

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

----------------------------------------

HỌ TÊN SINH VIÊN: BÙI ĐỨC KHANH

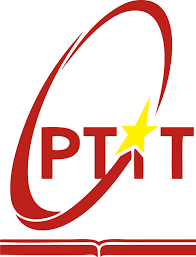
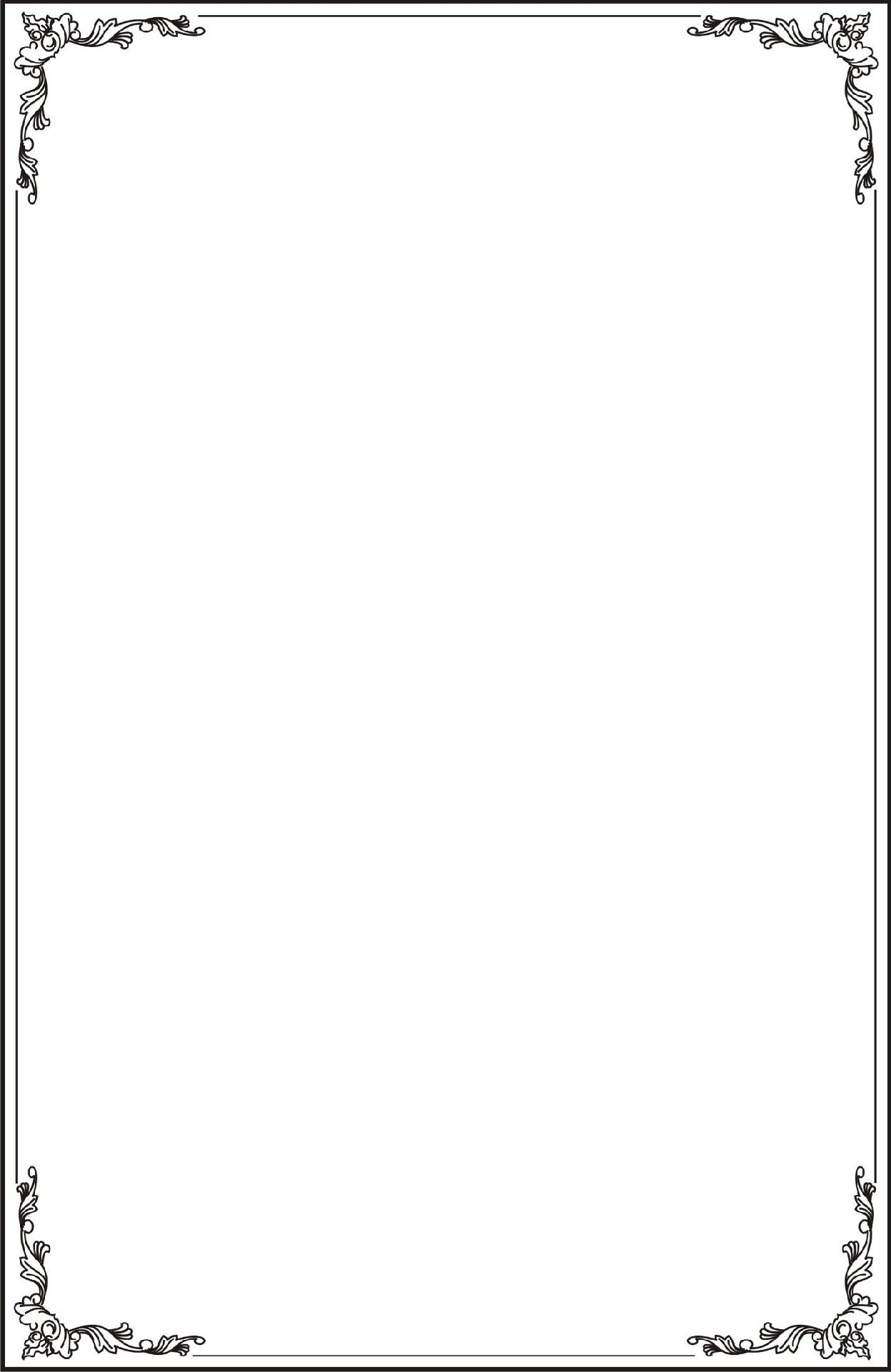
ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**Đề tài: XÂY DỰNG WEBSITE CHẤM BÀI TRỰC TUYẾN CHO VIỆC HỌC VÀ LUYỆN TẬP LẬP TRÌNH**

LỚP: D18CNPM03

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Đinh Xuân Trường |
| **Sinh viên thực hiện:** | Bùi Đức Khanh |
| **Mã sinh viên:** | B18DCCN303 |
| **Lớp:** | D18CNPM03 |
| **Niên khóa:** | 2018-2023 |
| **Hệ đào tạo:** | Đại học chính quy |

**Hà Nội 2022**



**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

--------------------------------------

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**Đề tài: XÂY DỰNG WEBSITE CHẤM BÀI TRỰC TUYẾN CHO VIỆC HỌC VÀ LUYỆN TẬP LẬP TRÌNH**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Đinh Xuân Trường |
| **Sinh viên:** | Bùi Đức Khanh |
| **Mã sinh viên:** | B18DCCN303 |
| **Lớp:** | D18CNPM03 |
| **Niên khóa:** | 2018-2023 |
| **Hệ đào tạo:** | Đại học chính quy |

**Hà Nội 2022**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

…………………………………………………………………………………………

**Điểm: (Bằng chữ: )**

Hà Nội, ngày tháng năm 20

**Giảng viên hướng dẫn**

**NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

…………………………………………………………………………………………

**Điểm: (Bằng chữ: )**

Hà Nội, ngày tháng năm 20

**Giảng viên phản biện**

**LỜI CẢM ƠN**

Đầu tiên, em xin cảm ơn các thầy, cô giáo của Học Viện Công nghệ Bưu chính Viễn thông, đặc biệt là các thầy cô trong khoa Công nghệ thông tin 1 đã dạy dỗ và chỉ bảo em trong suốt thời gian học tập tại trường.

Em xin chân thành cảm ơn ThS. Đinh Xuân Trường đã quan tâm, hướng dẫn và tận tình chỉ dạy em trong suốt quá trình thực hiện đề tài và viết báo cáo đồ án này.

Mặc dù đã cố gắng hoàn thành báo cáo trong phạm vi cho phép nhưng bài báo cáo chắc chắn sẽ không tránh được những sự thiếu sót. Em kính mong nhận được sự thông cảm của thầy cô và các bạn.

Em xin chân thành cảm ơn!

**Hà Nội, tháng 12 năm 2022**

**Sinh viên**

**Bùi Đức Khanh**

**MỤC LỤC**

[LỜI MỞ ĐẦU 13](#_Toc123246387)

[CHƯƠNG 1. GIỚI THIỆU CHUNG 13](#_Toc123246388)

[1.1 Khảo sát hiện trạng của ngành lập trình ở Việt Nam 13](#_Toc123246389)

[1.2 Xác định mục tiêu đề tài 13](#_Toc123246390)

[1.3 Mục tiêu nghiên cứu 14](#_Toc123246391)

[1.4 Phương pháp thực hiện 14](#_Toc123246392)

[1.5 Các công nghệ được sử dụng 14](#_Toc123246393)

[1.5.1 Tổng quan về Javascript 14](#_Toc123246394)

[1.5.2 Tổng quan về Node.js 15](#_Toc123246395)

[1.5.3 Tổng quan về ReactJs 16](#_Toc123246396)

[1.5.4 Tổng quan về REST và RESTful API 17](#_Toc123246397)

[1.5.5 Tổng quan về NoSQL và MongoDB 19](#_Toc123246398)

[1.6 Các công cụ 19](#_Toc123246399)

[1.6.1 Visual Studio Code 19](#_Toc123246400)

[1.6.2 Postman 20](#_Toc123246401)

[1.6.3 Chrome DevTools 20](#_Toc123246402)

[1.6.4 Mongo Atlas 21](#_Toc123246403)

[1.6.5 Cloudinary 21](#_Toc123246404)

[1.7 Tổng kết chương 21](#_Toc123246405)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 22](#_Toc123246406)

[2.1 Phân tích hệ thống 22](#_Toc123246407)

[2.1.1 Tên hệ thống 22](#_Toc123246408)

[2.1.2 Mục tiêu của hệ thống: 22](#_Toc123246409)

[2.1.3 Các tác nhân của hệ thống 22](#_Toc123246410)

[2.1.4 Yêu cầu của hệ thống: 22](#_Toc123246411)

[2.1.5 Phạm vi của hệ thống 22](#_Toc123246412)

[2.2 Thiết kế hệ thống 23](#_Toc123246413)

[2.2.1 Usecase tổng quan 23](#_Toc123246414)

[2.2.2 Biểu đồ lớp thực thể 24](#_Toc123246415)

[2.3 Phân rã các module 24](#_Toc123246416)

[2.3.1 Module đăng ký 24](#_Toc123246417)

[2.3.2 Module đăng nhập 27](#_Toc123246418)

[2.3.3 Module đăng xuất 29](#_Toc123246419)

[2.3.4 Module đổi mật khẩu 30](#_Toc123246420)

[2.3.5 Module xem danh sách bài toán 32](#_Toc123246421)

[2.3.6 Module lọc bài toán theo tiêu chí 34](#_Toc123246422)

[2.3.7 Module xem chi tiết bài toán 35](#_Toc123246423)

[2.3.8 Module xem xếp hạng thành viên 37](#_Toc123246424)

[2.3.9 Module chạy test 38](#_Toc123246425)

[2.3.10 Module submit lời giải 40](#_Toc123246426)

[2.3.11 Module xem lịch sử submit 42](#_Toc123246427)

[2.3.12 Bình luận về bài toán 43](#_Toc123246428)

[2.3.13 Module trả lời bình luận 45](#_Toc123246429)

[2.3.14 Module thích bình luận 46](#_Toc123246430)

[2.3.15 Module xem contest 48](#_Toc123246431)

[2.3.16 Đăng ký tham gia contest 49](#_Toc123246432)

[2.3.17 Tham gia contest 51](#_Toc123246433)

[2.3.18 Xem contest đã làm 53](#_Toc123246434)

[2.3.19 Module thêm danh mục 54](#_Toc123246435)

[2.3.20 Module sửa danh mục 56](#_Toc123246436)

[2.3.21 Module xóa danh mục 58](#_Toc123246437)

[2.3.22 Module thêm bài toán 60](#_Toc123246438)

[2.3.23 Module sửa bài toán 62](#_Toc123246439)

[2.3.24 Module thêm contest 63](#_Toc123246440)

[2.4 Thiết kế cơ sở dữ liệu 66](#_Toc123246441)

[2.4.1 Lược đồ cơ sở dữ liệu 66](#_Toc123246442)

[2.4.2 Mô tả cơ sở dữ liệu 66](#_Toc123246443)

[2.5 Tổng kết 71](#_Toc123246444)

[CHƯƠNG 3. CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG 72](#_Toc123246445)

[3.1 Yêu cầu hệ thống 72](#_Toc123246446)

[3.2 Một số công cụ, thư viện hỗ trợ 72](#_Toc123246447)

[3.3 Cài đặt 72](#_Toc123246448)

[3.3.1 Cài đặt Nodejs 72](#_Toc123246449)

[3.3.2 Đăng ký Mongo Atlas 73](#_Toc123246450)

[3.3.3 Đăng ký Cloudinary 74](#_Toc123246451)

[3.3.4 Cài đặt mã nguồn phía backend 75](#_Toc123246452)

[3.3.5 Cài đặt mã nguồn phía frontend 75](#_Toc123246453)

[3.4 Kết quả cài đặt 76](#_Toc123246454)

[3.4.1 Kết quả cài đặt database 76](#_Toc123246455)

[3.4.2 Kết quả cài đặt website 81](#_Toc123246456)

[3.5 Kết luận 88](#_Toc123246457)

[PHẦN TỔNG KẾT 89](#_Toc123246458)

[1. Đánh giá kết quả của đồ án 89](#_Toc123246459)

[2. Phương hướng phát triển 89](#_Toc123246460)

[TÀI LIỆU THAM KHẢO 90](#_Toc123246461)

[PHỤ LỤC 91](#_Toc123246462)

[A. Chi tiết sơ đồ mô phỏng 91](#_Toc123246463)

**DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Diễn giải** |
| 1 | API | Application Programming Interface |
| 2 | REST | REpresentational State Transfer |
| 3 | ECMA | European Computer Manufacturers Association |
| 4 | CPU | Central Processing Unit |
| 5 | HTML | HyperText Markup Language |
| 6 | JSX | Javascript Extensible Markup Language |
| 7 | MVC | Model-View-Controller |
| 8 | HATEOAS | Hypermedia As The Engine Of Application State |
| 9 | HTTP | HyperText Transfer Protocol |
| 10 | SQL | Structured Query Language |
| 11 | noSQL | Not-only SQL or No-relation SQL |
| 12 | IDE | Integrated Development Environment |

**DANH MỤC HÌNH VẼ**

[Hình 1.1 Khảo sát về công nghệ được sử dụng rộng rãi nhất năm 2022 [1] 15](#_Toc123246464)

[Hình 1.2 Tổng quan về nodejs [2] 16](#_Toc123246465)

[Hình 1.3 Danh sách thư viện được sử dụng rộng rãi nhất năm 2022 [1] 17](#_Toc123246466)

[Hình 1.4 Tổng quan về REST [3] 18](#_Toc123246467)

[Hình 1.5 Visual Studio Code 19](#_Toc123246468)

[Hình 1.6 Postman 20](#_Toc123246469)

[Hình 1.7 Chrome DevTools 20](#_Toc123246470)

[Hình 1.8 Mongo Atlas Site 21](#_Toc123246471)

[Hình 1.9 Cloudinary Site 21](#_Toc123246472)

[Hình 2.1 Usecase tổng quan của hệ thống 23](#_Toc123246473)

[Hình 2.2 Biểu đồ lớp thực thể của hệ thống 24](#_Toc123246474)

[Hình 2.3 Usecase chức năng đăng ký 24](#_Toc123246475)

[Hình 2.4 Biểu đồ tuần tự chức năng đăng ký (1) 26](#_Toc123246476)

[Hình 2.5 Usecase chi tiết module đăng nhập 27](#_Toc123246477)

[Hình 2.6 Biểu đồ tuần tự module đăng nhập 28](#_Toc123246478)

[Hình 2.7 Biểu đồ tuần tự module đăng xuất 30](#_Toc123246479)

[Hình 2.8 Usecase chi tiết module đổi mật khẩu 30](#_Toc123246480)

[Hình 2.9 Biểu đồ tuần tự module đổi mật khẩu 32](#_Toc123246481)

[Hình 2.10 Usecase chi tiết module xem danh sách bài toán 32](#_Toc123246482)

[Hình 2.11 Biểu đồ tuần tự module xem danh sách bài toán 33](#_Toc123246483)

[Hình 2.12 Usecase chi tiết module lọc bài toán 34](#_Toc123246484)

[Hình 2.13 Biểu đồ tuần tự module lọc bài toán 35](#_Toc123246485)

[Hình 2.14 Usecase tiết module xem chi tiết bài toán 35](#_Toc123246486)

[Hình 2.15 Biểu đồ tuần tự module xem chi tiết bài toán 36](#_Toc123246487)

[Hình 2.16 Usecase chi tiết chức năng xem xếp hạng thành viên 37](#_Toc123246488)

[Hình 2.17 Biểu đồ tuần tự module xem xếp hạng thành viên 38](#_Toc123246489)

[Hình 2.18 Usecase chi tiết module chạy test 38](#_Toc123246490)

[Hình 2.19 Biểu đồ tuần tự module chạy test 39](#_Toc123246491)

[Hình 2.20 Usecase chi tiết module submit lời giải 40](#_Toc123246492)

[Hình 2.21 Biểu đồ tuần tự module submit lời giải (2) 41](#_Toc123246493)

[Hình 2.22 Usecase chi tiết module xem lịch sử submit 42](#_Toc123246494)

[Hình 2.23 Biểu đồ tuần tự module xem lịch sử submit 43](#_Toc123246495)

[Hình 2.24 Usecase chi tiết module bình luận về bài toán 43](#_Toc123246496)

[Hình 2.25 Biểu đồ tuần tự module bình luận về bài toán 44](#_Toc123246497)

[Hình 2.26 Usecase chi tiết module trả lời bình luận 45](#_Toc123246498)

[Hình 2.27 Biểu đồ tuần tự module trả lời bình luận 46](#_Toc123246499)

[Hình 2.28 Usecase chi tiết module thích bình luận 46](#_Toc123246500)

[Hình 2.29 Biểu đồ tuần tự module thích bình luận 48](#_Toc123246501)

[Hình 2.30 Usecase chi tiết module xem contest 48](#_Toc123246502)

[Hình 2.31 Biểu đồ tuần tự module xem contest 49](#_Toc123246503)

[Hình 2.32 Usecase chi tiết module đăng ký tham gia contest 49](#_Toc123246504)

[Hình 2.33 Biểu đồ tuần tự module đăng ký tham gia contest 50](#_Toc123246505)

[Hình 2.34 Usecase chi tiết module tham gia contest 51](#_Toc123246506)

[Hình 2.35 Biểu đồ tuần tự module tham gia contest (3) 52](#_Toc123246507)

[Hình 2.36 Usecase chi tiết module xem contest đã làm 53](#_Toc123246508)

[Hình 2.37 Biểu đồ tuần tự module xem contest đã làm) 54](#_Toc123246509)

[Hình 2.38 Usecase chi tiết module thêm danh mục 54](#_Toc123246510)

[Hình 2.39 Biểu đồ tuần tự module thêm danh mục (6) 56](#_Toc123246511)

[Hình 2.40 Usecase chi tiết module sửa danh mục 56](#_Toc123246512)

[Hình 2.41 Biểu đồ tuần tự module sửa danh mục 58](#_Toc123246513)

[Hình 2.42 Usecase chi tiết module xóa danh mục 58](#_Toc123246514)

[Hình 2.43 Biểu đồ tuần tự module xóa danh mục 59](#_Toc123246515)

[Hình 2.44 Usecase chi tiết module thêm bài toán 60](#_Toc123246516)

[Hình 2.45 Biểu đồ tuần tự module thêm bài toán 61](#_Toc123246517)

[Hình 2.46 Usecase chi tiết module sửa bài toán 62](#_Toc123246518)

[Hình 2.47 Biểu đồ tuần tự module sửa bài toán 63](#_Toc123246519)

[Hình 2.48 Usecase chi tiết module thêm contest 63](#_Toc123246520)

[Hình 2.49 Biểu đồ tuần tự module thêm contest  (4) 65](#_Toc123246521)

[Hình 2.50 Lược đồ cơ sở dữ liệu 66](#_Toc123246522)

[Hình 3.1 Hướng dẫn tải Nodejs 72](#_Toc123246523)

[Hình 3.2 Màn hình cài đặt Nodejs 73](#_Toc123246524)

[Hình 3.3 Giao diện trang đăng ký Mongo Atlas 73](#_Toc123246525)

[Hình 3.4 Giao diện quản lý trang Mongo Atlas 74](#_Toc123246526)

[Hình 3.5 Giao diện trang hướng dẫn kết nối 74](#_Toc123246527)

[Hình 3.6 Giao diện đăng ký cloudinary 74](#_Toc123246528)

[Hình 3.7 Cấu trúc môi trường của dự án backend 75](#_Toc123246529)

[Hình 3.8 Kết quả cài đặt database bằng Mongo Atlas 76](#_Toc123246530)

[Hình 3.9 Kết quả cài đặt bảng users 76](#_Toc123246531)

[Hình 3.10 Kết quả cài đặt bảng uploads 77](#_Toc123246532)

[Hình 3.11 Kết quả cài đặt bảng testcases 77](#_Toc123246533)

[Hình 3.12 Kết quả cài đặt bảng submissions 78](#_Toc123246534)

[Hình 3.13 Kết quả cài đặt bảng Replies 78](#_Toc123246535)

[Hình 3.14 Kết quả cài đặt bảng problems 79](#_Toc123246536)

[Hình 3.15 Kết quả cài đặt bảng contests 79](#_Toc123246537)

[Hình 3.16 Kết quả cài đặt bảng contest\_histories 80](#_Toc123246538)

[Hình 3.17 Kết quả cài đặt bảng comments 80](#_Toc123246539)

[Hình 3.18 Kết quả cài đặt bảng categories 81](#_Toc123246540)

[Hình 3.19 Giao diện trang xem bài toán 81](#_Toc123246541)

[Hình 3.20 Giao diện xem chi tiết bài toán – chạy test 82](#_Toc123246542)

[Hình 3.21 Giao diện xem chi tiết bài toán 82](#_Toc123246543)

[Hình 3.22 Giao diện xem contest 83](#_Toc123246544)

[Hình 3.23 Giao diện làm contest (1) 83](#_Toc123246545)

[Hình 3.24 Giao diện làm contest (2) 84](#_Toc123246546)

[Hình 3.25 Giao diện xem lịch sử contest 84](#_Toc123246547)

[Hình 3.26 Giao diện xem lịch sử submit 85](#_Toc123246548)

[Hình 3.27 Giao diện xem xếp hạng 86](#_Toc123246549)

[Hình 3.28 Giao diện quản lý thông tin cá nhân 86](#_Toc123246550)

[Hình 3.29 Giao diện quản lý contest 86](#_Toc123246551)

[Hình 3.30 Giao diện quản lý danh mục 87](#_Toc123246552)

[Hình 3.31 Giao diện thêm danh mục 87](#_Toc123246553)

[Hình 3.32 Giao diện quản lý submissions 87](#_Toc123246554)

[Hình 3.33 Giao diện thêm contest 88](#_Toc123246555)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.1 Kịch bản module đăng ký 25](#_Toc123246556)

[Bảng 2.2 Kịch bản module đăng nhập 27](#_Toc123246557)

[Bảng 2.3 Kịch bản module đăng xuất 29](#_Toc123246558)

[Bảng 2.4 Kịch bản module đổi mật khẩu 31](#_Toc123246559)

[Bảng 2.5 Kịch bản module xem danh sách bài toán 33](#_Toc123246560)

[Bảng 2.6 Kịch bản module lọc bài toán theo tiêu chí 34](#_Toc123246561)

[Bảng 2.7 Kịch bản module xem chi tiết bài toán 35](#_Toc123246562)

[Bảng 2.8 Kịch bản module xem xếp hạng thành viên 37](#_Toc123246563)

[Bảng 2.9 Kịch bản module chạy test 38](#_Toc123246564)

[Bảng 2.10 Kịch bản module submit lời giải 40](#_Toc123246565)

[Bảng 2.11 Kịch bản module xem lịch sử submit 42](#_Toc123246566)

[Bảng 2.12 Kịch bản module bình luận về bài toán 43](#_Toc123246567)

[Bảng 2.13 Kịch bản module trả lời bình luận 45](#_Toc123246568)

[Bảng 2.14 Kịch bản module thích bình luận 47](#_Toc123246569)

[Bảng 2.15 Kịch bản module xem contest 48](#_Toc123246570)

[Bảng 2.16 Kịch bản module đăng ký tham gia contest 50](#_Toc123246571)

[Bảng 2.17 Kịch bản module tham gia contest 51](#_Toc123246572)

[Bảng 2.18 Kịch bản module xem contest đã làm 53](#_Toc123246573)

[Bảng 2.19 Kịch bản module thêm danh mục 55](#_Toc123246574)

[Bảng 2.20 Kịch bản module sửa danh mục 57](#_Toc123246575)

[Bảng 2.21 Kịch bản module xóa danh mục 59](#_Toc123246576)

[Bảng 2.22 Kịch bản module thêm bài toán 60](#_Toc123246577)

[Bảng 2.23 Kịch bản module sửa bài toán 62](#_Toc123246578)

[Bảng 2.24 Kịch bản module thêm contest 64](#_Toc123246579)

[Bảng 2.25 Cơ sở dữ liệu bảng users 66](#_Toc123246580)

[Bảng 2.26 Cơ sở dữ liệu bảng upload 67](#_Toc123246581)

[Bảng 2.27 Cơ sở dữ liệu bảng category 67](#_Toc123246582)

[Bảng 2.28 Cơ sở dữ liệu bảng comment 68](#_Toc123246583)

[Bảng 2.29 Cơ sở dữ liệu bảng reaction 68](#_Toc123246584)

[Bảng 2.30 Cơ sở dữ liệu bảng contest 68](#_Toc123246585)

[Bảng 2.31 Cơ sở dữ liệu bảng problem 69](#_Toc123246586)

[Bảng 2.32 Cơ sở dữ liệu bảng ContestHistory 70](#_Toc123246587)

[Bảng 2.33 Cơ sở dữ liệu bảng testcase 70](#_Toc123246588)

[Bảng 2.34 Cơ sở dữ liệu bảng submission 70](#_Toc123246589)

# LỜI MỞ ĐẦU

Hiện nay, với sự phát triển mạnh mẽ của ngành công nghệ thông tin và những ứng dụng của nó ở khắp các lĩnh vực trong đời sống thì nhu cầu về nhân lực trong ngành luôn ở mức cao. Các công việc liên quan đến vị trí lập trình cũng đang được rất nhiều công ty, tổ chức săn đón với mức thu nhập vô cùng hấp dẫn.

Hơn nữa, công nghệ thông tin đang là một trong số những ngành “hot” trong những năm gần đây. Số lượng các bạn sinh viên yêu thích và theo học ngành này cũng tăng vọt. Khi bước vào cánh cửa của ngành công nghệ thông tin, bên cạnh những kiến thức đại cương nền tảng, việc học lập trình là bắt buộc với những sinh viên theo học ngành này.

Không chỉ học sinh, sinh viên trong ngành công nghệ thông tin mới biết đến lập trình. Trên thực tế có rất nhiều sinh viên hay những người trái ngành yêu thích và mong muốn tiếp cận với công việc lập trình.

Do đó, nhu cầu học tập và luyện tập lập trình của học sinh, sinh viên và những người yêu thích lập trình cũng tăng cao. Để phục vụ cho nhu cầu đó, đồng thời để tăng khả năng tư duy giải quyết vấn đề cũng như chất lượng của những lập trình viên tương lai, cần có những hệ thống giúp học tập và luyện tập lập trình có khả năng đánh giá, kiểm tra code của người sử dụng. Đó chính là lý do lớn nhất em muốn xây dựng đồ án này với đề tài: **“Xây dựng Website chấm bài trực tuyến cho việc học và luyện tập lập trình”**. Thực hiện đề tài này, em mong muốn sử dụng các công cụ phổ biến trong thời điểm hiện tại như: Git, Visual studio code, Postman, … để xây dựng một hệ thống chấm code giúp học và luyện tập lập trình online.

Đồ án sẽ tập trung trình bày một số nội dung chính như sau:

* **Chương 1: Giới thiệu:**

Nội dung chương 1 sẽ mô tả khái quát về đề tài, giới thiệu chung về các công nghệ: Javascript, Node.js, ReactJs, REST, RESTful API, cơ sở dữ liệu không quan hệ và MongoDB.

* **Chương 2: Phân tích và thiết kế hệ thống:**

Nội dung của chương 2 sẽ trình bày về phân tích thiết kế các chức năng của hệ thống.

* **Chương 3: Cài đặt hệ thống:**

Nội dung của chương 3 sẽ tập trung vào việc hướng dẫn cài đặt các công cụ liên quan và chi tiết các bước cài đặt source code của hệ thống và những kết quả đạt được sau khi cài đặt thành công hệ thống.

* **Phần tổng kết:**

Tổng kết bài toán, tóm tắt những kết quả đã đạt được và còn chưa đạt được. Từ đó đề xuất mục tiêu hướng tới cũng như hướng nghiên cứu, phát triển tiếp theo.

# GIỚI THIỆU CHUNG

Trong chương này, đồ án sẽ trình bày về việc khảo sát về hiện trạng của ngành lập trình ở Việt Nam trong những năm gần đây và giới thiệu tổng quan về đề tài: **Xây dựng website chấm bài trực tuyến cho việc học và luyện tập lập trình** cũng như cơ sở lý thuyết của các công nghệ và giới thiệu về các công cụ được sử dụng để thực hiện đề tài.

## Khảo sát hiện trạng của ngành lập trình ở Việt Nam

Theo báo cáo thị trường của *TopDev*, năm 2021 Việt Nam cần 450.000 nhân lực ngành công nghệ thông tin. Hiện tại tổng số lập trình viên ở Việt Nam là vào khoảng 430.000 nhân lực, tương đương với việc đang thiếu hụt khoảng 20.000 nhân lực. Năm 2022, Việt Nam sẽ thiếu đến 150.000 nhân lực khi nhu cầu thị trường tăng lên đến 530.000 người.

Theo *Jobstreet.com,* lập trình viên đang nằm trong top 10 những ngành nghề có mức lương cao nhất tại Việt Nam.  Dải lương phổ biến của lập trình viên từ 8,4 triệu đồng đến 27,4 triệu đồng. Với những sinh viên mới tốt nghiệp chuyên ngành công nghệ thông tin, lương cơ bản nằm trong khoảng 8 - 10 triệu đồng. Những người 1 - 2 năm kinh nghiệm sẽ nhận lương mỗi tháng khoảng 10 - 15 triệu đồng. Con số này có thể lên hơn 20 triệu đồng nếu có tối thiểu 3 năm kinh nghiệm và kỹ năng chuyên sâu, đa dạng.

Theo báo *Thanh Niên*, số liệu xét tuyển năm nay cho thấy Công nghệ thông tin là một trong số những ngành thu hút nhiều thí sinh đăng ký ở nhiều trường. Kết quả này không quá bất ngờ bởi trên thực tế, ngành công nghệ thông tin đang có sức hút lớn với mức lương hấp dẫn và nhu cầu về nhân lực cao.

Tuy nhiên, theo *career.gpo.vn* chỉ có khoảng 16.500 sinh viên trong tổng số 55.000 sinh viên chuyên ngành công nghệ thông tin tốt nghiệp mỗi năm có khả năng đáp ứng được nhu cầu của doanh nghiệp (khoảng 30%).

Nguyên nhân thiếu hụt được cho là xuất phát từ sự chênh lệch giữa trình độ của lập trình viên và yêu cầu của doanh nghiệp về những kỹ năng cần thiết như khả năng lập trình và giải quyết các vấn đề liên quan đến các thuật toán trong thực tế.

## Xác định mục tiêu đề tài

* Xây dựng một môi trường luyện tập dành cho học sinh, sinh viên và những người yêu thích công việc lập trình.
* Cung cấp hệ thống hỗ trợ biên dịch, kiểm tra và đánh giá mã nguồn tự động, giúp cho những người yêu thích lập trình rèn luyện khả năng lập trình và giải quyết các vấn đề liên quan đến các thuật toán.
* Đối tượng hướng đến: các bạn học sinh, sinh viên và những người yêu thích lập trình.

## Mục tiêu nghiên cứu

Mục tiêu nghiên cứu của đồ án bao gồm 4 mục tiêu chính:

* Tìm hiểu về kiến trúc REST và tiêu chuẩn RESTful API trong việc thiết kế ứng dụng web.
* Tìm hiểu về ngôn ngữ lập trình Javascript thông qua việc xây dựng ứng dụng web sử dụng Nodejs kết hợp với Reactjs.
* Tìm hiểu về cơ sở dữ liệu không quan hệ Mongodb.
* Áp dụng những kiến thức đã tìm hiểu để xây dựng một hệ thống giúp người sử dụng học tập và luyện tập lập trình.

## Phương pháp thực hiện

* Khảo sát, tìm hiểu và xác định những thông tin cần thiết để hình thành ý tưởng, phạm vi, yêu cầu cần đạt của đề tài.
* Dựa vào phạm vi, yêu cầu để thực hiện phân tích thiết kế hệ thống.
* Tìm hiểu các công nghệ, kỹ thuật để đáp ứng được yêu cầu của đề tài.
* Dựa trên những kiến thức tìm hiểu được, tiến hành xây dựng và triển khai đề tài.
* Đánh giá kết quả đạt được để từ đó đưa ra phương hướng phát triển trong tương lai cho đề tài.

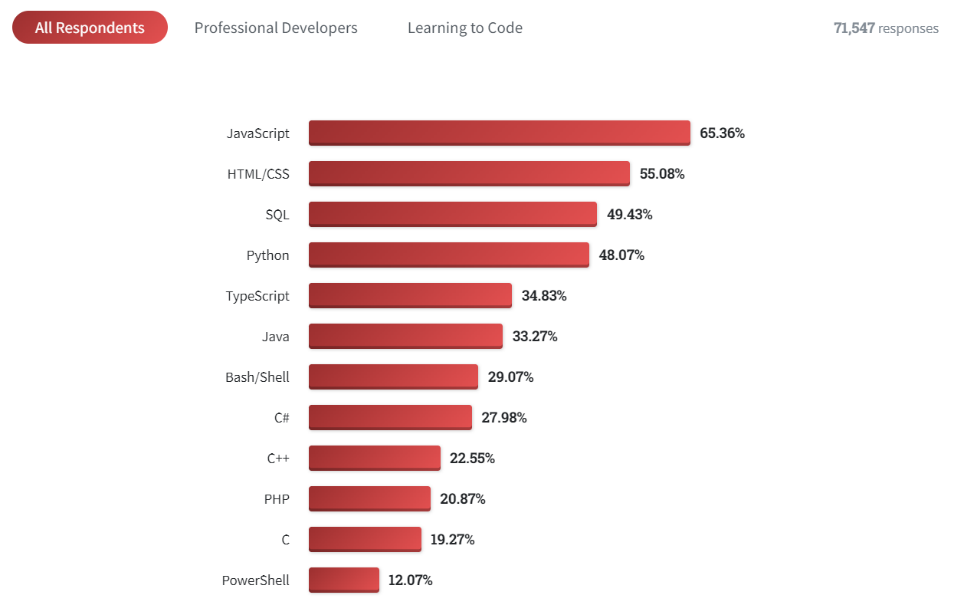
## Các công nghệ được sử dụng

### Tổng quan về Javascript

Javascript là một ngôn ngữ lập trình, ban đầu nó được tạo ra để giúp cho các trang web trở nên sống động hơn và được sử dụng chủ yếu phía trình duyệt. Các tập lệnh javascript được cung cấp và thực thi dưới dạng văn bản thuần túy, không cần chuẩn bị hay biên dịch để chạy. Hiện nay, javascript không chỉ được thực thi ở phía trình duyệt mà còn trên phía máy chủ hoặc trên thực tế là trên bất cứ thiết bị nào có chương trình đặc biệt được gọi là công cụ Javascript.

Quá trình phát triển của javascript:

* Năm 1995, javascript được phát minh bởi Brendan Eich.
* Năm 1996, Netscape 2 được phát hành với javascript 1.0
* Năm 1997, javascript trở thành tiêu chuẩn ECMA-262
* Trải qua nhiều giai đoạn, Javascript ngày càng phát triển mạnh mẽ và được hỗ trợ ngày càng nhiều trên các trình duyệt khác nhau.
* Cho đến năm 2018, Javascript phiên bản ECMAScript 6 được hỗ trợ trên tất cả các trình duyệt thông dụng: Chrome, FireFox, Safari.
* Ngày nay, javascript đang trở thành một trong những ngôn ngữ lập trình phổ biến nhất thế giới. Theo khảo sát của stackoverflow năm 2022, có đến 65.36% nhà phát triển sử dụng ngôn ngữ này cho việc lập trình, đứng đầu trong cuộc khảo sát về công nghệ được sử dụng rộng rãi nhất năm 2022.



Hình . Khảo sát về công nghệ được sử dụng rộng rãi nhất năm 2022 [1]

Ưu điểm của javascript:

* Javascript là ngôn ngữ lập trình có cấu trúc rất đơn giản, mã javascript dễ hiểu và dễ tiếp thu.
* Javascript là ngôn ngữ thông dịch, không cần biên dịch để thực thi nên tốc độ thực thi chương trình nhanh.
* Bất kể JavaScript được lưu trữ ở đâu, nó luôn chạy trong môi trường máy khách để giảm mức sử dụng băng thông và tăng tốc độ thực thi.
* Javascript hoạt động được trên nhiều trình duyệt, nền tảng.
* Thật dễ dàng trong việc giải quyết các vấn đề thắc mắc bởi Javascript được sử dụng bởi rất nhiều nhà phát triển.

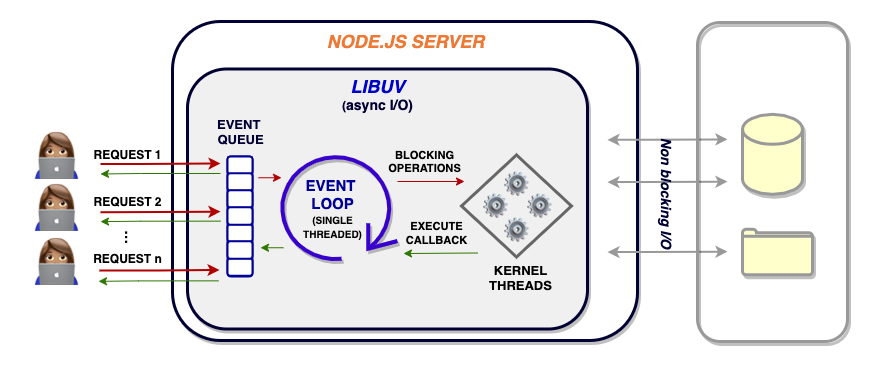
Nhược điểm của javascript:

* Khó khăn trong việc gỡ lỗi.
* Người dùng có thể xem được mã javascript phía máy khách, nó có thể bị người khác lạm dụng. Ngoài ra không khó để chèn mã vào trang web làm suy giảm tính bảo mật.
* Kiểu dữ liệu trong javascript ở dạng kiểu dữ liệu động nên khả năng gặp những lỗi tiềm ẩn là rất cao.

