia cuộc thi lập trình thi đấu

Quá trình kết thúc khi tất cả các thao tác hoàn thành thành công và người dùng nhận được thông báo xác nhận rõ ràng từ hệ thống.

Lược đồ tuần tự usecase đăng kí tham gia cuộc thi:



Hình 9: Sơ đồ tuần tự usecase đăng kí tham gia cuộc thi trên hệ thống

Lược đồ tuần tự thể hiện chi tiết quá trình tương tác giữa thí sinh và các thành phần trong hệ thống khi thực hiện đăng ký tham gia một cuộc thi. Các bước thực hiện cụ thể như sau:

1. Thí sinh gửi yêu cầu đăng ký tham gia:

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- Thí sinh nhấn nút hoặc kích hoạt chức năng tham gia cuộc thi (Contest participant trigger).
- Hệ thống gửi thông tin yêu cầu tới Contest controller bao gồm ID cuộc thi và email thí sinh.

2. Xác thực thông tin cuộc thi:

- Contest controller gửi yêu cầu đến Contest service để tìm kiếm thông tin cuộc thi bằng mã cuộc thi.
- Contest service trả về kết quả tìm kiếm.

Trường hợp ngoại lệ (alt):

- Nếu không tìm thấy cuộc thi, hệ thống hiển thị thông báo lỗi tới thí sinh và quá trình kết thúc.

3. Kiểm tra loại cuộc thi và xác thực thành viên:

- Nếu cuộc thi được tìm thấy, hệ thống tiếp tục kiểm tra loại cuộc thi (cá nhân hoặc đội nhóm).

Trường hợp cuộc thi theo nhóm:

- Contest participation service yêu cầu Contestant service cung cấp danh sách thành viên nhóm của thí sinh.
- Nếu không tìm thấy nhóm, hệ thống trả thông báo lỗi “Team not found”.
- Hệ thống thực hiện kiểm tra từng thành viên:
 - Xác thực sự tồn tại của thành viên trong nhóm. Nếu không tìm thấy, trả lỗi không tìm thấy tài khoản thí sinh.
 - Xác thực trạng thái đăng ký trước đó của thành viên. Nếu thành viên đã đăng ký, hệ thống thông báo lỗi tương ứng.

Trường hợp cuộc thi cá nhân:

- Hệ thống tìm kiếm và xác thực sự tồn tại của thí sinh.
 - Nếu không tìm thấy thí sinh, trả thông báo lỗi.
- Kiểm tra trạng thái đăng ký của thí sinh. Nếu đã tham gia trước đó, hệ thống trả lỗi thông báo thí sinh đã đăng ký.

4. Thực hiện ghi danh cuộc thi:

- Nếu các bước xác thực đều thành công, Contest participation service tiến hành ghi danh thí sinh (hoặc các thành viên trong nhóm) vào cuộc thi.

Trường hợp ngoại lệ (alt):

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- Nếu ghi danh thất bại (do lỗi hệ thống hoặc dữ liệu), hệ thống trả thông báo lỗi “Contestant registration error” về cho thí sinh.
- Nếu ghi danh thành công, hệ thống hiển thị thông báo xác nhận thành công cho thí sinh.

2. 3. Thiết kế hệ thống

2. 3. 1. CRC card của các đối tượng thành phần trong hệ thống

a) Các lớp đóng vai trò ánh xạ dữ liệu

Lớp “Affiliation”:

Class	Affiliation
Responsibilities	<ul style="list-style-type: none">• Lưu trữ thông tin liên khóa khi ánh xạ đến bảng “affiliations” trong CSDL thông qua các ORM framework
Collaborators	<ul style="list-style-type: none">• Lớp AffiliationService• Lớp AffiliationRepository

Lớp “Contestant”:

Class	Contestant
Responsibilities	<ul style="list-style-type: none">• Lưu trữ thông tin thí sinh khi ánh xạ đến bảng “contestants” trong CSDL thông qua các ORM framework
Collaborators	<ul style="list-style-type: none">• Lớp ContestantService• Lớp ContestantRepository• Lớp AffiliationService• Lớp TeamService

Lớp “Team”:

Class	Team
Responsibilities	<ul style="list-style-type: none">• Lưu trữ thông tin nhóm khi ánh xạ đến bảng “teams” trong CSDL
Collaborators	<ul style="list-style-type: none">• Lớp TeamService• Lớp TeamRepository

Lớp “Contest”:

Class	Contest
Responsibilities	<ul style="list-style-type: none">• Lưu trữ thông tin cuộc thi khi ánh xạ đến bảng “contests” trong CSDL
Collaborators	<ul style="list-style-type: none">• Lớp ContestService• Lớp ContestRepository

Lớp “ContestParticipants”

Class	ContestParticipant
Responsibilities	<ul style="list-style-type: none">• Lưu trữ thông tin thí sinh tham gia cuộc thi khi ánh xạ đến bảng “contest-participants” trong CSDL
Collaborators	<ul style="list-style-type: none">• Lớp ContestParticipantService• Lớp ContestantParticipationRepository• Lớp ContestService• Lớp ContestantService

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

b) Các lớp đóng vai trò truy cập dữ liệu (data access) thông qua ORM

Lớp “AffiliationRepository”:

Class	AffiliationRepository
Responsibilities	<ul style="list-style-type: none">Thực hiện các thao tác truy cập dữ liệu (data access) đến bảng “affiliations” trong CSDL, được implement thông qua ORM framework.
Collaborators	<ul style="list-style-type: none">Lớp AffiliationLớp AffiliationService

Lớp “ContestantRepository”:

Class	ContestantRepository
Responsibilities	<ul style="list-style-type: none">Thực hiện các thao tác truy cập dữ liệu (data access) đến bảng “contestants” trong CSDL, được implement thông qua ORM framework.
Collaborators	<ul style="list-style-type: none">Lớp ContestantLớp ContestantServiceLớp ContestLớp ContestServiceLớp TeamLớp TeamService

Lớp “TeamRepository”:

Class	TeamRepository
Responsibilities	<ul style="list-style-type: none">Thực hiện các thao tác truy cập dữ liệu (data access) đến bảng “teams” trong CSDL, được implement thông qua ORM framework.
Collaborators	<ul style="list-style-type: none">Lớp TeamLớp TeamService

Lớp “ContestRepository”:

Class	ContestRepository
Responsibilities	<ul style="list-style-type: none">Thực hiện các thao tác truy cập dữ liệu (data access) đến bảng “contests” trong CSDL, được implement thông qua ORM framework.
Collaborators	<ul style="list-style-type: none">Lớp ContestLớp ContestService

Lớp “ContestRegistrationRepository”:

Class	ContestRegistrationRepository
Responsibilities	<ul style="list-style-type: none">Thực hiện các thao tác truy cập dữ liệu (data access) đến bảng “contest-registrations” trong CSDL, được implement thông qua ORM framework.
Collaborators	<ul style="list-style-type: none">Lớp ContestRegistrationLớp ContestRegistrationServiceLớp ContestLớp ContestServiceLớp ContestantLớp ContestantService

c) Các lớp chứa business logic

Lớp “AffiliationService”:

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

Class	AffiliationService
Responsibilities	<ul style="list-style-type: none"> Tìm danh sách các niên khóa
Collaborators	<ul style="list-style-type: none"> Lớp Affiliation Lớp AffiliationRepository

Lớp “TeamService”:

Class	TeamService
Responsibilities	<ul style="list-style-type: none"> Tạo nhóm mới Tìm thành viên theo mã nhóm Cập nhật thông tin nhóm
Collaborators	<ul style="list-style-type: none"> Lớp Team Lớp TeamRepository Lớp Contestant Lớp ContestantService

Lớp “ContestantService”:

Class	ContestantService
Responsibilities	<ul style="list-style-type: none"> Tạo thí sinh mới Tìm thí sinh Cập nhật thông tin thí sinh Xác thực email, username và student ID Xác thực nhóm của thí sinh tham gia
Collaborators	<ul style="list-style-type: none"> Lớp Contestant Lớp ContestantRepository Lớp Affiliation Lớp AffiliationService Lớp Team Lớp TeamService

Lớp “ContestService”:

Class	ContestService
Responsibilities	<ul style="list-style-type: none"> Tạo cuộc thi mới Tìm danh sách các cuộc thi
Collaborators	<ul style="list-style-type: none"> Lớp Contest Lớp ContestRepository

Lớp “ContestRegistrationService”:

Class	ContestRegistrationService
Responsibilities	<ul style="list-style-type: none"> Tạo thông tin thí sinh tham gia cuộc thi Tìm danh sách các thí sinh tham gia cuộc thi theo mã cuộc thi Tìm danh sách các cuộc thi thí sinh tham gia theo mã thí sinh
Collaborators	<ul style="list-style-type: none"> Lớp ContestRegistration Lớp ContestRegistrationRepository

2. 3. 2. Usecase 1 – Đăng ký tài khoản thí sinh tham gia hệ thống

Hình ảnh form đăng ký tài khoản thí sinh tham gia hệ thống

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

Email address*

Enter email address

Password*

Enter password

User name*

Enter user name

First Name*

Enter first name

Last name*

Enter last name

Student ID*

Enter student ID

Gender*

School Year*

Sign in

Hình 10: Hình ảnh form đăng ký tài khoản thí sinh trên hệ thống

Các API sử dụng cho trang đăng ký tài khoản thí sinh trên hệ thống:

POST /api/v1/auth/sign-up Sign up

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{ "email": "john@example.com", "username": "john.doe", "password": "password123", "first_name": "John", "last_name": "Doe", "student_id": "12345678", "gender": "Male", "affiliation_id": "string" }
```

Responses

Code	Description	Links
200	Media type application/json Controls Accept header. Example Value Schema	No links
400	Email, username or student ID is taken	No links

Hình 11: API đăng ký tài khoản thí sinh trên hệ thống

Báo cáo đồ án Trang đăng kí tham gia cuộc thi lập trình thi đấu

2. 3. 3. Usecase 2 – Tạo nhóm tham gia cuộc thi trên hệ thống

Hình ảnh form tạo nhóm tham gia cuộc thi trên hệ thống:

The screenshot shows a 'Create Team' form. It includes fields for Team Name (AgentP), Team Description (Best of the best), Your Email (phuongnamtran1902@gmail.com), Member Email 2 (nnminhhaise@gmail.com), Member Email 3 (Enter third member email), and buttons for Create Team and Cancel.

Hình 12: Hình ảnh form tạo nhóm thi tham gia các cuộc thi

API đăng kí tạo nhóm trên hệ thống:

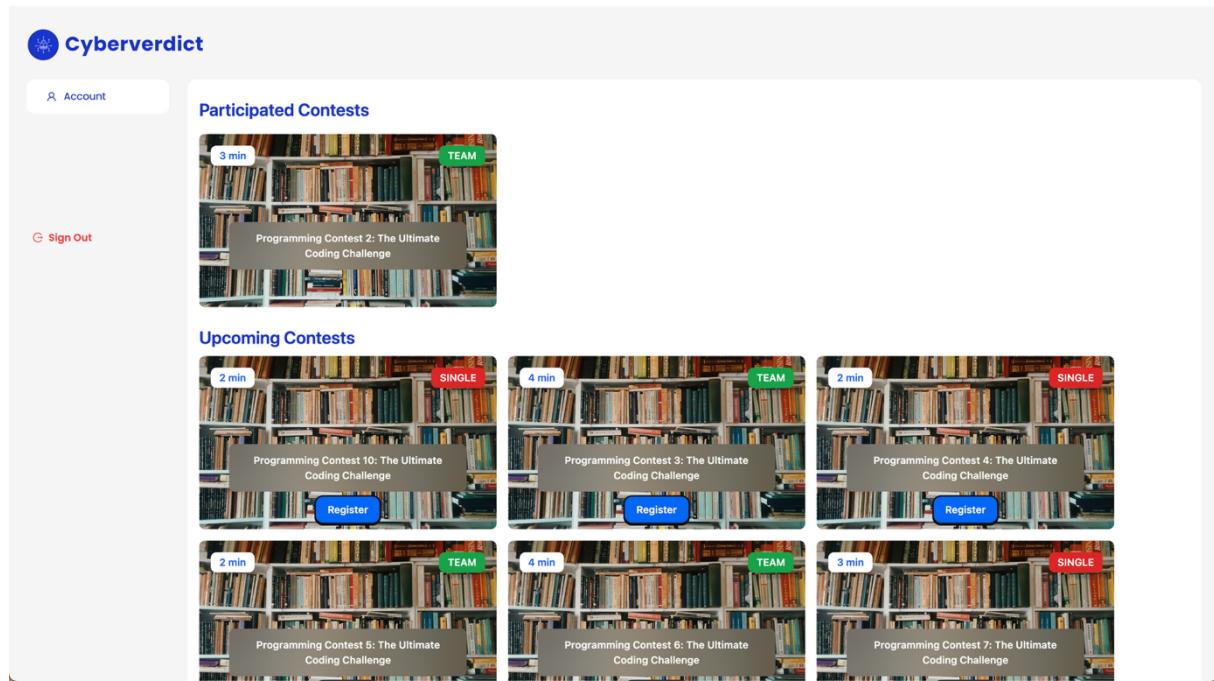
The screenshot shows a POST /api/v1/teams API endpoint. It includes sections for Parameters (None), Request body (application/json required), Example Value | Schema (JSON object with name, description, and members arrays), Responses, and a 200 OK response section. The 200 OK response shows a JSON schema for the created team data, including fields like id, name, description, and members (each member having fields like id, email, username, password, first_name, last_name, student_id, affiliation_id, gender, and team).

Hình 13: API tạo nhóm thi tham gia các cuộc thi

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

2. 3. 4. Usecase 3 – Đăng kí tham gia cuộc thi trên hệ thống

Hình ảnh trang chủ hiển thị danh sách các cuộc thi và nút ghi danh tham gia cuộc thi:



Hình 14: Hình ảnh trang chủ hiển thị danh sách các cuộc thi và nút đăng kí thi

API ghi danh tham gia cuộc thi trên hệ thống:

POST /api/v1/contests/{id}/participations [Role: Contestant] Contestant participate for a contest

Parameters

Name	Description
Id * required string (path)	id

Responses

Code	Description	Links
200	Media type application/json Controls Accept header. Example Value Schema	No links