### Tổng quan về Node.js

Node.js là một Javascript runtime mã nguồn mở được xây dựng dựa trên V8 Javascript engine của Chrome, công cụ này chịu trách nhiệm cho việc biên dịch mã javascript sang mã máy tính để giúp cho máy tính có thể hiểu và chạy được mã javascript.

Node.js ra mắt vào năm 2009, bởi Ryan Dahl và nhận sự bảo trợ từ công ty Joyent (Hoa Kỳ).



Hình . Tổng quan về nodejs [2]

Node.js có những đặc điểm nổi bật như sau:

* **Bất đồng bộ và phát sinh sự kiện**: Tất các các APIs của thư viện Node.js đều bất đồng bộ. NodeJs không cần đợi một API trả về dữ liệu. Máy chủ chuyển sang một API khác sau khi gọi nó và có cơ chế riêng để gửi thông báo và nhận phản hồi về các hoạt động của Node.js và API đã gọi.
* **Tốc độ nhanh**: Phần core phía dưới được viết gần như toàn bộ bằng C++ kết hợp Chrome V8 Engine nên tốc độ xử lý công việc của Node.js cực nhanh, nhưng vẫn đảm bảo được tính chuẩn xác
* **Đơn giản – Hiệu năng cao**: Node.js sử dụng một mô hình luồng đơn luồng và các sự kiện lặp. Cơ chế sự kiện cho phép phía máy chủ trả về phản hồi theo non-blocking, đồng thời tăng hiệu quả sử dụng. Các luồng đơn cung cấp dịch vụ cho nhiều yêu cầu hơn hẳn máy chủ truyền thống.

Những ưu điểm của Nodejs:

* Phần core bên dưới của Node.js được viết chủ yếu bằng C++ nên hiệu năng và tốc độ xử lý cao. Do đó Node.js thích hợp cho việc tạo ra ứng dụng có tốc độ xử lý nhanh với thời gian thực.
* Thích hợp để tạo các sản phẩm có lượng truy cập lớn, mở rộng nhanh và đổi mới công nghệ… hoặc tạo dự án startup tiết kiệm thời gian nhất.
* Ngoài ra, Nodejs sử dụng javascript, là ngôn ngữ lập trình phổ biến nhất trong thời điểm những năm gần đây nên cộng đồng hỗ trợ lớn. Dễ dàng giải đáp những vướng mắc.

Những nhược điểm của Nodejs:

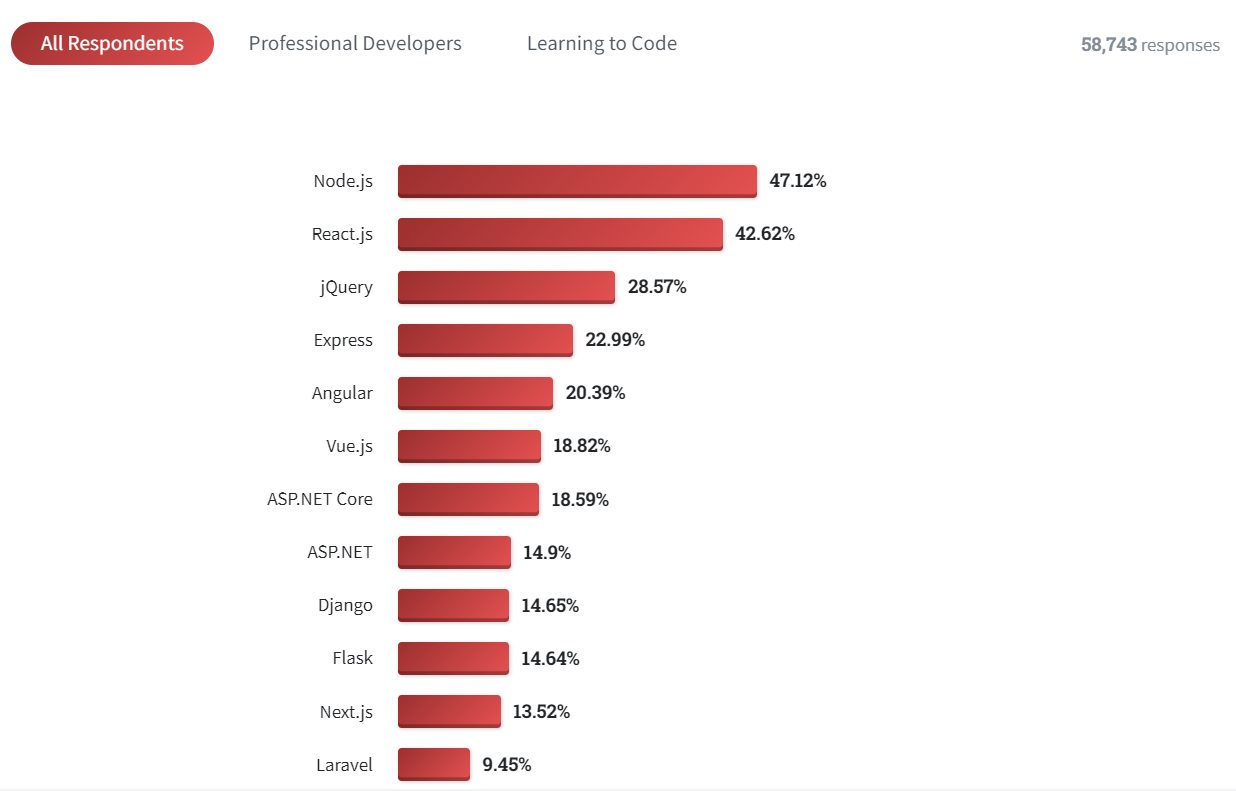
* Nền tảng nodejs không có vùng nhớ đệm, không cung cấp khả năng lưu trữ dữ liệu buffer.
* Không thích hợp để xây dựng ứng dụng gây tiêu tốn tài nguyên và CPU.

### Tổng quan về ReactJs

ReactJs được biết đến là một thư viện Javascript có tính hiệu quả, linh hoạt dùng để xây dựng giao diện người dùng. Nó cho phép xây dựng giao diện người dùng phức tạp từ những thành phần nhỏ được gọi là component.

Cha đẻ của ReactJs là Jordan Walke**,** một kỹ sư phần mềm tại Facebook.Ban đầu, nó được Facebook phát triển và duy trì, sau đó được sử dụng trong các sản phẩm của họ như WhatsApp hay Instagram**.**

Facebook đã phát triển ReactJS vào năm 2011 trong phần newsfeed của mình, nhưng nó được được phát hành ra công chúng vào tháng 5 năm 2013. Sau đó ReactJs đã phát triển nhanh chóng và trở thành một trong những thư viện được sử dụng rộng rãi nhất năm 2022.



Hình . Danh sách thư viện được sử dụng rộng rãi nhất năm 2022 [1]

Về ưu điểm, ReactJs có những đặc điểm sau:

* ReactJs sử dụng Virtual DOM, điều này giúp cải thiện rất nhiều về mặt hiệu suất. Ngoài ra nó còn giúp việc viết các đoạn mã nguồn Javascript dễ dàng hơn, ReactJs cung cấp cú pháp đặc biệt gọi là JSX có cấu trúc tương tự như các thẻ HTML nhưng điều đặc biệt là nó có thể trộn lẫn mã Javascript. Ngoài xây dựng ứng dụng web, chúng ta còn có thể sử dụng React để xây dựng ứng dụng dành cho các thiết bị di động.
* ReactJs là một thư viện của Javascript, với cộng đồng sử dụng đông đảo, việc giải đáp những vấn đề thắc mắc cũng dễ dàng hơn.

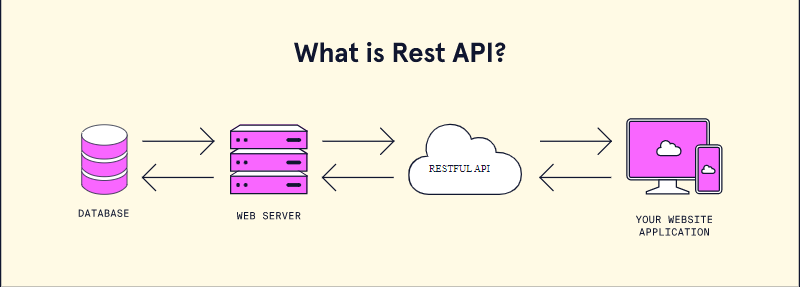
Về nhược điểm:

* ReactJs chỉ phục vụ cho tầng view, để kết hợp với các mô hình truyền thống như MVC cần phải cấu hình lại.
* ReactJs còn khá nặng so với các thư viện khác nhưng angular hay vuejs.

### Tổng quan về REST và RESTful API

REST là tập hợp các hướng dẫn và kiến trúc sử dụng cho việc truyền dữ liệu. REST áp dụng phổ biến cho các Web app, nhưng cũng hoàn toàn có thể sử dụng cho các phần mềm nói chung.

RESTful API là một tiêu chuẩn trong việc thiết kế ứng dụng web, nó tuân thủ theo kiến trúc và quy tắc của REST.



Hình . Tổng quan về REST [3]

Kiến trúc REST có những đặc điểm sau:

* Thứ nhất, REST có cấu trúc khách – chủ: Giao diện người dùng trên máy khách tách biệt và độc lập với việc lưu trữ dữ liệu trên máy chủ. Điều này cho phép máy khách được triển khai và sửa đổi bất kể điều gì đang xảy ra trên máy chủ. Tương tự như vậy, dữ liệu trên máy chủ có thể được sử dụng và sửa đổi bất kể máy khách đang truy cập bằng cách nào. Thiết kế như vậy cho phép cả hệ thống máy khách và máy chủ phát triển với tốc độ riêng của chúng, độc lập với nhau.
* Thứ hai, REST là không trạng thái: Không có dữ liệu phiên (Session Data) nào được lưu trữ trên máy chủ giữa các yêu cầu từ máy khách.
* Thứ ba, REST có khả năng cache: Trong mọi trao đổi, dữ liệu phải được đánh dấu là dữ liệu có thể lưu vào bộ nhớ cache hoặc không thể lưu trong bộ nhớ cache. Dữ liệu có thể lưu vào bộ nhớ cache có thể được máy khách lưu trữ và sử dụng lại. Không có dữ liệu nào được lưu trữ trên máy chủ vì điều này sẽ vi phạm ràng buộc nguyên tắc không trạng thái (Stateless).
* Thứ tư, REST đảm bảo giao diện thống nhất: Có bốn ràng buộc phụ đảm bảo giao diện thống nhất: Định danh tài nguyên, thao tác tài nguyên thông qua biểu tả, thông điệp tự mô tả và siêu phương tiện đóng vai trò công cụ của ứng dụng HATEOAS.
* Thứ năm là phân lớp hệ thống: Trong hệ thống các lớp phân cấp, không thành phần nào có thể tương tác với hoặc xem Bất kì dữ liệu hoặc giao diện nào ngoại trừ trên lớp hiện tại của chính nó. Do đó, máy khách không cần biết cách, thậm chí không cần thiết phải kết nối với Bất kì máy chủ, proxy, tường lửa, bộ định tuyến hoặc điểm cuối bổ sung nào. Thay vào đó, Bất kì trung gian nào sẽ tiếp tục kết nối với các máy chủ tiếp theo theo các ràng buộc REST và kết quả là phản hồi cho Bất kì yêu cầu nào được trả lại qua trung gian cho máy khách cũng sẽ tuân thủ REST.
* Thứ sáu là mã theo yêu cầu: các máy khách trong kiến trúc REST sẽ có thể tải xuống và thực thi các tập lệnh. Điều này cho phép mở rộng các chức năng và hệ thống phức tạp hơn.

Ưu điểm của REST:

* Một trong số những đặc điểm của REST là không trạng thái. Máy chủ không lưu bất cứ trạng thái nào của yêu cầu. Điều này làm tăng hiệu suất làm việc
* Các hệ thống REST có khả năng mở rộng dễ dàng hơn do sự tách biệt giữa các thành phần và các quy ước giao tiếp được quy định sẵn.
* Các yêu cầu trong REST rất rõ ràng, dễ hiểu

Nhược điểm:

* REST là không trạng thái, nên mọi tác vụ quản lý trạng thái phải được máy khách thực hiện, ngoài ra REST chỉ hoạt động dựa trên các giao thức HTTP

### Tổng quan về NoSQL và MongoDB

Cơ sở dữ liệu noSQL là Cơ sở dữ liệu được xây dựng dành riêng cho mô hình dữ liệu và có sơ đồ linh hoạt để xây dựng các ứng dụng hiện đại. Cơ sở dữ liệu noSQL được công nhận rộng rãi vì khả năng dễ phát triển, chức năng cũng như hiệu năng ở quy mô lớn.

Trong noSQL, dữ liệu được lưu trữ theo dạng cặp giá trị “key – value”. Sử dụng số lượng lớn các node để lưu trữ thông tin. noSQL là phi quan hệ, không có ràng buộc nào cho việc nhất quán dữ liệu.

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là cơ sở dữ liệu thuộc noSQL và được hàng triệu người sử dụng.

Ưu điểm của việc sử dụng noSQL:

* NoSQL giải quyết được các vấn đề dữ liệu lớn về các hệ thống thông tin hoặc là phân tán dữ liệu. Việc mở rộng phạm vi của noSQL là rất dễ dàng.

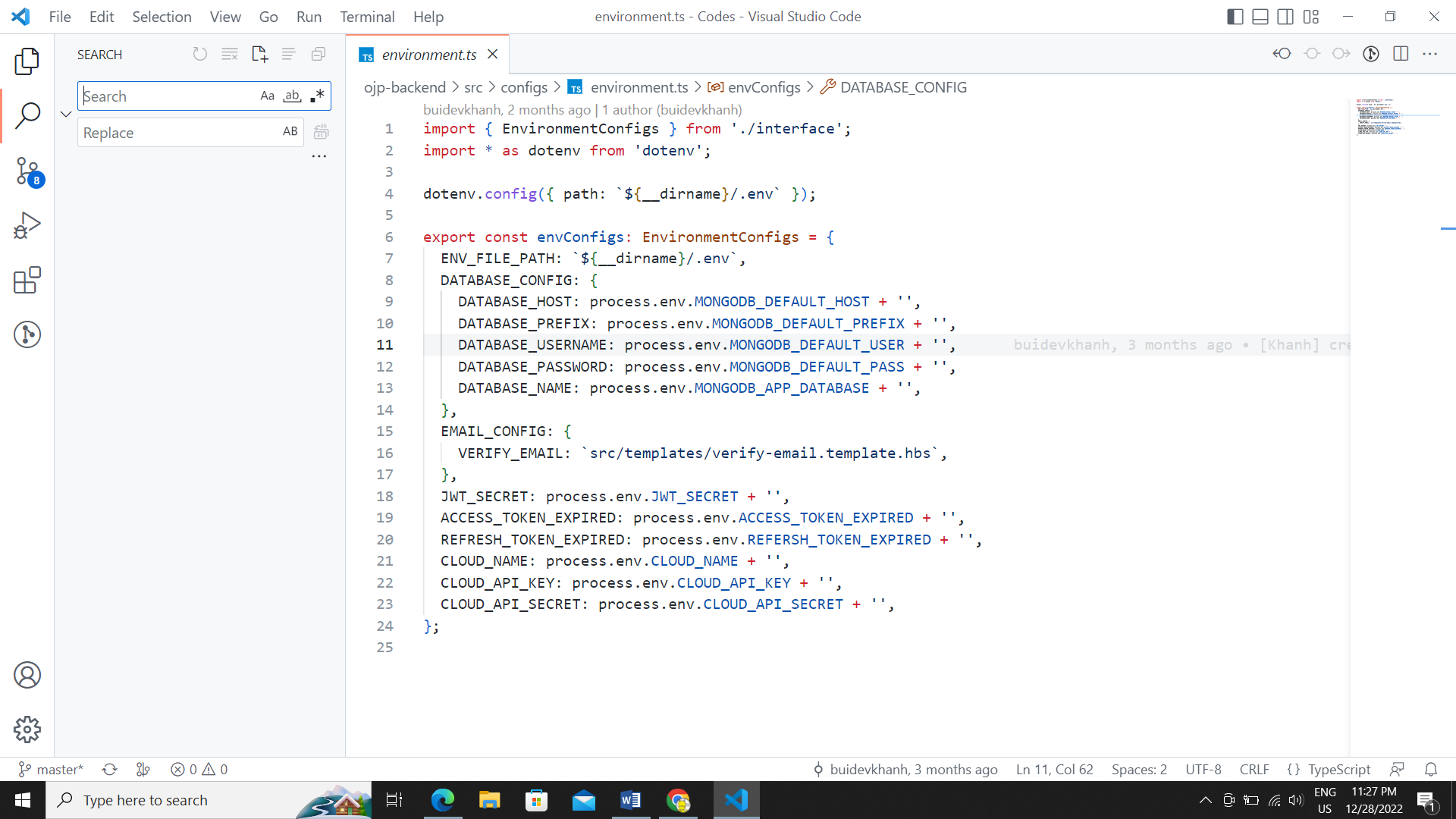
Nhược điểm của noSQL:

* NoSQL đánh đổi sự nhất quán để đạt được hiệu năng và tốc độ.
* Việc quản lý dữ liệu trong noSQL được đánh giá là phức tạp hơn so với SQL.

## Các công cụ

### Visual Studio Code

Visual Studio Code là một trình soạn thảo, biên tập mã nguồn được Microsoft phát triển cho các lập trình viên và có mặt trên hầu hết các hệ điều hành phổ biến như: Windows, Linux và macOS. Có thể nói rằng, Visual Studio Code là một sự kết hợp giữa IDE và Code Editor.

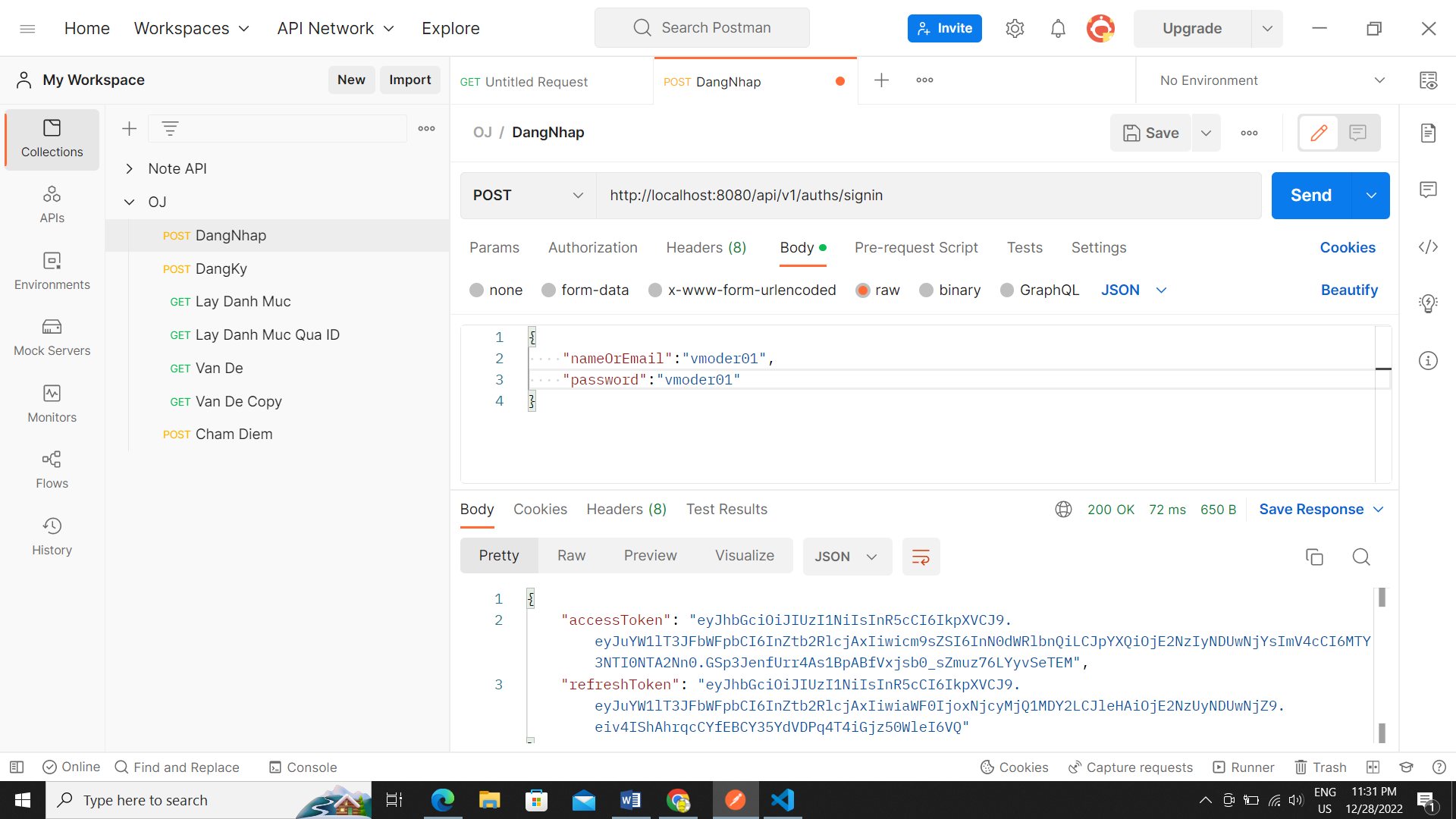


Hình . Visual Studio Code

### Postman

Postman hiện là một trong những công cụ phổ biến nhất được sử dụng trong thử nghiệm các API. Với Postman, ta có thể gọi Rest API mà không cần viết bất cứ mã nguồn nào.

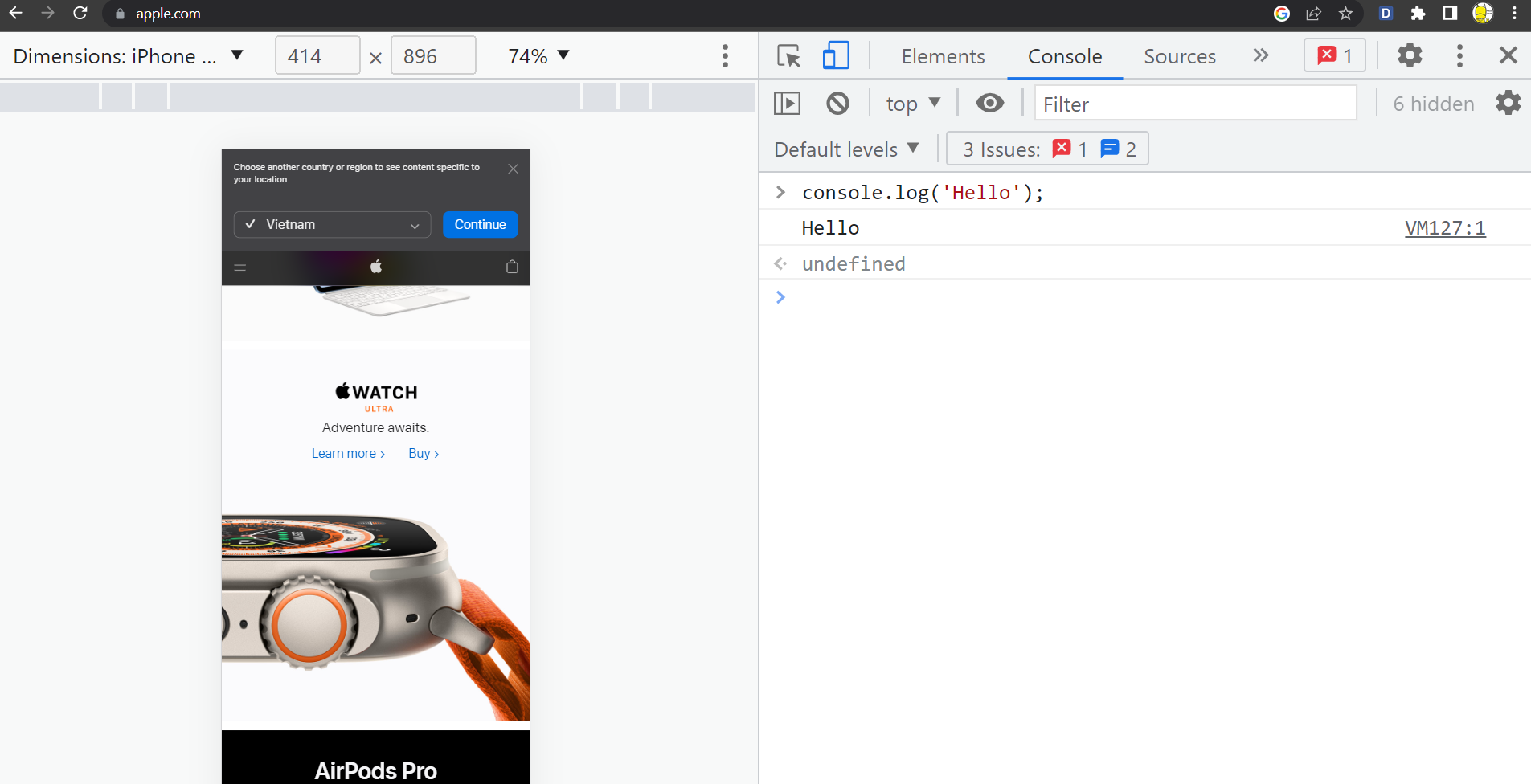
Postman hỗ trợ tất cả các phương thức HTTP như: GET, POST, PUT, PATCH, DELETE, ... Bên cạnh đó, nó còn cho phép lưu lại lịch sử các yêu cầu của người dùng, điều này rất tiện lợi cho việc sử dụng lại khi cần.



Hình . Postman

### Chrome DevTools

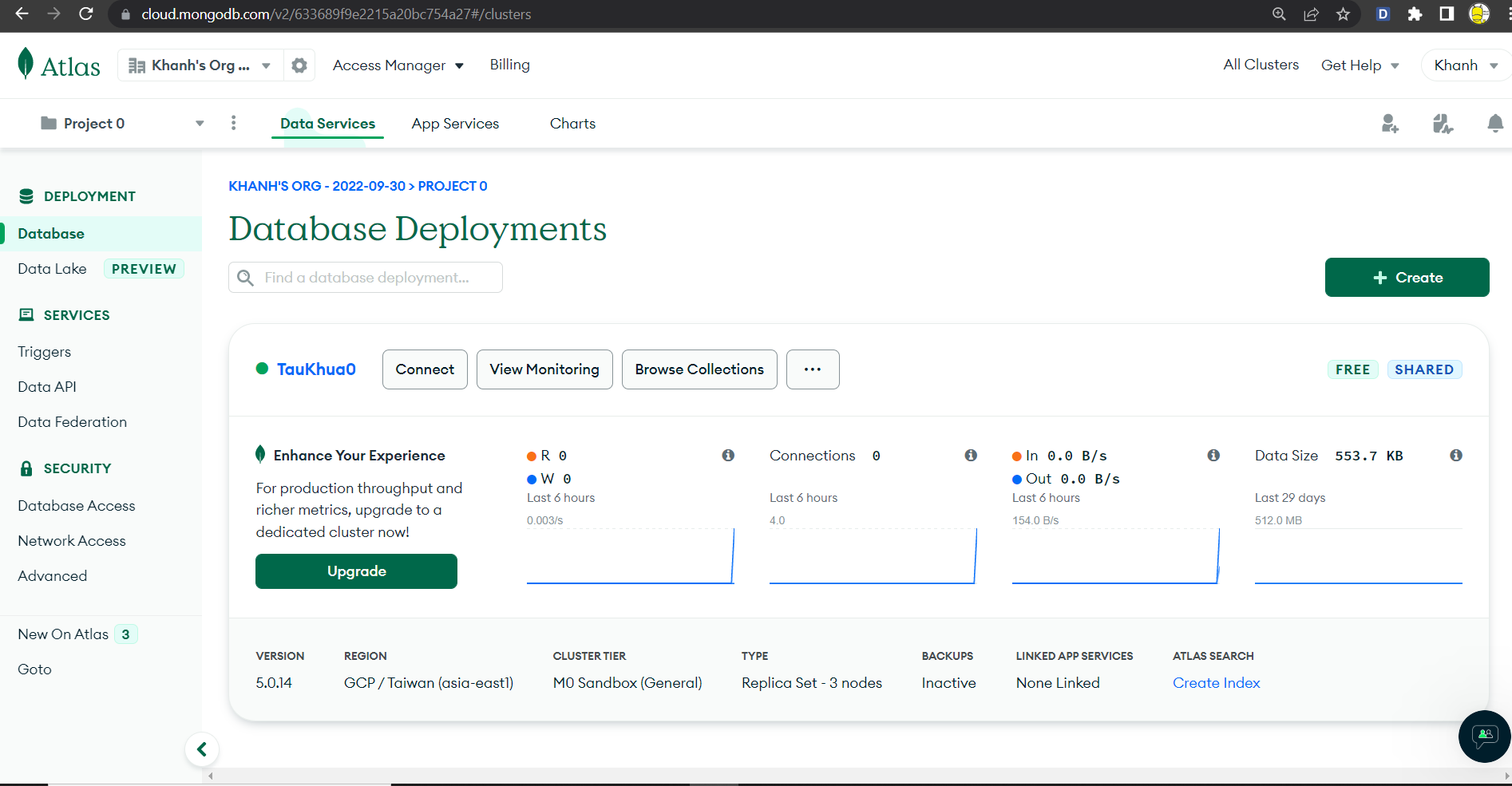
Chrome DevTools là một bộ công cụ phát triển web được tích hợp trực tiếp vào trình duyệt Google Chrome. DevTools có thể giúp chúng ta chuẩn đoán vấn đề một cách nhanh chóng, điều này giúp xây dựng trang web tốt hơn, nhanh hơn. Với DevTools, bạn có thể xem và thay đổi Bất kì trang web nào, ngay cả trang chủ Google.



Hình . Chrome DevTools

### Mongo Atlas

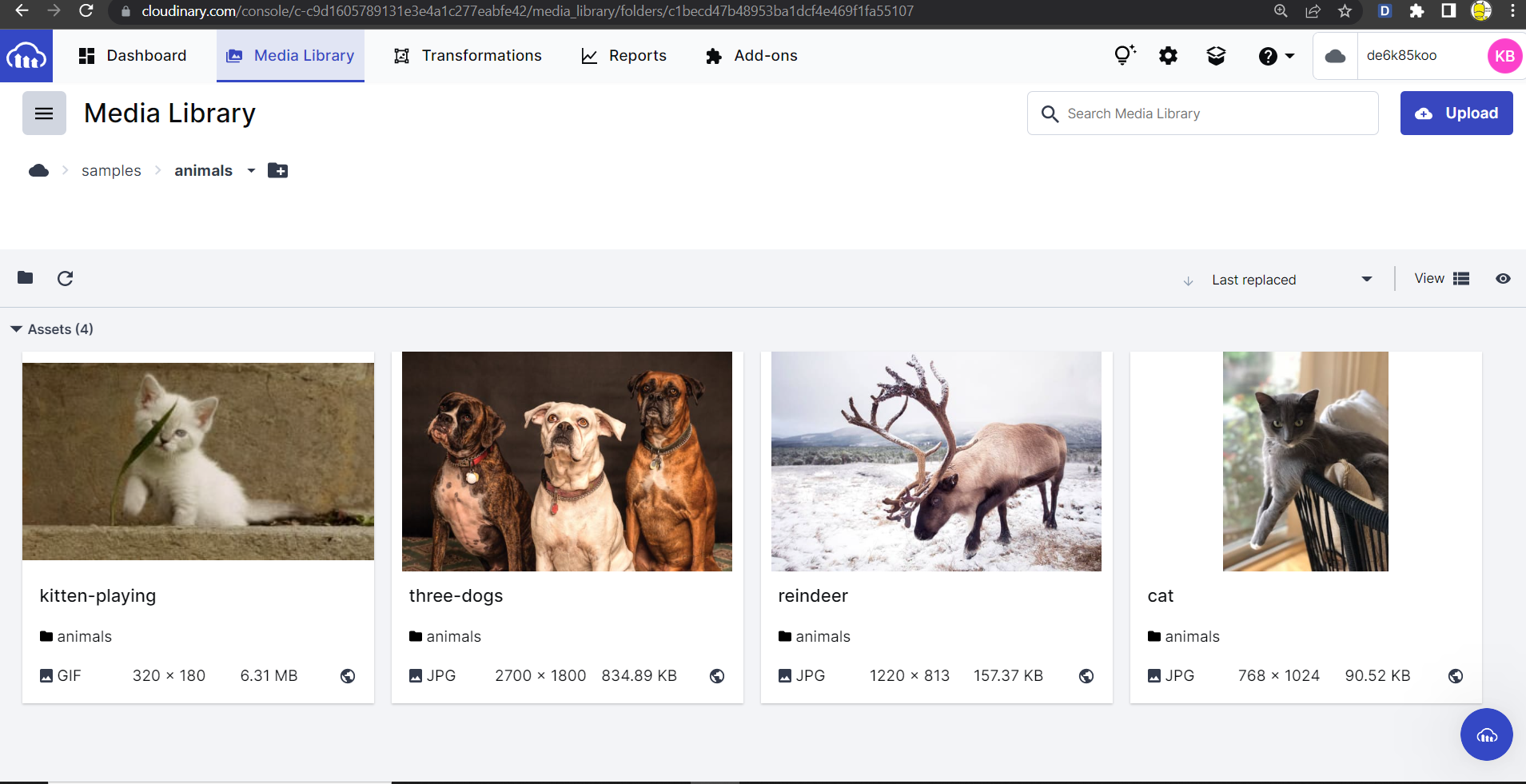
MongoDB Atlas là cơ sở dữ liệu đám mây của MongoDB được ra mắt vào năm 2016 chạy trên AWS, Microsoft Azure và Google Cloud Platform.



Hình . Mongo Atlas Site

### Cloudinary

Cloudinary là một cloud-based service, nó cung cấp một giải pháp quản lý hình ảnh bao gồm upload, lưu trữ, thao tác, và tối ưu hóa. Trong phạm vi của hệ thống, cloudinary được sử dụng để lưu trữ hình ảnh.



Hình . Cloudinary Site

## Tổng kết chương

Như vậy, ở chương 1 đồ án đã giới thiệu tổng quan về đề tài bao gồm những nội dung sau:

* Khảo sát về hiện trạng của ngành lập trình ở Việt Nam
* Xác định mục tiêu và mục tiêu nghiên cứu của đề tài.
* Xác định phương pháp để thực hiện đề tài.
* Giới thiệu về các công nghệ và công cụ được sử dụng để thực hiện đề tài.

Dựa vào bài toán đã được đặt ra ở chương này, chương tiếp theo đồ án sẽ thực hiện xây dựng phân tích và thiết kế cho hệ thống.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Trong chương này, đồ án sẽ tập trung vào việc trình bày phân tích thiết kế cho toàn bộ hệ thống và phân tích thiết kế chi tiết cho từng module có mặt trong trong hệ thống.

## Phân tích hệ thống

### Tên hệ thống

Website chấm bài trực tuyến cho việc học và luyện tập lập trình.

### Mục tiêu của hệ thống:

Hệ thống được xây dựng nhằm phục vụ cho việc học tập và luyện tập lập trình của các bạn học sinh, sinh viên và những người yêu thích lập trình. Hệ thống giúp kiểm tra, đánh giá chương trình của người sử dụng, thông qua đó giúp người sử dụng rèn luyện tư duy lập trình và khả năng giải quyết vấn đề một cách toàn diện nhất.

### Các tác nhân của hệ thống

Hệ thống bao gồm những tác nhân chính sau đây:

* Người dùng: Các bạn học sinh, sinh viên ngành công nghệ thông tin và những người yêu thích công việc lập trình.
* Quản trị viên của hệ thống: Người quản lý, chịu trách nhiệm bảo trì, cấu hình và vận hành hệ thống.

### Yêu cầu của hệ thống:

Hệ thống có khả năng kiểm tra, đánh giá bài làm của thành viên trong hệ thống và đáp ứng đồng thời yêu cầu của nhiều người sử dụng.