```
{ "data": { "id": "string", "name": "string", "formal_name": "string", "description": "string", "banner": "string", "start_at": "string", "duration": 0, "type": "SINGLE", "created_at": "2025-06-16T15:47:03.125Z", "updated_at": "2025-06-16T15:47:03.125Z" }, "count": 0, "timestamp": "2025-03-11T12:00:00Z", "path": "string" }
```

400 Contest not found, contestant not found or contestant has been registered

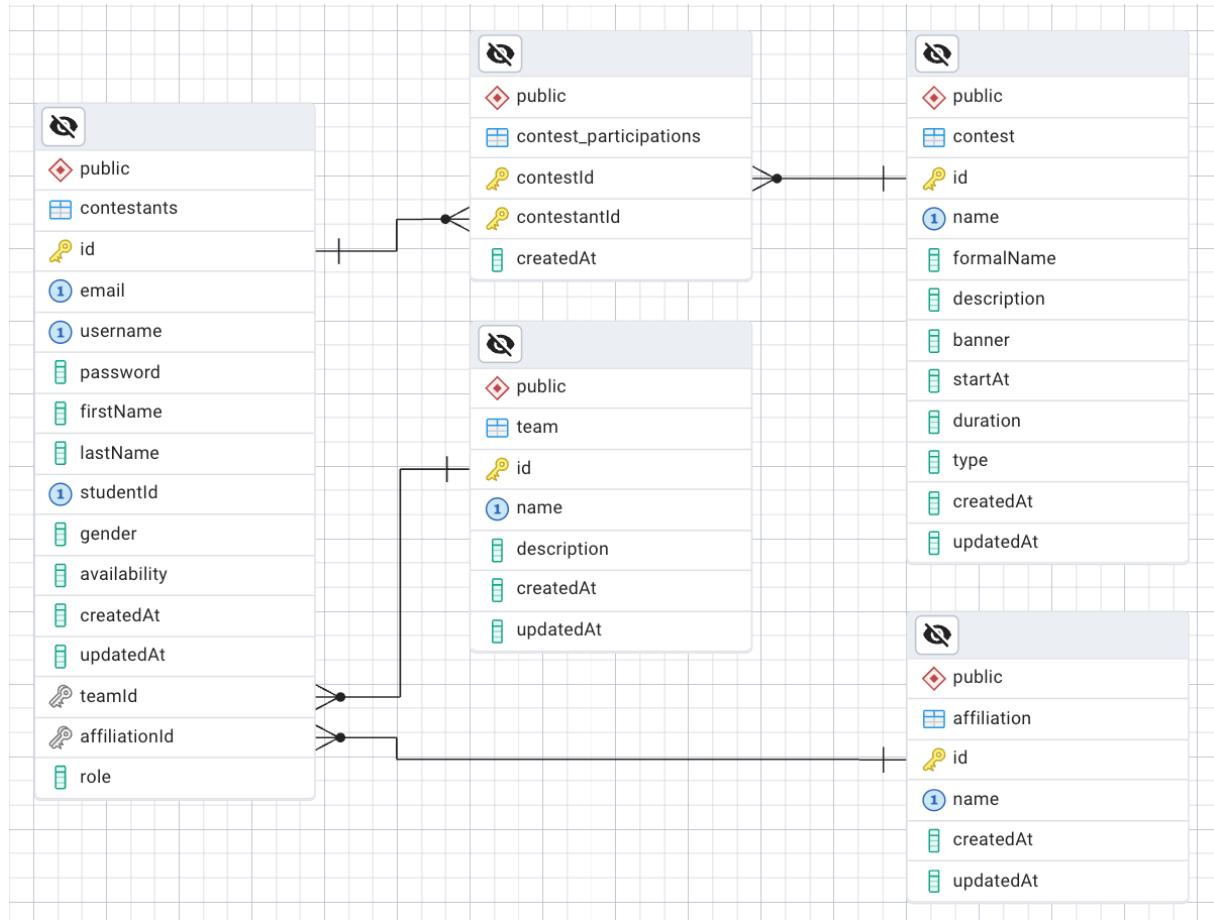
Hình 15: API đăng kí tham gia cuộc thi

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

2. 3. 5. Database của phần mềm

Loại cơ sở dữ liệu mà nhóm chọn sử dụng là cơ sở dữ liệu quan hệ và sử dụng hệ quản trị cơ sở dữ liệu PostgreSQL.

Lược đồ ERD:



Hình 16: Sơ đồ ERD của hệ thống

Tùy diễn dữ liệu:

Bảng niên khóa (affiliations):

STT	Tên thuộc tính	Ràng buộc	Kiểu dữ liệu	Ý nghĩa
1	id	PK	UUID	Mã niên khóa
2	name	NOT NULL, UNIQUE	Varchar	Tên niên khóa
3	createdAt	NOT NULL, Default now()	Timestamp	Thời điểm tạo niên khóa
4	updatedAt	NOT NULL, Default now()	Timestamp	Thời điểm cập nhật thông tin niên khóa

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

Bảng cuộc thi (contests):

STT	Tên thuộc tính	Ràng buộc	Kiểu dữ liệu	Ý nghĩa
1	id	PK	UUID	Mã cuộc thi
2	name	NOT NULL, UNIQUE	Varchar	Tên viết tắt cuộc thi
3	formalName	NOT NULL	Varchar	Tên đầy đủ chính thức của cuộc thi
4	description	NOT NULL	Varchar	Mô tả chi tiết cuộc thi
5	banner	NOT NULL	Varchar	Đường dẫn banner cuộc thi
6	startAt	NOT NULL	Timestamp	Thời điểm bắt đầu cuộc thi
7	duration	NOT NULL	Integer	Thời lượng cuộc thi bằng phút
8	type	NOT NULL, Default “Single”	Varchar	Loại cuộc thi (cá nhân hoặc theo nhóm).
9	createdAt	NOT NULL, Default now()	Timestamp	Thời điểm tạo niêm khóa
10	updatedAt	NOT NULL, Default now()	Timestamp	Thời điểm cập nhật thông tin niêm khóa

Bảng thí sinh (contestants):

STT	Tên thuộc tính	Ràng buộc	Kiểu dữ liệu	Ý nghĩa
1	id	PK	UUID	Mã thí sinh
2	email	NOT NULL, UNIQUE	Varchar	Địa chỉ email của thí sinh (mỗi email cho một tài khoản duy nhất trong hệ thống)
3	username	NOT NULL, UNIQUE	Varchar	Username của tài khoản (duy nhất cho một tài khoản trong hệ thống)
4	password	NOT NULL	Varchar	Mật khẩu của tài khoản
5	firstName	NOT NULL	Varchar	Họ của thí sinh
6	lastName	NOT NULL	Varchar	Tên của thí sinh
7	studentId	NOT NULL, UNIQUE	Varchar	Mã số sinh viên của thí sinh. Đúng với pattern mã số sinh viên của học viên
8	gender	NOT NULL, Default “Other”	Varchar	Giới tính của thí sinh
9	availability	NOT NULL, Default “true”	Boolean	Tình trạng tài khoản. True đối với tài khoản hoạt động và False với tài khoản không hoạt động trên hệ thống
10	teamId	FK	UUID	Mã nhóm của sinh viên
11	affiliationId	NOT NULL, FK	UUID	Mã niêm khóa của sinh viên
12	role	Default 0	Varchar	Vai trò của tài khoản trong hệ thống (mặc định 0, 1 đối với tài khoản admin)
13	createdAt	NOT NULL, Default now()	Timestamp	Thời điểm tạo niêm khóa
14	updatedAt	NOT NULL, Default now()	Timestamp	Thời điểm cập nhật thông tin niêm khóa

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

Bảng nhóm (teams):

STT	Tên thuộc tính	Ràng buộc	Kiểu dữ liệu	Ý nghĩa
1	id	PK	UUID	Mã nhóm
2	name	NOT NULL, UNIQUE	UUID	Tên nhóm
3	description	NOT NULL	Varchar	Mô tả nhóm
4	createdAt	NOT NULL, Default now()	Timestamp	Thời điểm tạo nhóm
5	updatedAt	NOT NULL, Default now()	Timestamp	Thời điểm cập nhật thông tin nhóm

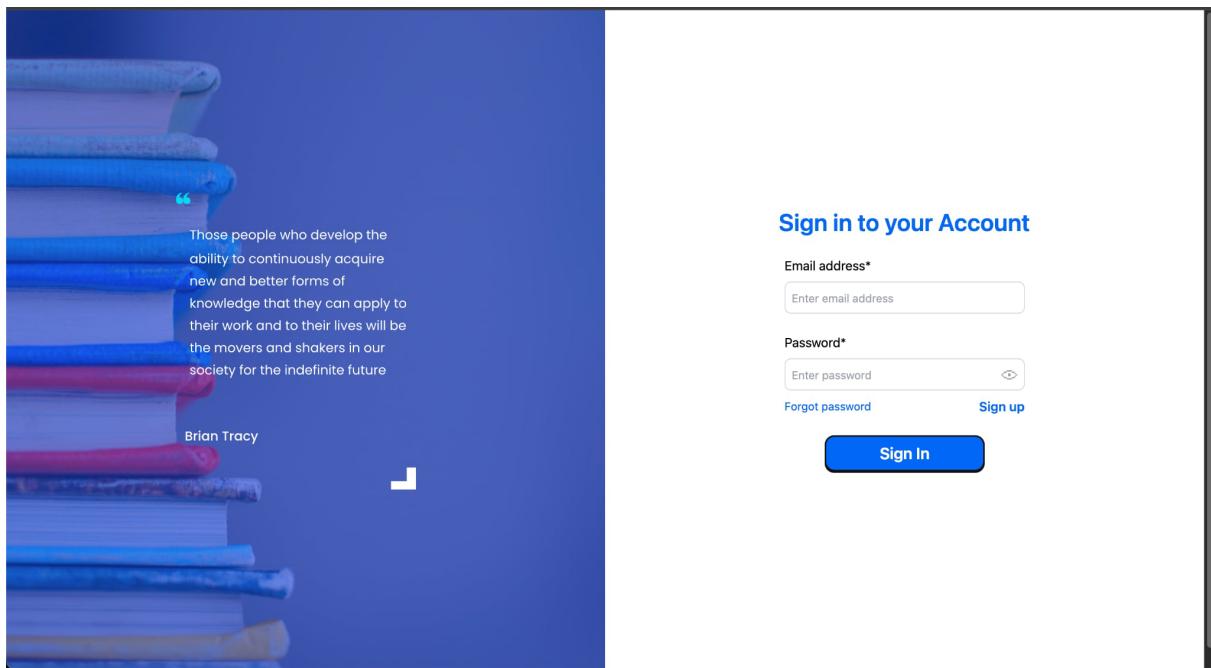
Bảng ghi danh tham gia cuộc thi (contest_participations):

STT	Tên thuộc tính	Ràng buộc	Kiểu dữ liệu	Ý nghĩa
1	contestId	PK	UUID	Mã cuộc thi
2	contestantId	PK	UUID	Mã thí sinh tham gia cuộc thi
3	createdAt	NOT NULL, Default now()	Timestamp	Thời điểm thí sinh đăng kí tham gia cuộc thi

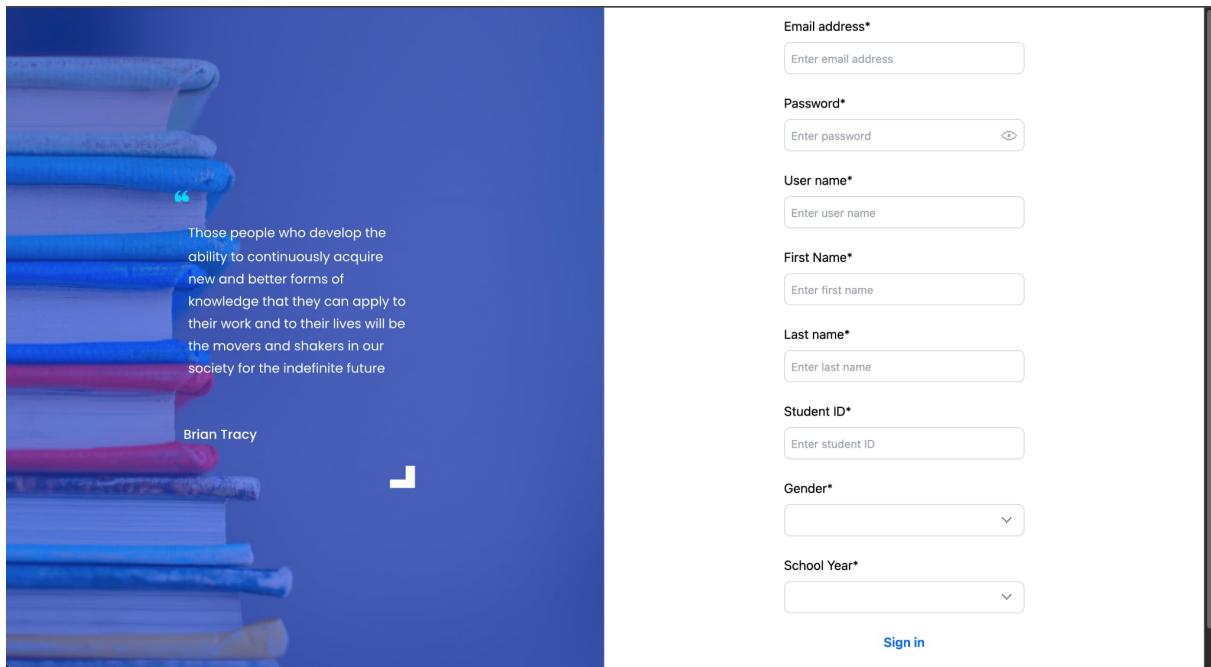
CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG

3. 1. Phần mềm ứng dụng

3. 1. 1. Giao diện của ứng dụng phần mềm

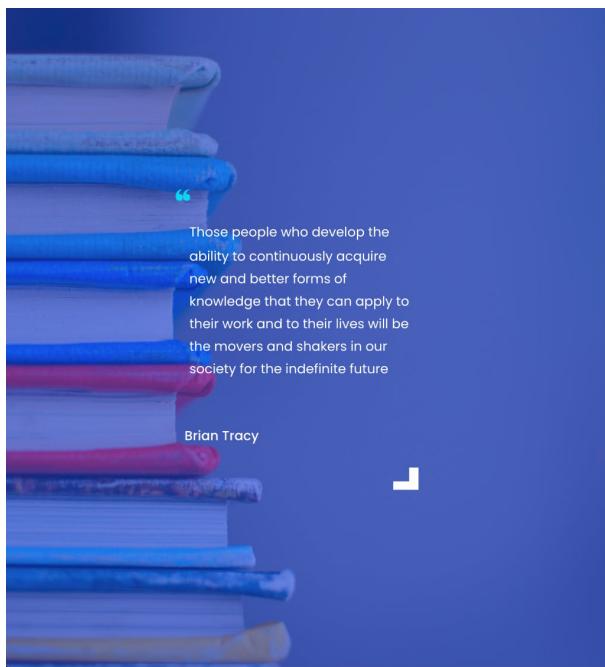


Hình 17: Giao diện trang đăng nhập của ứng dụng



Hình 18: Giao diện trang đăng ký tài khoản của ứng dụng

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu



Forgot Password

Email address*

Enter email address

Sign In

Send Email

Hình 19: Giao diện trang quên mật khẩu của ứng dụng

Hình 20: Giao diện trang chủ của ứng dụng

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

The screenshot shows the 'Account' section of the Cyberverdict platform. It includes fields for Email address (phuongnamtran1902@gmail.com), First name (TRAN), Last name (NAM), Username (fungoman), Student ID (n21dccn151), Gender (Male), School Year (admin), and a 'Save' button. Below this is a 'Create Team' section with a message: 'You haven't joined any team yet. Create a new team to start collaborating!' and a 'Create Team' button.

Hình 21: Giao diện trang thông tin cá nhân của thí sinh

The screenshot shows the 'Create Team' form. It includes fields for Team Name (AgentP), Team Description (Best of the best), Your Email (phuongnamtran1902@gmail.com), Member Email 2 (nnminhaise@gmail.com), Member Email 3 (Enter third member email), and two buttons: 'Create Team' (blue) and 'Cancel' (grey).

Hình 22: Giao diện form đăng ký tạo nhóm thi

3. 1. 2. Cài đặt hệ thống phần mềm

Hệ thống phần mềm gồm 4 thành phần Frontend, Backend, DBMS PostgreSQL và Redis. Trong đó, DBMS PostgreSQL và Redis có thể được triển khai nhanh với Docker qua docker compose.

Yêu cầu phần mềm:

- Node.js phiên bản 20.x LTS
- Docker phiên bản mới nhất

Triển khai Backend, DBMS PostgreSQL và Redis:

1. Clone GitHub repository của Backend ở đây: [Competitive Contest Utils Server](#).

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- Thiết lập các biến môi trường như hình dưới vào file “.env” ở trong thư mục gốc.

```
1 PORT=3001 # Backend application running port
2
3 # Backend config variables for connecting to Database
4 POSTGRES_HOST=localhost
5 POSTGRES_PORT=5432
6 POSTGRES_DB=contest-system
7 POSTGRES_USER=postgres
8 POSTGRES_PASSWORD=postgres
9
10 # Secret key for generating tokens and expiration time
11 SECRET=secret
12 EXPIRED_IN=1h
13
14 # Redis config for connecting to Redis server
15 REDIS_BLACKLIST_HOST=localhost
16 REDIS_BLACKLIST_PORT=6380
17
18 # Google Gmail API config variables
19 EMAIL_HOST=smtp.gmail.com
20 EMAIL_USER=nnminh.sam.1803@gmail.com
21 EMAIL_PORT=587
22 EMAIL_PASS=cdlc kvkh bboq wuol
23
24 # Frontend hosting URL for CORS config
25 FE_HOST=localhost:5173
26
27 # Admin account config
28 ADMIN_EMAIL=admin@example.com
29 ADMIN_PASSWORD=your_secure_password
```

Hình 23: Cấu hình biến môi trường cho Backend

- Mở Terminal và chạy lệnh bên dưới để tạo Docker containers cho DBMS PostgreSQL và Redis server.

```
docker compose --env-file .env -f docker-compose.yml up -d
```

Nếu lệnh “docker compose” không chạy thì có thể thử “docker-compose”.

Sau một lát, một DBMS PostgreSQL và Redis sẽ được tạo ra.

- Cài đặt các dependencies với lệnh “yarn” hoặc “npm install”.
- Chạy Backend với lệnh “yarn start”.

Triển khai Frontend:

- Clone GitHub repository của Frontend tại đây: [Competitive contest web app](#).
- Tạo file “.env” chứa biến môi trường tại thư mục gốc.

```
1 VITE_BASE_URL=http://localhost:3001/api
```

- Cài đặt các dependencies với lệnh “npm install”.
- Chạy Frontend với lệnh “npm run preview”.

3. 2. Kiểm thử

Bảng danh sách các testcases của Usecase 1:

Mã testcase	Tên gọi
TC01	Đăng ký thành công với thông tin hợp lệ
TC02	Đăng ký thất bại do email đã được sử dụng
TC03	Đăng ký thất bại do MSSV đã được sử dụng
TC04	Đăng ký thất bại do username đã tồn tại
TC05	Đăng ký thất bại do bỏ trống các trường bắt buộc
TC06	Đăng ký thất bại do sai định dạng email

Bảng danh sách các testcases của Usecase 2:

Mã testcase	Tên gọi
TC07	Tạo nhóm thành công với 3 thành viên hợp lệ chưa có nhóm
TC08	Tạo nhóm thất bại do có thành viên không tồn tại
TC09	Tạo nhóm thất bại do có thành viên đã thuộc nhóm khác
TC10	Tạo nhóm thất bại do nhập email trùng nhau
TC11	Tạo nhóm thất bại do bỏ trống thông tin thành viên
TC12	Tạo nhóm thất bại do email thành viên không đúng định dạng
TC13	Tạo nhóm thất bại do tên nhóm bị trùng

Bảng danh sách các testcases của Usecase 3:

Mã testcase	Tên gọi
TC14	Ghi danh thành công cuộc thi cá nhân với thí sinh hợp lệ
TC15	Ghi danh thất bại vì đã tham gia cuộc thi cá nhân
TC16	Ghi danh thành công cuộc thi theo nhóm
TC17	Ghi danh thất bại vì nhóm đã tham gia cuộc thi nhóm
TC18	Ghi danh thất bại vì chưa có nhóm khi ghi danh thí sinh cuộc thi nhóm
TC19	Ghi danh thất bại do cuộc thi không tồn tại

3. 2. 1. Đặc tả các testcases cho Usecase 1

Test Case TC01

- Tên testcase: Đăng ký thành công với thông tin hợp lệ
- Mô tả: Thí sinh chưa có tài khoản, điền đúng và đầy đủ thông tin gồm: họ tên, email, username, password, MSSV, giới tính và niên khóa → hệ thống xử lý và trả về thông báo đăng ký thành công.
- Tiền điều kiện: Email, username, và MSSV chưa tồn tại trong cơ sở dữ liệu.
- Các bước thực hiện:
 1. Truy cập giao diện trang đăng ký.
 2. Nhập đầy đủ thông tin: Họ và tên, email, username, password, MSSV, giới tính, chọn niên khóa.
 3. Nhấn nút “Đăng ký”.
 4. Hệ thống xác thực các thông tin.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

5. Nếu hợp lệ → tạo tài khoản và hiển thị thông báo thành công.
- Kỹ thuật kiểm thử: Phân vùng tương đương (Equivalence Partitioning), Positive Testing.
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Email hợp lệ	Username hợp lệ	MSSV hợp lệ	Kết quả
Có	Có	Có	Đăng ký thành công

Test Case TC02

- Tên testcase: Đăng ký thất bại do email đã được sử dụng
- Mô tả: Người dùng nhập thông tin với email đã tồn tại trong hệ thống → hệ thống phát hiện trùng lặp và từ chối đăng ký.
- Tiền đề kiện: Một tài khoản với email này đã tồn tại trong CSDL.
- Các bước thực hiện:
 1. Truy cập giao diện đăng ký.
 2. Nhập email đã được dùng trước đó.
 3. Nhập các trường còn lại hợp lệ.
 4. Nhấn “Đăng ký”.
 5. Hệ thống thông báo lỗi “Email đã được sử dụng”.
- Kỹ thuật kiểm thử: Negative Testing, Phân tích giá trị biên (Boundary Value Analysis).
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Email hợp lệ	Email tồn tại	Username hợp lệ	MSSV hợp lệ	Kết quả
Có	Có	Có	Có	Thông báo lỗi email trùng

Test Case TC03

- Tên testcase: Đăng ký thất bại do MSSV đã được sử dụng
- Mô tả: MSSV đã tồn tại trong hệ thống nhưng người dùng cố gắng đăng ký bằng MSSV đó → hệ thống từ chối.
- Tiền đề kiện: MSSV đã được đăng ký bởi một tài khoản khác.
- Các bước thực hiện:
 1. Truy cập trang đăng ký.
 2. Nhập MSSV trùng đã có trong hệ thống.
 3. Nhập các thông tin còn lại hợp lệ.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- 4. Nhấn “Đăng ký”.
- 5. Nhận thông báo lỗi “Mã số sinh viên đã tồn tại”.
- Kỹ thuật kiểm thử: Negative Testing
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Email hợp lệ	Username hợp lệ	MSSV tồn tại	Kết quả
Có	Có	Có	Thông báo lỗi MSSV trùng

Test Case TC04

- Tên testcase: Đăng ký thất bại do username đã tồn tại
- Mô tả: Hệ thống kiểm tra trùng username và từ chối nếu đã được dùng bởi người khác.
- Tiền điều kiện: Username đã tồn tại trong hệ thống.
- Các bước thực hiện:
 1. Mở trang đăng ký.
 2. Nhập username trùng lặp.
 3. Nhập các thông tin còn lại hợp lệ.
 4. Nhấn “Đăng ký”.
 5. Hệ thống trả thông báo “Username đã tồn tại”.
- Kỹ thuật kiểm thử: Negative Testing
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Email hợp lệ	Username tồn tại	MSSV hợp lệ	Kết quả
Có	Có	Có	Thông báo lỗi username trùng

Test Case TC05

- Tên testcase: Đăng ký thất bại do bỏ trống các trường bắt buộc
- Mô tả: Người dùng không điền đủ thông tin (ví dụ: bỏ trống email hoặc MSSV).
- Tiền điều kiện: Không có.
- Các bước thực hiện:
 1. Truy cập giao diện đăng ký.
 2. Để trống một số trường như email, username hoặc MSSV.
 3. Nhấn nút “Đăng ký”.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

4. Hệ thống yêu cầu điền đủ thông tin và hiển thị lỗi tương ứng.
- Kỹ thuật kiểm thử: Phân tích giá trị biên, Negative Testing
 - Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Email	Username	MSSV	Kết quả
Trống	Có	Có	Lỗi thiếu email
Có	Trống	Có	Lỗi thiếu username
Có	Có	Trống	Lỗi thiếu MSSV

Test Case TC06

- Tên testcase: Đăng ký thất bại do sai định dạng email
- Mô tả: Email không tuân theo định dạng abc@xyz.com → hệ thống báo lỗi định dạng.
- Tiền điều kiện: Không có
- Các bước thực hiện:
 1. Nhập một email không hợp lệ (ví dụ: abc@@gmail, abc.com).
 2. Điền các thông tin còn lại hợp lệ.
 3. Nhấn “Đăng ký”.
 4. Hệ thống thông báo lỗi định dạng email không hợp lệ.
- Kỹ thuật kiểm thử: Kiểm thử biên, Negative Testing
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Email hợp lệ	Định dạng đúng	Kết quả
Có	Không	Thông báo lỗi định dạng email

Bảng truy vết yêu cầu -> thiết kế -> testcase cho các testcase TC01 – TC06:

Testcase	Yêu cầu hệ thống	Thiết kế	Thực hiện kiểm thử
TC01	Yêu cầu hệ thống cho phép người dùng đăng ký tài khoản mới với thông tin hợp lệ	Form đăng ký thí sinh, API POST auth/sign-up	TC01
TC02	Yêu cầu hệ thống không cho phép trùng lặp email	Form đăng ký thí sinh, API POST auth/sign-up	TC02
TC03	Yêu cầu hệ thống không cho phép trùng lặp mã số sinh viên	Form đăng ký thí sinh, API POST auth/sign-up	TC03
TC04	Yêu cầu hệ thống không cho phép trùng lặp username	Form đăng ký thí sinh, API POST auth/sign-up	TC04
TC05	Yêu cầu hệ thống yêu cầu điền đầy đủ thông tin bắt buộc	Form đăng ký thí sinh, API POST auth/sign-up	TC05

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

TC06	Yêu cầu hệ thống kiểm tra định dạng hợp lệ của email	Form đăng ký thí sinh, API POST auth/sign-up	TC06
------	--	--	------

Bảng impact/derive/coverage:

Testcase	Vết	Impact	Derived from	Coverage
TC01	Yêu cầu hệ thống cho phép người dùng đăng ký tài khoản mới với thông tin hợp lệ	Form giao diện, Backend, CSDL	Yêu cầu chức năng hệ thống về đăng ký tài khoản	Giao diện, API, Validation logic, CSDL
TC02	Yêu cầu hệ thống không cho phép trùng lặp email	Form giao diện, Backend, CSDL	Yêu cầu chức năng hệ thống về đăng ký tài khoản	Giao diện, API, Validation logic, CSDL
TC03	Yêu cầu hệ thống không cho phép trùng lặp mã số sinh viên	Form giao diện, Backend, CSDL	Yêu cầu chức năng hệ thống về đăng ký tài khoản	Giao diện, API, Validation logic, CSDL
TC04	Yêu cầu hệ thống không cho phép trùng lặp username	Form giao diện, Backend, CSDL	Yêu cầu chức năng hệ thống về đăng ký tài khoản	Giao diện, API, Validation logic, CSDL
TC05	Yêu cầu hệ thống yêu cầu điền đầy đủ thông tin bắt buộc	Form giao diện, Backend, CSDL	Yêu cầu chức năng hệ thống về đăng ký tài khoản	Giao diện, API, Validation logic, CSDL
TC06	Yêu cầu hệ thống kiểm tra định dạng hợp lệ của email	Form giao diện, Backend, CSDL	Yêu cầu chức năng hệ thống về đăng ký tài khoản	Giao diện, API, Validation logic, CSDL

3. 2. 2. ĐẶC TẢ CÁC TESTCASES CHO USECASE 2

Test Case TC07

- Tên testcase: Tạo nhóm thành công với 3 thành viên hợp lệ chưa có nhóm
- Mô tả: Thí sinh có tài khoản hợp lệ tạo nhóm với hai thí sinh khác chưa thuộc nhóm nào, tất cả đều hợp lệ và nhóm được tạo thành công.
- Tiền điều kiện:
 - Cả ba thí sinh đã có tài khoản hợp lệ trong hệ thống.
 - Không thí sinh nào trong ba người thuộc bất kỳ nhóm nào khác.
- Các bước thực hiện:
 - Thí sinh đăng nhập hệ thống.
 - Truy cập giao diện tạo nhóm.
 - Nhập email của 2 thí sinh khác.
 - Nhấn nút “Tạo nhóm”.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- 5. Hệ thống kiểm tra tính hợp lệ của email và trạng thái nhóm của các thí sinh.
- 6. Nếu hợp lệ → Lưu vào cơ sở dữ liệu → Hiển thị thông báo tạo nhóm thành công.
- Kỹ thuật kiểm thử: Phân vùng tương đương (Equivalence Partitioning), Positive Testing
- Công cụ hiện thực hóa: Selenium (UI), Postman (API POST /team)

Bảng quyết định kiểm thử:

Thành viên A hợp lệ	Thành viên B hợp lệ	Thành viên C hợp lệ	Tất cả chưa có nhóm	Kết quả
Có	Có	Có	Có	Tạo nhóm thành công

Test Case TC08

- Tên testcase: Tạo nhóm thất bại do có thành viên không tồn tại
- Mô tả: Một hoặc cả hai email nhập vào không khớp với bất kỳ tài khoản thí sinh nào trong hệ thống.
- Tiền điều kiện: Có ít nhất một email không tồn tại trong bảng contestants.
- Các bước thực hiện:
 1. Đăng nhập hệ thống.
 2. Truy cập form tạo nhóm.
 3. Nhập email của ít nhất 1 người chưa có tài khoản.
 4. Nhấn “Tạo nhóm”.
 5. Hệ thống kiểm tra và phát hiện email không tồn tại → trả về thông báo lỗi.
- Kỹ thuật kiểm thử: Negative Testing
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Email A tồn tại	Email B tồn tại	Email C tồn tại	Kết quả
Có	Có	Không	Lỗi không tìm thấy thí sinh
Có	Không	Có	Lỗi không tìm thấy thí sinh

Test Case TC09

- Tên testcase: Tạo nhóm thất bại do có thành viên đã thuộc nhóm khác
- Mô tả: Một trong các thí sinh đã thuộc nhóm khác → hệ thống từ chối tạo nhóm mới.
- Tiền điều kiện:
- Có ít nhất một thí sinh (A, B, hoặc C) đã có trường teamId khác null.
- Các bước thực hiện:
 1. Đăng nhập hệ thống.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

2. Truy cập form tạo nhóm.
 3. Nhập email hợp lệ, nhưng một trong số đó đã có nhóm.
 4. Nhấn nút “Tạo nhóm”.
 5. Hệ thống phát hiện trạng thái đã thuộc nhóm → hiện thông báo lỗi.
- Kỹ thuật kiểm thử: Negative Testing
 - Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Thành viên A đã có nhóm	Thành viên B đã có nhóm	Thành viên C đã có nhóm	Kết quả
Không	Có	Không	Lỗi thành viên đã có nhóm
Có	Không	Không	Lỗi thành viên đã có nhóm

Test Case TC10

- Tên testcase: Tạo nhóm thất bại do nhập email trùng nhau
- Mô tả: Người dùng nhập cùng một email cho nhiều thành viên, vi phạm tính duy nhất của từng thành viên trong nhóm.
- Tiền điều kiện: Người tạo nhóm có tình nhập trùng email của chính mình hoặc người khác.
- Các bước thực hiện:
 1. Đăng nhập hệ thống.
 2. Truy cập form tạo nhóm.
 3. Nhập cùng một email ở hai trường email khác nhau.
 4. Nhấn “Tạo nhóm”.
 5. Hệ thống phát hiện lỗi trùng → trả thông báo lỗi.
- Kỹ thuật kiểm thử: Negative Testing
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Email A = B	Email A = C	Email B = C	Kết quả
Có	Không	Không	Lỗi email bị trùng
Không	Có	Không	Lỗi email bị trùng

Test Case TC11

- Tên testcase: Tạo nhóm thất bại do bỏ trống thông tin thành viên

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- Mô tả: Người dùng không điền email cho một hoặc cả hai thí sinh khi tạo nhóm → hệ thống từ chối xử lý.
- Tiền điều kiện: Không có
- Các bước thực hiện:
 - Đăng nhập hệ thống.
 - Truy cập form tạo nhóm.
 - Để trống email của thành viên 1 hoặc 2.
 - Nhấn nút “Tạo nhóm”.
 - Hệ thống trả về lỗi “thiếu thông tin bắt buộc”.
- Kỹ thuật kiểm thử: Phân tích giá trị biên (Boundary Value Analysis), Negative Testing
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Email A có nhập	Email B có nhập	Email C có nhập	Kết quả
Có	Có	Trống	Lỗi thiếu thông tin
Có	Trống	Có	Lỗi thiếu thông tin

Test Case TC12

- Tên testcase: Tạo nhóm thất bại do email thành viên không đúng định dạng
- Mô tả: Người dùng điền email của thành viên muốn đăng kí nhưng email không đúng định dạng email yêu cầu.
- Tiền điều kiện: Không có
- Các bước thực hiện:
 - Đăng nhập hệ thống
 - Truy cập form tạo nhóm.
 - Điền chuỗi bất kì không phải email hoặc email sai định dạng vào email của thành viên muốn đăng kí bất kì.
 - Nhấn nút “Tạo nhóm”.
 - Hệ thống trả về lỗi “Sai định dạng email”.
- Kỹ thuật kiểm thử: Phân tích giá trị biên (Boundary Value Analysis), Negative Testing
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Email A có nhập	Email B sai định dạng, có nhập	Email C sai định dạng, có nhập	Kết quả
Có	Có	Không	Lỗi định dạng email
Có	Không	Có	Lỗi định dạng email

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

Test Case TC13

- Tên testcase: Tạo nhóm thất bại do tên nhóm bị trùng.
- Mô tả: Người dùng trong lúc tạo nhóm điền tên nhóm muốn tạo nhưng tên đã bị sử dụng trước đó.
- Tiền điều kiện: Tên của nhóm đã được sử dụng.
- Các bước thực hiện:
 - Đăng nhập hệ thống
 - Truy cập vào form tạo nhóm
 - Điền tên nhóm đã được sử dụng vào tên nhóm
 - Nhấn nút “Tạo nhóm”
 - Hệ thống trả về lỗi “Tên nhóm đã được sử dụng”
- Kỹ thuật kiểm thử: Phân tích giá trị biên (Boundary Value Analysis), Negative testing
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Tên nhóm đã sử dụng	Kết quả
Có	Lỗi tên nhóm đã sử dụng

Bảng truy vết yêu cầu -> thiết kế -> testcase cho các testcase TC07 – TC11:

Testcase	Yêu cầu hệ thống	Thiết kế	Thực hiện kiểm thử
TC07	Cho phép tạo nhóm gồm 3 thành viên hợp lệ chưa thuộc nhóm nào	Form tạo nhóm, API POST /teams, TeamController	TC07
TC08	Hệ thống kiểm tra sự tồn tại của thí sinh khi tạo nhóm	Form tạo nhóm, kiểm tra email trong bảng “contestants”	TC08
TC09	Không cho phép thí sinh đã có nhóm tham gia nhóm mới	Kiểm tra teamId trước khi tạo nhóm, TeamService	TC09
TC10	Các thành viên trong nhóm phải có email khác nhau	Validate unique email khi tạo nhóm	TC10
TC11	Không được bỏ trống thông tin bắt buộc khi tạo nhóm	Kiểm tra dữ liệu đầu vào khi tạo nhóm	TC11
TC12	Không cho tạo nhóm khi email thành viên không đúng định dạng	Kiểm tra định dạng email ở Frontend	TC12
TC13	Không cho tạo nhóm khi tên nhóm bị trùng	Kiểm tra tên nhóm đã được sử dụng hay chưa ở database	TC13

Bảng impact/derive/coverage:

Testcase	Vết	Impact	Derived from	Coverage

Báo cáo đồ án Trang đăng kí tham gia cuộc thi lập trình thi đấu

TC07	Tạo nhóm thành viên hợp lệ	Form UI, TeamService, CSDL bảng “teams”	Yêu cầu chức năng về tạo nhóm thi đấu	Form, API, Validation, DB
TC08	Kiểm tra tồn tại thí sinh	TeamController, ContestantService	Ràng buộc hệ thống về tính hợp lệ thành viên	API, DB lookup
TC09	Ràng buộc nhóm duy nhất	TeamService, CSDL bảng “contestants”	Quy định hệ thống: mỗi thí sinh chỉ thuộc 1 nhóm	Logic nhóm, DB kiểm tra, API phản hồi
TC10	Ràng buộc dữ liệu đầu vào không trùng lặp	Form frontend, validation backend	Yêu cầu kiểm soát tính duy nhất	Form, validation server-side
TC11	Kiểm tra dữ liệu đầu vào đầy đủ	Form UI, validation controller	Yêu cầu kiểm tra dữ liệu nhập	Form, validation, phản hồi lỗi
TC12	Email thành viên không đúng định dạng	Form UI, API tạo nhóm	Yêu cầu chức năng hệ thống về đăng kí nhóm	Giao diện, API, CSDL
TC13	Ràng buộc tên nhóm là duy nhất	ContestService, CSDL bảng “contests”	Ràng buộc hệ thống về tên nhóm	API, DB lookup

3. 2. 3. Đặc tả các testcases cho Usecase 3

Test Case TC14

- Tên testcase: Ghi danh thành công cuộc thi cá nhân với thí sinh hợp lệ
- Mô tả: Thí sinh có tài khoản hợp lệ và chưa từng tham gia cuộc thi cá nhân → hệ thống xử lý và ghi danh thành công.
- Tiền điều kiện:
 - Cuộc thi loại cá nhân còn hiệu lực và có thể đăng ký.
 - Thí sinh đã đăng ký tài khoản hợp lệ.
 - Thí sinh chưa đăng ký cuộc thi đó trước đó.
- Các bước thực hiện:
 1. Thí sinh đăng nhập vào hệ thống.
 2. Truy cập danh sách cuộc thi.
 3. Chọn cuộc thi cá nhân chưa đăng ký.
 4. Nhấn “Đăng ký”.
 5. Hệ thống xác thực điều kiện và ghi danh thành công.
- Kỹ thuật kiểm thử: Phân vùng tương đương, Positive Testing
- Công cụ hiện thực hóa: Selenium

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

Bảng quyết định kiểm thử:

Cuộc thi cá nhân	Thí sinh hợp lệ	Đã ghi danh	Kết quả
Có	Có	Chưa	Ghi danh thành công

Test Case TC15

- Tên testcase: Ghi danh thất bại vì đã tham gia cuộc thi cá nhân
- Mô tả: Thí sinh đã từng đăng ký tham gia cuộc thi đó → hệ thống không cho ghi danh lại.
- Tiền điều kiện:
 - Cuộc thi là loại cá nhân.
 - Thí sinh đã đăng ký cuộc thi đó trước đó.
- Các bước thực hiện:
 1. Đăng nhập hệ thống.
 2. Truy cập danh sách cuộc thi.
 3. Nhấn “Đăng ký” cuộc thi đã đăng ký trước đó.
 4. Hệ thống hiển thị lỗi “Đã ghi danh”.
- Kỹ thuật kiểm thử: Negative Testing
- Công cụ hiện thực hóa: Postman

Bảng quyết định kiểm thử:

Cuộc thi cá nhân	Thí sinh hợp lệ	Đã ghi danh	Kết quả
Có	Có	Có	Lỗi đã ghi danh cuộc thi

Test Case TC16

- Tên testcase: Ghi danh thành công cuộc thi theo nhóm
- Mô tả: Một thành viên trong nhóm hợp lệ đăng ký cuộc thi nhóm, toàn bộ nhóm được hệ thống ghi danh.
- Tiền điều kiện:
 - Cuộc thi là loại nhóm.
 - Người đăng ký và cả nhóm đã được tạo hợp lệ.
 - Nhóm chưa tham gia cuộc thi trước đó.
- Các bước thực hiện:
 1. Thành viên nhóm đăng nhập.
 2. Truy cập danh sách cuộc thi.
 3. Nhấn “Đăng ký” tại cuộc thi nhóm chưa tham gia.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- 4. Hệ thống kiểm tra nhóm và ghi danh toàn nhóm vào cuộc thi.
- Kỹ thuật kiểm thử: Phân vùng tương đương, Positive Testing
- Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Cuộc thi nhóm	Có nhóm hợp lệ	Đã ghi danh	Kết quả
Có	Có	Chưa	Ghi danh cả nhóm thành công

Test Case TC17

- Tên testcase: Ghi danh thất bại vì nhóm đã tham gia cuộc thi
- Mô tả: Nhóm thí sinh đã đăng ký cuộc thi trước đó, hệ thống từ chối ghi danh lại.
- Tiền điều kiện:
- Cuộc thi là nhóm.
- Nhóm đã đăng ký cuộc thi trước đó.
- Các bước thực hiện:
 1. Thành viên nhóm đăng nhập.
 2. Chọn cuộc thi nhóm đã tham gia.
 3. Nhấn “Đăng ký”.
 4. Hệ thống trả thông báo lỗi “Đã ghi danh”.
- Kỹ thuật kiểm thử: Negative Testing
- Công cụ hiện thực hóa: Postman

Bảng quyết định kiểm thử:

Cuộc thi nhóm	Có nhóm hợp lệ	Đã ghi danh	Kết quả
Có	Có	Có	Lỗi nhóm đã tham gia cuộc thi

Test Case TC18

- Tên testcase: Ghi danh thất bại vì chưa có nhóm khi ghi danh thí thi nhóm.
- Mô tả: Thí sinh muốn ghi danh cuộc thi nhóm nhưng chưa lập nhóm → hệ thống từ chối.
- Tiền điều kiện:
- Cuộc thi là nhóm.
- Thí sinh chưa tạo hoặc tham gia nhóm nào.
- Các bước thực hiện:
 1. Thí sinh đăng nhập.

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

2. Chọn cuộc thi nhóm bất kỳ.
 3. Nhấn “Đăng ký”.
 4. Hệ thống hiển thị lỗi “Chưa có nhóm hợp lệ”.
- Kỹ thuật kiểm thử: Negative Testing
 - Công cụ hiện thực hóa: Selenium

Bảng quyết định kiểm thử:

Cuộc thi nhóm	Có nhóm hợp lệ	Kết quả
Có	Không	Lỗi chưa có nhóm để ghi danh

Test Case TC19

- Tên testcase: Ghi danh thất bại do cuộc thi không tồn tại
- Mô tả: Người dùng cố gắng ghi danh cuộc thi bằng ID không hợp lệ hoặc cuộc thi đã bị xóa.
- Tiền điều kiện:
- Mã cuộc thi không có trong hệ thống hoặc đã bị xoá.
- Các bước thực hiện:
 1. Đăng nhập hệ thống.
 2. Gửi yêu cầu ghi danh với mã cuộc thi không tồn tại.
 3. Hệ thống kiểm tra và trả thông báo lỗi.
- Kỹ thuật kiểm thử: Negative Testing
- Công cụ hiện thực hóa: Postman

Bảng quyết định kiểm thử:

Cuộc thi tồn tại	Kết quả
Không	Lỗi không tìm thấy cuộc thi

Bảng truy vết yêu cầu -> thiết kế -> testcase cho các testcase TC01 – TC06:

Testcase	Yêu cầu hệ thống	Thiết kế	Thực hiện kiểm thử
TC14	Cho phép thí sinh hợp lệ ghi danh cuộc thi cá nhân nếu chưa tham gia	Form danh sách cuộc thi, API POST contests/participate, kiểm tra loại cuộc thi	TC14
TC15	Không cho phép thí sinh ghi danh cuộc thi cá nhân nếu đã tham gia	Kiểm tra trạng thái ghi danh trong bảng “contest_participations”	TC15
TC16	Cho phép nhóm hợp lệ ghi danh vào cuộc thi nhóm	API kiểm tra danh sách thành viên nhóm, ghi danh toàn nhóm	TC16

Báo cáo đồ án Trang đăng kí tham gia cuộc thi lập trình thi đấu

TC17	Không cho phép nhóm ghi danh vào cuộc thi nếu đã tham gia	Kiểm tra trạng thái trong bảng “contest_participations” theo từng thành viên	TC17
TC18	Không cho phép ghi danh vào cuộc thi nhóm nếu chưa có nhóm	Kiểm tra contestant.teamId	TC18
TC19	Không ghi danh nếu cuộc thi không tồn tại	API kiểm tra sự tồn tại của cuộc thi trong bảng “contests”	TC19

Bảng impact/derive/coverage:

Testcase	Vết	Impact	Derived from	Coverage
TC14	Ghi danh thành công cuộc thi cá nhân	Form UI, ContestService, bảng “contest_participations”	Yêu cầu cho phép đăng kí cuộc thi cá nhân	Form + API + DB + Kiểm tra trạng thái
TC15	Ngăn chặn ghi danh trùng cho cuộc thi cá nhân	Validation ghi danh, bảng “contest_participations”	Quy tắc mỗi thí sinh chỉ tham gia một lần	Logic kiểm tra lịch sử + API phản hồi
TC16	Ghi danh cuộc thi theo nhóm hợp lệ	ContestService, TeamService, DB bảng “teams” và “contest_participations”	Yêu cầu ghi danh cho nhóm	Form + API + DB + Lặp theo thành viên
TC17	Ngăn chặn ghi danh lại nếu nhóm đã tham gia	ContestService và bảng “contest_participations”	Không ghi danh nhiều lần	API + Truy vấn DB + Logic nhóm
TC18	Kiểm tra tồn tại nhóm trước khi ghi danh nhóm	ContestController, contestant.teamId	Ràng buộc nhóm phải tồn tại	Kiểm tra điều kiện trước khi xử lý
TC19	Kiểm tra mã cuộc thi tồn tại	ContestService, bảng “contests”	Ràng buộc dữ liệu phải tồn tại	API + DB truy xuất + Kiểm tra đầu vào

3. 2. 4. Hiện thực các testcases

Kiểm thử Usecase đăng kí tài khoản thí sinh

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