### Phạm vi của hệ thống

Người dùng ẩn danh có thể:

* Đăng ký, đăng nhập
* Xem danh sách các bài toán
* Lọc bài toán theo tiêu chí
* Xem xếp hạng thành viên
* Xem chi tiết các bài toán
* Chạy test: thực thi chương trình với đầu vào mong muốn

Người dùng đã đăng nhập có thể:

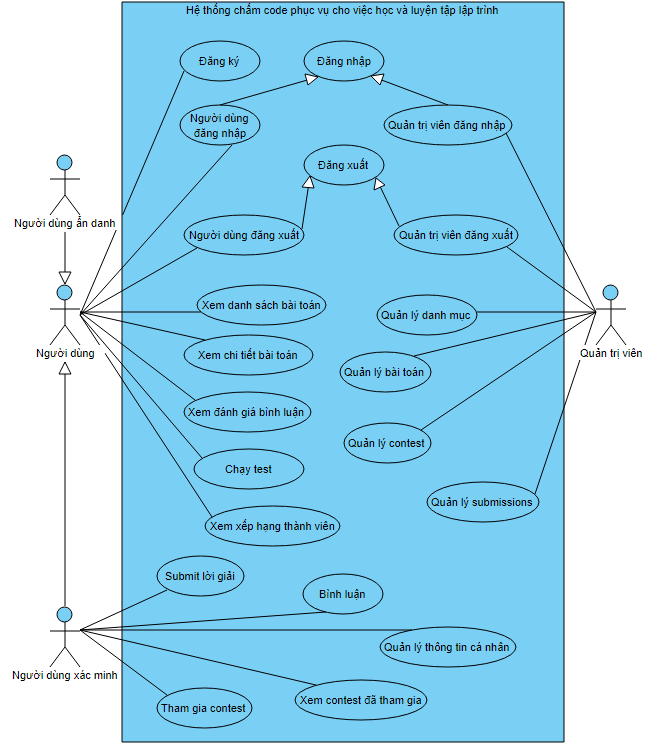
* Đăng xuất
* Thực hiện các chức năng giống như người dùng ẩn danh
* Quản lý thông tin cá nhân
* Đổi mật khẩu
* Submit lời giải
* Xem lịch sử các submissions của hệ thống
* Xem danh sách các contest
* Bình luận về bài toán
* Trả lời bình luận
* Thích hoặc hủy thích bình luận
* Đăng ký tham gia contest
* Tham gia contest
* Xem lại các contest đã tham gia

Quản trị viên của hệ thống có thể:

* Đăng nhập, đăng xuất hệ thống
* Quản lý các danh mục: Xem, thêm, sửa, xóa danh mục
* Quản lý các bài toán: Xem, thêm, sửa, xóa bài toán
* Quản lý các contest: Xem, thêm contest
* Quản lý các submission của hệ thống: Xem submission.

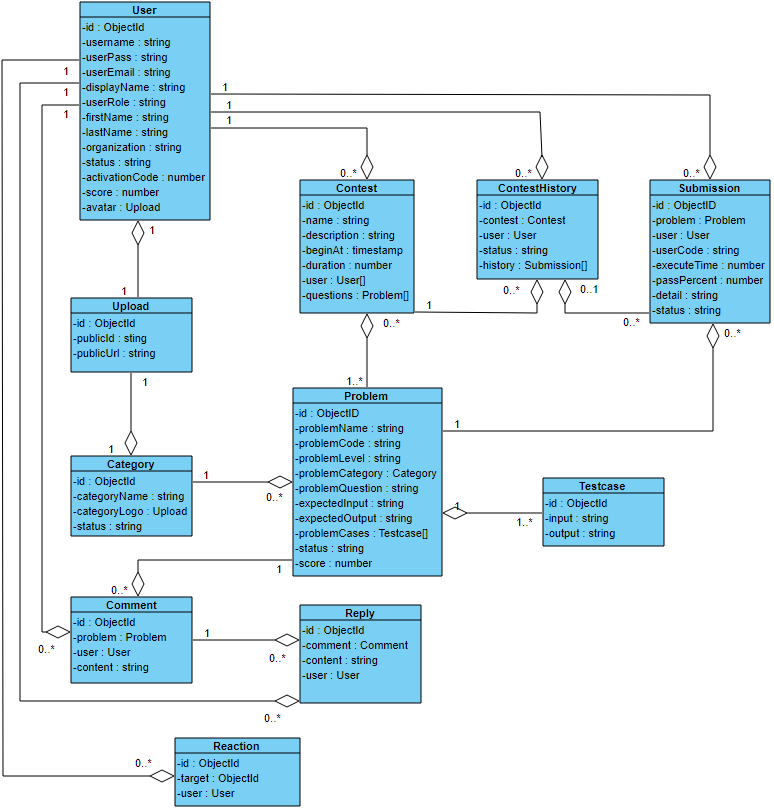
## Thiết kế hệ thống

### Usecase tổng quan



Hình . Usecase tổng quan của hệ thống

### Biểu đồ lớp thực thể

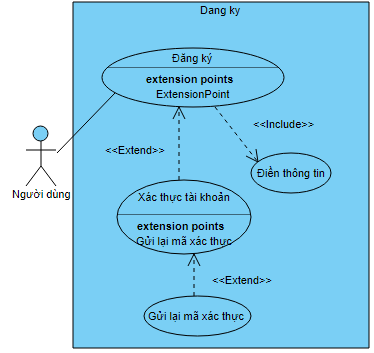


Hình . Biểu đồ lớp thực thể của hệ thống

## Phân rã các module

### Module đăng ký

#### Usecase chi tiết



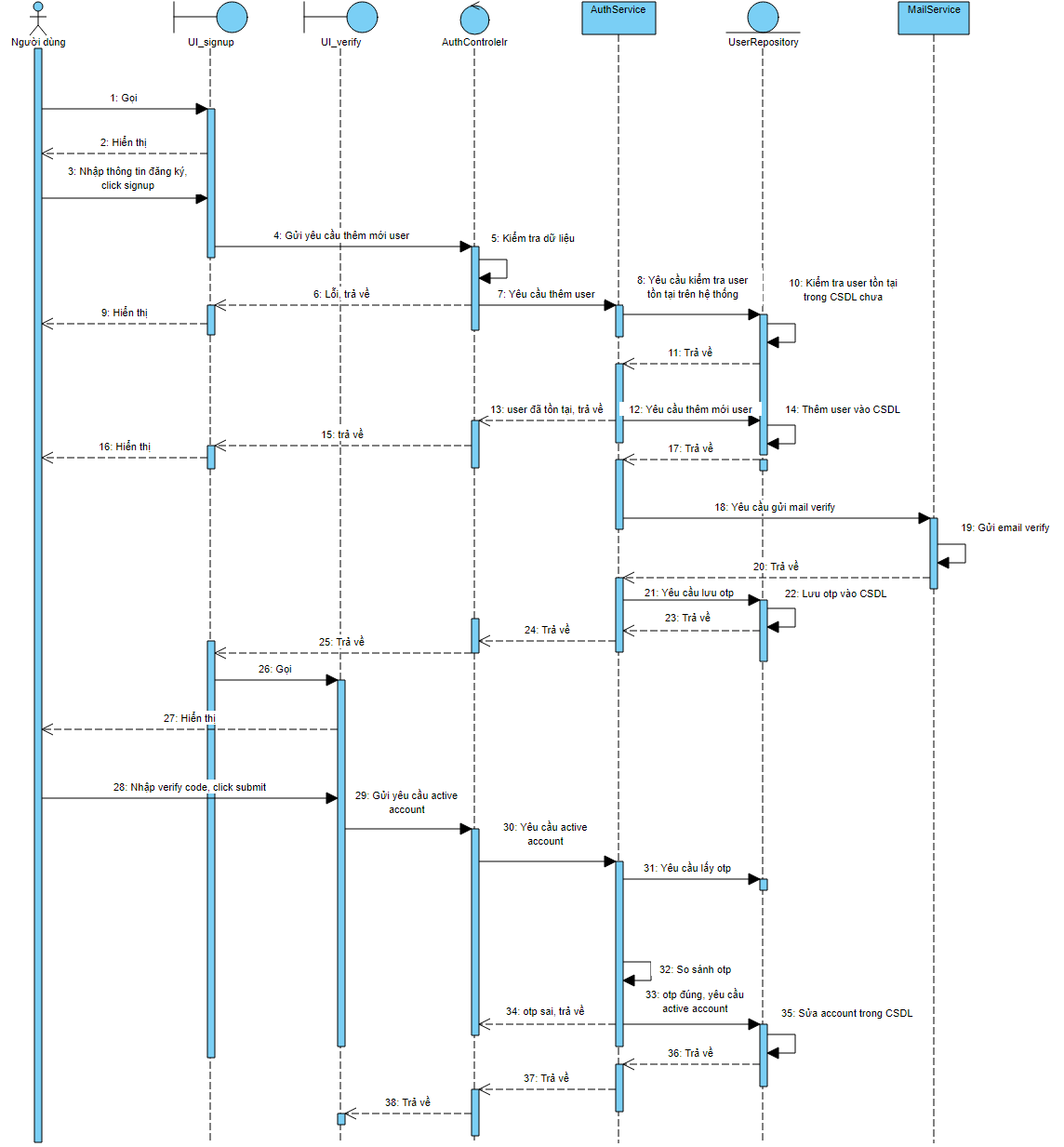
Hình . Usecase chức năng đăng ký

#### Kịch bản

Bảng . Kịch bản module đăng ký

|  |  |
| --- | --- |
| Tên usecase | Đăng ký |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng chưa có tài khoản trên hệ thống |
| Hậu điều kiện | Người dùng đăng ký tài khoản thành công |
| Kịch bản | 1. Người dùng truy cập vào trang đăng ký của hệ thống  2. Giao diện trang đăng ký hiện lên với các ô nhập: tên đăng nhập, mật khẩu, email, xác nhận mật khẩu, tên hiển thị và nút sign up.  3. Người dùng điền thông tin đăng ký rồi click nút sign up  4. Hệ thống tiếp nhận thông tin đăng ký của người dùng, gửi email xác thực về tài khoản email người dùng đã nhập và chuyển sang trang xác nhận email.  5. Trang xác nhận email hiện lên với ô nhập mã otp, nút verify và link resend verify code.  6. Người dùng nhập mã otp và ấn nút verify  7. Hệ thống kích hoạt tài khoản của người dùng sau đó chuyển hướng sang trang đăng nhập. |
| Ngoại lệ | **3.1** Tên đăng nhập đã tồn tại  3.1.1 Hệ thống thông báo tên đăng nhập đã tồn tại trên hệ thống  3.1.2 Người dùng nhập tên đăng nhập mới rồi click sign up  **3.2** Mật khẩu không trùng khớp  3.2.1 Hệ thống thông báo mật khẩu không trùng khớp  3.2.2 Người dùng nhập lại thông tin mật khẩu rồi click sign up  **3.3** Email đã tồn tại trên hệ thống  3.3.1 Người dùng nhập lại email rồi click sign up  **3.4** Tên hiển thị đã tồn tại trên hệ thống  3.4.1 Hệ thống thông báo tên hiển thị đã tồn tại.  3.4.2 Người dùng nhập tên hiển thị mới rổi click sign up.  **3.5** Tên đăng nhập không đúng định dạng  3.5.1 Hệ thống hiển thị thông báo thể hiện rằng: tên đăng nhập không được để trống, bao gồm chữ số và chữ cái, tối thiểu 6 kí tự.  3.5.2 Người dùng nhập tên đăng nhập đúng định dạng rồi ấn sign up  **3.6** Mật khẩu không đúng định dạng  3.6.1 Hệ thống hiển thị thông báo thể hiện rằng: mật khẩu không được để trống, 8 đến 20 kí tự chứa chữ, số và kí tự đặc biệt.  3.6.2 Người dùng nhập lại mật khẩu rồi ấn sign up  **3.7** Email không đúng định dạng  3.7.1 Hệ thống hiển thị thông báo email không hợp lệ  3.7.2 Người dùng nhập email đúng định dạng và ấn sign up  **5.1** Người dùng không verify  5.1.1 Hệ thống tự động chuyển sang trang verify khi người dùng đăng nhập vào tài khoản vào lần sau. |

#### Biểu đồ tuần tự

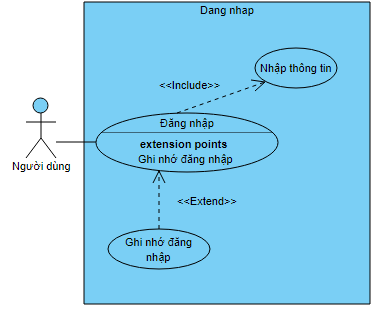
­­

Hình . Biểu đồ tuần tự chức năng đăng ký (1)

1. *Chi tiết xem tại phụ lục A trang 91*

### Module đăng nhập

#### Usecase chi tiết



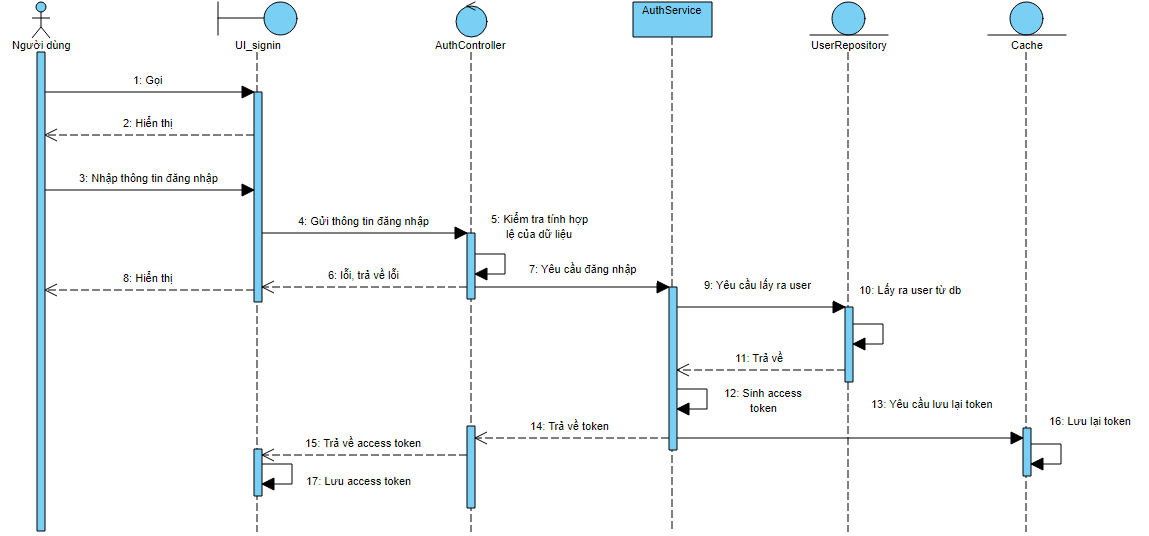
Hình . Usecase chi tiết module đăng nhập

#### Kịch bản

Bảng . Kịch bản module đăng nhập

|  |  |
| --- | --- |
| Tên usecase | Đăng nhập |
| Actor | Người dùng, quản trị viên |
| Tiền điều kiện | Người dùng đã có tài khoản trên hệ thống |
| Hậu điều kiện | Người dùng, quản trị viên đăng nhập thành công vào hệ thống |
| Kịch bản | 1. Người dùng truy cập vào trang đăng nhập của hệ thống  2. Trang đăng nhập hiện ra với ô nhập tên đăng nhập, mật khẩu, checkbox ghi nhớ đăng nhập và nút sign in  3. Người dùng nhập thông tin đăng nhập, tích vào ô ghi nhớ đăng nhập nếu muốn tự động đăng nhập trên trình duyệt vào những lần tiếp theo, sau đó click sign in  4. Hệ thống kiểm tra thông tin đăng nhập của người dùng và chuyển sang trang chủ tương ứng |
| Ngoại lệ | **3.1** Tên đăng nhập không đúng định dạng  3.1.1 Hệ thống hiển thị thông báo tên đăng nhập không đúng định dạng thể hiện rằng: tên đăng nhập không được để trống, bao gồm chữ số và chữ cái, tối thiểu 6 kí tự.  3.1.2 Người dùng nhập lại thông tin đăng nhập rồi click signin  **3.2** Mật khẩu không đúng định dạng  3.2.1 Hệ thống hiển thị thông báo thể hiện rằng: mật khẩu không được để trống, 8 đến 20 kí tự chứa chữ, số và kí tự đặc biệt.  3.2.2 Người dùng nhập lại mật khẩu rồi ấn sign in  **3.3** Tài khoản không chính xác  3.3.1 Hệ thống hiển thị thông báo thể hiện rằng tài khoản không chính xác.  3.3.2 Người dùng nhập thông tin tài khoản chính xác rồi click sign in  **3.4** Tài khoản chưa được kích hoạt  3.4.1 Hệ thống hiển thị thông báo thể hiện tài khoản chưa được kích hoạt, gửi email kích hoạt rồi chuyển hướng sang trang xác thực tài khoản  3.4.2 Người dùng xác thực tài khoản sau đó tiến hành đăng nhập lại. |

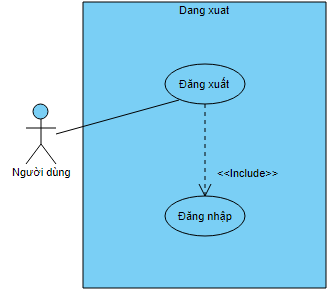
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module đăng nhập

### Module đăng xuất

#### Usecase chi tiết



#### Kịch bản

Bảng . Kịch bản module đăng xuất

|  |  |
| --- | --- |
| Tên usecase | Đăng xuất |
| Actor | Người dùng, quản trị viên |
| Tiền điều kiện | Người dùng, quản trị viên đã đăng nhập và hệ thống |
| Hậu điều kiện | Người dùng, quản trị viên đăng xuất thành công |
| Kịch bản | 1. Người dùng đăng nhập vào hệ thống  2. Tại giao diện trang bất kì có thanh điều hướng, người dùng click vào biểu tượng hình đại diện trên thanh điều hướng.  3. Danh sách chức năng sổ ra với các lựa chọn: Quản lý trang cá nhân, đổi mật khẩu, đăng xuất.  4. Người dùng click chọn đăng xuất  5. Hệ thống đăng xuất người dùng. |
| Ngoại lệ | Không có ngoại lệ |

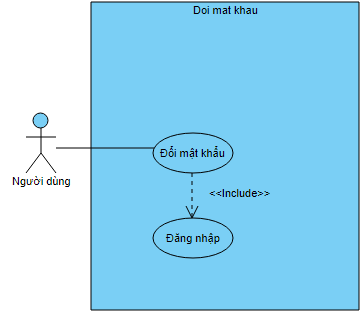
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module đăng xuất

### Module đổi mật khẩu

#### Usecase chi tiết



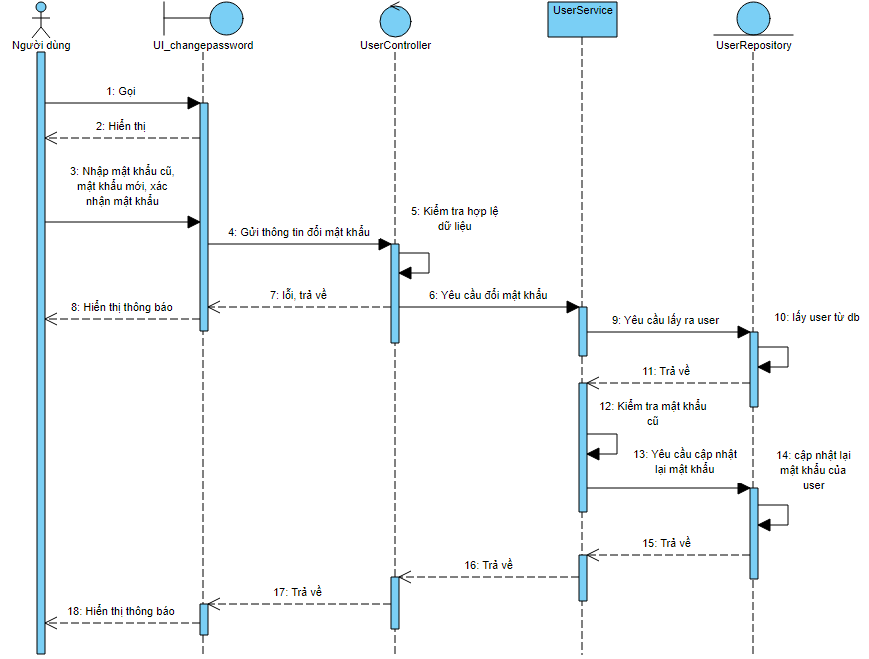
Hình . Usecase chi tiết module đổi mật khẩu

#### Kịch bản

Bảng . Kịch bản module đổi mật khẩu

|  |  |
| --- | --- |
| Tên usecase | Đổi mật khẩu |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập và hệ thống |
| Hậu điều kiện | Người dùng đổi mật khẩu thành công |
| Kịch bản | 1. Người dùng đăng nhập vào hệ thống  2. Tại giao diện trang bất kì có thanh điều hướng, người dùng click vào biểu tượng hình đại diện trên thanh điều hướng.  3. Danh sách chức năng sổ ra với các lựa chọn: Quản lý trang cá nhân, đổi mật khẩu, đăng xuất.  4. Người dùng click chọn đổi mật khẩu  5. Giao diện đổi mật khẩu hiện lên với ô nhập: mật khẩu cũ, mật khẩu mới, xác nhận mật khẩu và ô submit.  6. Người dùng nhập mật khẩu cũ, mật khẩu mới và xác nhận mật khẩu rồi click submit.  7. Hệ thống kiểm tra mật khẩu của người dùng và lưu mật khẩu mới vào CSDL. Sau đó hiển thị thông báo đổi mật khẩu thành công |
| Ngoại lệ | **6.1** Mật khẩu không đúng định dạng  6.1.1 Hệ thống hiển thị thông báo thể hiện rằng: mật khẩu không được để trống, 8 đến 20 kí tự chứa chữ, số và kí tự đặc biệt.  6.1.2 Người dùng nhập lại thông tin đổi mật khẩu rồi click submit  **6.2** Mật khẩu cũ không chính xác  6.2.1 Hệ thống hiển thị thông báo biểu thị mật khẩu cũ không đúng  6.2.2 Người dùng nhập lại mật khẩu chính xác rồi click submit  **6.3** Mật khẩu mới không khớp  6.3.1 Hệ thống hiển thị thông báo biểu thị mật khẩu mới không trùng khớp  6.3.2 Người dùng nhập lại mật khẩu mới và xác nhận mật khẩu rồi nhấn submit |

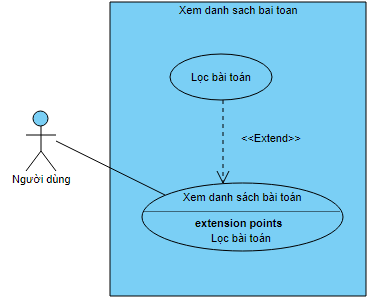
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module đổi mật khẩu

### Module xem danh sách bài toán

#### Usecase chi tiết



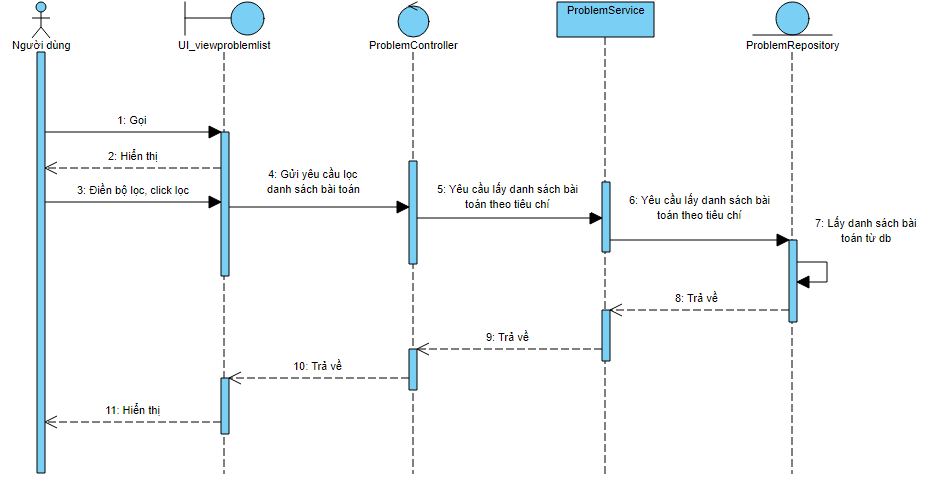
Hình . Usecase chi tiết module xem danh sách bài toán

#### Kịch bản

Bảng . Kịch bản module xem danh sách bài toán

|  |  |
| --- | --- |
| Tên usecase | Xem danh sách bài toán |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã truy cập vào hệ thống |
| Hậu điều kiện | Người dùng xem được danh sách bài toán |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang bất kì, người dùng click chọn bài toán trên thanh điều hướng.  3. Giao diện trang danh sách bài toán hiện ra với danh sách bài toán bao gồm các trường thông tin: tên bài toán, mã bài toán, mức độ và nút visit, icon thể hiện trạng thái làm bài toán: dấu tích thể hiện đã bài toán đã được giải quyết và dấu x thể hiện bài toán chưa giải quyết được. Bên dưới danh sách là ô phân trang và bên cạnh danh sách là bộ lọc bao gồm các trường: ô nhập tên bài toán, ô nhập mã bài toán, ô chọn danh mục bài toán, ô chọn mức độ bài toán, nút filter và nút clear. |
| Ngoại lệ | Không có ngoại lệ |

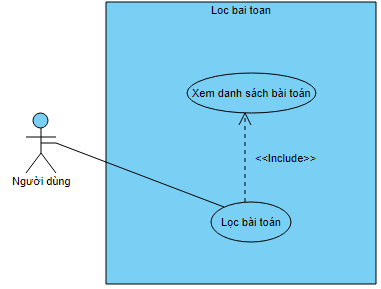
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module xem danh sách bài toán

### Module lọc bài toán theo tiêu chí

#### Usecase chi tiết



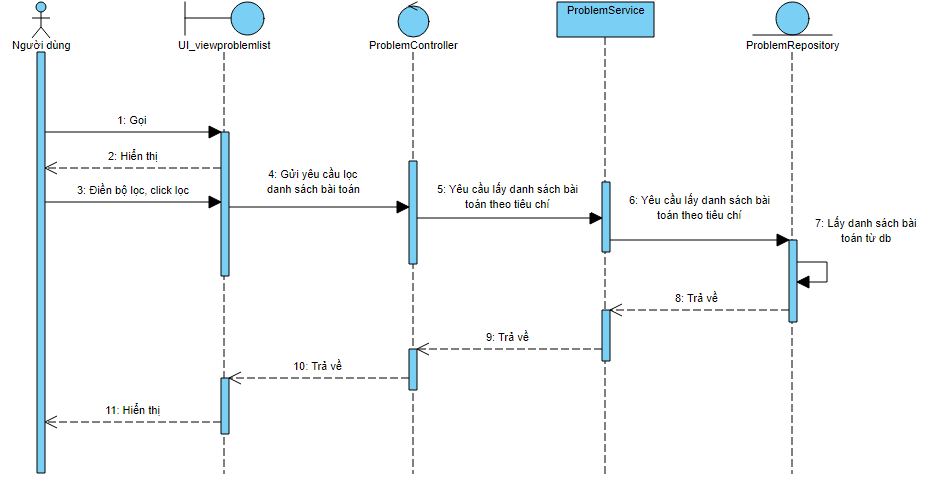
Hình . Usecase chi tiết module lọc bài toán

#### Kịch bản

Bảng . Kịch bản module lọc bài toán theo tiêu chí

|  |  |
| --- | --- |
| Tên usecase | Lọc bài toán theo tiêu chí |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã truy cập vào hệ thống |
| Hậu điều kiện | Người dùng xem được danh sách bài toán theo tiêu chí |
| Kịch bản | 1. Tại giao diện trang bất kì, người dùng click chọn bài toán trên thanh điều hướng.  2. Giao diện trang danh sách bài toán hiện ra với danh sách bài toán với các thông tin: tên bài toán, mã bài toán, mức độ và nút view, icon thể hiện trạng thái. cùng với các ô phân trang và bộ lọc bao gồm: ô nhập tên bài toán, ô nhập mã bài toán, ô chọn danh mục bài toán, ô chọn mức độ bài toán, nút filter và nút clear.  3. Người dùng nhập thông tin tiêu chí của bài toán vào bộ lọc và click filter.  4. Danh sách các bài toán phù hợp với tiêu chí hiện lên với các thông tin: tên bài toán, mã bài toán, mức độ và nút resolve. |
| Ngoại lệ | **4.1** Không có bài toán phù hợp với tiêu chí  4.1.1 Hệ thống hiển thị thông báo: Không có bài toán được tìm thấy |

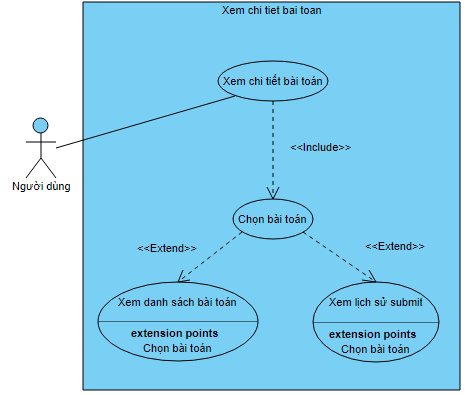
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module lọc bài toán

### Module xem chi tiết bài toán

#### Usecase chi tiết



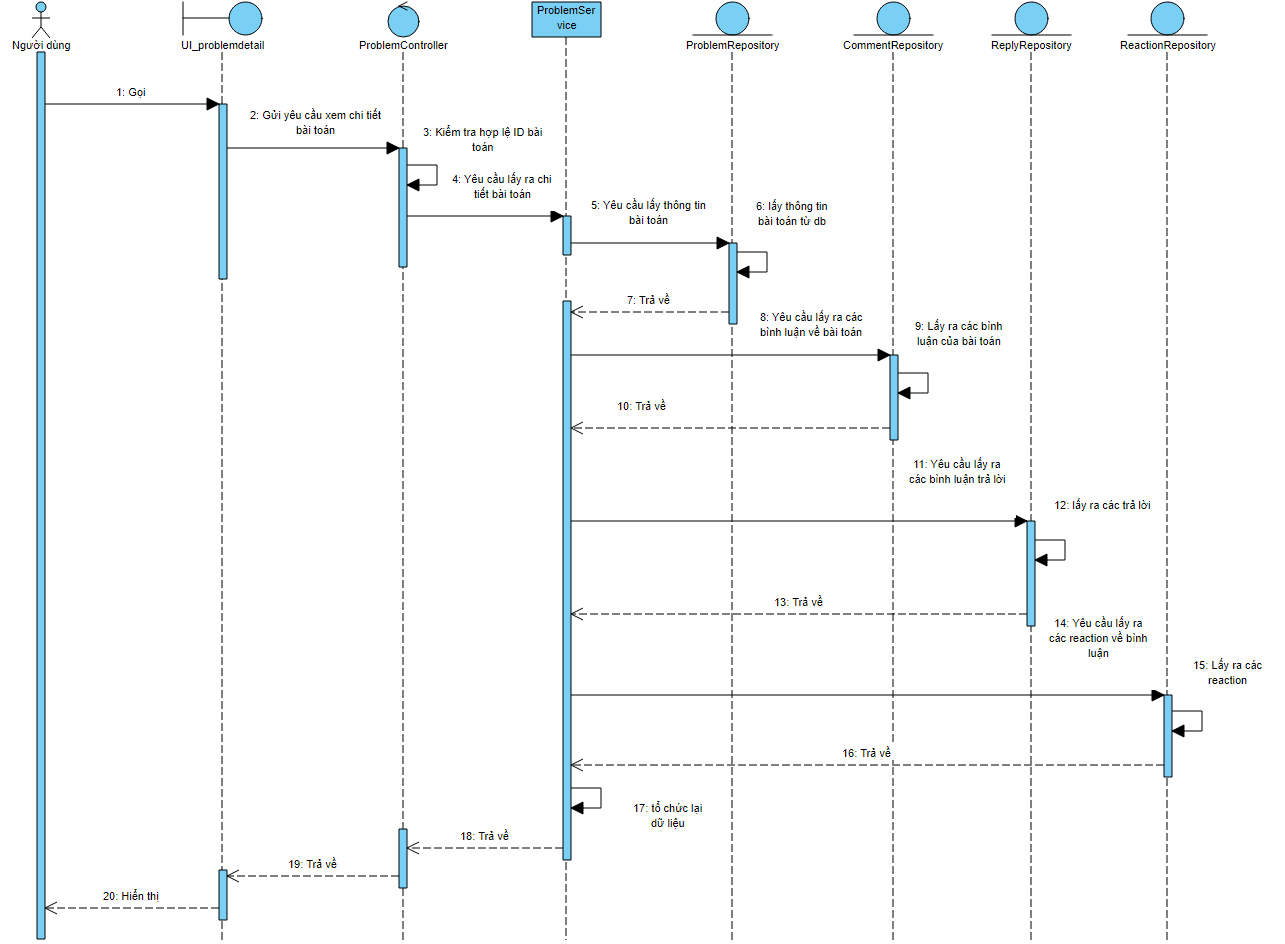
Hình . Usecase tiết module xem chi tiết bài toán

#### Kịch bản

Bảng . Kịch bản module xem chi tiết bài toán

|  |  |
| --- | --- |
| Tên usecase | Xem chi tiết bài toán |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống |
| Hậu điều kiện | Người dùng xem được chi tiết bài toán |
| Kịch bản | 1. Người dùng truy cập vào trang xem bài toán hoặc xem lịch sử submit của hệ thống.  2. Hệ thống hiển thị trang tương ứng.  3. Tại giao diện trang xem bài toán, người dùng click chọn visit trên một bài toán bất kì hoặc tại giao diện của trang xem lịch sử, người dùng click chọn edit một bài làm trên lịch sử submit bài toán của chính mình.  4. Giao diện trang xem chi tiết bài toán hiện ra với thông tin về bài toán bao gồm: tên bài toán, mã bài toán, mức độ, câu hỏi, yêu cầu đầu vào, yêu cầu đầu ra, ví dụ về bộ test của bài toán, ô nhập mã nguồn, ô chọn ngôn ngữ, nút run, nút submit, ô nhập bình luận, nút post comment và danh sách các bình luận và trả lời bình luận về bài toán. Nếu người dùng chưa đăng nhập, ô submit lời giải, ô nhập bình luận và nút post comment sẽ bị ẩn. |
| Ngoại lệ | Không có ngoại lệ |

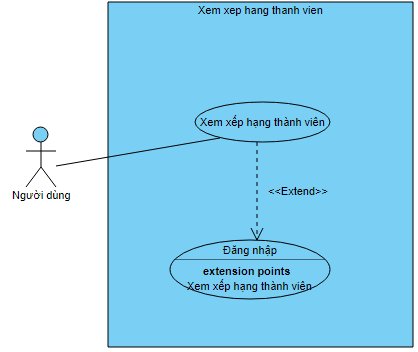
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module xem chi tiết bài toán

### Module xem xếp hạng thành viên

#### Usecase chi tiết



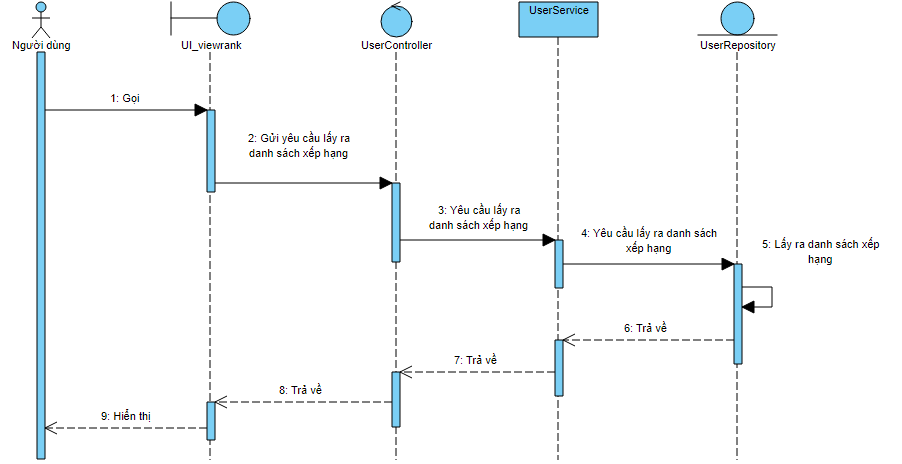
Hình . Usecase chi tiết chức năng xem xếp hạng thành viên

#### Kịch bản

Bảng . Kịch bản module xem xếp hạng thành viên

|  |  |
| --- | --- |
| Tên usecase | Xem xếp hạng thành viên |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống |
| Hậu điều kiện | Người dùng xem được xếp hạng người dùng |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang bất kì, người dùng click chọn xếp hạng trên thanh điều hướng.  3. Giao diện trang xem xếp hạng thành viên hiện ra với xếp hạng của chính người dùng và danh sách 10 thành viên có thành tích tốt nhất bao gồm các trường chứa thông tin: tên hiển thị, điểm số, số lần giải, số bài toán giải chính xác. Nếu người dùng chưa đăng nhập, xếp hạng của chính người dùng sẽ không được hiển thị. |
| Ngoại lệ | Không có ngoại lệ |

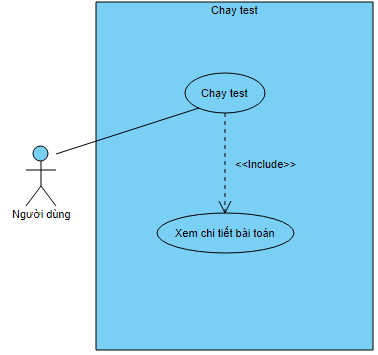
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module xem xếp hạng thành viên