```
39  async function findSignUpElements(driver: WebDriver): Promise<{ emailInput: WebElement; passwordInput: WebElement; usernameInput: WebElement; firstNameInput: WebElement; lastNameInput: WebElement; studentIdInput: WebElement; genderSelect: WebElement; schoolYearSelect: WebElement; signUpButton: WebElement }> {
40    const emailInput: WebElement = await driver.findElement(By.css('input[type="email"]'));
41    const passwordInput: WebElement = await driver.findElement(
42      By.css('input[type="password"]')
43    );
44    const usernameInput: WebElement = await driver.findElement(
45      By.css('input[name="username"]')
46    );
47    const firstNameInput: WebElement = await driver.findElement(
48      By.css('input[name="first_name"]')
49    );
50    const lastNameInput: WebElement = await driver.findElement(
51      By.css('input[name="last_name"]')
52    );
53    const studentIdInput: WebElement = await driver.findElement(
54      By.css('input[name="student_id"]')
55    );
56    const genderSelect: WebElement = await driver.findElement(By.id("gender"));
57    const schoolYearSelect: WebElement = await driver.findElement(By.id("school_year"));
58    const signUpButton: WebElement = await driver.findElement(
59      By.css("button[name='submit']")
60    );
61
62    return {
63      emailInput, passwordInput, usernameInput,
64      firstNameInput, lastNameInput, studentIdInput,
65      genderSelect, schoolYearSelect, signUpButton,
66    };
67  }
```

Hình 24: Hàm tìm thành phần trên trang đăng kí

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

```
75  async function clearAndFillInputs(  
76    elements: any,  
77    data: {  
78      email: string; password: string; username: string;  
79      firstName: string; lastName: string; studentId: string;  
80    }  
81  ) : Promise<void> {  
82    await sleep(500);  
83    await forceClearInput(elements.emailInput);  
84    await forceClearInput(elements.passwordInput);  
85    await forceClearInput(elements.usernameInput);  
86    await forceClearInput(elements.firstNameInput);  
87    await forceClearInput(elements.lastNameInput);  
88    await forceClearInput(elements.studentIdInput);  
89    await sleep(500);  
90    await elements.emailInput.sendKeys(data.email);  
91    await elements.passwordInput.sendKeys(data.password);  
92    await elements.usernameInput.sendKeys(data.username);  
93    await elements.firstNameInput.sendKeys(data.firstName);  
94    await elements.lastNameInput.sendKeys(data.lastName);  
95    await elements.studentIdInput.sendKeys(data.studentId);  
96  
97    await elements.genderSelect.click(); await sleep(300);  
98    const genderOption : any = await elements.genderSelect.findElement(  
99      By.xpath("/html/body/div[2]/div/div/div[2]/div/div/div[1]/div")  
100 );  
101    await genderOption.click(); await sleep(300);  
102  
103    await elements.schoolYearSelect.click(); await sleep(300);  
104    const schoolYearOption : any = await elements.schoolYearSelect.findElement(  
105      By.xpath("/html/body/div[3]/div/div/div[2]/div[1]/div/div/div[1]/div")  
106 );  
107    await sleep(300); await schoolYearOption.click(); await sleep(500);  
108 }
```

Hình 25: Hàm điền dữ liệu vào các thành phần nhập liệu trang đăng kí

```
120  async function testInvalidEmail(driver: WebDriver, elements: any) : Promise<void> {  
121    logTestData("Invalid Email Format", SignUpTest.InvalidEmail);  
122  
123    await clearAndFillInputs(elements, SignUpTest.InvalidEmail);  
124    await elements.signUpButton.click();  
125  
126    try {  
127      await driver.wait(until.elementLocated(By.id("email-error")), 3000);  
128      console.log("✅ Invalid Email Test Passed");  
129    } catch {  
130      console.error("✖ Invalid Email Test Failed: No error displayed");  
131    }  
132 }
```

Hình 26: Hàm kiểm tra email không đúng định dạng

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

```
134  async function testUsernameTaken(driver: WebDriver, elements: any) : Promise<void> {
135    logTestData("Username Already Taken", SignUpTest.UsernameTaken);
136
137    await clearAndFillInputs(elements, SignUpTest.UsernameTaken);
138    await elements.signUpButton.click();
139
140    try {
141      await driver.wait(until.elementLocated(By.id("username-error")), 3000);
142      console.log("✅ Username Taken Test Passed");
143    } catch {
144      console.error("✖ Username Taken Test Failed: No error displayed");
145    }
146 }
```

Hình 27: Hàm kiểm tra username đã được sử dụng

```
148  async function testStudentIdTaken(driver: WebDriver, elements: any) : Promise<void> {
149    logTestData("Student ID Already Taken", SignUpTest.StudentIdTaken);
150
151    await clearAndFillInputs(elements, SignUpTest.StudentIdTaken);
152    await elements.signUpButton.click();
153
154    try {
155      await driver.wait(until.elementLocated(By.id("student-id-error")), 3000);
156      console.log("✅ Student ID Taken Test Passed");
157    } catch {
158      console.error("✖ Student ID Taken Test Failed: No error displayed");
159    }
160 }
```

Hình 28: Hàm kiểm tra mã sinh viên đã sử dụng