### Module chạy test

#### Usecase chi tiết



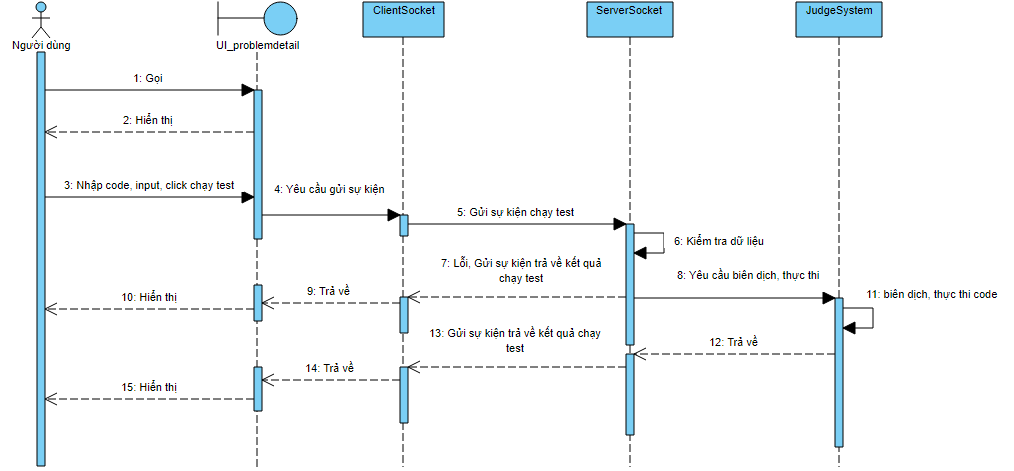
Hình . Usecase chi tiết module chạy test

#### Kịch bản

Bảng . Kịch bản module chạy test

|  |  |
| --- | --- |
| Tên usecase | Chạy test |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống |
| Hậu điều kiện | Người dùng chạy test được lời giải |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang xem bài toán, người dùng click chọn view của một bài toán hoặc tại giao diện của trang xem lịch sử, người dùng click chọn edit một bài làm của chính mình.  3. Giao diện trang xem chi tiết bài toán hiện ra với thông tin về bài toán bao gồm: tên bài toán, mã bài toán, mức độ, câu hỏi, yêu cầu đầu vào, yêu cầu đầu ra, ví dụ, ô input lời giải, select ngôn ngữ, nút run, nút submit, ô nhập bình luận, nút post comment và danh sách các bình luận về bài toán. Nếu người dùng chưa đăng nhập, ô submit lời giải, ô nhập bình luận và nút post comment sẽ bị ẩn.  4. Người dùng nhập mã nguồn của mình vào ô nhập mã nguồn và chọn ngôn ngữ thích hợp, sau đó click vào nút run.  5. Giao diện chạy test hiện ra với ô nhập input đầu vào, nút execute và icon thoát  6. Người dùng nhập input đầu vào rồi click execute  7. Hệ thống chạy mã nguồn của người dùng và hiển thị output ứng với input đầu vào  8. Người dùng tiếp tục thực hiện lại bước 6 cho đến khi chọn nút thoát. |
| Ngoại lệ | **4.1** Người dùng không nhập mã nguồn  4.1.1 Hệ thống hiển thị thông báo biểu thị rằng người dùng phải nhập mã nguồn để chạy được test.  4.1.2 Người dùng nhập mã nguồn của mình vào ô nhập mã nguồn sau đó nhấn nút run. |

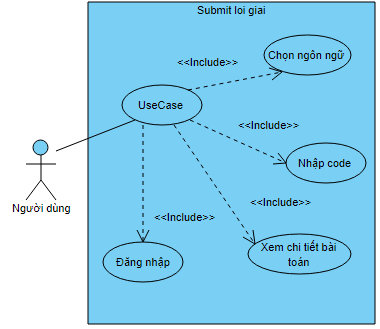
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module chạy test

### Module submit lời giải

#### Usecase chi tiết



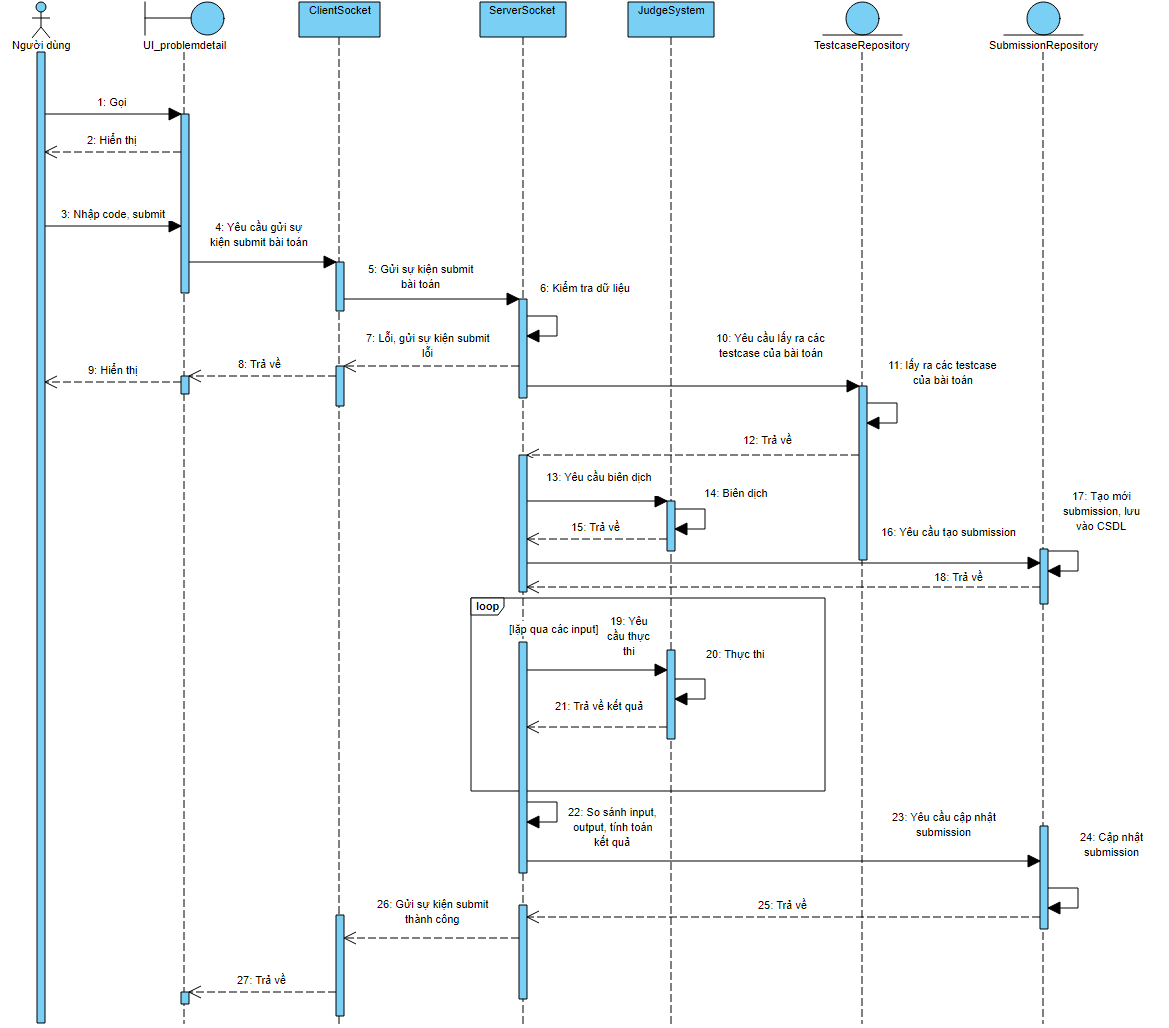
Hình . Usecase chi tiết module submit lời giải

#### Kịch bản

Bảng . Kịch bản module submit lời giải

|  |  |
| --- | --- |
| Tên usecase | Submit lời giải |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Hậu điều kiện | Người dùng submit lời giải thành công |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang xem bài toán, người dùng click chọn visit trên một bài toán bất kì hoặc tại giao diện của trang xem lịch sử, người dùng click chọn edit trên một lịch sử submit bài toán của chính mình.  3. Giao diện trang xem chi tiết bài toán hiện ra với thông tin về bài toán bao gồm: tên bài toán, mã bài toán, mức độ, câu hỏi, yêu cầu đầu vào, yêu cầu đầu ra, ví dụ, ô nhập lời giải, ô chọn ngôn ngữ lập trình, nút run, nút submit, ô nhập bình luận, nút post comment, danh sách các bình luận và trả lời bình luận về bài toán. Nếu người dùng chưa đăng nhập, ô submit lời giải, ô nhập bình luận và nút post comment sẽ bị ẩn.  4. Người dùng nhập mã nguồn của mình vào ô nhập mã nguồn và chọn ngôn ngữ thích hợp, sau đó click vào nút submit.  5. Hệ thống biên dịch và thực thi mã nguồn của người dùng sau đó chuyển hướng đến trang xem lịch sử submit để xem kết quả. |
| Ngoại lệ | **4.1** Người dùng không nhập mã nguồn  4.1.1 Hệ thống hiển thị thông báo biểu thị rằng người dùng phải nhập mã nguồn để chạy được test.  4.1.2 Người dùng nhập mã nguồn của mình vào ô nhập code rồi click submit. |

#### Biểu đồ tuần tự

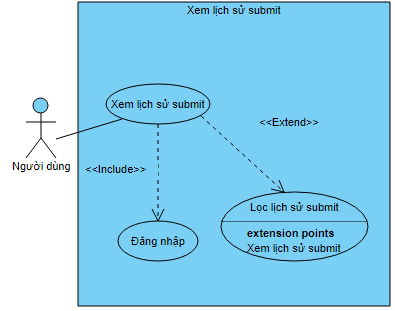


Hình . Biểu đồ tuần tự module submit lời giải (2)

1. *Chi tiết xem tại phụ lục A trang 92*

### Module xem lịch sử submit

#### Usecase chi tiết



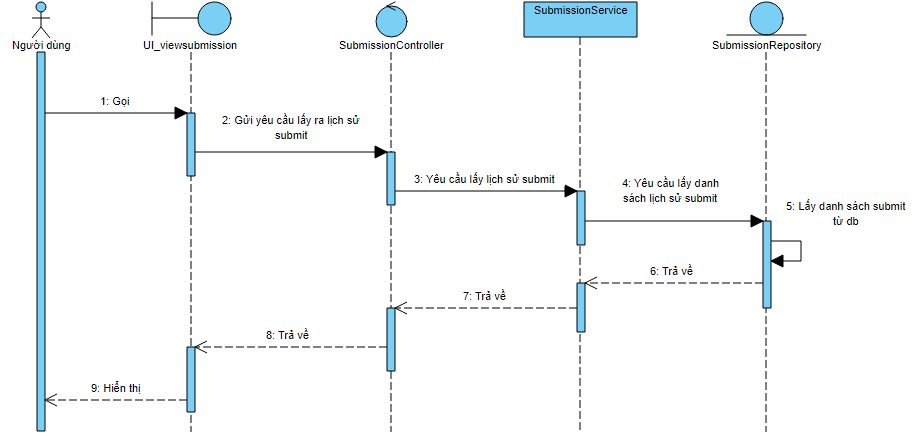
Hình . Usecase chi tiết module xem lịch sử submit

#### Kịch bản

Bảng . Kịch bản module xem lịch sử submit

|  |  |
| --- | --- |
| Tên usecase | Xem lịch sử submit |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Hậu điều kiện | Người dùng xem được lịch sử submit |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang bất kì, người dùng click chọn lịch sử trên thanh điều hướng.  3. Giao diện trang xem lịch sử submit xuất hiện với danh sách các submit bao gồm các thông tin: mã submit, tên bài toán, tên tác giả, thời gian chạy, kết quả và nút edit nếu đó là submit của chính người dùng và nút chọn: all ( cho tất cả submit) và you (cho submit của người dùng). |
| Ngoại lệ | **3.1** Người dùng không có submit nào  3.1.1 Hệ thống hiển thị thông báo biểu thị người dùng không có submit nào được tìm thấy. |

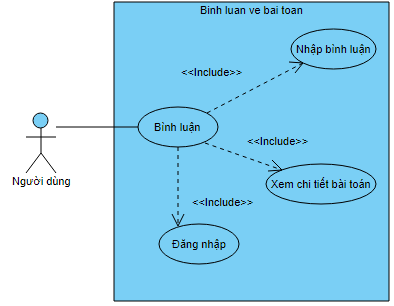
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module xem lịch sử submit

### Bình luận về bài toán

#### Usecase chi tiết



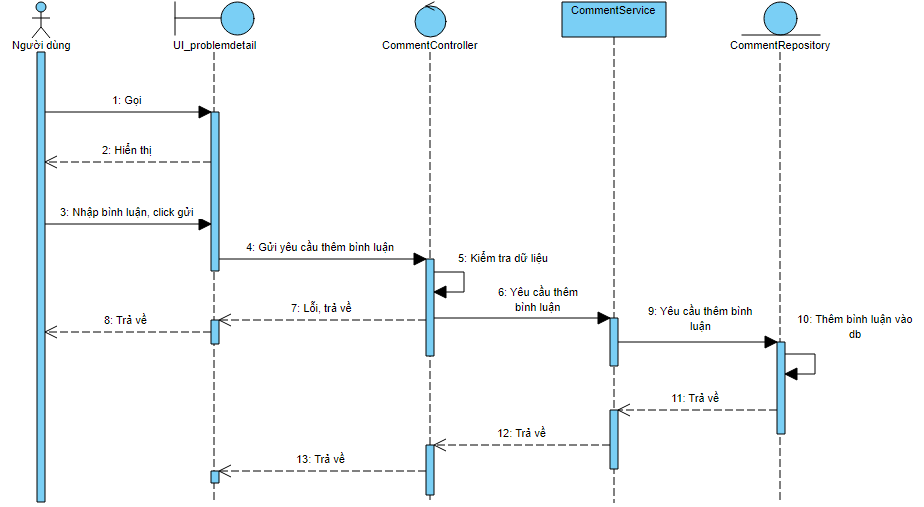
Hình . Usecase chi tiết module bình luận về bài toán

#### Kịch bản

Bảng . Kịch bản module bình luận về bài toán

|  |  |
| --- | --- |
| Tên usecase | Bình luận về bài toán |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng truy cập vào hệ thống |
| Hậu điều kiện | Người dùng bình luận thành công |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang xem bài toán, người dùng click chọn view của một bài toán hoặc tại giao diện của trang xem lịch sử, người dùng click chọn edit một bài làm của chính mình.  3. Giao diện trang xem chi tiết bài toán hiện ra với thông tin về bài toán bao gồm: tên bài toán, mã bài toán, mức độ, câu hỏi, yêu cầu đầu vào, yêu cầu đầu ra, ví dụ, ô input lời giải, select ngôn ngữ, nút run, nút submit, ô nhập bình luận, nút post comment và danh sách các bình luận về bài toán. Nếu người dùng chưa đăng nhập, ô submit lời giải, ô nhập bình luận và nút post comment sẽ bị ẩn.  4. Người dùng nhập bình luận vào ô nhập bình luận và click post comment  5. Hệ thống lưu lại bình luận của người dùng và tải lại danh sách bình luận |
| Ngoại lệ | **4.1** Người dùng không nhập bình luận  4.1.1 Hệ thống hiển thị thông báo biểu thị rằng người dùng phải nhập bình luận để có thể post comment.  4.1.2 Người dùng nhập lại bình luận của mình vào ô nhập bình luận sau đó nhấn nút post comment. |

#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module bình luận về bài toán

### Module trả lời bình luận

#### Usecase chi tiết



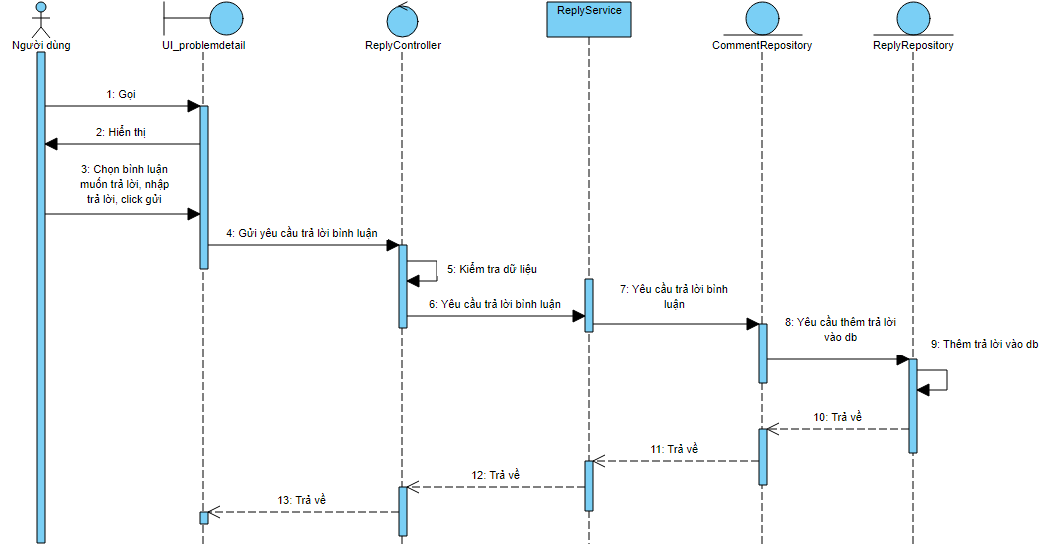
Hình . Usecase chi tiết module trả lời bình luận

#### Kịch bản

Bảng . Kịch bản module trả lời bình luận

|  |  |
| --- | --- |
| Tên usecase | Trả lời bình luận |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Hậu điều kiện | Người dùng trả lời bình luận thành công |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang xem bài toán, người dùng click chọn view của một bài toán hoặc tại giao diện của trang xem lịch sử, người dùng click chọn edit một bài làm của chính mình.  3. Giao diện trang xem chi tiết bài toán hiện ra với thông tin về bài toán bao gồm cả danh sách bình luận của bài toán. Mỗi bình luận đều có nút reaction, trả lời bình luận, người bình luận, nội dung và thời gian bình luận. Nếu người dùng chưa đăng nhập, ô submit lời giải, ô nhập bình luận và nút post comment sẽ bị ẩn.  4. Tại bình luận muốn trả lời. Người dùng chọn nút trả lời  5. Ô input trả lời bình luận hiện ra cùng biểu tượng gửi và nút cancel  6. Người dùng nhập bình luận và ấn gửi.  7. Hệ thống lưu lại bình luận trả lời đồng thời đóng ô input trả lời bình luận. |
| Ngoại lệ | **6.1** Người dùng không nhập bình luận  6.1.1 Hệ thống hiển thị thông báo biểu thị rằng người dùng phải nhập bình luận để có thể gửi  6.1.2 Người dùng nhập lại bình luận của mình vào ô nhập bình luận sau đó nhấn nút gửi |

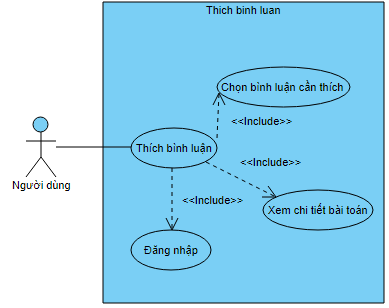
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module trả lời bình luận

### Module thích bình luận

#### Usecase chi tiết



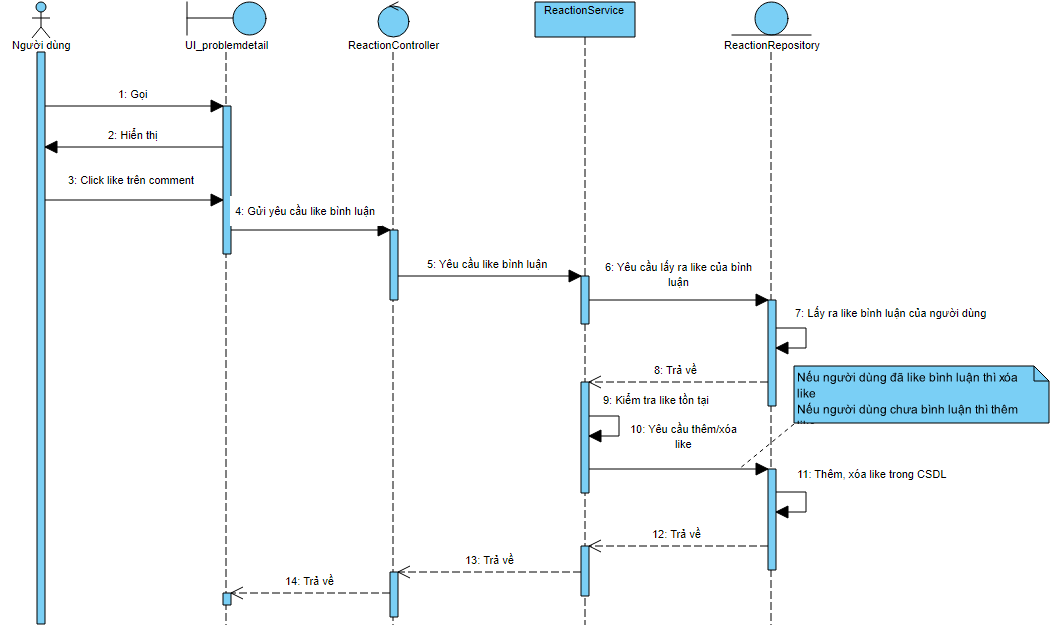
Hình . Usecase chi tiết module thích bình luận

#### Kịch bản

Bảng . Kịch bản module thích bình luận

|  |  |
| --- | --- |
| Tên usecase | Thích hoặc hủy thích bình luận |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Hậu điều kiện | Người dùng thích hoặc hủy thích bình luận thành công |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang xem bài toán, người dùng click chọn nút visit trên một bài toán bất kỳ hoặc tại giao diện của trang xem lịch sử submit, người dùng click chọn edit trên một lịch sử submit của chính mình.  3. Giao diện trang xem chi tiết bài toán hiện ra với thông tin về bài toán bao gồm các trường thông tin chứa tên bài toán, mã bài toán, mức độ, câu hỏi, yêu cầu đầu vào, yêu cầu đầu ra, ví dụ, ô input lời giải, select ngôn ngữ, nút run, nút submit, ô nhập bình luận, nút post comment và danh sách các bình luận về bài toán. Mỗi bình luận đều có nút thích, trả lời bình luận, người bình luận, nội dung và thời gian bình luận. Nếu người dùng chưa đăng nhập, ô submit lời giải, ô nhập bình luận và nút post comment sẽ bị ẩn.  4. Tại bình luận mà người dùng muốn thích hoặc hủy thích. Người dùng chọn biểu tượng thích bình luận.  5. Nếu người dùng đã thích bình luận trước đó, hệ thống sẽ hủy thích bình luận và lưu trạng thái vào cơ sở dữ liệu và ngược lại, nếu người dùng chưa thích bình luận trước đó hệ thống sẽ ghi nhận lượt yêu thích và lưu lại thông tin lượt thích của người dùng vào cơ sở dữ liệu. |
| Ngoại lệ | Không có ngoại lệ |

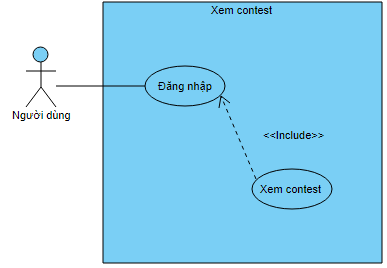
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module thích bình luận

### Module xem contest

#### Usecase tổng quan



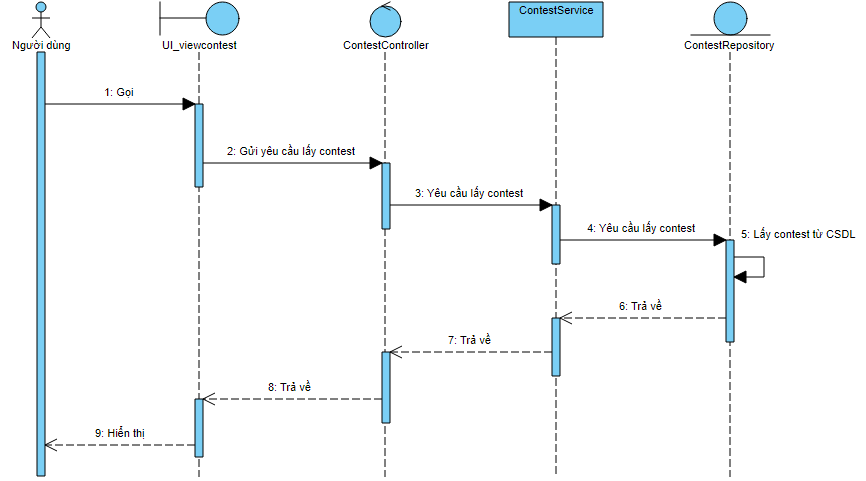
Hình . Usecase chi tiết module xem contest

#### Kịch bản

Bảng . Kịch bản module xem contest

|  |  |
| --- | --- |
| Tên usecase | Xem contest |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Hậu điều kiện | Người dùng xem được danh sách contest |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang bất kì, người dùng click chọn contest trên thanh điều hướng.  3. Giao diện trang xem contest hiện ra với các loại danh sách: danh sách các contest sắp đến, danh sách contest đã đăng ký bao gồm: tên contest, trạng thái, đối với contest sắp đến có thêm nút register. Đối với contest đã đăng ký và sắp bắt đầu có thêm nút Join. Đối với contest đã đăng ký và kết thúc có nút view result. |
| Ngoại lệ | **3.1** Không có contest thỏa mãn  3.1.1 Hệ thống hiển thị thông báo biểu thị không có contest nào được tìm thấy. |

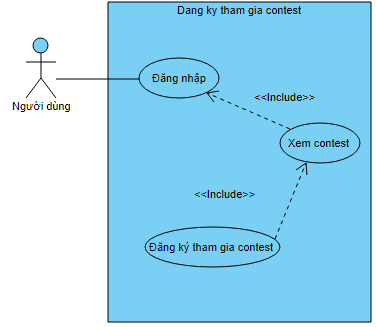
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module xem contest

### Đăng ký tham gia contest

#### Usecase chi tiết



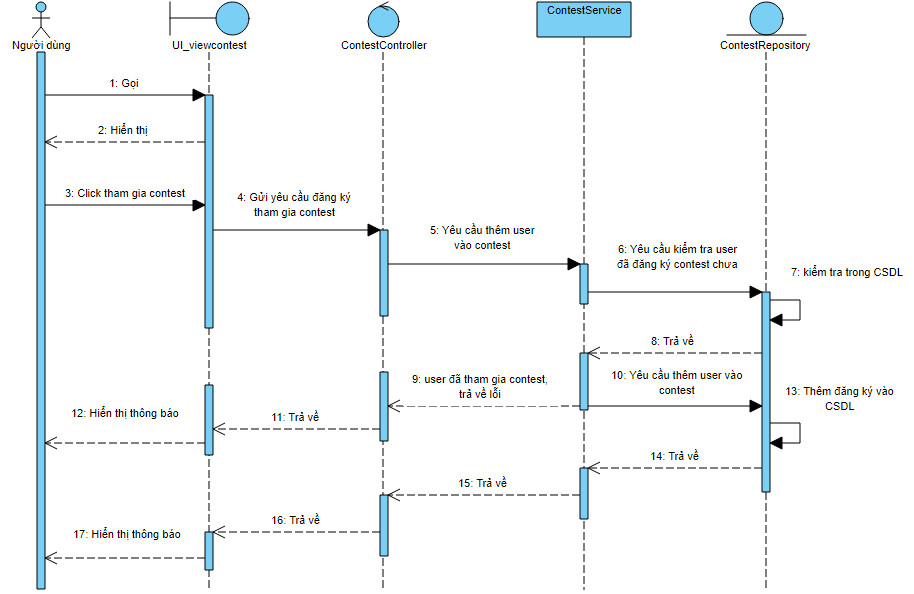
Hình . Usecase chi tiết module đăng ký tham gia contest

#### Kịch bản

Bảng . Kịch bản module đăng ký tham gia contest

|  |  |
| --- | --- |
| Tên usecase | Đăng ký tham gia contest |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Hậu điều kiện | Người dùng đăng ký tham gia contest thành công |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang bất kì, người dùng click chọn contest trên thanh điều hướng.  3. Giao diện trang xem contest hiện ra với các loại danh sách: danh sách các contest sắp đến, danh sách contest đã đăng ký bao gồm: tên contest, trạng thái, số người giới hạn, số người còn lại. Đối với contest sắp đến có thêm nút register. Đối với contest đã đăng ký và sắp bắt đầu có thêm nút Join. Đối với contest đã đăng ký và kết thúc có nút view result.  4. Trên danh sách contest sắp đến, người dùng click chọn contest muốn đăng ký tham gia.  5. Hệ thống lưu lại thông tin Đăng ký và hiển thị thông báo đăng ký thành công. |
| Ngoại lệ | **4.1** Người dùng đã đăng ký trước đó  4.1.1 Hệ thống hiển thị thông báo người dùng đã đăng ký contest trước đó.  **4.2** Số lượng người tham gia đã đạt đến tối đa  4.2.1 Hệ thống hiển thị thông báo thể hiện số lượng người tham gia đã đạt đến tối đa |

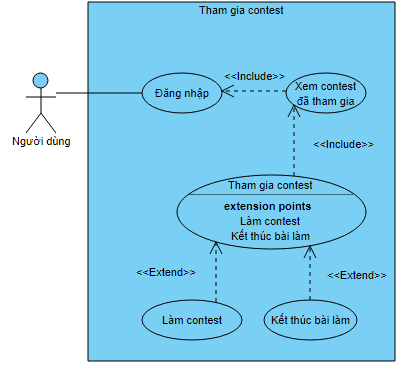
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module đăng ký tham gia contest

### Tham gia contest

#### Usecase chi tiết



Hình . Usecase chi tiết module tham gia contest

#### Kịch bản

Bảng . Kịch bản module tham gia contest

|  |  |
| --- | --- |
| Tên usecase | Đăng ký tham gia contest |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Hậu điều kiện | Người dùng hoàn thành contest |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang bất kì, người dùng click chọn contest trên thanh điều hướng.  3. Giao diện trang xem contest hiện ra với các loại danh sách: danh sách các contest sắp đến, danh sách contest đã đăng ký bao gồm: tên contest, trạng thái, đối với contest sắp đến có thêm nút register. Đối với contest đã đăng ký và sắp bắt đầu có thêm nút Join. Đối với contest đã đăng ký và kết thúc có nút view result.  4. Trên danh sách contest đã đăng ký có trạng thái đã bắt đầu, người dùng click nút join.  5. Giao diện trang contest hiện ra với tên contest, nút finish, nút cancel và danh sách các bài tập trong contest. Mỗi bài tập bao gồm các thông tin: tên bài, điểm số, câu hỏi, input, output, ví dụ. ô tải file, nút submit và danh sách các submit của người tham gia.  5. Người dùng giải bài tập trong contest và submit lời giải lên hệ thống  6. Hệ thống ghi nhận lời giải và cập nhật lại lịch sử submit của người dùng đang thực hiện contest  7. Khi kết thúc, người dùng ấn finish để kết thúc làm contest  8. Hệ thống hiển thị hộp thoại xác nhận kết thúc contest gồm thông báo xác nhận kết thúc, nút đồng ý và nút cancel.  9. Người dùng click chọn đồng ý  10. Hệ thống cập nhật trạng thái người dùng đã hoàn thành contest và lưu vào hệ thống. |
| Ngoại lệ | **4.1** Contest đã hoàn thành  4.1.1 Hệ thống tự chuyển hướng sang trang xem lịch sử contest. |

#### Biểu đồ tuần tự

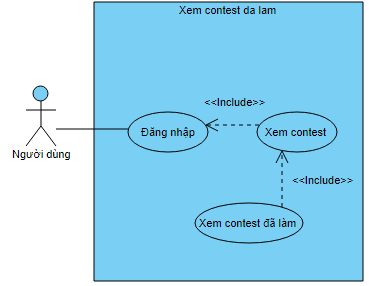


Hình . Biểu đồ tuần tự module tham gia contest (3)

1. *Chi tiết xem tại phụ lục A trang 93*

### Xem contest đã làm

#### Usecase chi tiết



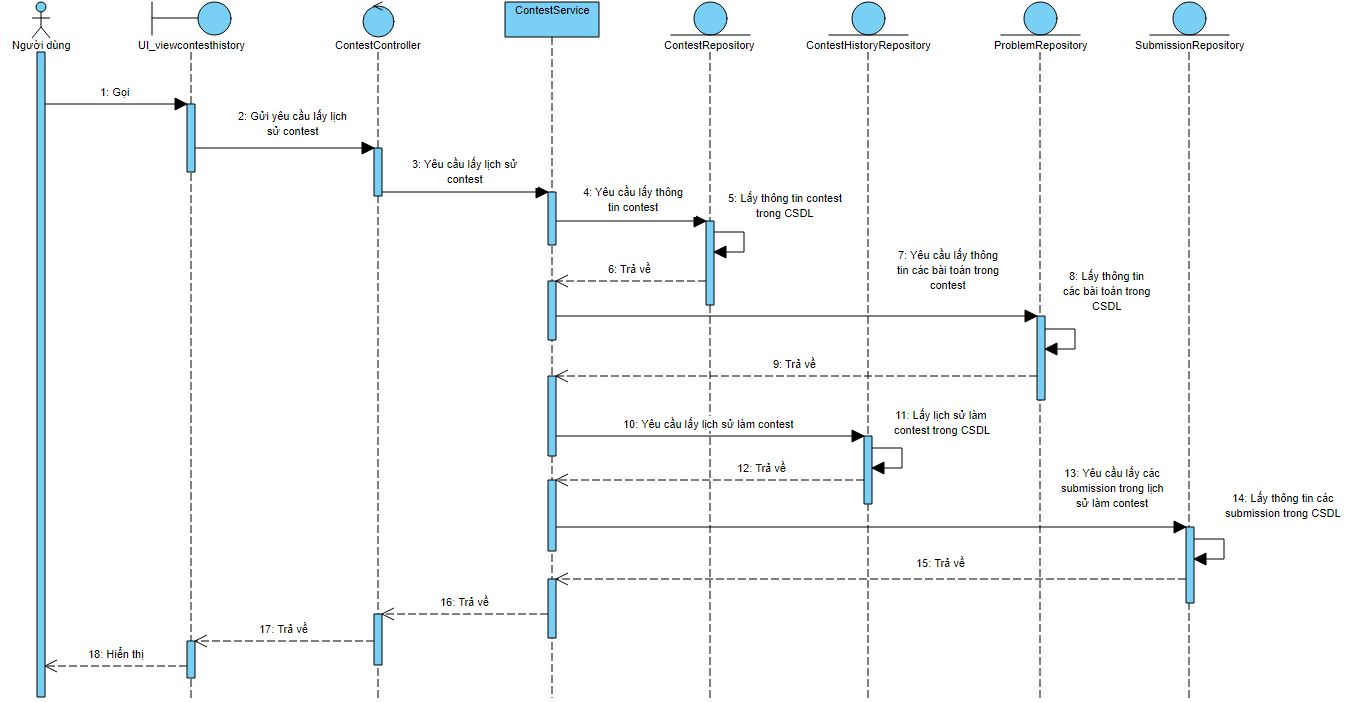
Hình . Usecase chi tiết module xem contest đã làm

#### Kịch bản

Bảng . Kịch bản module xem contest đã làm

|  |  |
| --- | --- |
| Tên usecase | Xem contest đã làm |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng đã đăng nhập vào hệ thống |
| Hậu điều kiện | Người dùng xem được contest đã làm |
| Kịch bản | 1. Người dùng truy cập vào hệ thống  2. Tại giao diện trang bất kì, người dùng click chọn contest trên thanh điều hướng.  3. Giao diện trang xem contest hiện ra với các loại danh sách: danh sách các contest sắp đến, danh sách contest đã đăng ký bao gồm: tên contest, trạng thái, đối với contest sắp đến có thêm nút register. Đối với contest đã đăng ký và sắp bắt đầu có thêm nút Join. Đối với contest đã đăng ký và kết thúc có nút view result.  4. Trên danh sách contest đã đăng ký có trạng thái đã hoàn thành, người dùng chọn một contest muốn xem rồi click nút view result.  5. Giao diện trang xem contest đã làm hiện ra với tên contest, nút cancel và danh sách các bài tập trong contest. Mỗi bài tập bao gồm các thông tin: tên bài, điểm số, câu hỏi, input, output, ví dụ. ô tải file bị disabled, nút submit bị disablevà danh sách các submit của người tham gia. |
| Ngoại lệ | Không có ngoại lệ |