```
162  async function testEmailTaken(driver: WebDriver, elements: any) : Promise<void> {
163    logTestData("Email Already Taken", SignUpTest.EmailTaken);
164
165    await clearAndFillInputs(elements, SignUpTest.EmailTaken);
166    await elements.signUpButton.click();
167
168    try {
169      await driver.wait(until.elementLocated(By.id("email-taken-error")), 3000);
170      console.log("✅ Email Taken Test Passed");
171    } catch {
172      console.error("✖ Email Taken Test Failed: No error displayed");
173    }
174 }
```

Hình 29: Hàm kiểm tra email đã sử dụng

Báo cáo dossier Trang đăng ký tham gia cuộc thi lập trình thi đấu

```
176  async function testSuccessfulSignUp(driver: WebDriver, elements: any) : Promise<void> {
177    logTestData("Successful Sign Up", SignUpTest.SignUpSuccess);
178
179    await clearAndFillInputs(elements, SignUpTest.SignUpSuccess);
180    await elements.signUpButton.click();
181
182    try {
183      await driver.sleep(1000);
184      const currentUrl : string = await driver.getCurrentUrl();
185      if (currentUrl === `${BASE_URL}/`){
186        console.log("✓ Successfully navigated to home page");
187        console.log("✓ Sign-Up Success Test Passed");
188      } else {
189        throw new Error("Failed to navigate to home page");
190      }
191    } catch (error: any) {
192      console.error("✗ Sign-Up Success Test Failed:", error);
193    }
194 }
```

Hình 30: Hàm kiểm tra đăng nhập thành công

Kiểm thử UseCase tạo nhóm tham gia các cuộc thi

```
32  async function findTeamElements(driver: WebDriver) : Promise<{ nameInput: WebElement; descri... {
33    // Wait for form to be visible
34    await driver.wait(until.elementLocated(By.id("input-team-name")), 1000);
35
36    const nameInput : WebElement = await driver.findElement(By.id("input-team-name"));
37    const descriptionInput : WebElement = await driver.findElement(
38      By.id("input-team-description")
39    );
40    const member2Input : WebElement = await driver.findElement(By.id("input-member-2"));
41    const member3Input : WebElement = await driver.findElement(By.id("input-member-3"));
42    const createButton : WebElement = await driver.findElement(
43      By.xpath("//button[contains(text(), 'Create Team')]")
44    );
45
46    // Wait for elements to be interactable
47    await driver.wait(until.elementIsVisible(nameInput), 5000);
48    await driver.wait(until.elementIsVisible(descriptionInput), 5000);
49    await driver.wait(until.elementIsVisible(createButton), 5000);
50
51    return {
52      nameInput, descriptionInput, member2Input, member3Input, createButton,
53    };
54 }
```

Hình 31: Hàm tìm các thành phần trang thông tin nhóm

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

```
56  async function clearAndFillInputs(
57    driver: WebDriver,
58    elements: any,
59    data: {
60      name: string;
61      description: string;
62      members: any;
63    }
64  ): Promise<void> {
65    await sleep(1000); // Increased wait time
66
67    // Wait for and clear name input
68    await driver.wait(until.elementIsEnabled(elements.nameInput), 5000);
69    await forceClearInput(elements.nameInput);
70    await elements.nameInput.sendKeys(data.name);
71
72    // Wait for and clear description input
73    await driver.wait(until.elementIsEnabled(elements.descriptionInput), 5000);
74    await forceClearInput(elements.descriptionInput);
75    await elements.descriptionInput.sendKeys(data.description);
76
77    // Fill member emails (only second and third members)
78    if (data.members[1]) {
79      await driver.wait(until.elementIsEnabled(elements.member2Input), 5000);
80      await forceClearInput(elements.member2Input);
81      await elements.member2Input.sendKeys(data.members[1]);
82    }
83
84    if (data.members[2]) {
85      await driver.wait(until.elementIsEnabled(elements.member3Input), 5000);
86      await forceClearInput(elements.member3Input);
87      await elements.member3Input.sendKeys(data.members[2]);
88    }
89 }
```

Hình 32: Hàm điền thông tin vào các thành phần nhập liệu

Báo cáo dossier Trang đăng ký tham gia cuộc thi lập trình thi đấu

```
91  async function testInvalidMemberEmail(driver: WebDriver, elements: any) : Promise<void> {
92    logTestData("Invalid Member Email", TeamTest.InvalidMemberEmail);
93
94    await clearAndFillInputs(driver, elements, TeamTest.InvalidMemberEmail);
95    await sleep(2000);
96    await elements.createButton.click();
97
98    try {
99      await driver.wait(
100        until.elementLocated(
101          By.xpath("//*[contains(text(), 'Invalid email format')]")
102        ),
103        3000
104      );
105      console.log("✅ Invalid Member Email Test Passed");
106      await sleep(2000);
107    } catch {
108      console.error("❌ Invalid Member Email Test Failed: No error displayed");
109      await sleep(2000);
110    }
111 }
```

Hình 33: Hàm kiểm tra email không hợp lệ

```
113  async function testNonExistentMember(driver: WebDriver, elements: any) : Promise<void> {
114    logTestData("Non-existent Member", TeamTest.NonExistentMember);
115
116    await clearAndFillInputs(driver, elements, TeamTest.NonExistentMember);
117    await sleep(2000);
118    await elements.createButton.click();
119
120    try {
121      await driver.wait(
122        until.elementLocated(
123          By.xpath("//*[contains(text(), 'Contestants not found')]")
124        ),
125        3000
126      );
127      console.log("✅ Non-existent Member Test Passed");
128      await sleep(2000);
129    } catch {
130      console.error("❌ Non-existent Member Test Failed: No error displayed");
131      await sleep(2000);
132    }
133 }
```

Hình 34: Hàm kiểm tra thí sinh không tồn tại

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

```
135  async function testNameTaken(driver: WebDriver, elements: any) : Promise<void> {
136    logTestData("Team Name Taken", TeamTest.NameTaken);
137
138    await clearAndFillInputs(driver, elements, TeamTest.NameTaken);
139    await sleep(2000);
140    await elements.createButton.click();
141
142    try {
143      await driver.wait(
144        until.elementLocated(
145          By.xpath("//*[contains(text(), 'Name has been taken')]")
146        ),
147        3000
148      );
149      console.log("✅ Team Name Taken Test Passed");
150      await sleep(2000);
151    } catch {
152      console.error("❌ Team Name Taken Test Failed: No error displayed");
153      await sleep(2000);
154    }
155  }
```

Hình 35: Hàm kiểm tra tên nhóm đã tồn tại

```
157  async function testCreateTeamSuccess(driver: WebDriver, elements: any) : Promise<void> {
158    logTestData("Create Team Success", TeamTest.SuccessCreate);
159
160    await clearAndFillInputs(driver, elements, TeamTest.SuccessCreate);
161    await sleep(2000);
162    await elements.createButton.click();
163
164    try {
165      await driver.wait(
166        until.elementLocated(
167          By.xpath("//*[contains(text(), 'Team created successfully')]")
168        ),
169        3000
170      );
171      console.log("✅ Create Team Success Test Passed");
172      await sleep(2000);
173    } catch {
174      console.error(
175        "❌ Create Team Success Test Failed: No success message displayed"
176      );
177      await sleep(2000);
178    }
179  }
```

Hình 36: Hàm tạo nhóm thành công

Báo cáo dossier Trang đăng ký tham gia cuộc thi lập trình thi đấu

```
445 // Check for validation errors on required fields
446 try {
447     // Look for error messages on firstName and lastName fields
448     await driver.wait(until.elementLocated(By.id("first-name-error")), 3000);
449     console.log("✅ First Name validation error detected");
450 } catch {
451     console.log("⚠️ No first name validation error found");
452 }
453
454 try {
455     await driver.wait(until.elementLocated(By.id("last-name-error")), 3000);
456     console.log("✅ Last Name validation error detected");
457 } catch {
458     console.log("⚠️ No last name validation error found");
459 }
460
461 // Check if we're still on the sign-up page (form should not submit)
462 const currentUrl : string = await driver.getCurrentUrl();
463 if (currentUrl === `${BASE_URL}/sign-up`) {
464     console.log(
465         "✅ Form correctly prevented submission with missing information"
466     );
467     console.log("✅ Missing Information Test Passed");
468
469     addTestResult({
470         testName,
471         testId: SignUpTest.MissingInformation.testId,
472         testType: SignUpTest.MissingInformation.testType,
473         status: "PASSED",
474         duration: Date.now() - startTime,
475     });
476 } else {
477     throw new Error("Form submitted despite missing required fields");
478 }
479 } catch (error) {
480     console.error("❌ Missing Information Test Failed:", error);
481 }
```

Hình 37: Hàm kiểm tra các thông tin điền bị thiếu

Báo cáo dossier Trang đăng ký tham gia cuộc thi lập trình thi đấu

Kiểm thử Usecase ghi danh cuộc thi

```
180  async function testSuccessSingleRegistration(driver: WebDriver) : Promise<void> {
181    const startTime : number = Date.now();
182    const testName = "Success Single Registration";
183    const { testId, testType, contestName } =
184      RegistrationTest.SuccessSingleRegistration;
185    logTestData(testName, { testId, contestName, testType });
186    try {
187      const elements : { contestCard: WebElement; registerButton... = await findContestElements(driver, contestName);
188      await elements.registerButton.click();
189      await sleep(2000);
190      const modalElements : { modal: WebElement; confirmButton: WebElement... = await findModalElements(driver);
191      await modalElements.confirmButton.click();
192      await sleep(3000);
193      await driver.wait(
194        until.elementLocated(
195          By.xpath("//*[contains(text(), 'Register success')]")
196        ),
197        5000
198      );
199      console.log("✅ Success Single Registration Test Passed");
200      addTestResult({
201        testName,
202        testId,
203        testType,
204        status: "PASSED",
205        duration: Date.now() - startTime,
206      });
207      await sleep(2000);
208    } catch (error) {
209      console.error("❌ Success Single Registration Test Failed:", error);
210      addTestResult({
211        testName,
212        testId,
213        testType,
214        status: "FAILED",
215        duration: Date.now() - startTime,
216        error: error instanceof Error ? error.message : String(error),
217      });
218      await sleep(2000);
219    }
220  }
```

Hình 38: Hàm kiểm tra đăng kí cuộc thi đơn thành công

Báo cáo dossier Trang đăng ký tham gia cuộc thi lập trình thi đấu

```
222  async function testSuccessTeamRegistration(driver: WebDriver) : Promise<void> {
223    const startTime : number = Date.now();
224    const testName = "Success Team Registration";
225    const { testId, testType, contestName } =
226      RegistrationTest.SuccessTeamRegistration;
227    logTestData(testName, { testId, contestName, testType });
228    try {
229      const elements : { contestCard: WebElement; registerButton... = await findContestElements(driver, contestName);
230      await elements.registerButton.click();
231      await sleep(2000);
232      const modalElements : { modal: WebElement; confirmButton: WebElement... = await findModalElements(driver);
233      await modalElements.confirmButton.click();
234      await sleep(3000);
235      await driver.wait(
236        until.elementLocated(
237          By.xpath("//*[contains(text(), 'Register success')]")
238        ),
239        5000
240      );
241      console.log("✅ Success Team Registration Test Passed");
242      addTestResult({
243        testName,
244        testId,
245        testType,
246        status: "PASSED",
247        duration: Date.now() - startTime,
248      });
249      await sleep(2000);
250    } catch (error) {
251      console.error("🔴 Success Team Registration Test Failed:", error);
252      addTestResult({
253        testName,
254        testId,
255        testType,
256        status: "FAILED",
257        duration: Date.now() - startTime,
258        error: error instanceof Error ? error.message : String(error),
259      });
260      await sleep(2000);
261    }
262  }
```

Hình 39: Hàm kiểm tra cuộc thi nhóm thành công

Báo cáo dossier Trang đăng ký tham gia cuộc thi lập trình thi đấu