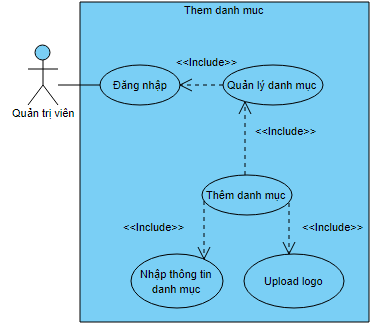
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module xem contest đã làm)

### Module thêm danh mục

#### Usecase chi tiết



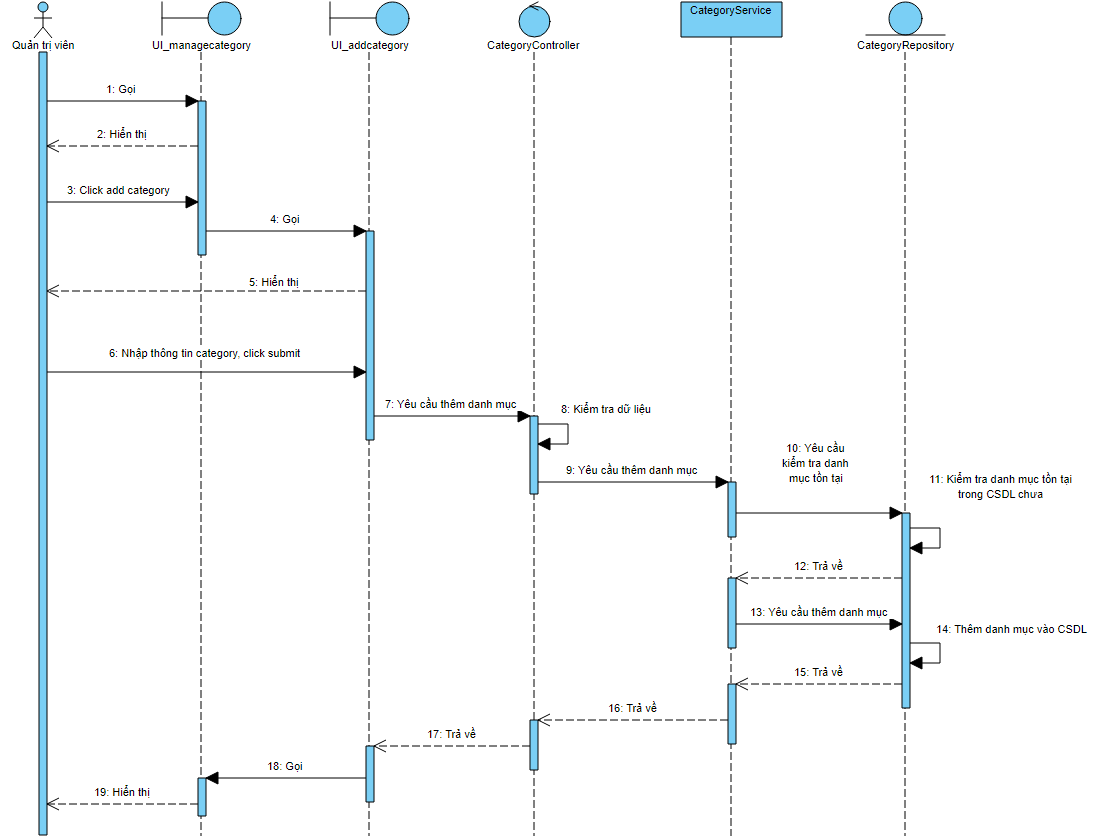
Hình . Usecase chi tiết module thêm danh mục

#### Kịch bản

Bảng . Kịch bản module thêm danh mục

|  |  |
| --- | --- |
| Tên usecase | Thêm danh mục |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống |
| Hậu điều kiện | Thêm danh mục thành công |
| Kịch bản | 1. Quản trị viên đăng nhập vào hệ thống  2. Tại giao diện trang chủ của quản trị viên, trên thanh điều hướng, quản trị viên chọn quản lý danh mục.  3. Trang quản lý danh mục hiện ra với nút thêm danh mục và danh sách các danh mục. Mỗi danh mục bao gồm các thông tin: mã danh mục, tên danh mục, logo danh mục, trạng thái danh mục, ngày cập nhật, ngày chỉnh sửa, nút sửa danh mục, xóa danh mục, ẩn/hiện danh mục.  4. Quản trị viên chọn thêm danh mục  5. Giao diện trang thêm danh mục hiện ra với ô nhập tên, ô upload logo, nút clear logo nút submit và nút cancel.  6. Quản trị viên nhập tên danh mục, upload ảnh sau đó nhấn nút submit.  7. Hệ thống lưu danh mục vào CSDL và thông báo thành công sau đó chuyển hướng về trang quản lý danh mục |
| Ngoại lệ | **6.1** Tên danh mục đã tồn tại trên hệ thống  6.1.1 Hệ thống hiển thị thông báo thể hiện danh mục đã tồn tại trên hệ thống  6.1.2 Quản trị viên nhập tên danh mục khác rồi nhất submit  **6.2** Tên danh mục để trống  6.2.1 Hệ thống hiển thị thông báo biểu thị tên danh mục không được để trống  6.2.2 Quản trị viên nhập tên danh mục rồi nhấn submit  **6.3** Quản trị viên không upload ảnh  6.3.1 Hệ thống hiển thị thông báo biểu thị logo của danh mục cần được upload  6.3.2 Quản trị viên upload ảnh rồi nhấn submit |

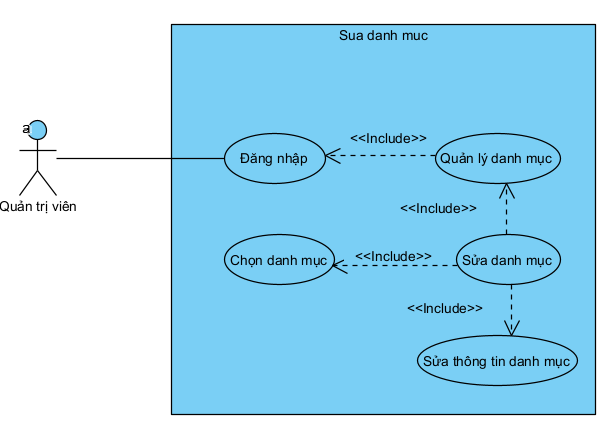
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module thêm danh mục (6)

### Module sửa danh mục

#### Usecase chi tiết



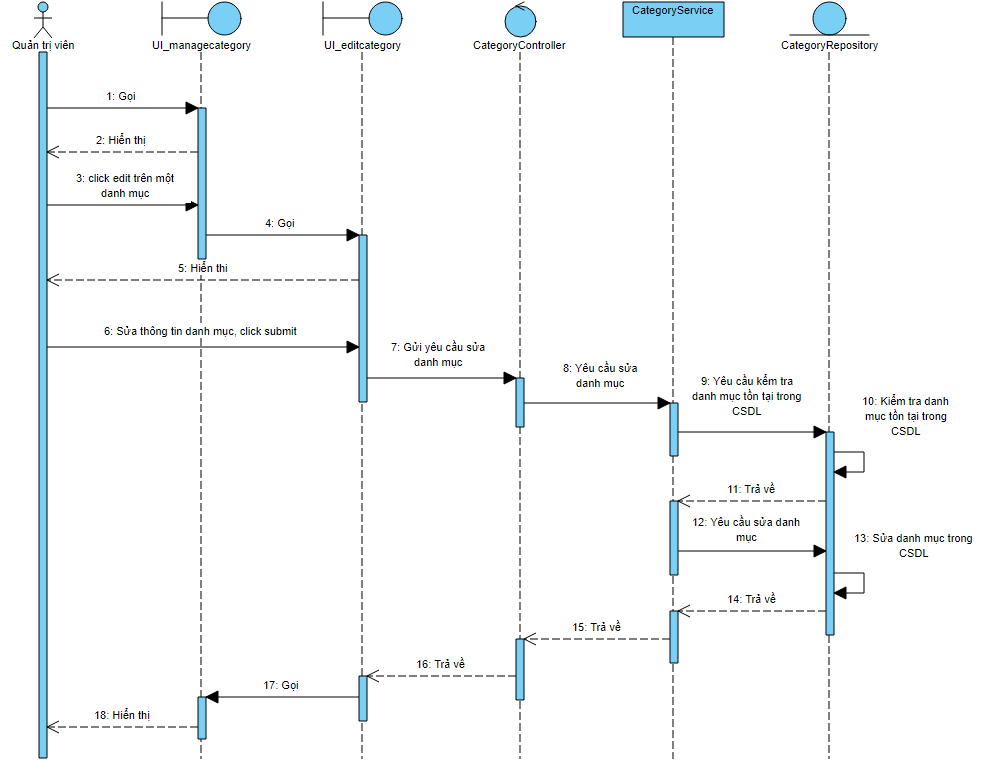
Hình . Usecase chi tiết module sửa danh mục

#### Kịch bản

Bảng . Kịch bản module sửa danh mục

|  |  |
| --- | --- |
| Tên usecase | Sửa danh mục |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống |
| Hậu điều kiện | Sửa danh mục thành công |
| Kịch bản | 1. Quản trị viên đăng nhập vào hệ thống  2. Tại giao diện trang chủ của quản trị viên, trên thanh điều hướng, quản trị viên chọn quản lý danh mục.  3. Trang quản lý danh mục hiện ra với nút thêm danh mục và danh sách các danh mục. Mỗi danh mục bao gồm các thông tin: mã danh mục, tên danh mục, logo danh mục, trạng thái danh mục, ngày cập nhật, ngày chỉnh sửa, nút sửa danh mục, xóa danh mục, ẩn/hiện danh mục.  4. Quản trị viên chọn nút sửa trên danh mục muốn sửa.  5. Giao diện trang sửa danh mục hiện ra với ô sửa tên danh mục, ô upload logo, nút clear logo nút submit và nút cancel.  6. Quản trị viên nhập tên danh mục, upload ảnh sau đó nhấn nút submit.  7. Hệ thống lưu danh mục vào CSDL và thông báo thành công sau đó chuyển hướng về trang quản lý danh mục |
| Ngoại lệ | **6.1** Tên danh mục đã tồn tại trên hệ thống  6.1.1 Hệ thống hiển thị thông báo thể hiện danh mục đã tồn tại trên hệ thống  6.1.2 Quản trị viên nhập tên danh mục khác rồi nhất submit  **6.2** Tên danh mục để trống  6.2.1 Hệ thống hiển thị thông báo biểu thị tên danh mục không được để trống  6.2.2 Quản trị viên nhập tên danh mục rồi nhấn submit  **6.3** Quản trị viên không upload ảnh  6.3.1 Hệ thống hiển thị thông báo biểu thị logo của danh mục cần được upload |

#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module sửa danh mục

### Module xóa danh mục

#### Usecase chi tiết



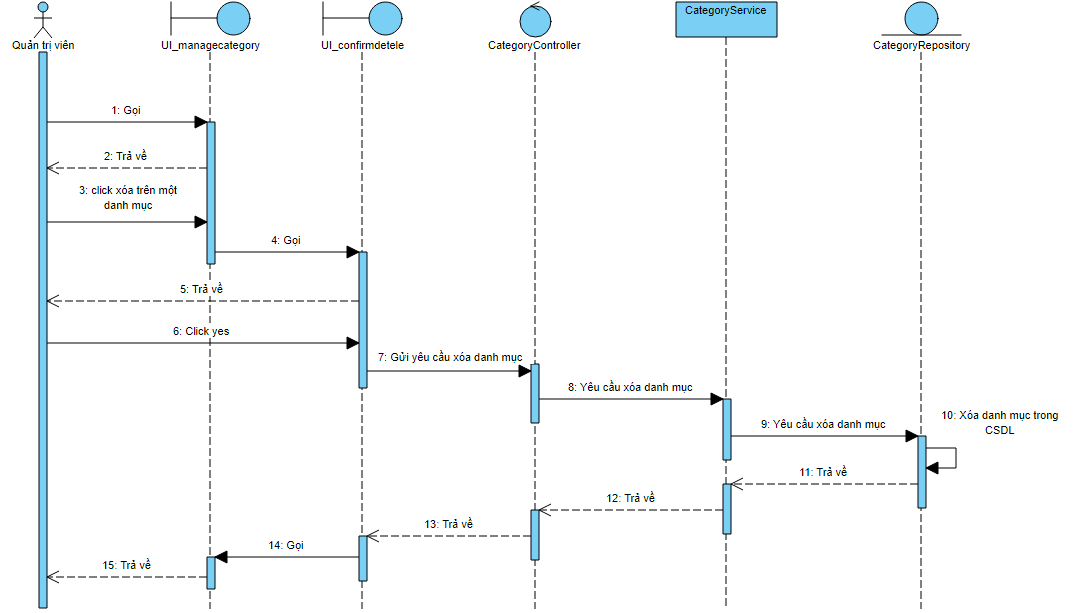
Hình . Usecase chi tiết module xóa danh mục

#### Kịch bản

Bảng . Kịch bản module xóa danh mục

|  |  |
| --- | --- |
| Tên usecase | Xóa danh mục |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống |
| Hậu điều kiện | Xóa danh mục thành công |
| Kịch bản | 1. Quản trị viên đăng nhập vào hệ thống  2. Tại giao diện trang chủ của quản trị viên, trên thanh điều hướng, quản trị viên chọn quản lý danh mục.  3. Trang quản lý danh mục hiện ra với nút thêm danh mục và danh sách các danh mục. Mỗi danh mục bao gồm các thông tin: mã danh mục, tên danh mục, logo danh mục, trạng thái danh mục, ngày cập nhật, ngày chỉnh sửa, nút sửa danh mục, xóa danh mục, ẩn/hiện danh mục.  4. Quản trị viên chọn nút xóa trên danh mục cần xóa  5. Hộp thoại xác nhận xóa hiện lên với thông báo xác nhận xóa, nút yes và nút cancel.  6. Quản trị viên chọn nút yes  7. Hệ thống xóa danh mục và chuyển hướng về trang quản lý danh muc |
| Ngoại lệ | **6.1** Danh mục đang có bài toán tham chiếu đến  6.1.1 Hệ thống hiển thị thông báo thể hiện không thể xóa danh mục đang có bài toán tham chiếu đến và chuyển về trang quản lý danh mục |

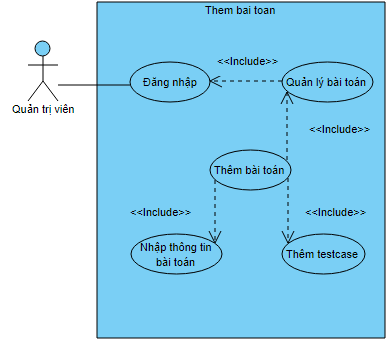
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module xóa danh mục

### Module thêm bài toán

#### Usecase chi tiết



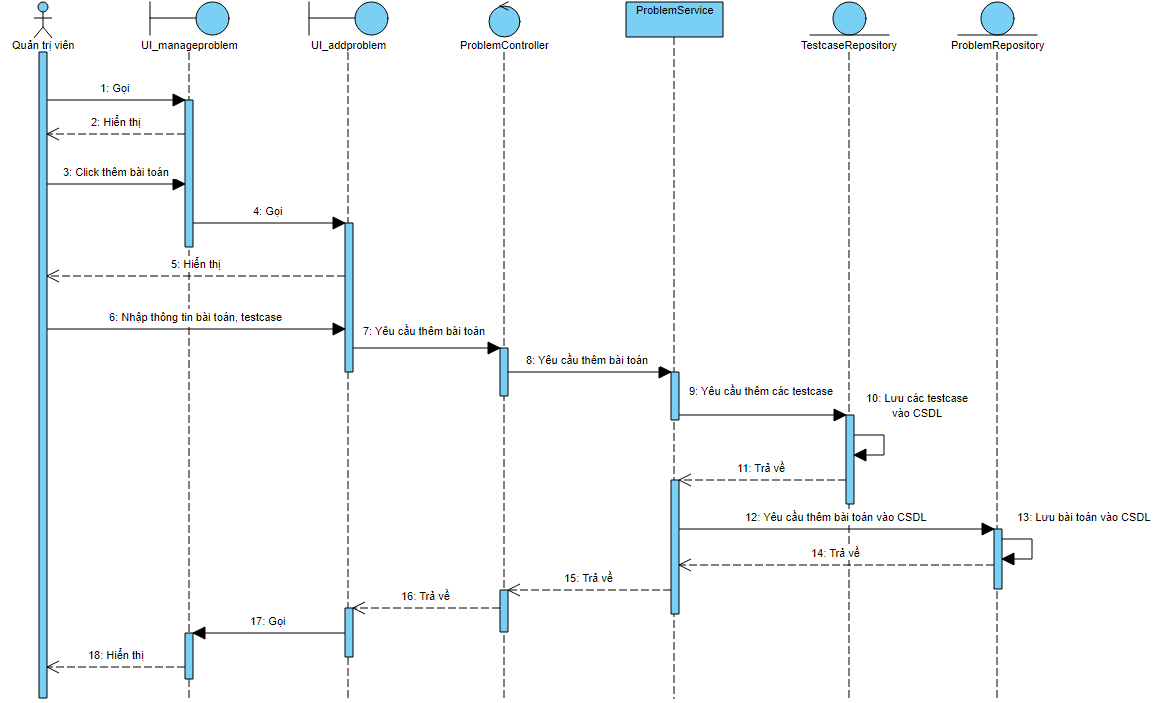
Hình . Usecase chi tiết module thêm bài toán

#### Kịch bản

Bảng . Kịch bản module thêm bài toán

|  |  |
| --- | --- |
| Tên usecase | Thêm bài toán |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống |
| Hậu điều kiện | Thêm bài toán thành công |
| Kịch bản | 1. Quản trị viên đăng nhập vào hệ thống  2. Tại giao diện trang chủ của quản trị viên, trên thanh điều hướng, quản trị viên chọn quản lý bài toán.  3. Trang quản lý bài toán hiện ra với nút thêm bài toán và danh sách các bài toán. Mỗi bài toán bao gồm: mã bài toán, tên bài toán, trạng thái bài toán, mức độ của bài toán, nút sửa và nút xóa.  4. Quản trị viên chọn thêm bài toán  5. Giao diện trang thêm bài toán hiện ra với ô nhập tên bài toán, chọn mức độ của bài toán, chọn danh mục chứa tất cả các danh mục của hệ thống, ô soạn thảo câu hỏi, ô nhập ràng buộc đầu vào, ô nhập ràng buộc đầu ra, nút cancel, nút submit, ô nhập input và output của testcase, nút thêm testcase và danh sách các testcase. Mỗi testcase bao gồm thông tin testcase: input của testcase, output của testcase, và nút xóa testcase.  6. Quản trị viên nhập hoàn thành đầy đủ thông tin bao gồm: tên bài toán, chọn mức độ, chọn danh mục, câu hỏi, ràng buộc đầu vào, ràng buộc đầu ra. Lặp lại các bước sau cho đến khi hết các testcase: Nhập input của testcase, nhập output của testcase, Sau đó click nút thêm testcase.  7. Với mỗi testcase được thêm mới, hệ thống tự động cập nhật lại danh sách các testcase.  8. Quản trị viên click nút submit.  9. Hệ thống lưu bài toán vào CSDL và thông báo thành công sau đó chuyển hướng về trang quản lý bài toán. |
| Ngoại lệ | Không có ngoại lệ |

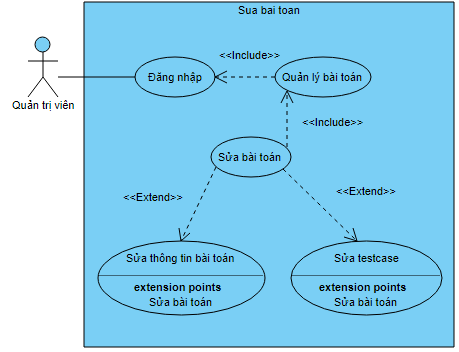
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module thêm bài toán

### Module sửa bài toán

#### Usecase chi tiết



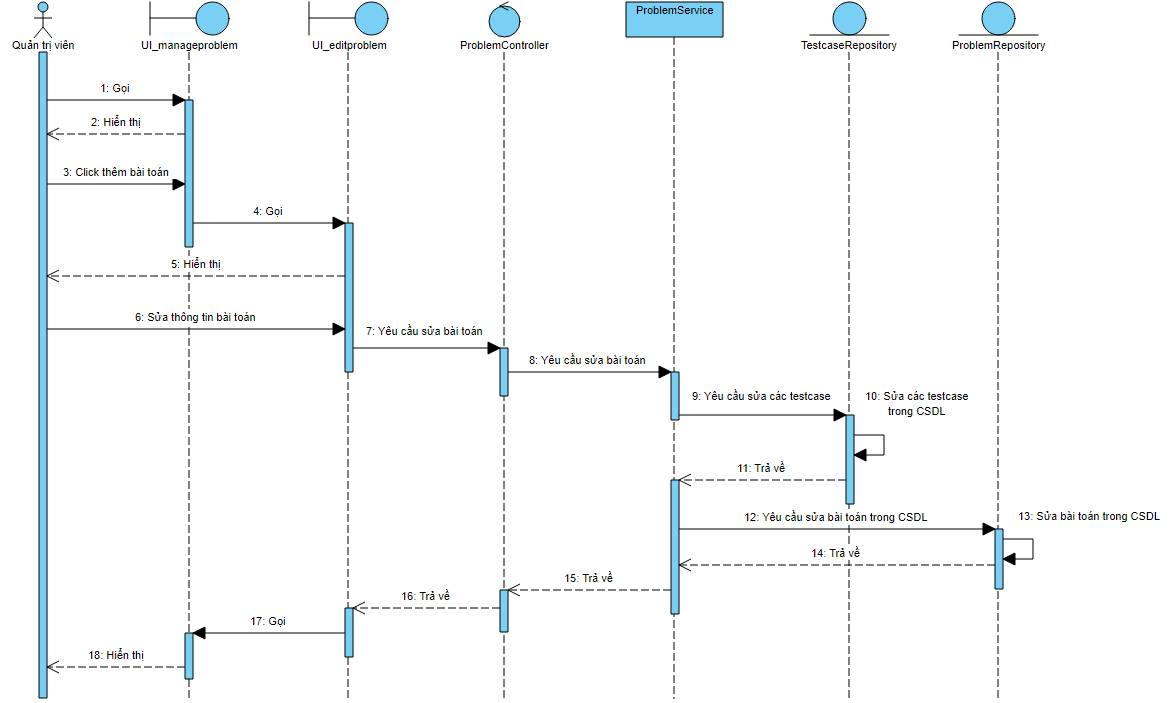
Hình . Usecase chi tiết module sửa bài toán

#### Kịch bản

Bảng . Kịch bản module sửa bài toán

|  |  |
| --- | --- |
| Tên usecase | Sửa bài toán |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống |
| Hậu điều kiện | Thêm bài toán thành công |
| Kịch bản | 1. Quản trị viên đăng nhập vào hệ thống  2. Tại giao diện trang chủ của quản trị viên, trên thanh điều hướng, quản trị viên chọn quản lý bài toán.  3. Trang quản lý bài toán hiện ra với nút thêm bài toán và danh sách các bài toán. Mỗi bài toán bao gồm: mã bài toán, tên bài toán, trạng thái bài toán, mức độ của bài toán, nút sửa và nút xóa.  4. Quản trị viên chọn một bài toán muốn sửa  5. Giao diện trang sửa bài toán hiện ra với ô sửa tên bài toán, sửa mức độ của bài toán, sửa danh mục, ô sửa câu hỏi, ràng buộc đầu vào, ràng buộc đầu ra, nút cancel, nút submit, ô nhập testcase, nút thêm testcase. Danh sách các testcase. Mỗi testcase bao gồm thông tin testcase: input, output, và nút xóa.  6. Quản trị viên nhập thông tin muốn sửa của tên bài toán, mức độ, danh mục, câu hỏi, ràng buộc đầu vào, ràng buộc đầu ra, thêm, xóa testcase  7. Quản trị viên click nút submit  9. Hệ thống lưu bài toán vào CSDL và thông báo thành công sau đó chuyển hướng về trang quản lý bài toán |
| Ngoại lệ | Không có ngoại lệ |

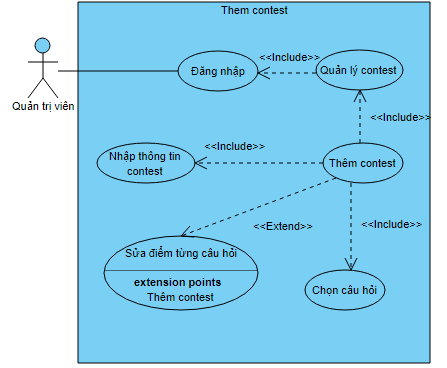
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module sửa bài toán

### Module thêm contest

#### Usecase chi tiết



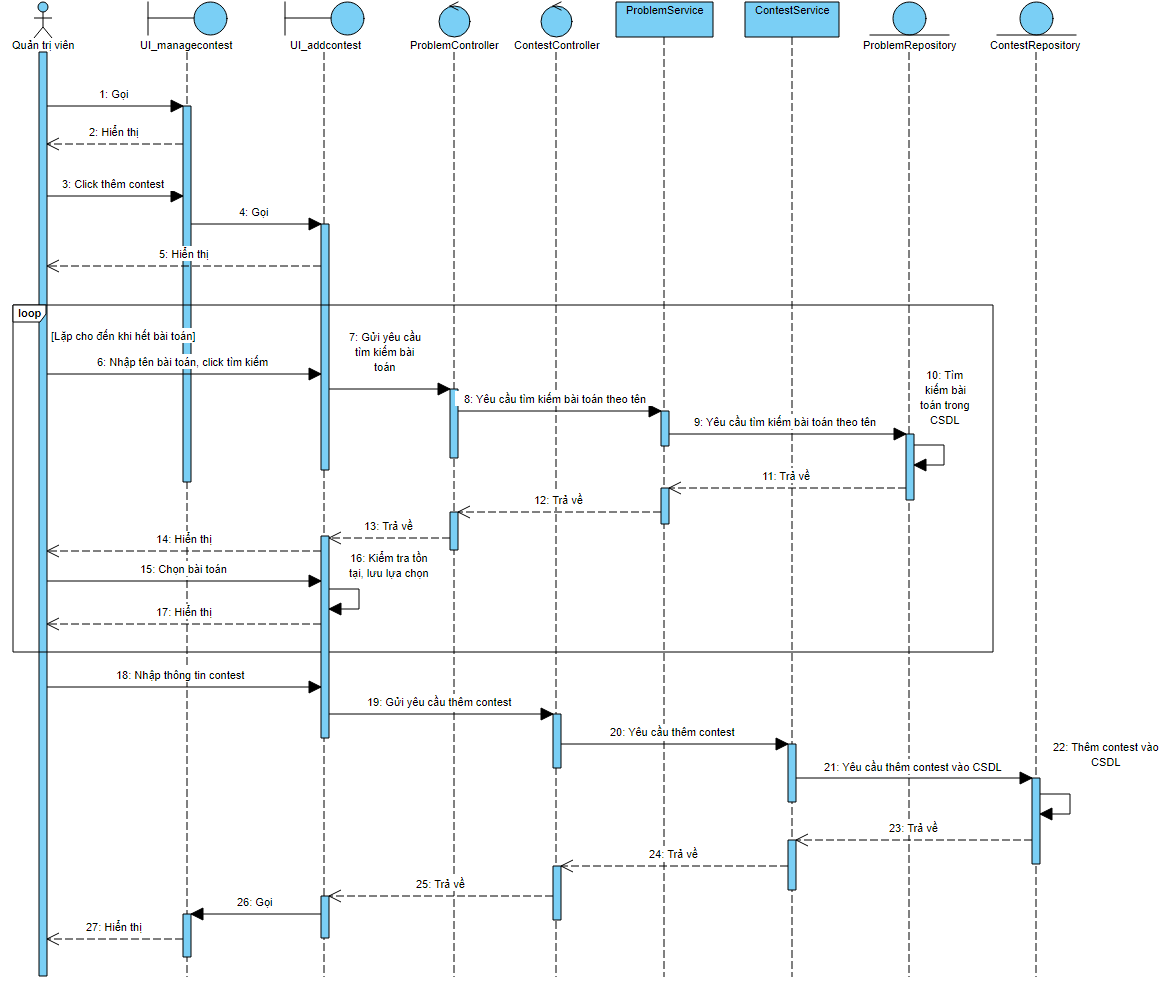
Hình . Usecase chi tiết module thêm contest

#### Kịch bản

Bảng . Kịch bản module thêm contest

|  |  |
| --- | --- |
| Tên usecase | Thêm contest |
| Actor | Quản trị viên |
| Tiền điều kiện | Quản trị viên đã đăng nhập vào hệ thống |
| Hậu điều kiện | Thêm bài toán thành công |
|  | 1. Quản trị viên đăng nhập vào hệ thống  2. Tại giao diện trang chủ của quản trị viên, trên thanh điều hướng, quản trị viên chọn quản contest  3. Trang quản lý contest hiện ra với thông tin của các contest bao gồm: mã contest, tên contest, thời gian bắt đầu, thời gian làm contest, số lượng người giới hạn, trạng thái contest, thời gian tạo.  4. Quản trị viên chọn thê contest  5. Giao diện trang thêm contest hiên ra với các ô nhập tên contest, số lượng người giới hạn, mô tả về contest, chọn thời gian bắt đầu, thời lượng contest, ô tìm kiếm bài toán, textbox xem trước bài toán, nút chọn bài toán, danh sách bài toán trong contest, nút submit và nút cancel. Mỗi bài toán trong danh sách gồm: tên, ô nhập điểm (mặc định để 1) và nút xóa.  6. Quản trị viên nhập thông tin contest bao gồm: tên, người giới hạn, mô tả, chọn thời gian bắt đầu, thời lượng. Quản trị viên lặp lại các bước 6a đến 6g cho đến khi hết bài toán rồi ấn submit:  6a Quản trị viên nhập từ khóa vào ô tìm kiếm bài toán  6b Hệ thống hiển thị danh sách bài toán khớp với tên vừa nhập  6c Quản trị viên chọn một bài toán trong danh sách  6d Hệ thống hiển thị câu hỏi của bài toán lên textbox xem trước.  6e Quản trị viên hệ thống nhấn select bài toán  6f Hệ thống cập nhật lại danh sách bài toán trong contest  6g Quản trị viên nhập điểm của bài toán  7. Hệ thống lưu contest vào CSQL, thông báo thành công và chuyển về trang quản lý contest. |
| Ngoại lệ | **6.1** Tên contest để trống  6.1.1 Hệ thống hiển thị thông báo biểu thị tên contest không được để trống  6.1.2 Quản trị viên nhập tên contest rồi nhấn submit  **6.2** Thời lượng contest lớn hơn 8  6.2.1 Hệ thống hiển thị thông báo biểu thị thời lượng contest quá lớn  6.2.2 Quản trị viên nhập thời lượng nhỏ hơn rồi nhấn submit.  **6.3** Tên contest đã tồn tài trên hệ thống  6.2.1 Hệ thống hiển thị thông báo biểu thị contest đã tồn tại trên hệ thống  6.2.2 Quản trị viên nhập tên contest khác rồi nhấn submit. |

#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module thêm contest  (4)

1. *Chi tiết xem tại phụ lục A trang 94*

## Thiết kế cơ sở dữ liệu

### 2.4.1 Lược đồ cơ sở dữ liệu



Hình . Lược đồ cơ sở dữ liệu

### 2.4.2 Mô tả cơ sở dữ liệu

+ Bảng user:

Bảng . Cơ sở dữ liệu bảng users

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| username | String | Tên đăng nhập của người dùng, sử dụng để đăng nhập |
| userPass | String | Mật khẩu của người dùng |
| userEmail | String | Email của người dùng, email là duy nhất |
| displayName | String | Tên hiển thị của người dùng, được sử dụng để hiển thị trên hệ thống. Tên hiển thị là duy nhất |
| userRole | String | Vai trò của người dùng, có 2 giá trị được cho phép là: admin và student |
| firstName | String | Họ của người dùng, có thể để trống |
| lastName | String | Tên của người dùng, có thể để trống |
| organization | String | Tổ chức của người dùng, có thể là tên trường, tên công ty, … và có thể để trống |
| status | String | Trạng thái tài khoản, có thể nhận 2 giá trị: verified, not\_verified |
| activationCode | Number | Mã kích hoạt tài khoản của người dùng |
| score | Number | Điểm của người dùng |
| avatar | ObjectId | Ảnh đại diện của người dùng, là khóa ngoại liên kết 1-1 với bảng Upload |

+ Bảng upload

Bảng . Cơ sở dữ liệu bảng upload

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| publicId | String | Id công khai của ảnh được upload lên cloud |
| publicUrl | String | Đường dẫn công khai của ảnh được upload lên cloud |

Bảng Category

Bảng . Cơ sở dữ liệu bảng category

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| categoryName | String | Tên danh mục |
| categoryLogo | ObjectId | Logo của danh mục, khóa ngoại được liên kết 1-1 với bảng Upload |
| status | String | Trạng thái của danh mục, nhận 2 giá trị: active, inactive |

+ Bảng Comment

Bảng . Cơ sở dữ liệu bảng comment

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| problem | ObjectId | Id của bài toán được bình luận, là khóa ngoại được liên kết n-1 với bảng Problem |
| user | ObjectId | Id của người bình luận, là khóa ngoại được liên kết n-1 với bảng User |
| content | String | Nội dung của bình luận |

+ Bảng Reaction

Bảng . Cơ sở dữ liệu bảng reaction

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| target | ObjectId | Id của đối tượng được reaction, có thể là bình luận hoặc reply bình luận |
| user | ObjectId | Id của người reaction, là khóa ngoại được liên kết n-1 với bảng User |

+ Bảng Contest

Bảng . Cơ sở dữ liệu bảng contest

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| name | String | Tên của contest |
| description | String | Mô tả contest |
| beginAt | Timestamp | Thời gian bắt đầu contest |
| duration | Number | Thời lượng của contest |
| user | ObjectId[] | Mảng lưu id người tham gia contest, được liên kết 1-n với bảng User |
| questions | ObjectId[] | Mảng lưu id của bài toán trong contest, được liên kết 1-n với bảng Problem |

+ Bảng Problem

Bảng . Cơ sở dữ liệu bảng problem

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| problemName | String | Tên của bài toán |
| problemCode | String | Mã dạng uuid của bài toán, có thể sử dụng thay cho id, mã bài toán là duy nhất |
| problemLevel | String | Mức độ của bài toán, có thể nhận 2 giá trị: easy hoặc medium |
| problemCategory | ObjectId | Id của danh mục mà bài toán thuộc về. Là khóa ngoại được liên kết n-1 với bảng Category |
| problemQuestion | String | Câu hỏi của bài toán |
| expectedInput | String | Ràng buộc dữ liệu đầu vào |
| expectedOutput | String | Ràng buộc dữ liệu đầu ra |
| problemCases | ObjectId[] | Mảng lưu id của các testcases của bài toán, là khóa ngoại được liên kết 1-n với bảng Testcase |
| status | String | Trạng thái của bài toán, có thể nhận giá trị active hoặc inactive |
| score | Number | Điểm dành được khi hoàn thành bài toán |

+ Bảng Reply

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| comment | ObjectId | Id của bình luận được trả lời, là khóa ngoại được liên kết n-1 với bảng Comment |
| user | ObjectId | Id của người trả lời bình luận, là khóa ngoại n-1 được liên kết với bảng User |
| content | String | Nội dung trả lời |

+ Bảng ContestHistory

Bảng . Cơ sở dữ liệu bảng ContestHistory

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| contest | ObjectId | Id của contest mà lịch sử làm contest thuộc về. Là khóa ngoại được liên kết n-1 với bảng Contest |
| user | ObjectID | Id của người làm contest, là khóa ngoại được liên kết n-1 với bảng User |
| status | String | Trạng thái làm contest |
| history | ObjectId[] | Mảng Id của submission thuộc về lịch sử làm contest, là khóa ngoại được liên kết 1-n với bảng Submission |

+ Bảng Testcase

Bảng . Cơ sở dữ liệu bảng testcase

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| input | String | Đầu vào của testcase |
| output | String | Đầu ra của testcase |

+ Bảng Submission

Bảng . Cơ sở dữ liệu bảng submission

|  |  |  |
| --- | --- | --- |
| **Thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| \_id | ObjectId | Một dãy gồm 24 kí tự hexa được sinh tự động, dãy này đảm bảo luôn là duy nhất. Là mã định danh người dùng đồng thời là khóa chính của bảng. |
| problem | ObjectId | Id của bài toán mà submission thuộc về. Là khóa ngoại liên kết n-1 với bảng Problem |
| user | ObjectId | Id của người tạo ra submission, là khóa ngoại liên kết n-1 với bảng User |
| userCode | String | Lưu lại code mà người dùng đã submit |
| executeTime | Number | Thời gian thực thi test lớn nhất |
| passPercent | Number | Phần trăm testcase đúng so với tổng số testcase của bài toán |
| detail | String | Chi tiết của submission, có giá trị khi biên dịch lỗi |
| status | String | Trạng thái của submission |

## Tổng kết

Chương 2 của đồ án đã trình bày phân tích thiết kế tổng quan của toàn bộ hệ thống bao gồm:

* Giới thiệu tổng quan về hệ thống
* Usecase tổng quan của hệ thống
* Biểu đồ lớp thực thể và thiết kế cơ sở dữ liệu cho hệ thống.

Chương 2 cũng tập trung vào phân tích thiết kế chi tiết từng module chức năng có mặt trong hệ thống bao gồm:

* Usecase chi tiết từng chức năng
* Kịch bản chuẩn cho module
* Biểu đồ tuần tự của module

Từ đó cung cấp cái nhìn cụ thể hơn về hệ thống và các chức năng trong hệ thống. Trên cơ sở phân tích thiết kế đó, trong chương tiếp theo đồ án sẽ trình bày về cách cài đặt, triển khai hệ thống và kết quả thu được khi triển khai hệ thống.

# CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

Trong chương này, đồ án sẽ trình bày về cách cài đặt và triển khai hệ thống và kết quả thu được sau khi cài đặt.

## Yêu cầu hệ thống

* Hệ điều hành: window, linux
* NodeJs: version 14 trở lên
* Cài trình quản lý gói yarn (hoặc sử dụng npm được tích hợp sẵn khi cài nodejs)
* Cài đặt các trình biên dịch: c++, java, python,...
* Git: Quản lý source code

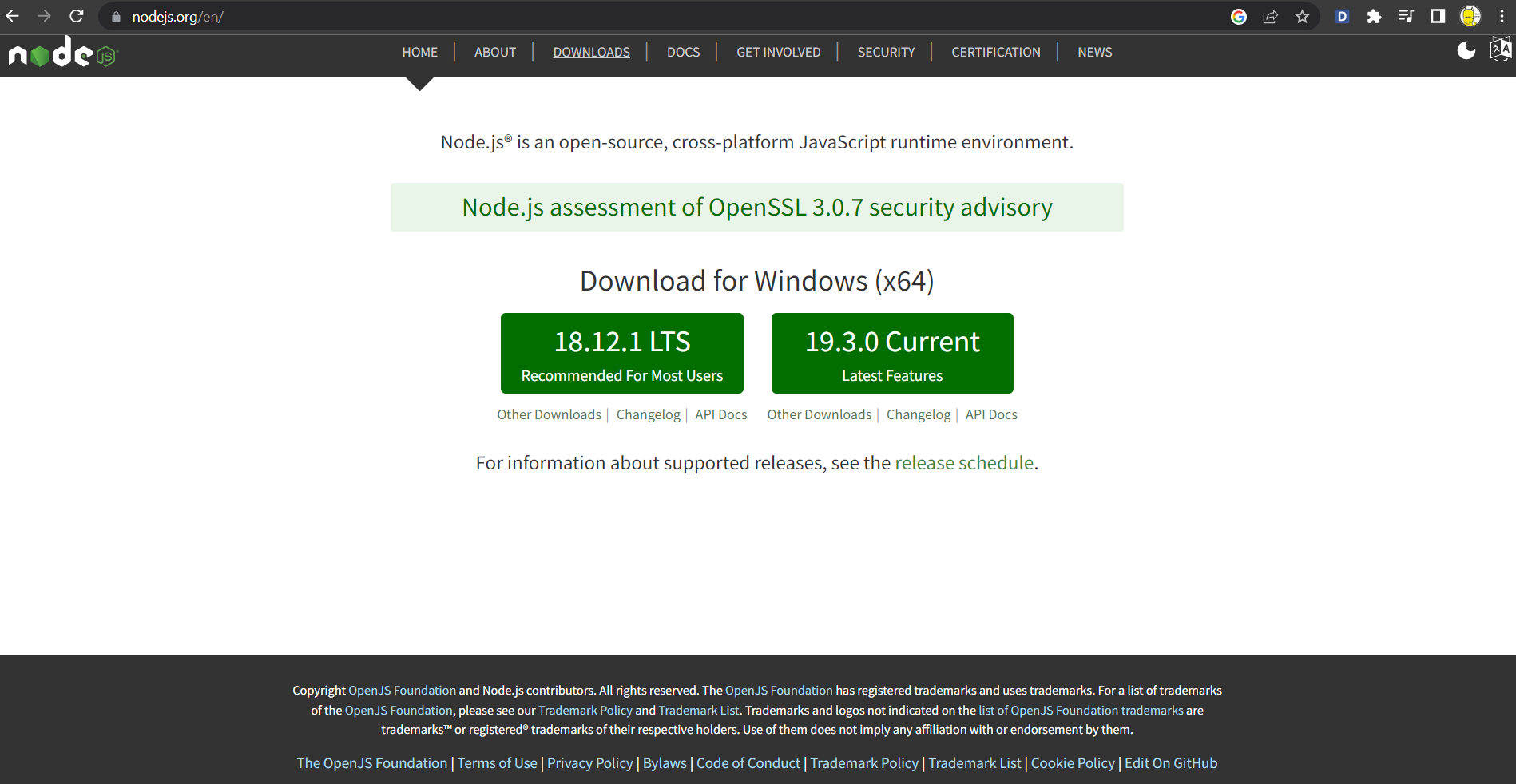
## Một số công cụ, thư viện hỗ trợ

* Visual studio code
* Bootstrap 5
* Mongo Atlas
* ReactJs
* Express
* Socket.io
* Axios

## Cài đặt

### 3.3.1 Cài đặt Nodejs

Bước 1: Tải NodeJS



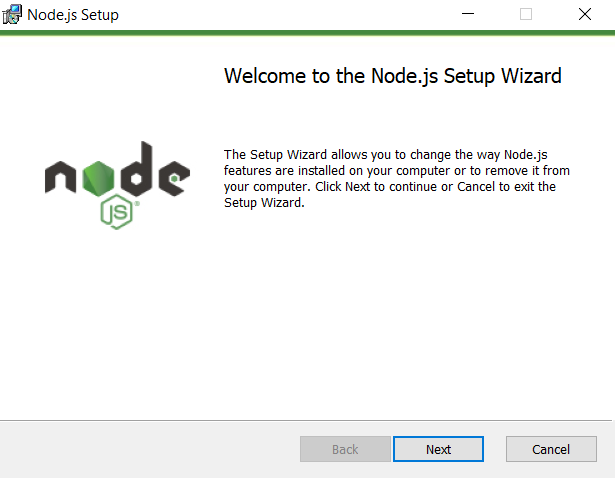
Hình . Hướng dẫn tải Nodejs

Để tải nodejs, hãy truy cập vào trang <https://nodejs.org>. Màn hình trang chủ sẽ hiện lên. Nodejs sẽ tự phát hiện ra hệ điều hành để đưa ra link tải phù hợp đối với từng thiết bị. Việc tiếp theo chỉ cần click vào button màu xanh lục để tải phiên bản mong muốn.

Nodejs gợi ý 2 phiên bản. Current là phiên bản mới nhất, LTS là phiên bản được khuyên dùng.

Bước 2: Cài đặt

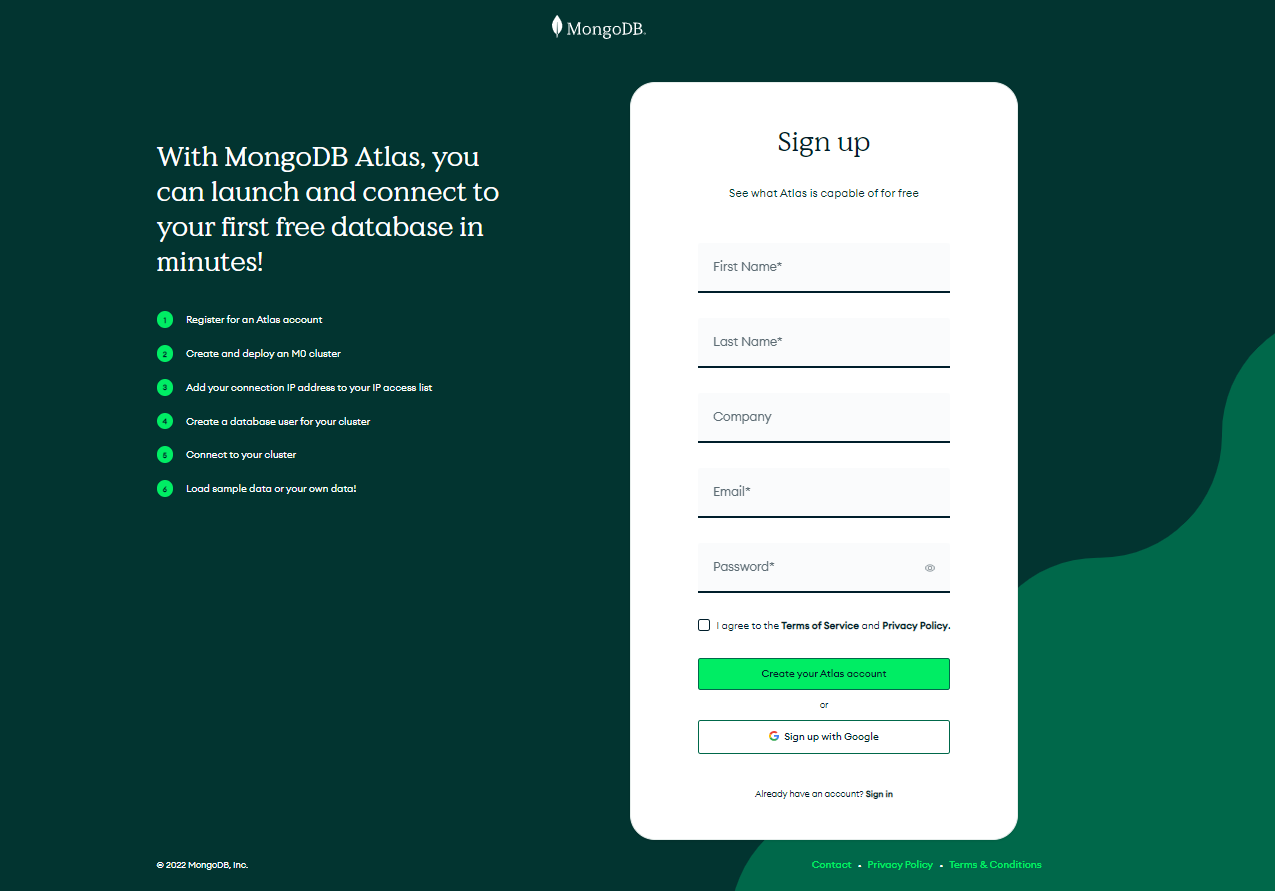
Sau khi đã tải về, hãy tiến hành cài đặt Nodejs (có thể để các options mặc định trong khi cài đặt).



Hình . Màn hình cài đặt Nodejs

### 3.3.2 Đăng ký Mongo Atlas

Bước 1: Truy cập trang: https://www.mongodb.com/cloud/atlas/register



Hình . Giao diện trang đăng ký Mongo Atlas

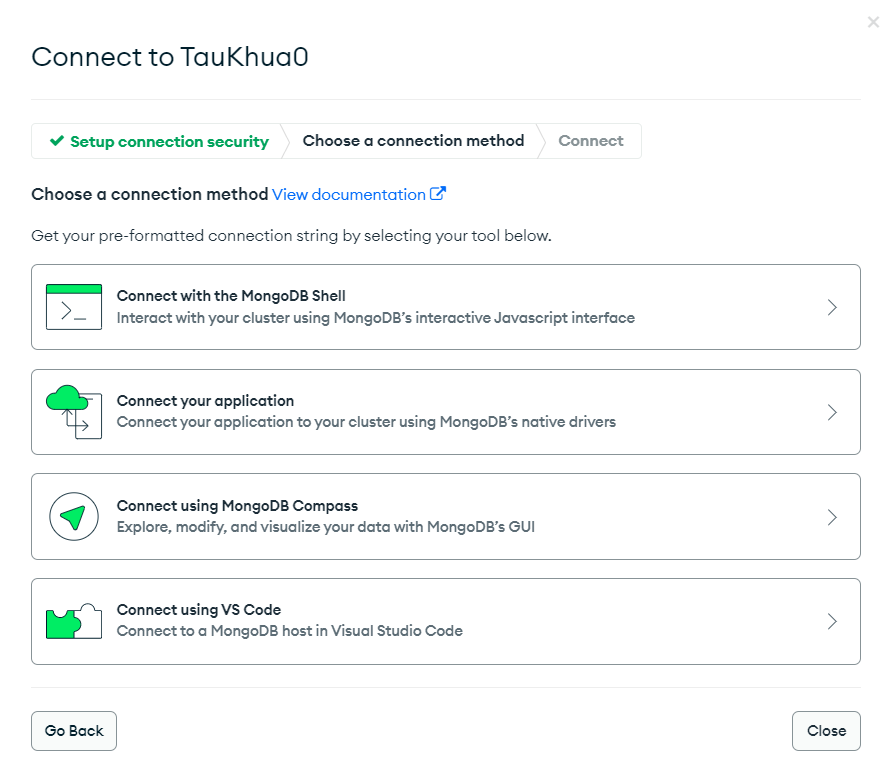
Bước 2: Điền đầy đủ thông tin rồi click create your Atlas Account, hoặc đơn giản hơn click nút sign up with Google để đăng ký bằng tài khoản google.

Bước 3: Giao diện quản lý cloud hiện lên, để tạo một database mới, click vào nút create ở tab database



Hình . Giao diện quản lý trang Mongo Atlas

Bước 4: Để lấy link kết nối với Mongo Atlas, tại giao diện quản lý cloud, ở tab database, click connect

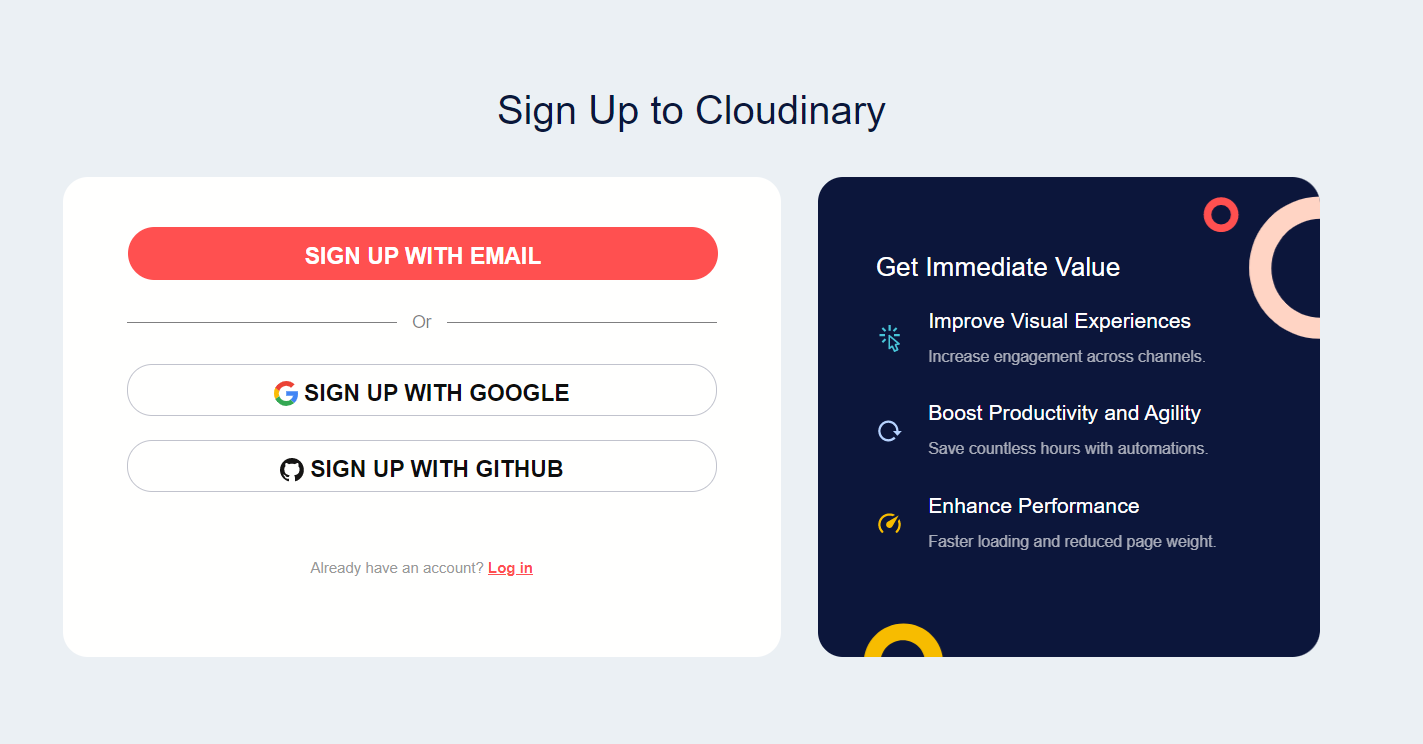


Hình . Giao diện trang hướng dẫn kết nối

Bước 5: Giao diện hướng dẫn kết nối hiện lên, chọn phương thức kết nối và làm theo hướng dẫn.

### 3.3.3 Đăng ký Cloudinary

Bước 1: Truy cập trang: https://cloudinary.com/users/register\_free



Hình . Giao diện đăng ký cloudinary

Bước 2: Giao diện trang đăng ký cloudinary hiện ra, click chọn sign up with google

Bước 3: Lấy các key kết nối với cloudinary. Tại giao diện trang dashboard, cloudinary cung cấp api key và secret key để kết nối tới cloudinary. Việc còn lại chỉ cần copy đúng key và sử dụng.

### 3.3.4 Cài đặt mã nguồn phía backend

Bước 1: Mở terminal trên visual studio code, hoặc cmd trên windows.

Bước 2: Chuyển hướng đến thư mục chứa mã nguồn:

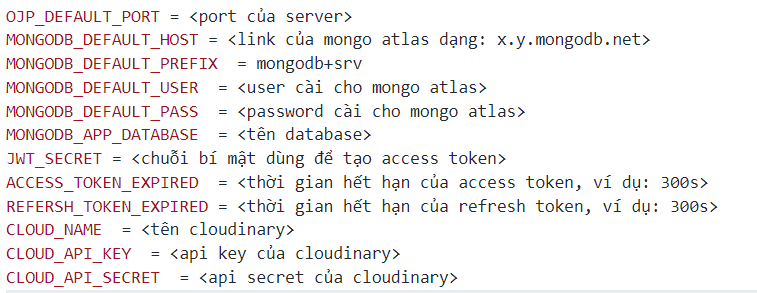
Sử dụng câu lệnh: cd <directory path>

Bước 3: Cài đặt toàn bộ thư viện phụ thuộc của dự án.

Sử dụng câu lệnh: yarn install hoặc npm install (hoặc thêm tên package muốn cài sau chữ install nếu muốn cài một thư viện cụ thể)

Bước 4: Cài đặt môi trường:

Chuyển hướng tới thư mục src/configs trong dự án sau đó tạo mới 1 file có tên .env, sau đó viết mã cài đặt môi trường theo form:



Hình . Cấu trúc môi trường của dự án backend

Bước 5: Vào thư mục src/libs/sockets tạo mới thư mục rỗng có tên temps ( đây là thư mục tạm dùng để lưu trữ mã nguồn của người dùng)

Bước 6: Chạy ứng dụng

Sử dụng câu lệnh: yarn start:dev hoặc npm run start:dev để chạy ứng dụng backend

Lưu ý: Khi chạy dự án trên windows có thể gặp lỗi:

TypeError: app.use() requires a middleware function

Để fix lỗi này, mở file package.json và sửa version của glob thành 7.2.0 sau đó install lại thư viện của dự án và chạy lại

### 3.3.5 Cài đặt mã nguồn phía frontend

Bước 1: Mở terminal của visual studio code hoặc cmd trên windows

Bước 2: Chuyển đến thư mục chứa mã nguồn:

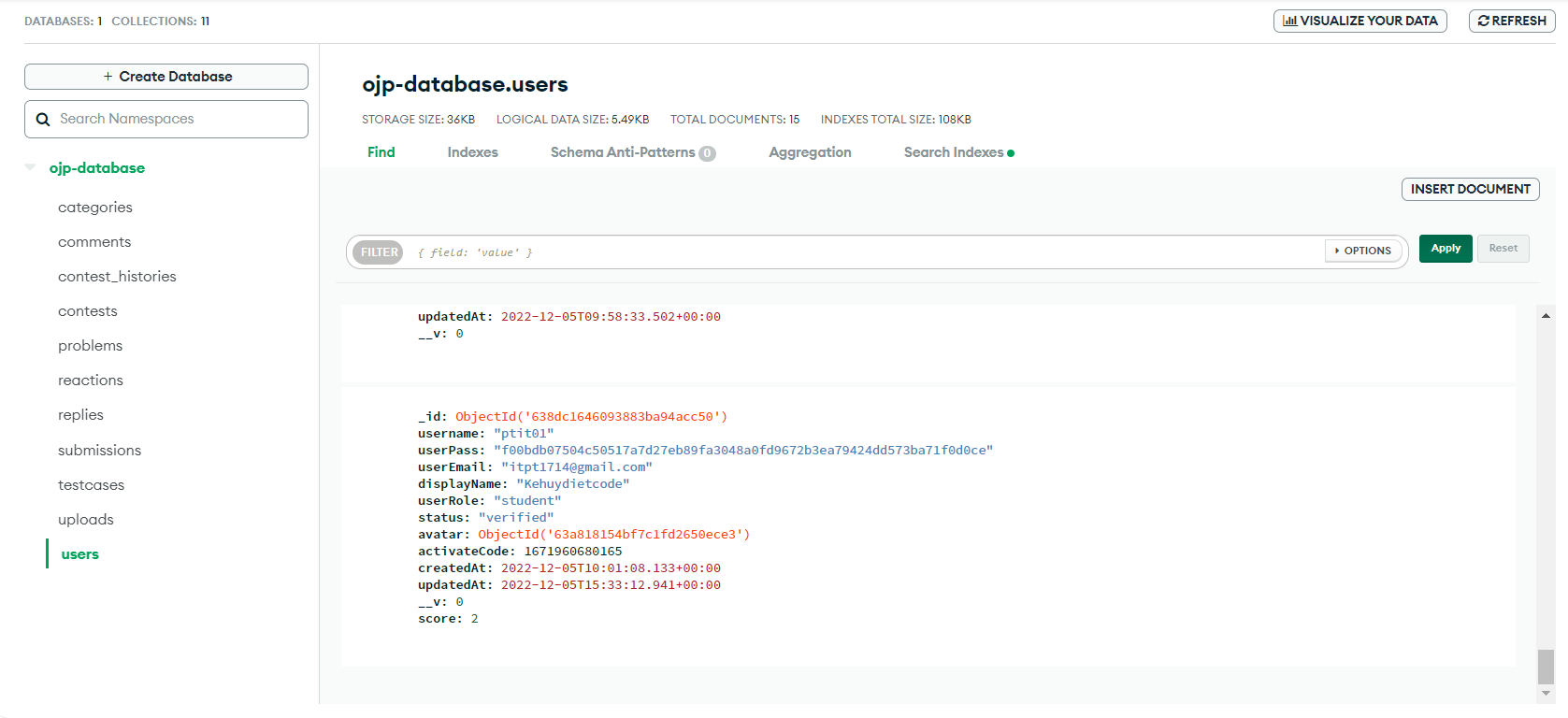
Sử dụng lệnh: cd <directory path>

Bước 3: Cài đặt toàn bộ thư viện bằng lệnh: yarn install hoặc npm install

Bước 4: Chạy ứng dụng bằng lệnh: yarn start hoặc npm run start

## Kết quả cài đặt

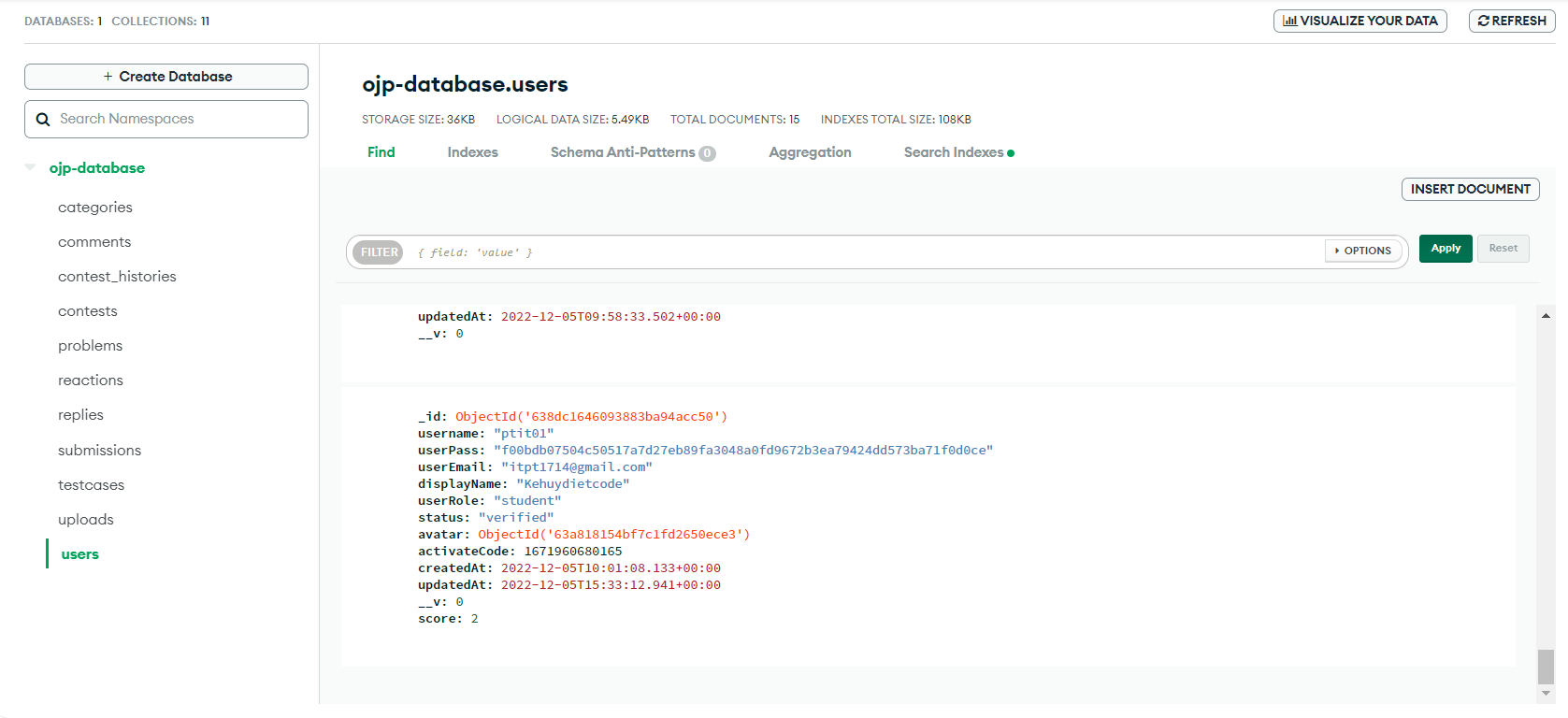
### Kết quả cài đặt database



Hình . Kết quả cài đặt database bằng Mongo Atlas

Sau khi cài đặt database, có 11 bảng được tự động sinh ra bao gồm: categories, comments, contest\_histories, contests, problems, reactions, replies, submissions, testcases, uploads, users. Tất cả cả bảng đều có 3 trường thông tin được sinh ra tự động gồm createdAt là thời gian tạo, updatedAt là thời gian cập nhật và \_\_v là phiên bản của bản ghi. \_\_v sẽ được tăng thủ công khi cập nhật một bản ghi.

Bảng users

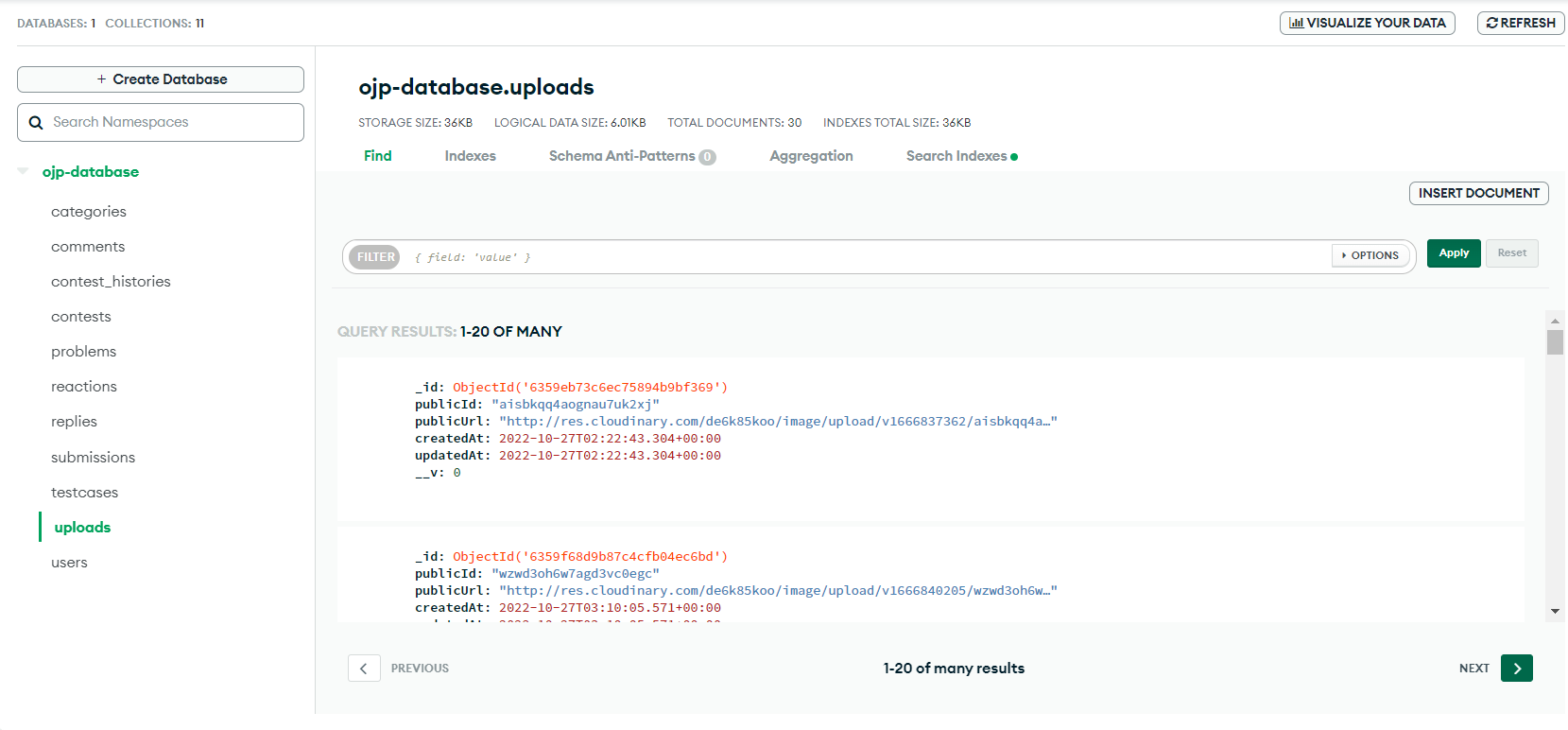


Hình . Kết quả cài đặt bảng users

Nhìn vào bảng dữ liệu trên, ta có thể thấy bảng users bao gồm các trường thông tin: \_id là mã định danh của user, username là tên đăng nhập của user, userPass là mật khẩu của user dưới dạng mã hóa, userEmail là email của user, displayName là tên hiển thị của user, userRole là vai trò của user, status là trạng thái của user, avatar là hình đại diện của user. activationCode là verify code của user được lấy từ thời gian hết hạn của verify code, score lưu điểm của user.

Ngoài ra bảng users còn có các trường createdAt, updatedAt và \_\_v. Trường createdAt dùng để lưu thông tin thời gian tạo của bản ghi. Trường updatedAt lưu thông tin thời gian bản ghi cập nhật. Trường \_\_v dùng để lưu thông tin phiên bản của bản ghi. Trường \_\_v sẽ được cập nhật thủ công khi một bản ghi trong cơ sở dữ liệu được cập nhật.

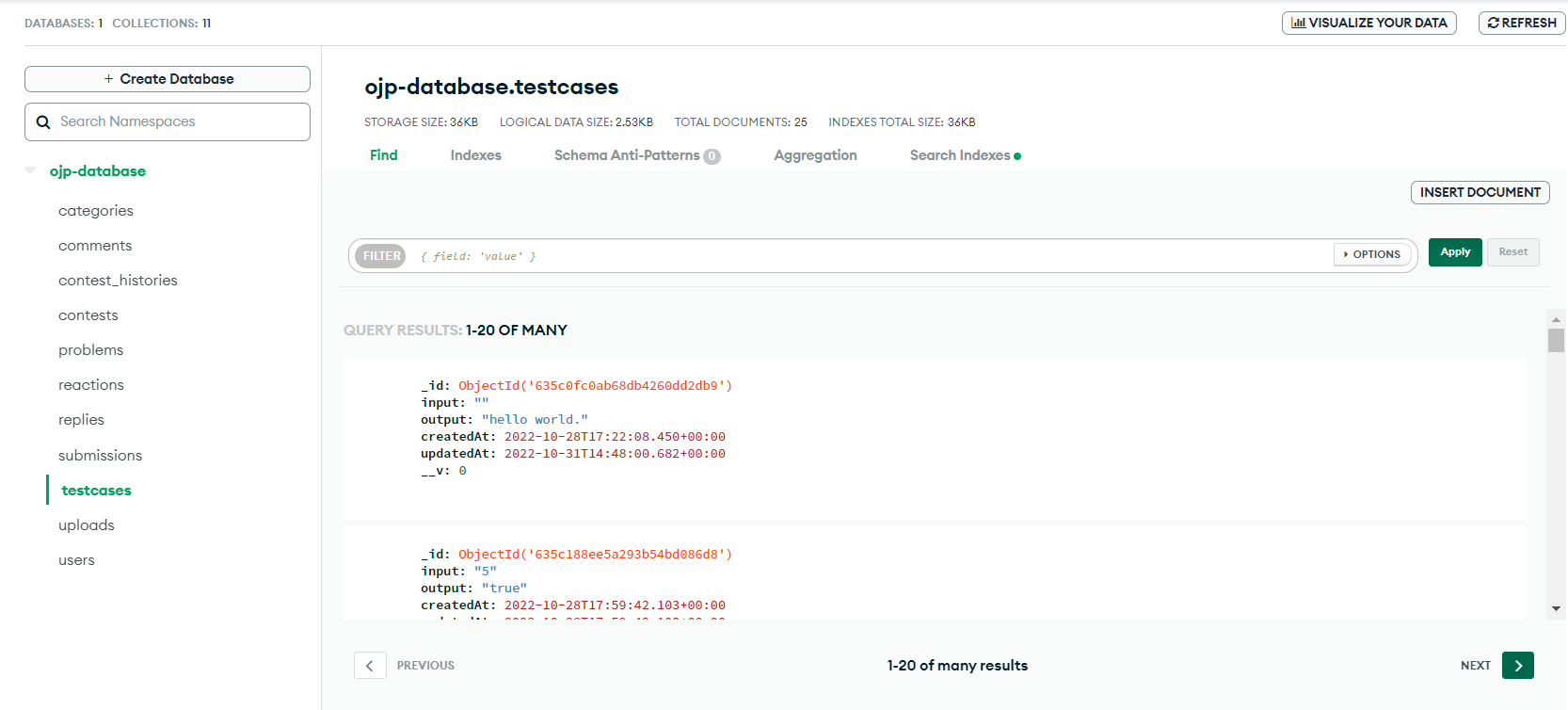
Bảng uploads



Hình . Kết quả cài đặt bảng uploads

Bảng uploads bao gồm các trường thông tin: \_id là mã định danh của upload, publicId là id công khai của ảnh được upload lên cloud, publicUrl là url công khai của ảnh được upload lên cloud. Các trường createdAt, updatedAt và \_\_v được sinh tự động khi tạo mới upload.