```
264  async function testNoTeamRegistrationFailed(driver: WebDriver) : Promise<void> {
265    const startTime : number = Date.now();
266    const testName = "No Team Registration Failed";
267    const { testId, testType, contestName } =
268      RegistrationTest.NoTeamRegistrationFailed;
269    logTestData(testName, { testId, contestName, testType });
270    try {
271      const elements : { contestCard: WebElement; registerButton... = await findContestElements(driver, contestName);
272      await elements.registerButton.click();
273      await sleep(2000);
274      const modalElements : { modal: WebElement; confirmButton: WebEl... = await findModalElements(driver);
275      await modalElements.confirmButton.click();
276      await sleep(3000);
277      await driver.wait(
278        until.elementLocated(
279          By.xpath(
280            "//*[contains(text(), 'Contestant must joined a team to be able to participate in the contest')]"
281          )
282        ),
283        5000
284      );
285      console.log("✅ No Team Registration Failed Test Passed");
286    > addTestResult({ ...
287    });
288    await sleep(2000);
289  } catch (error) {
290    console.error("❌ No Team Registration Failed Test Failed:", error);
291    addTestResult({ ...
292    });
293    await sleep(2000);
294  }
295  >
296  >
297  >
298  >
299  >
300  >
301  >
302  >
303  >
304  >
305  >
306 }
```

Hình 40: Hàm kiểm tra ghi danh cuộc thi nhóm nhưng không có nhóm

```
1  const loginUrl = 'http://localhost:3001/api/v1/auth/sign-in';
2  const loginPayload = {
3    email: 'nnminh@gmail.com',
4    password: '12345678'
5  };
6
7  pm.sendRequest({
8    url: loginUrl,
9    method: 'POST',
10   header: {
11     'Content-Type': 'application/json'
12   },
13   body: {
14     mode: 'raw',
15     raw: JSON.stringify(loginPayload)
16   }
17 }, function (err, res) {
18   if (err) {
19     console.error('Login failed:', err);
20     return;
21   }
22
23   const jsonResponse = res.json();
24
25   if (jsonResponse.data.jwt) {
26     pm.environment.set('access_token', jsonResponse.data.jwt);
27     console.log('Token saved:', jsonResponse.data.jwt);
28   } else {
29     console.warn('Access token not found in response');
30   }
31});
```

Hình 41: Code đăng nhập để test trong Postman

Báo cáo dossier án Trang đăng ký tham gia cuộc thi lập trình thi đấu

The screenshot shows a Postman API test configuration for a 'Contestant Participated Single Contest'. The method is POST, the URL is `{{host}}/contests/:id/participations`, and the host is set to `localhost:3001/api/v1`. The 'Params' tab shows a 'Key' named 'id' with a value of `cf1f2c88-bda2-468e-9d8e-5e4811cf8106`. The 'Path Variables' section also lists 'id' with the same value. The 'Send' button is visible at the top right.

Hình 42: API request để kiểm tra thí sinh ghi danh cuộc thi đơn đã ghi danh

The screenshot shows a Postman API test configuration for a 'Contestant Participated Single Contest'. The method is POST, the URL is `{{host}}/contests/:id/participations`. The 'Scripts' tab contains a 'Pre-request' script:

```

1 pm.test("Should return 400 if contestant already participated in a Single type contest", function () {
2     pm.response.to.have.status(400);
3 }
4 const jsonData = pm.response.json();
5
6 pm.expect(jsonData.message).to.eql("Contestant is already participated in the contest");
7 });

```

Hình 43: JS Code kiểm tra kết quả API request khi thí sinh ghi danh 2 lần cho cuộc thi đơn

The screenshot shows a Postman API test configuration for a 'Contestant Participated Team Contest'. The method is POST, the URL is `{{host}}/contests/:id/participations`. The 'Params' tab shows a 'Key' named 'id' with a value of `58e156c2-96e9-4939-8601-02cf52a32be6`. The 'Path Variables' section also lists 'id' with the same value. The 'Send' button is visible at the top right.

Hình 44: API request để kiểm tra thí sinh ghi danh cuộc thi nhóm đã ghi danh

The screenshot shows a Postman API test configuration for a 'Contestant Participated Team Contest'. The method is POST, the URL is `{{host}}/contests/:id/participations`. The 'Scripts' tab contains a 'Pre-request' script:

```

1 pm.test("Should return 400 if contestant already participated in a Team type contest", function () {
2     pm.response.to.have.status(400);
3 }
4 const jsonData = pm.response.json();
5
6 pm.expect(jsonData.message).to.eql("Contestant is already participated in the contest");
7 });

```

Hình 45: JS Code kiểm tra kết quả API request khi thí sinh ghi danh 2 lần cho cuộc thi nhóm

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

The screenshot shows a Postman API test configuration. The method is POST, the URL is {{host}}/contests/:id/participations, and the path variable id is set to 58e156c2-96e9-4939-8601-02cf52a32be7. The Headers tab shows 9 items. The Scripts tab contains the following JavaScript code:

```
pm.test("Should return 404 if contest is not found", function () {
    pm.response.to.have.status(404);

    const jsonData = pm.response.json();

    pm.expect(jsonData.message).to.eql("Contest not found");
});
```

Hình 46: API request để kiểm tra thí sinh ghi danh vào cuộc thi không tồn tại

The screenshot shows a Postman API test configuration. The method is POST, the URL is {{host}}/contests/:id/participations, and the path variable id is set to 58e156c2-96e9-4939-8601-02cf52a32be7. The Headers tab shows 9 items. The Scripts tab contains the same JavaScript code as in Figure 46.

Hình 47: JS code kiểm tra kết quả API khi thí sinh ghi danh vào cuộc thi không tồn tại

CHƯƠNG 4. KẾT LUẬN

4. 1. Điểm mạnh và điểm yếu của dự án

Điểm mạnh

1. Phân tích yêu cầu rõ ràng, có hệ thống

Dự án đã xác định được các chức năng cốt lõi của hệ thống: đăng ký tài khoản, tạo nhóm, và ghi danh cuộc thi – phù hợp với mục tiêu thực tế của một hệ thống thi đấu lập trình.

2. Tổ chức use case hợp lý, bám sát luồng người dùng

Các use case được mô hình hóa rõ ràng bằng sơ đồ, minh họa luồng nghiệp vụ trực quan, giúp dễ hiểu và dễ kiểm thử.

3. Thực hiện kiểm thử bài bản

Hệ thống kiểm thử được xây dựng đầy đủ với test case chi tiết, bảng quyết định, truy vết yêu cầu, và đánh giá coverage.

Điểm yếu

1. Chưa xử lý đầy đủ các trường hợp lỗi nâng cao

Một số trường hợp như “đăng nhập sai nhiều lần”, “hết hạn đăng ký”, “trùng tên nhóm”, v.v. chưa được mô tả trong test case hay luồng phụ.

2. Chưa tích hợp kiểm thử hiệu năng và tải

Dự án chưa đánh giá khả năng chịu tải hoặc phản hồi của hệ thống khi có nhiều người đăng ký cùng lúc, vốn rất quan trọng trong hệ thống thi đấu.

3. Chưa mô tả rõ cơ chế xác thực, bảo mật

Dự án mới dừng ở bước xử lý logic và xác thực đơn giản, chưa làm rõ các cơ chế mã hóa, bảo mật thông tin tài khoản, hay xử lý token đăng nhập.

4. 2. Khả năng cải tiến cách tiến hành đồ án

Để nâng cao chất lượng sản phẩm và tối ưu hiệu quả làm việc trong các lần phát triển tiếp theo, nhóm nhận thấy một số điểm trong cách tiến hành dự án có thể được cải tiến như sau:

1. Áp dụng quy trình phát triển phần mềm có mô hình rõ ràng

Dự án hiện được triển khai theo hướng tuyến tính, dễ thiếu phản hồi sớm từ kiểm thử hoặc người dùng. Trong tương lai, nhóm có thể áp dụng mô hình Iterative hoặc Agile/Scrum, giúp:

Báo cáo đồ án Trang đăng ký tham gia cuộc thi lập trình thi đấu

- Phát triển và kiểm thử song song theo từng vòng lặp nhỏ
- Tiếp nhận phản hồi liên tục từ giảng viên/người dùng
- Dễ thích nghi khi thay đổi yêu cầu hoặc ưu tiên

2. Phân công và quản lý tiến độ bằng công cụ chuyên nghiệp

Việc giao việc và kiểm soát tiến độ trong dự án được thực hiện khá thủ công. Nhóm có thể sử dụng các công cụ như Trello, Jira hoặc Notion để:

- Giao việc rõ ràng cho từng thành viên
- Theo dõi trạng thái công việc theo sprint
- Giảm nguy cơ trễ deadline hoặc trùng lặp nhiệm vụ

3. Tăng cường kiểm thử sớm và liên tục

Thay vì kiểm thử ở giai đoạn cuối, nhóm có thể áp dụng phương pháp Test-Driven Development (TDD) hoặc Continuous Testing để:

- Viết test case song song với quá trình lập trình
- Phát hiện lỗi sớm hơn và tiết kiệm chi phí sửa lỗi
- Đảm bảo chất lượng hệ thống ngay từ các vòng phát triển đầu tiên

4. Ghi nhật ký dự án đầy đủ hơn

Trong dự án hiện tại, việc ghi nhận tiến độ hoặc các khó khăn chưa được hệ thống hóa. Trong tương lai nhóm sẽ áp dụng thêm:

- Tạo log nhật ký cho từng lần họp, lần commit chức năng
- Ghi lại lỗi phát sinh và cách khắc phục
- Từ đó tạo tiền đề cho việc phân tích và rút kinh nghiệm sau dự án

5. Tổ chức phiên kiểm thử chéo giữa các thành viên

Để giảm thiên kiến trong kiểm thử, nhóm có thể triển khai kiểm thử chéo (cross-testing), điều đó có nghĩa là:

- Thành viên A phát triển → thành viên B kiểm thử độc lập
- Kết hợp với checklist hoặc bộ tiêu chí rõ ràng
- Phát hiện lỗi logic hoặc thiếu sót mà người viết code có thể bỏ qua

Việc cải tiến cách tiến hành dự án không chỉ giúp tăng chất lượng sản phẩm, mà còn nâng cao tính chuyên nghiệp, kỹ năng làm việc nhóm và khả năng phản ứng linh hoạt của từng thành viên. Đây là bước chuẩn bị cần thiết cho các dự án quy mô lớn hoặc trong môi trường làm việc thực tế sau này.