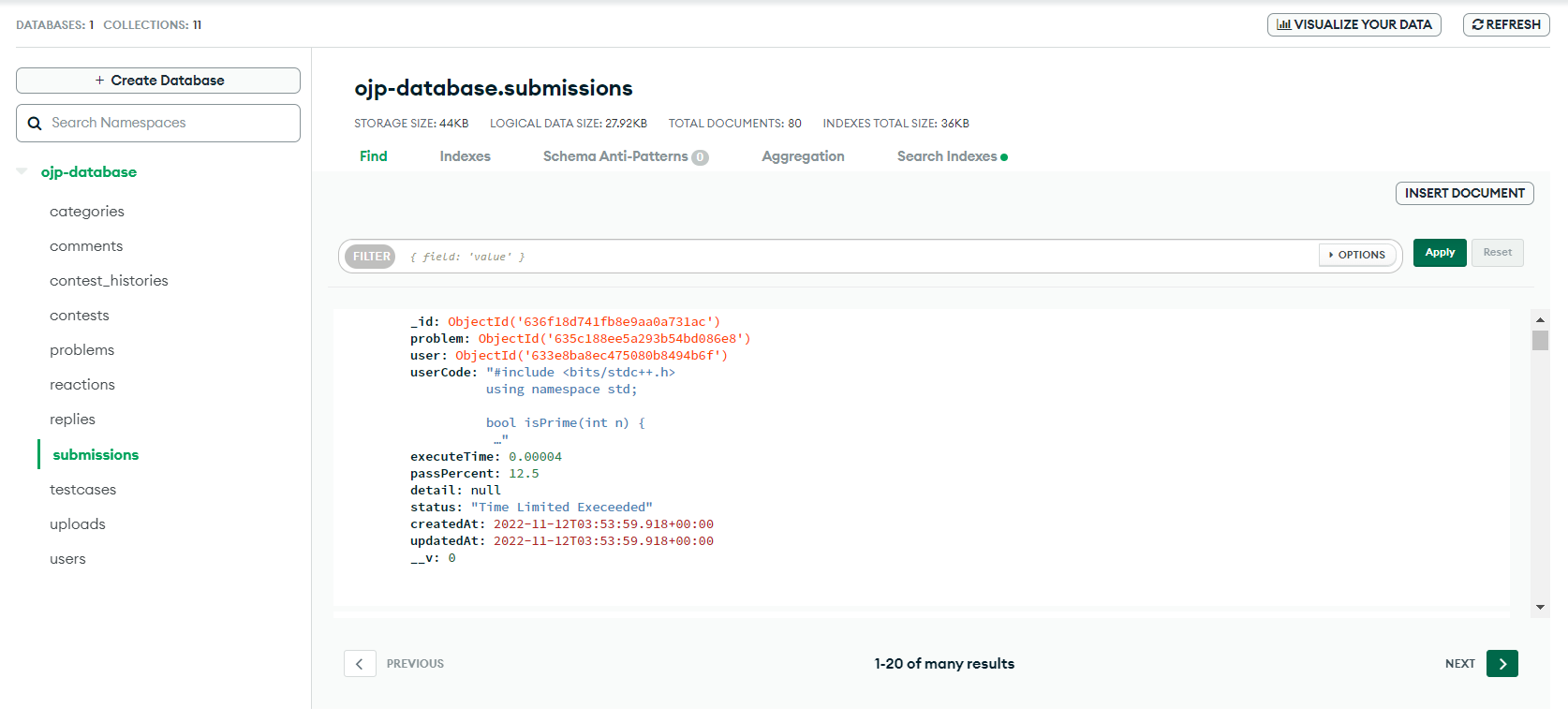
Bảng testcases:



Hình . Kết quả cài đặt bảng testcases

Bảng testcases bao gồm các trường thông tin: \_id là mã định danh của testcase, input là dữ liệu đầu vào của testcase dùng để truyền vào stdin khi chạy chương trình của người dung. Trường output là dữ liệu đầu ra của testcase dùng để so sánh với kết quả đầu ra khi chạy chương trình của người dùng. Ngoài ra, các trường createdAt, updatedAt và \_\_v được sinh ra tự động khi tạo mới bảng testcases

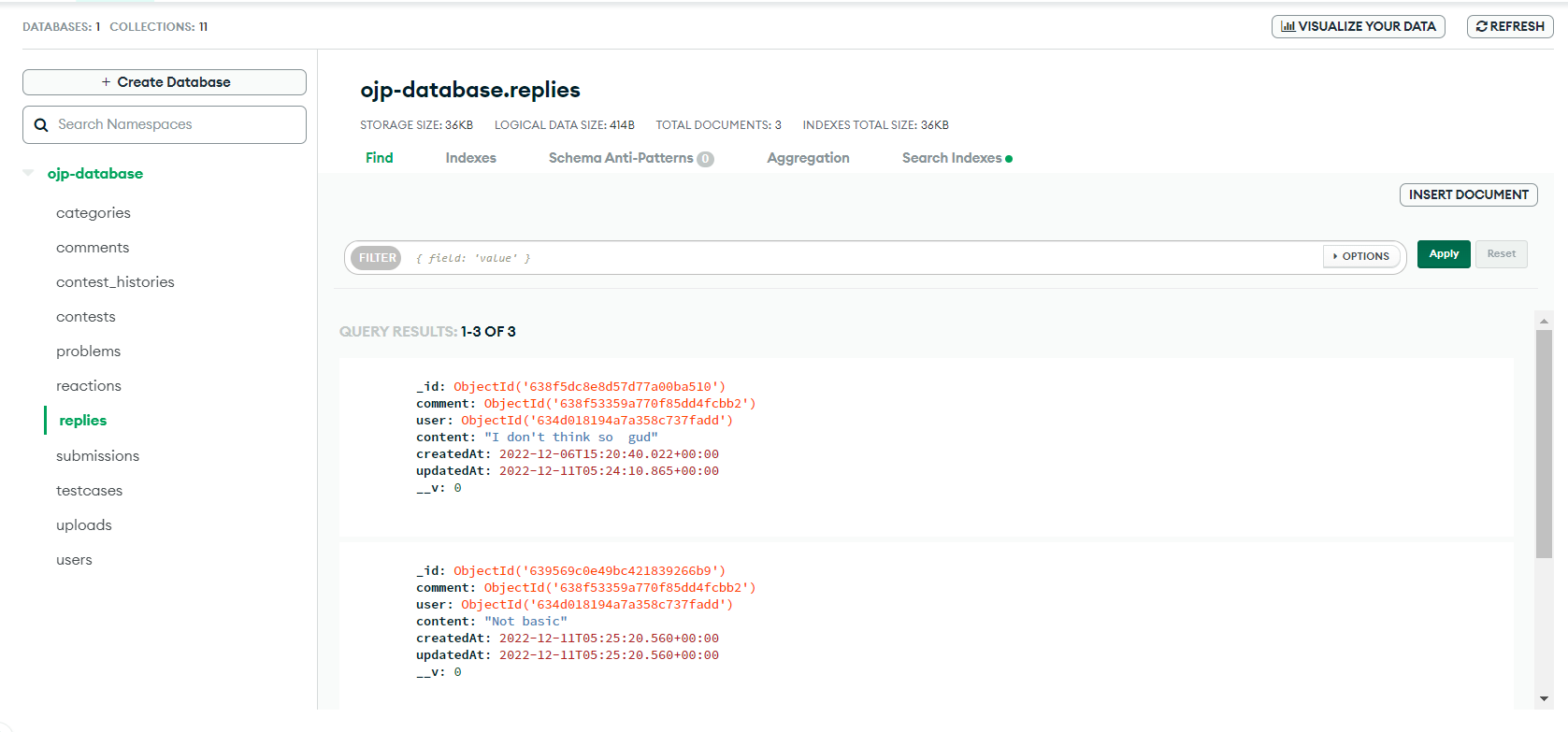
Bảng submissions:



Hình . Kết quả cài đặt bảng submissions

Bảng submissions bao gồm các trường thông tin: \_id là mã định danh của submission, problem là mã định danh bài toán, user là mã định danh của người submit, userCode là mã nguồn của người submit, executeTime là thời gian thực thi mã nguồn. passPercent là phần trăm testcase được chấp nhận, detail lưu thông tin biên dịch lỗi, status là trạng thái của submission. Ngoài ra các trường dateCreated, dateUpdated, \_\_v được cập nhật tự động khi tạo mới submissions

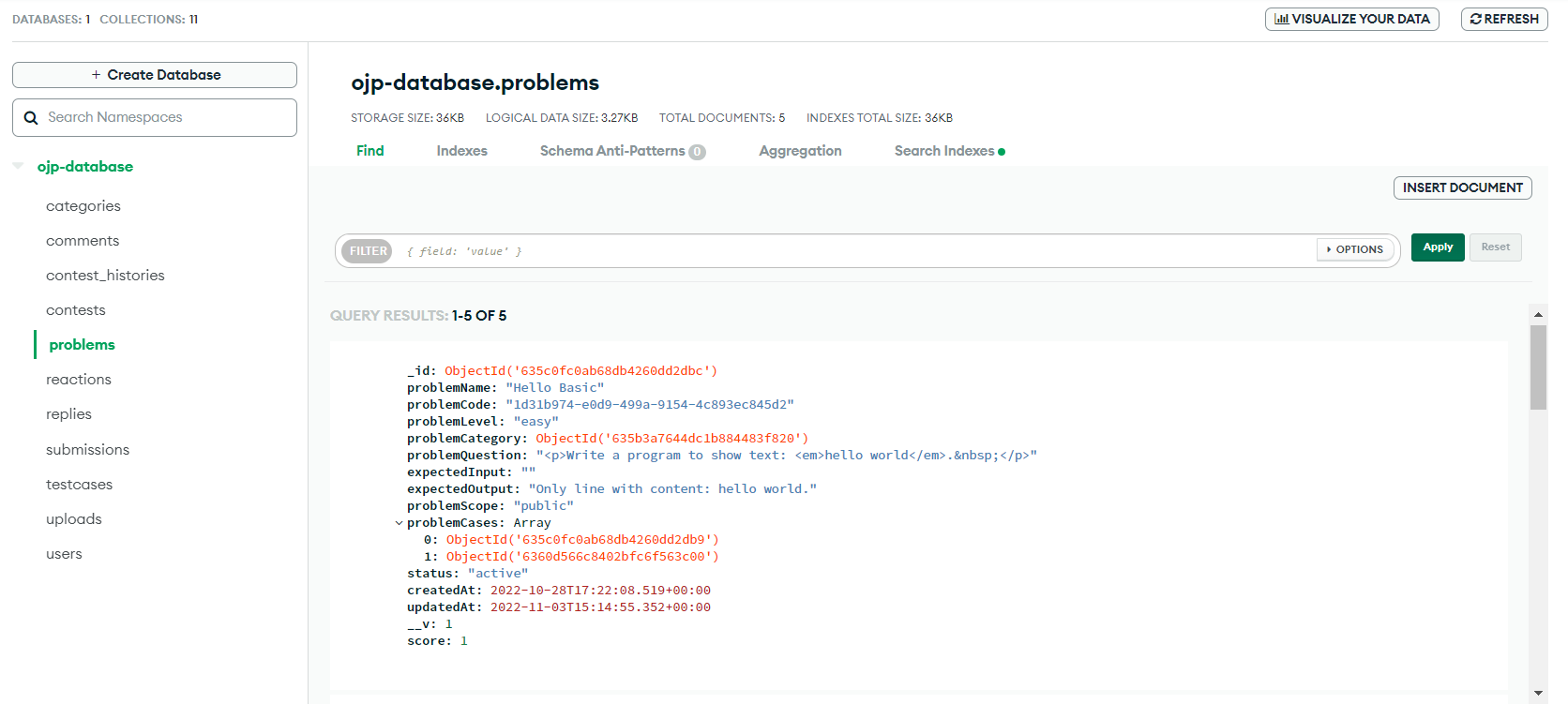
Bảng replies



Hình . Kết quả cài đặt bảng Replies

Bảng replies bao gồm các trường thông tin: \_id là mã định danh của replies, comment là mã định danh của Comment được trả lời, content là nội dung trả lời, user là mã định danh của người trả lời. Ngoài ra, các trường createdAt, updatedAt, \_\_v được sinh ra tự động khi tạo mới replies

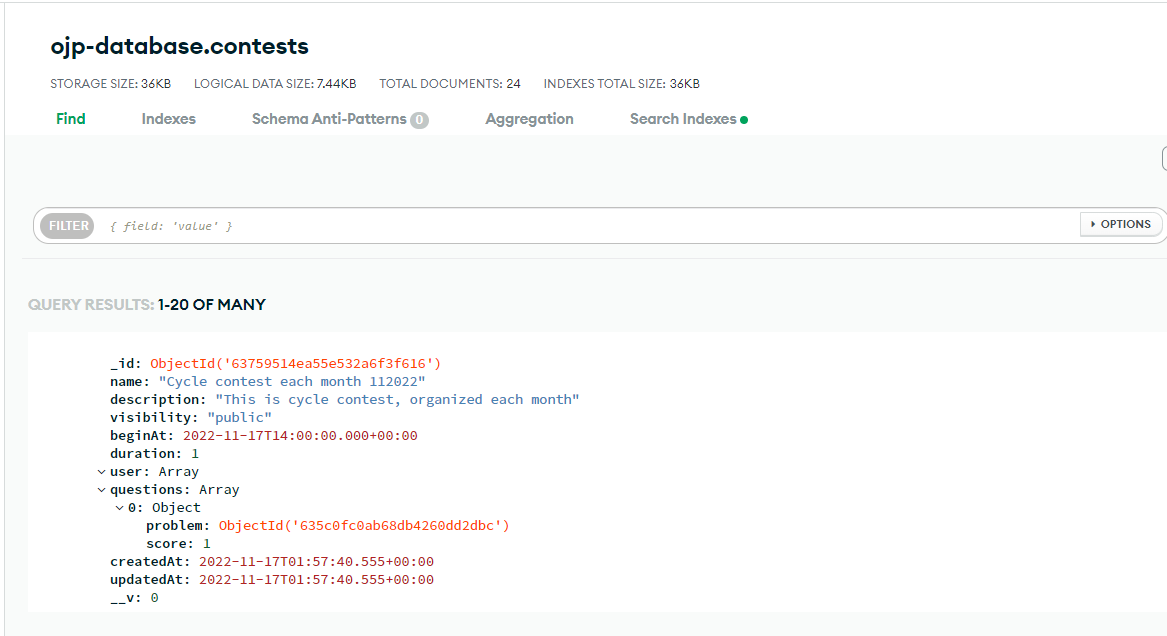
Bảng problems:



Hình . Kết quả cài đặt bảng problems

Bảng problems bao gồm các trường thông tin: \_id là mã định danh của bài toán, problemName là tên của bài toán, problemCode là mã của bài toán dưới dạng uuid, problemLevel là mức độ của bài toán, problemCategory lưu mã định danh của danh mục mà bài toán thuộc về, problemQuestion là câu hỏi của bài toán. expectedInput là ràng buộc đầu vào, expectOutput lưu ràng buộc đầu ra, problemsCases là mảng lưu mã định danh của các testcases, status là trạng thái của bài toán, score là điểm của bài toán. Ngoài ra các trường dateCreated, dateUpdated, \_\_v được tạo tự động khi tạo mới problems

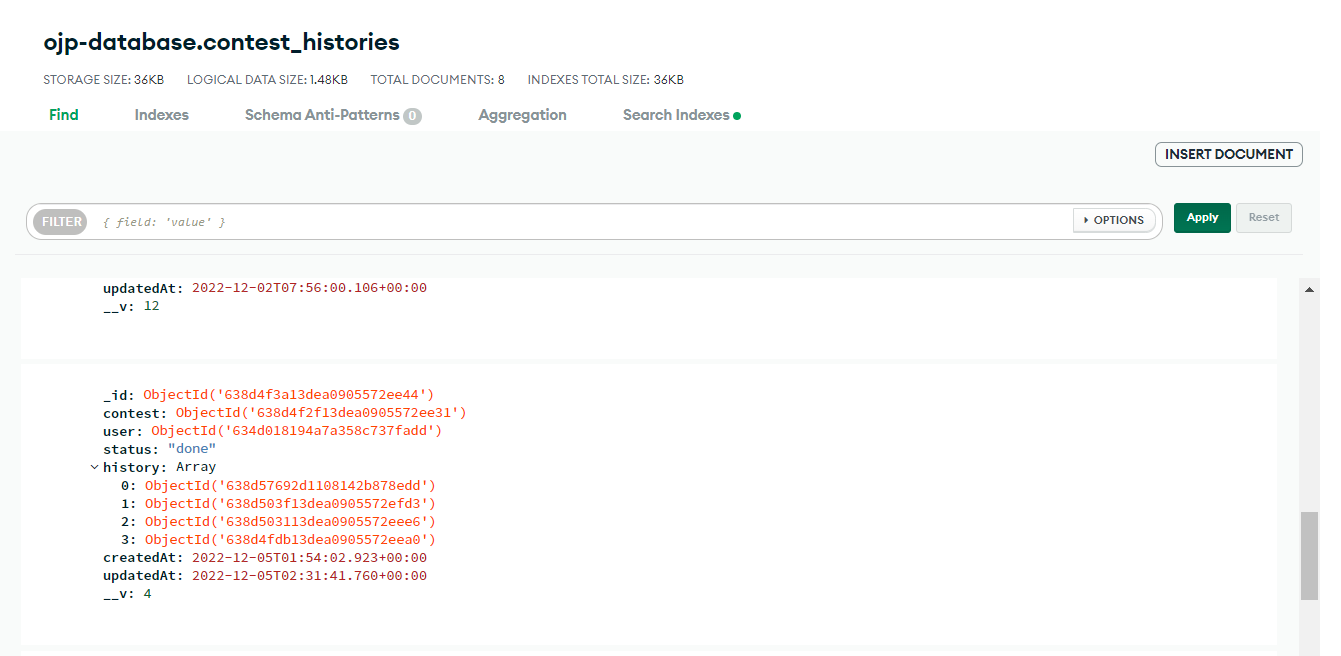
Bảng contests



Hình . Kết quả cài đặt bảng contests

Bảng contests bao gồm các trường thông tin: \_id lưu mã định danh của contest, name lưu tên của contest, description là mô tả của contest, beginAt lưu thời gian bắt đầu của contest, duration lưu thời lượng của contest, user lưu mảng định danh của người tham gia contest, questions là mảng lưu mã định danh của các bài toán trong contest và điểm của bài toán. Ngoài ra các trường dateCreated, dateUpdated và \_\_v được sinh ra tự động khi tạo mới contests

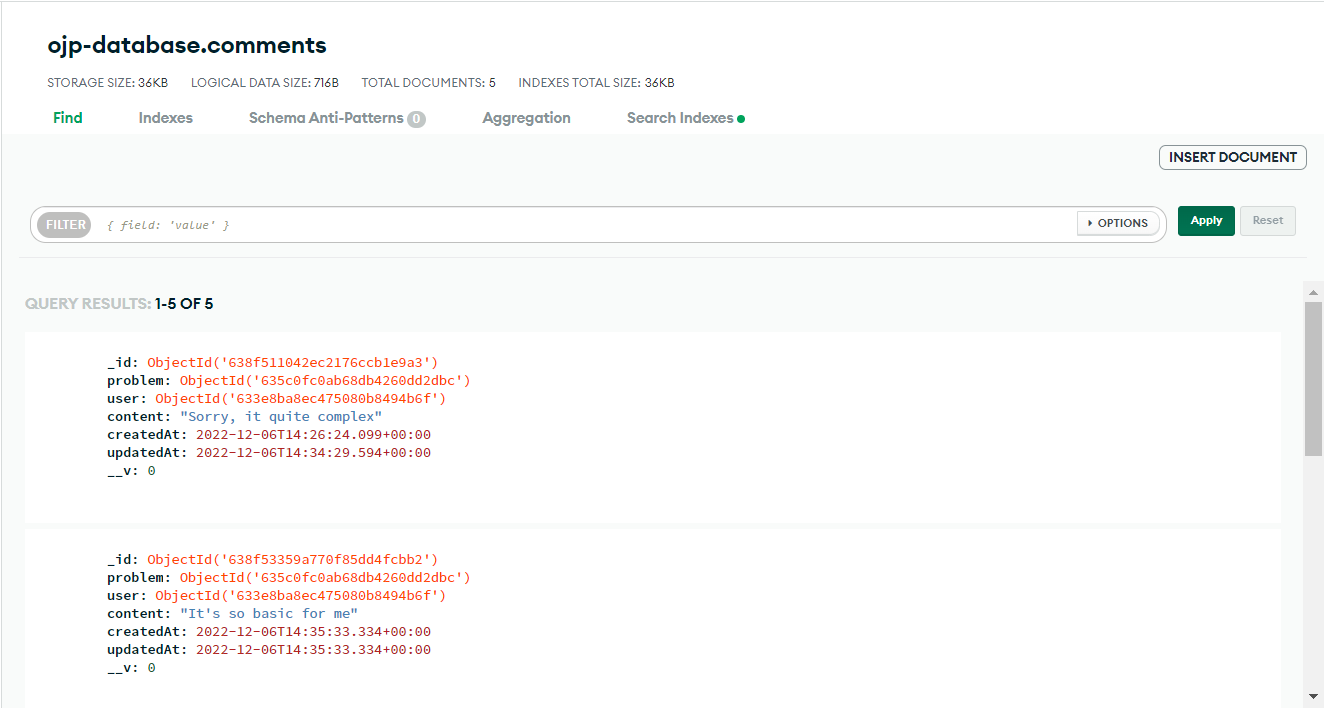
Bảng contest\_histories



Hình . Kết quả cài đặt bảng contest\_histories

Bảng contest\_histories bao gồm các trường thông tin: \_id lưu mã định danh của lịch sử làm contest, contest lưu mã định danh của contest, user lưu mã định danh người làm contest, status lưu trạng thái làm contest của người đó, history là mảng lưu các submission của contest. Ngoài ra, các trường createdAt, updatedAt, \_\_v được sinh ra tự động khi tạo mới contest\_histories

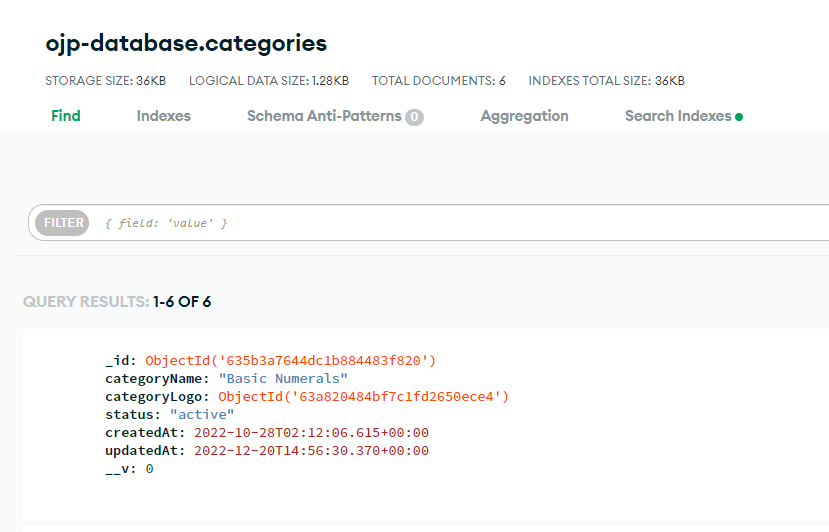
Bảng comments



Hình . Kết quả cài đặt bảng comments

Nhìn vào hình mô tả, ta có thể thấy bảng comments bao gồm các trường thông tin: \_id lưu mã định danh của comment, problem lưu mã định danh của bài toán, user lưu mã định danh của người bình luận, content lưu nội dung bình luận. Ngoài ra các trường dateCreated, dateUpdated, \_\_v được sinh ra tự động khi tạo mới comments

Bảng categories



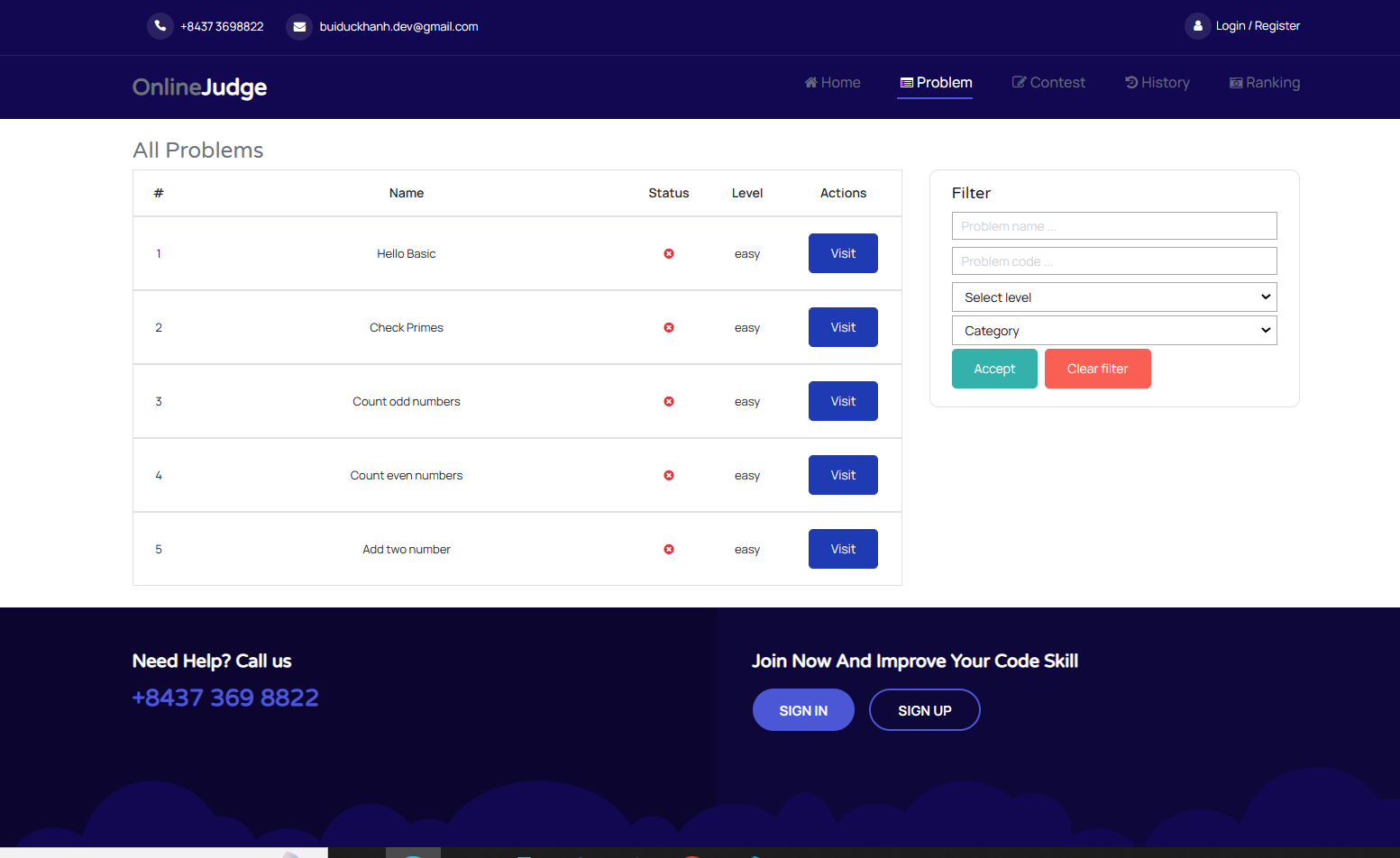
Hình . Kết quả cài đặt bảng categories

Bảng categories bao gồm các trường thông tin: \_id lưu mã định danh của category, categoryName lưu tên category, categoryLogo lưu mã định danh logo của category, status lưu trạng thái category.

### Kết quả cài đặt website

Sau khi chạy dự án, dự án sẽ được chạy trên port 3000 cho frontend và port 8080 cho backend. Truy cập trang <http://localhost:3000> ta sẽ có thể sử dụng website trên máy cục bộ. Giao diện chính của website bao gồm những giao diện sau:

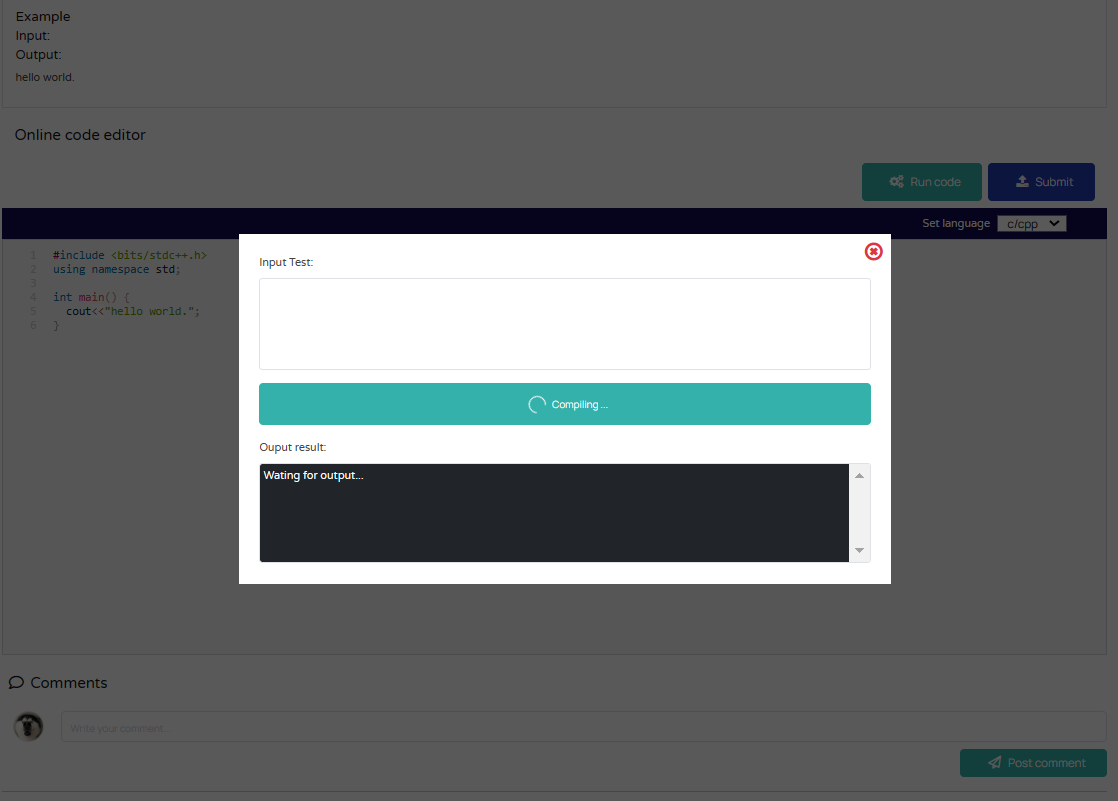
+ Giao diện xem bài toán



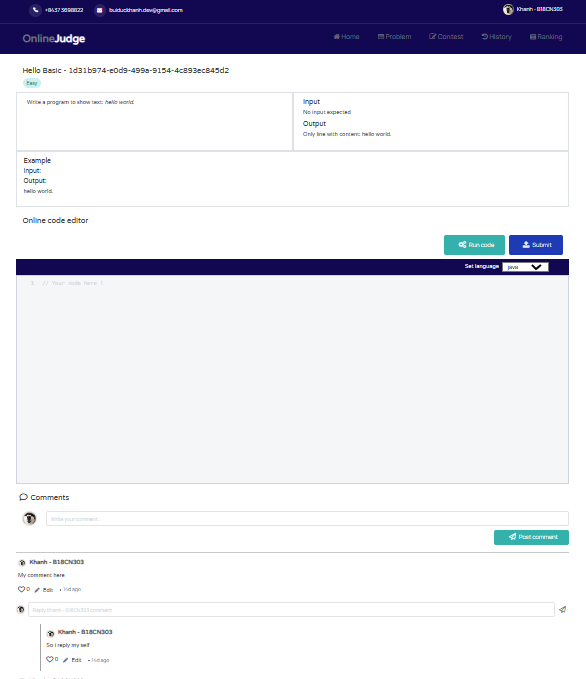
Hình . Giao diện trang xem bài toán

Giao diện trang xem bài toán bao gồm các thành phần chính là bộ lọc bài toán theo tên, mã bài toán, mức độ, danh mục và danh sách các bài toán, mỗi phần tử trong danh sách thể hiện các thông tin: tên bài toán, trạng thái làm bài, mức độ của bài toán và nút visit, ngoài ra khi tổng số bài toán có số trang lớn hơn 1, bên dưới danh sách xem bài toán sẽ xuất hiện thanh phân trang.

+ Giao diện xem chi tiết bài toán:



Hình . Giao diện xem chi tiết bài toán – chạy test



Hình . Giao diện xem chi tiết bài toán

Giao diện xem chi tiết bài toán bao gồm thông tin chi tiết về bài toán như: mã bài, tên bài, đề bài, ràng buộc đầu vào, ràng buộc đầu ra, ví dụ. Nút run code, nút submit, code editor, ô chọn ngôn ngữ, phần bình luận bao gồm các bình luận và trả lời bình luận.

Giao diện xem chi tiết bài toán, chức năng chạy test gồm ô input để nhập test, output hiển thị kết quả chạy test, icon thoát và nút chạy test.

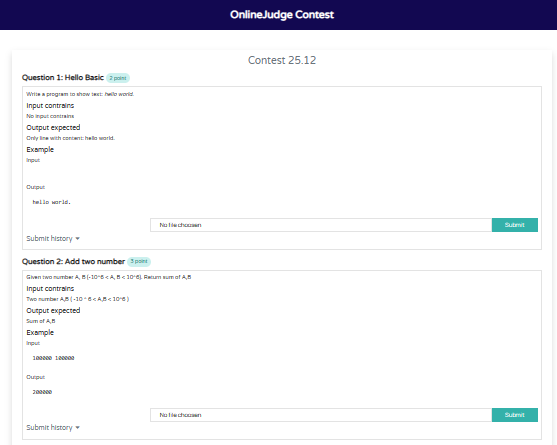
+ Giao diện xem contest



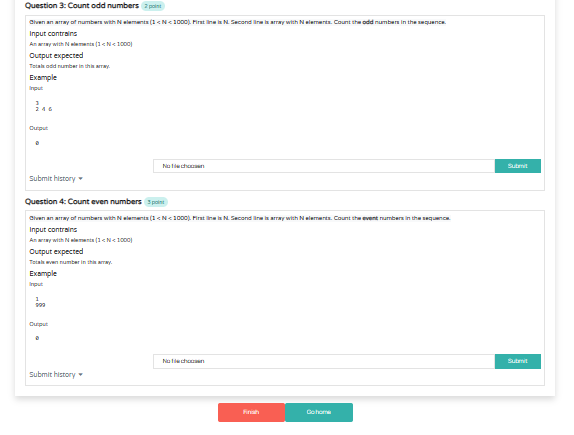
Hình . Giao diện xem contest

Giao diện xem contest bao gồm thông tin các contest đã làm: số người tham dự, tổng số lượng người cho phép, trạng thái contets. Với contest đã đăng ký sắp tới có nút tham gia nếu đã đến giờ, nút waiting khi chưa đến giờ, contest sắp tới, contest đã từng tham gia.

+ Giao diện làm contest:



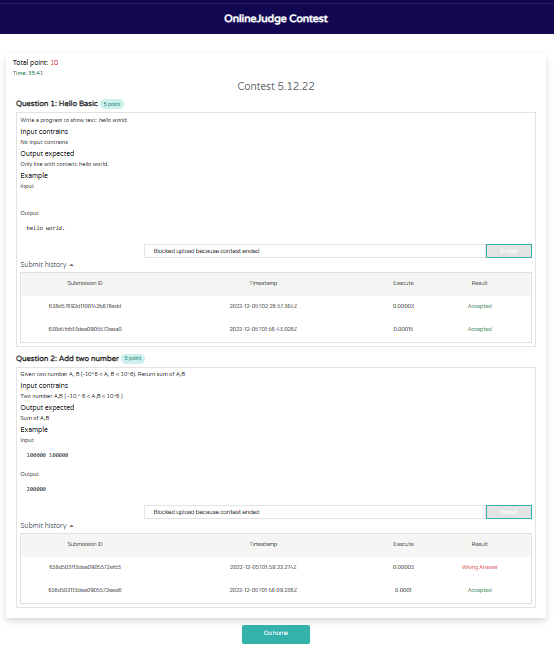
Hình . Giao diện làm contest (1)



Hình . Giao diện làm contest (2)

Giao diện làm contest bao gồm thông tin về tên contest, các câu hỏi trong contest. Mỗi câu hỏi trong contest bao gồm thông tin về câu hỏi, điểm của câu hỏi, ô tải file, nút submit và danh sách các submission của câu hỏi. Ở cuối cùng của trang có nút finish và nút go home.

+ Giao diện xem lịch sử contest:



Hình . Giao diện xem lịch sử contest

Giao diện xem lịch sử contest tương đối giống với giao diện làm contest tuy nhiên có một số điểm khác biệt giữa hai giao diện này: Giao diện xem lịch sử contest có thêm thông tin thời gian làm contest, tổng số điểm mà người làm contest đã đạt được, các ô input, submit đều bị disable.

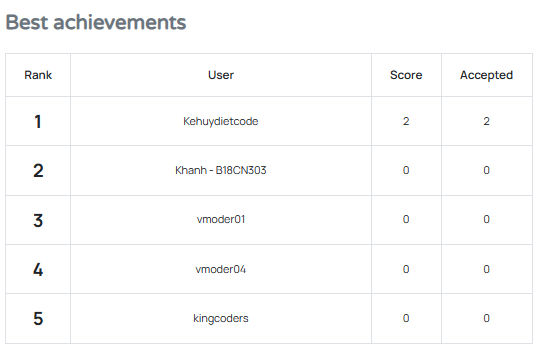
+ Giao diện xem lịch sử submit



Hình . Giao diện xem lịch sử submit

Giao diện xem lịch sử submit bao gồm lựa chọn: you để xem lịch sử của chính mình, hoặc all để xem toàn bộ lịch sử submit cùng với danh sách các submission tương ứng, mỗi submission bao gồm thông tin: trạng thái, tên bài toán, tên người làm, thời gian chạy, phần trăm vượt qua test, thời gian tạo. Với submit của người dùng, có thêm nút edit ở ô thể hiện kết quả submit, các submit đúng sẽ có ô thể hiện kết quả màu xanh, các submit sai sẽ có ô thể hiện kết quả màu đỏ, các submit gặp lỗi biên dịch sẽ có ô thể hiện kết quả submit màu xám. Ngoài ra bên dưới danh sách còn có thanh phân trang.

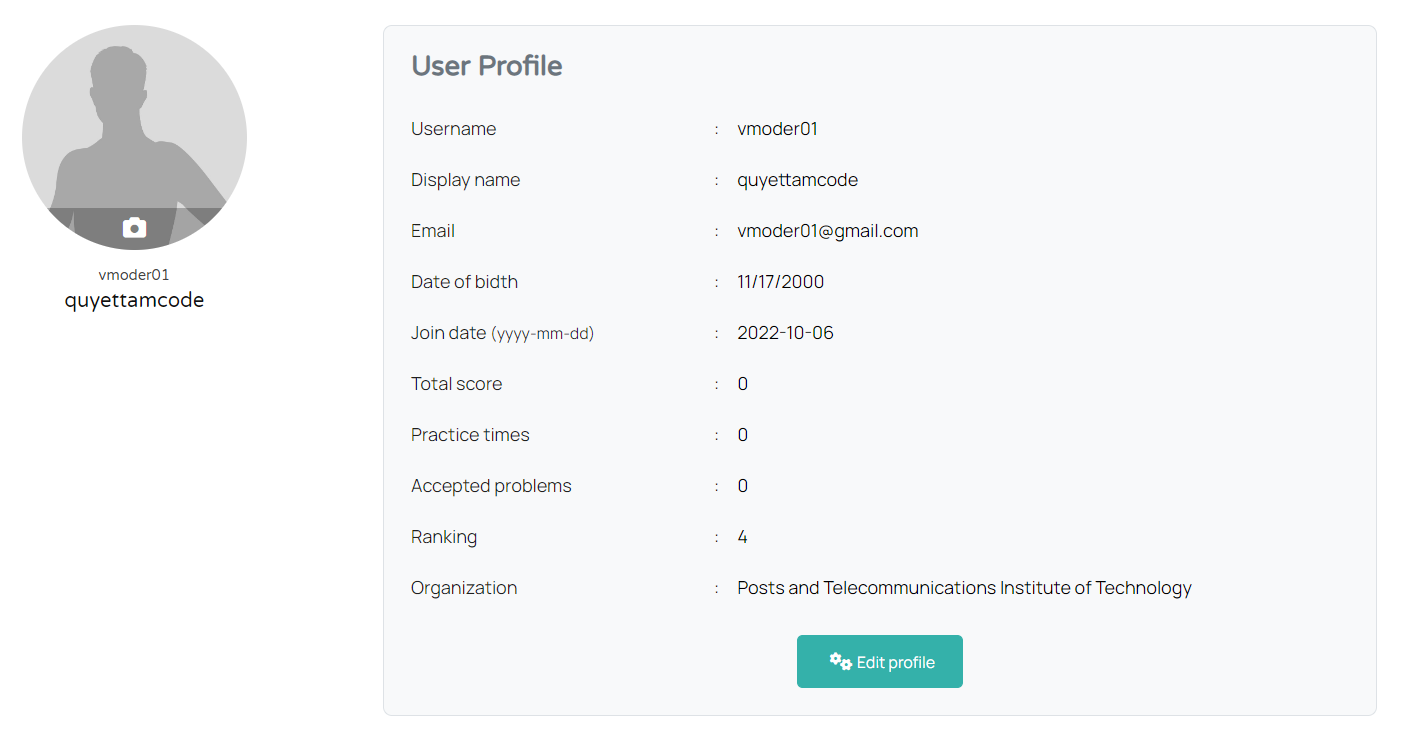
+ Giao diện xem xếp hạng



Hình . Giao diện xem xếp hạng

Giao diện xếp hạng bao gồm thông tin xếp hạng của các thành viên hệ thống, mỗi xếp hạng bao gồm số tự tự, tên người dùng, điểm và số bài được accept.

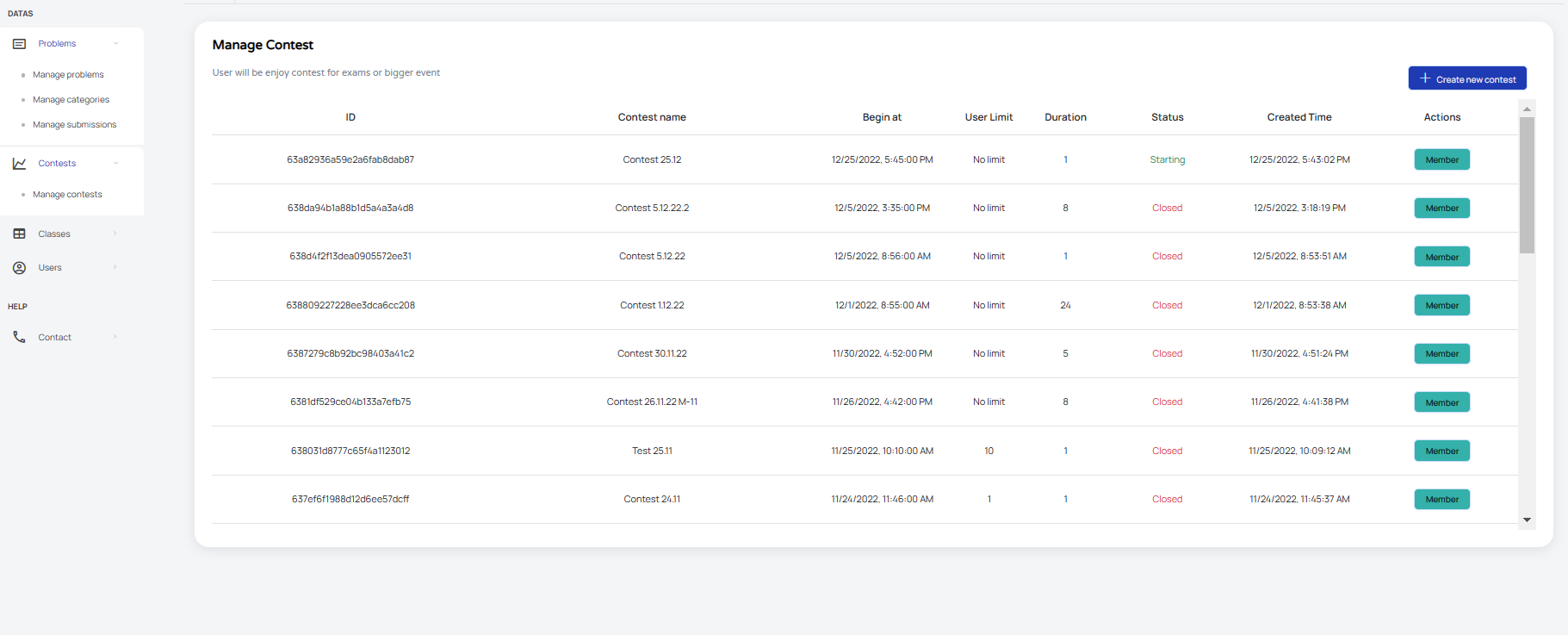
+ Giao diện quản lý thông tin cá nhân:



Hình . Giao diện quản lý thông tin cá nhân

Giao diện quản lý thông tin cá nhân bao gồm thông tin cá nhân của người dùng , nút đổi avatar và nút edit profile để thay đổi thông tin của user.

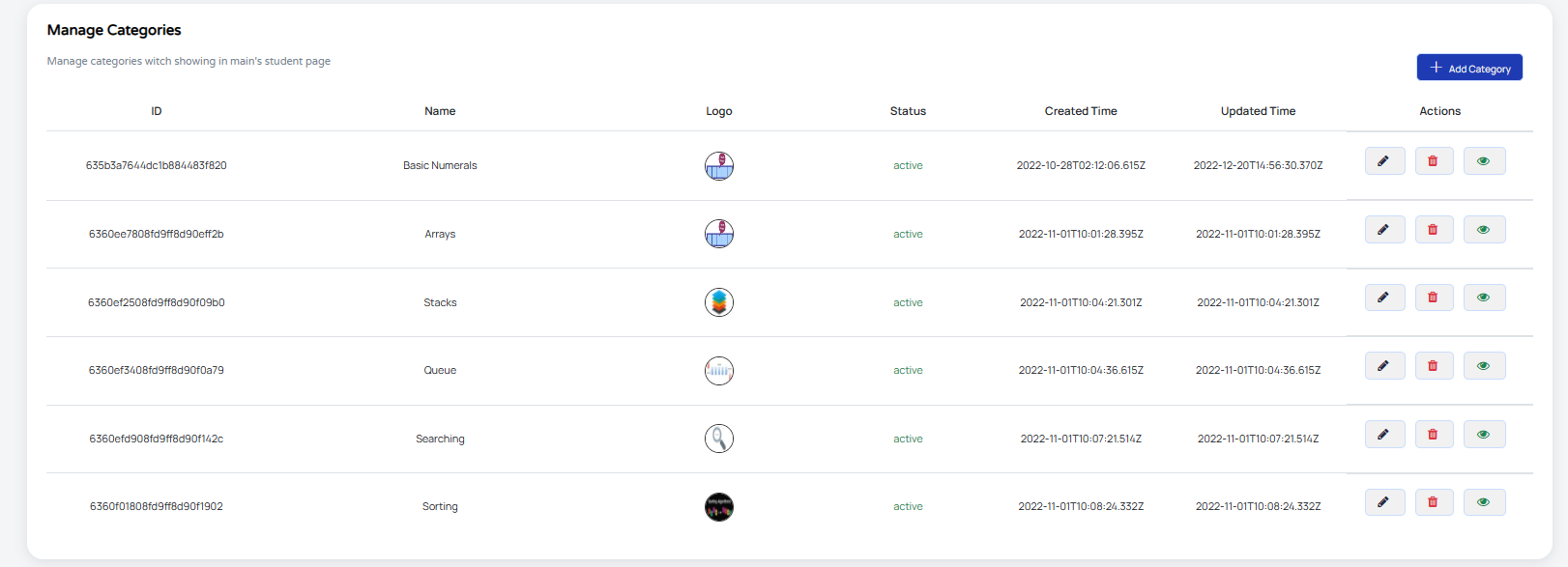
+ Giao diện quản lý contest



Hình . Giao diện quản lý contest

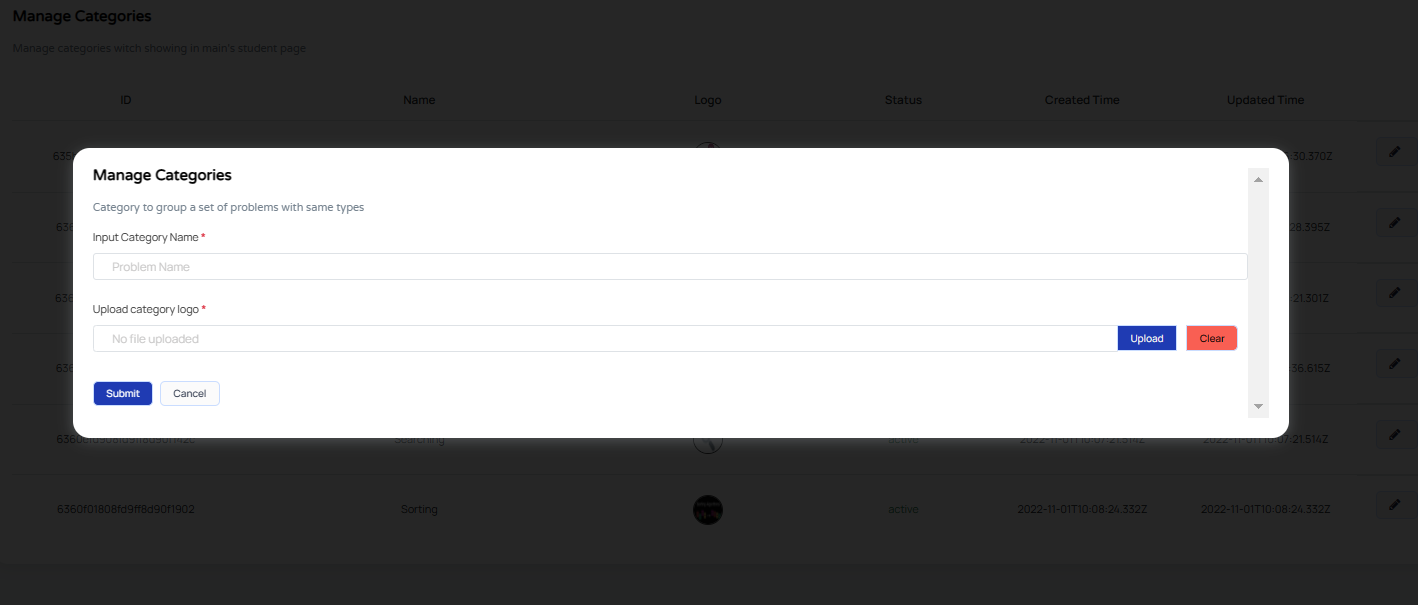
Giao diện quản lý contest bao gồm các thông tin về contest gồm: mã định danh, tên contets, thời gian bắt đầu, số người, thời lượng, trạng thái contest, thời gian tạo và nút thêm contest.

+ Giao diện quản lý category:



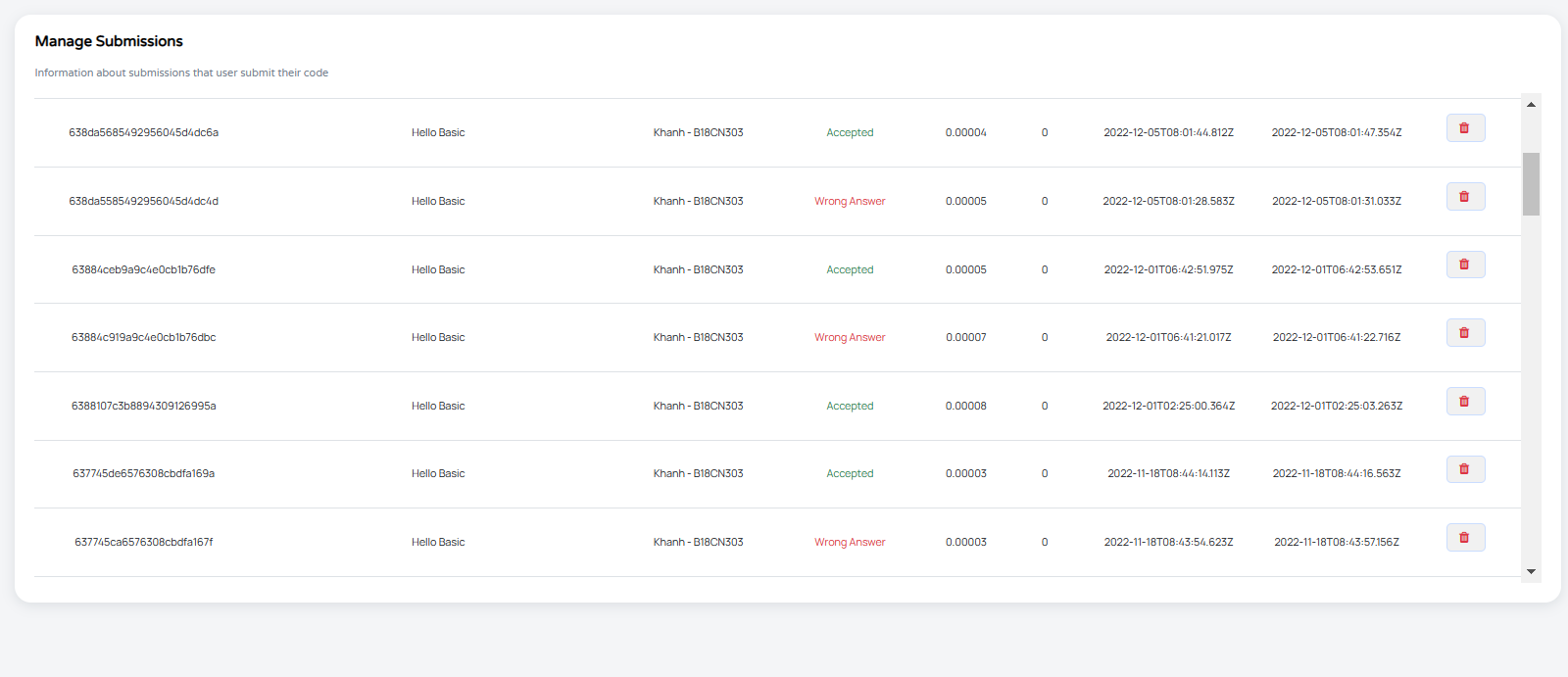
Hình . Giao diện quản lý danh mục

Giao diện quản lý danh mục bao gồm nút thêm danh mục và danh sách các danh mục bao gồm các thông tin: mã danh mục, tên danh mục, logo, trạng thái, thời gian tạo, thời gian cập nhật, nút sửa danh mục, xóa danh mục, ẩn hiện danh mục. Khi click thêm danh mục, giao diện thêm danh mục hiện ra với ô nhập tên danh mục và upload ảnh danh mục cùng nút submit và cancel.

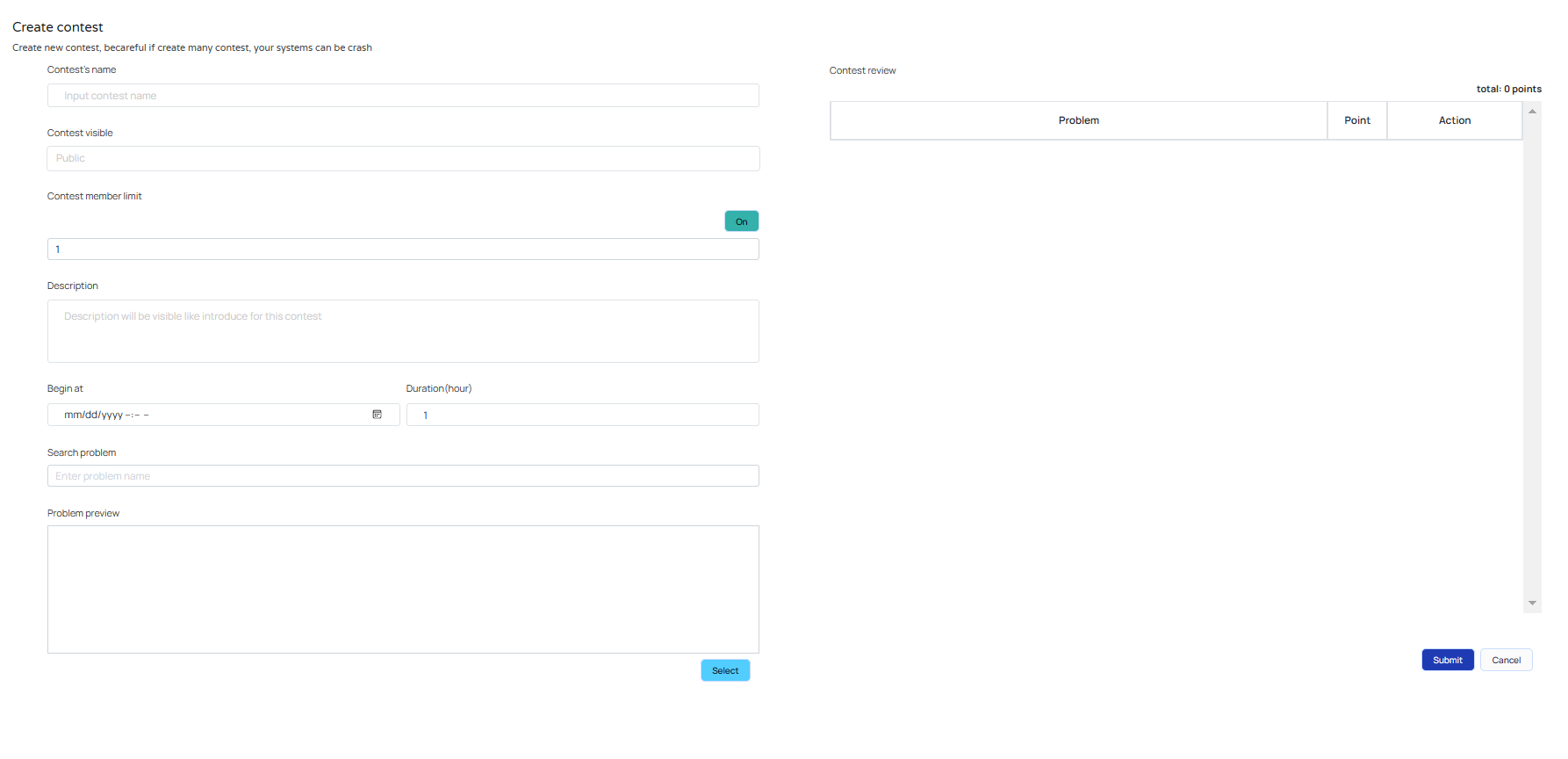


Hình . Giao diện thêm danh mục

+ Giao diện quản lý submission



Hình . Giao diện quản lý submissions



Hình . Giao diện thêm contest

Giao diện quản lý submission bao gồm thông tin của các submission bao gồm mã định danh, tên bài toán, tên người làm, trạng thái làm bài, thời gian thực hiện, thời gian tạo, thời gian cập nhật và nút xóa. Khi click vào nút thêm contest, giao diện thêm contest hiện lên với ô nhập thông tin contest, ô tìm kiếm bài toán, nút chọn bài toán, danh sách bài toán được chọn, nút submit và nút cancel.

## Kết luận

Chương 3 đã trình bày hướng dẫn về cách cài đặt hệ thống bao gồm cài đặt mã nguồn cho frontend, backend cùng các công cụ liên quan và kết quả cài đặt về database và giao diện của hệ thống. Dựa vào kết quả của chương 3, chương tiếp theo sẽ đưa ra kết luận, đánh giá về kết quả cài đặt hệ thống và phương hướng phát triển của hệ thống sau này.

# PHẦN TỔNG KẾT

Nội dung chính của chương là đưa ra đánh giá về kết quả thu được khi thực hiện đồ án và đề ra phương hướng phát triển cho đồ án

## Đánh giá kết quả của đồ án

Sau thời gian tìm hiểu, nghiên cứu các công nghệ để thực hiện đồ án với đề tài xây dựng website chấm bài trực tuyến cho việc học và luyện tập lập trình, bản thân em đã thu được những kết quả cụ thể như sau:

* Xây dựng hoàn chỉnh hệ thống chấm code online phục vụ cho việc học và luyện tập lập trình.
* Hiểu được cách giao tiếp giữa máy khách và máy chủ trong mô hình REST.
* Vận dụng được các kiến thức đã học để xây dựng các API theo chuẩn REST.
* Tìm hiểu thêm được những công nghệ mới: mongoDB, reactJs, express, socket.io.
* Hiểu hơn về ngôn ngữ lập trình javascript đặc biệt là vấn đề về bất đồng bộ trong javascript.
* Vận dụng được các công cụ hỗ trợ cho javascript để phát triển ứng dụng nhanh hơn.

Bên cạnh đó, đồ án còn có nhiều hạn chế, cụ thể là:

* Số lượng bài tập trên hệ thống còn hạn chế
* Ngôn ngữ lập trình hỗ trợ trên hệ thống còn ít
* Hiệu năng hệ thống chưa đảm bảo
* Chưa tính toán được bộ nhớ sử dụng cho chương trình của người dùng
* Cài đặt trên các môi trường khác nhau còn xuất hiện nhiều lỗi

## Phương hướng phát triển

Với kết quả thu được, em đã đề ra phương hướng phát triển cho đồ án trong lai cần triển khai, bao gồm các nội dung sau:

* Thêm các tính năng mới để hệ thống có sức hút và mạnh mẽ hơn như chia sẻ màn hình code, chống gian lận trong khi làm contest, hỗ trợ thêm actor: giảng viên, thêm chức năng tạo lớp học, tạo task cho thành viên trong lớp học, thêm blog chia sẻ kiến thức.
* Cải thiện hiệu năng của hệ thống.
* Hỗ trợ thêm nhiều ngôn ngữ lập trình và phiên bản của các ngôn ngữ lập trình.
* Sử dụng docker cho việc cài đặt hệ thống để có thể dễ dàng thay đổi môi trường cài đặt.

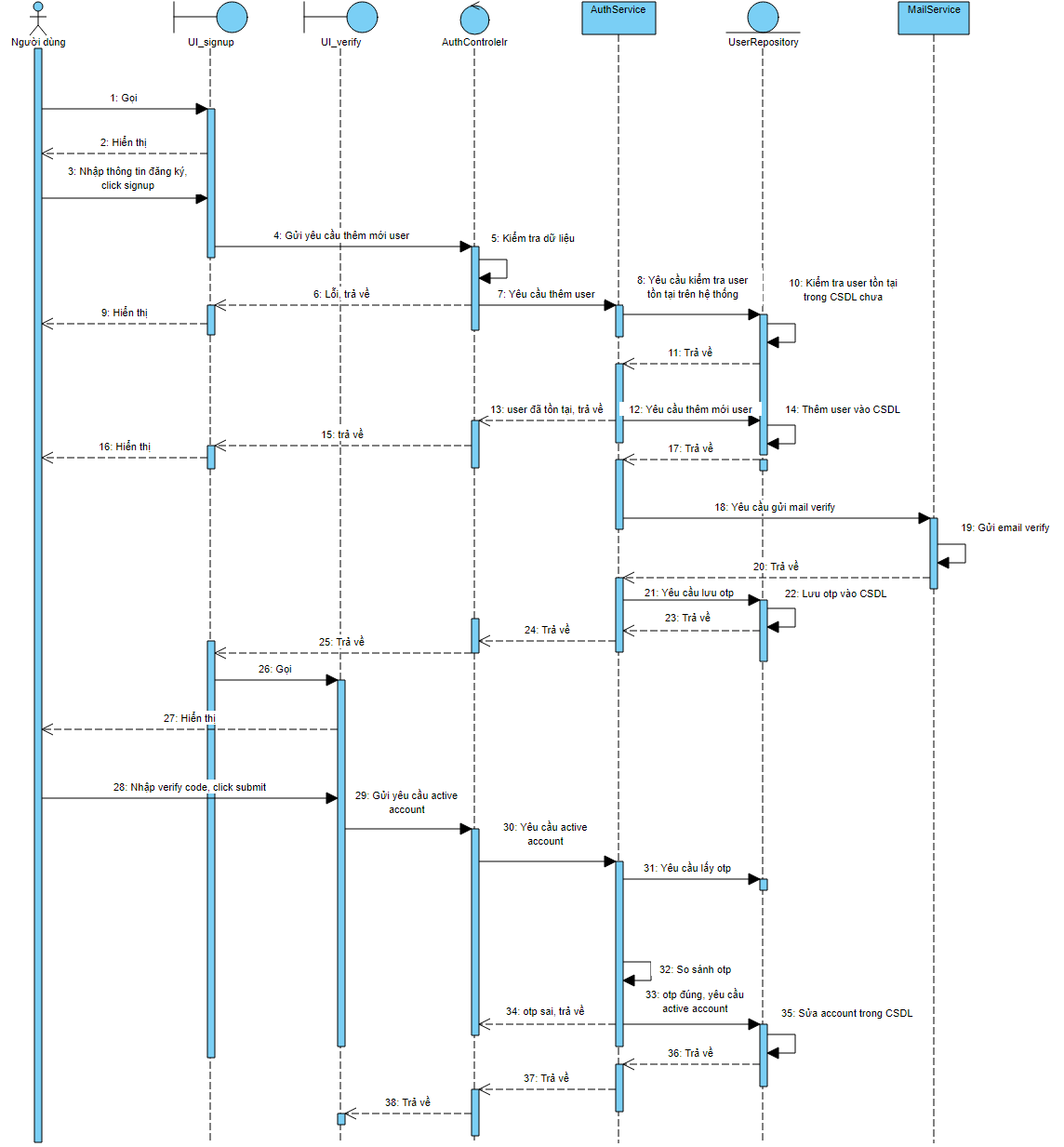
# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | "Stack Overflow Developer Survey 2022," [Online]. Available: https://survey.stackoverflow.co/2022/#technology. [Accessed 12 26 2022]. |
| [2] | "Node.js Architecture and 12 Best Practices for Node.js Development," [Online]. Available: https://scoutapm.com/blog/nodejs-architecture-and-12-best-practices-for-nodejs-development. [Accessed 12 26 2022]. |
| [3] | "What is REST? | Codecademy," [Online]. Available: https://www.codecademy.com/article/what-is-rest. [Accessed 12 26 2022]. |
| [4] | Trần Đình Quế, Nguyễn Mạnh Sơn, "Phân tích thiết kế Hệ Thống Thông Tin," in *Phân tích thiết kế Hệ Thống Thông Tin*, Học Viện Công Nghệ Bưu Chính Viễn Thông. |
| [5] | Stephen R.Schach, "Object-Oriented and Classical Software Engineering," in *Object-Oriented and Classical Software Engineering*, McGraw Hill. |
| [6] | Nguyễn Mạnh Hùng, Đỗ Thị Bích Ngọc, "Phân tích thiết kế và đảm bảo chất lượng phần mềm," in *Phân tích thiết kế và đảm bảo chất lượng phần mềm*, Học Viện Công Nghệ Bưu Chính Viễn Thông. |

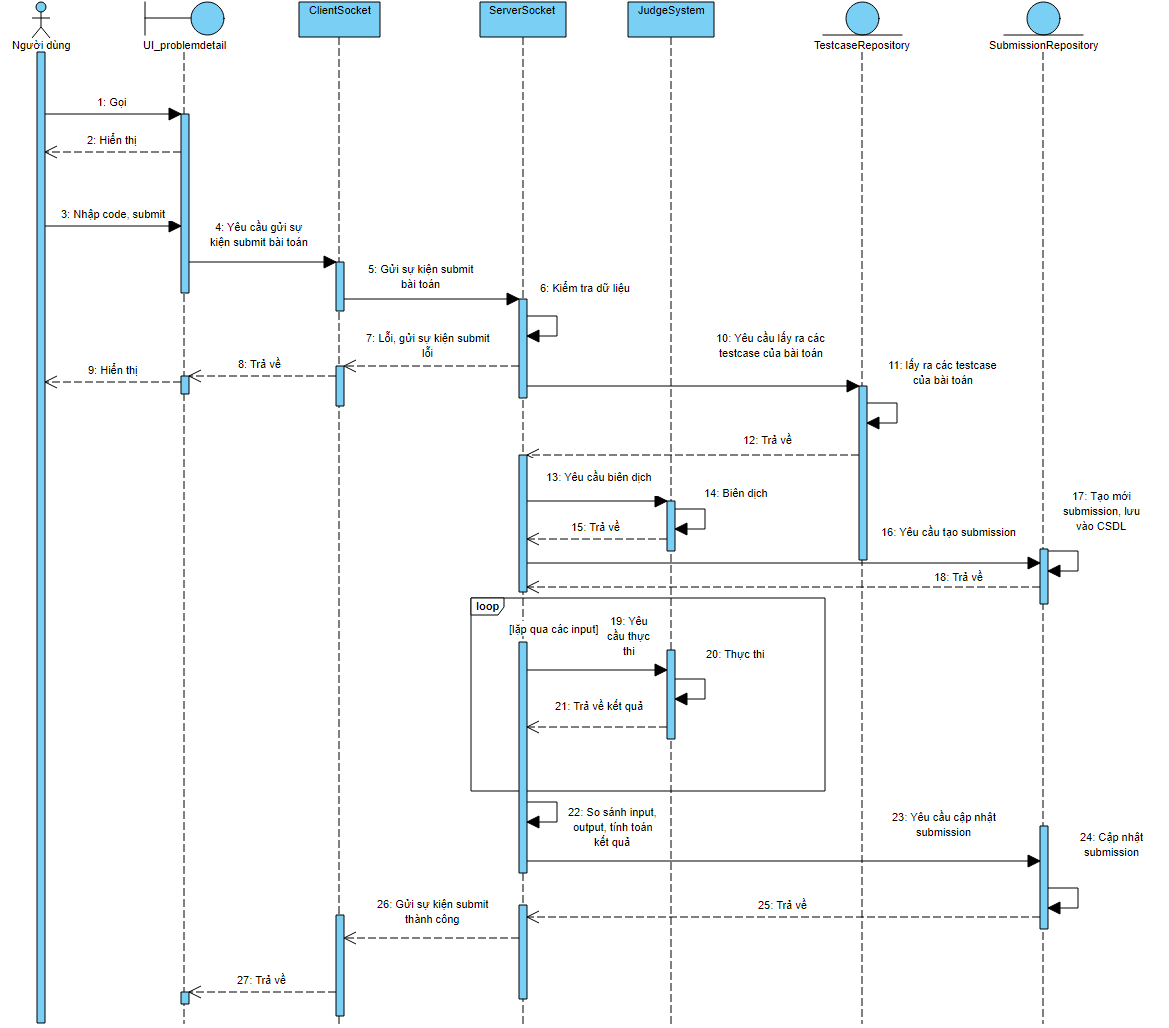
# PHỤ LỤC

## Chi tiết sơ đồ mô phỏng

*Biểu đồ tuần tự module đăng ký*



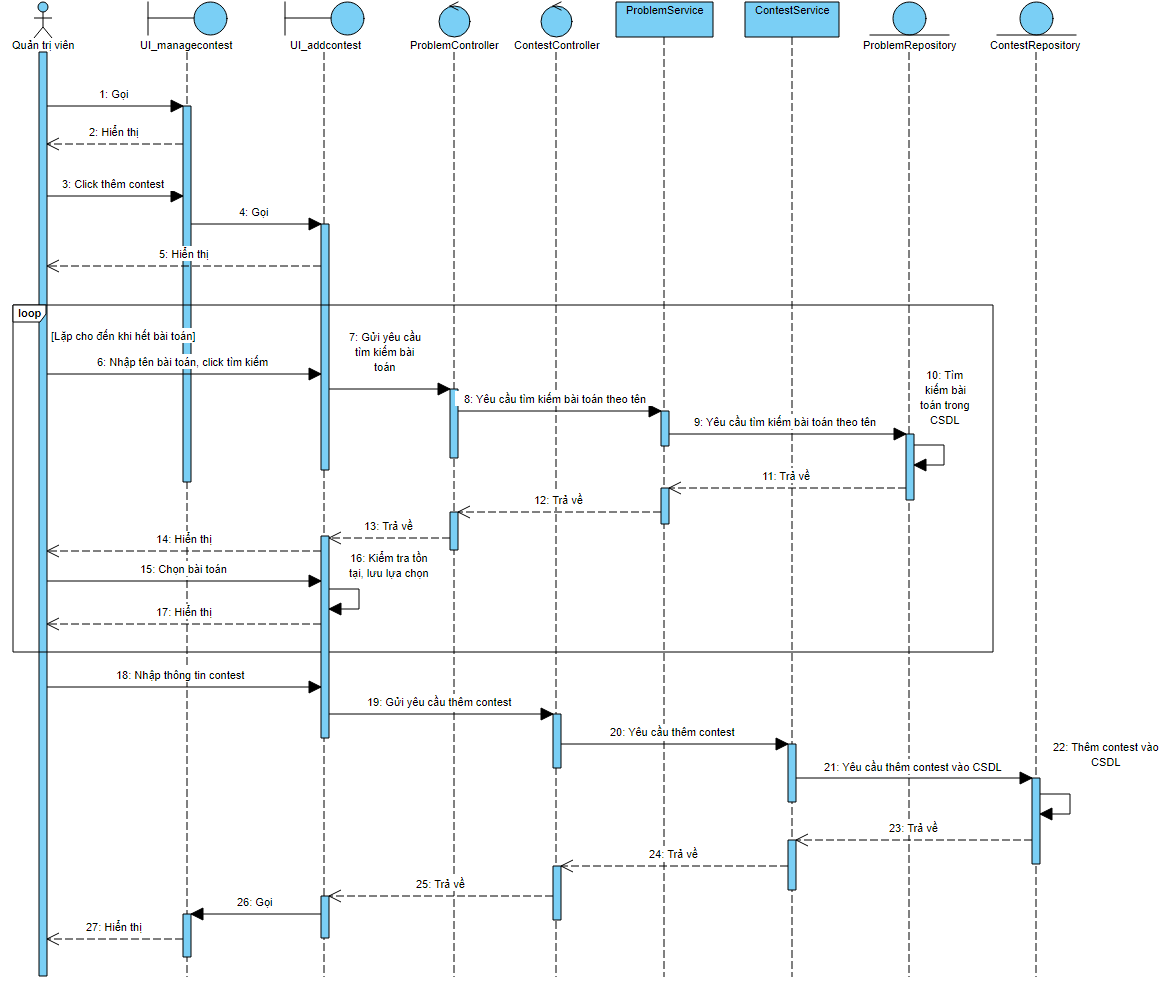
*Biểu đồ tuần tự module submit lời giải*

**

*Biểu đồ tuần tự module tham gia contest*

**

*Biểu đồ tuần tự module thêm contest*

**