

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

**KHOA CÔNG NGHỆ THÔNG TIN**

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

HỌ TÊN SINH VIÊN: NGUYỄN HỒNG PHÚC

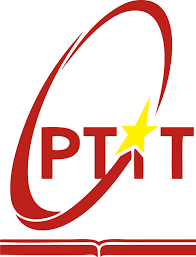
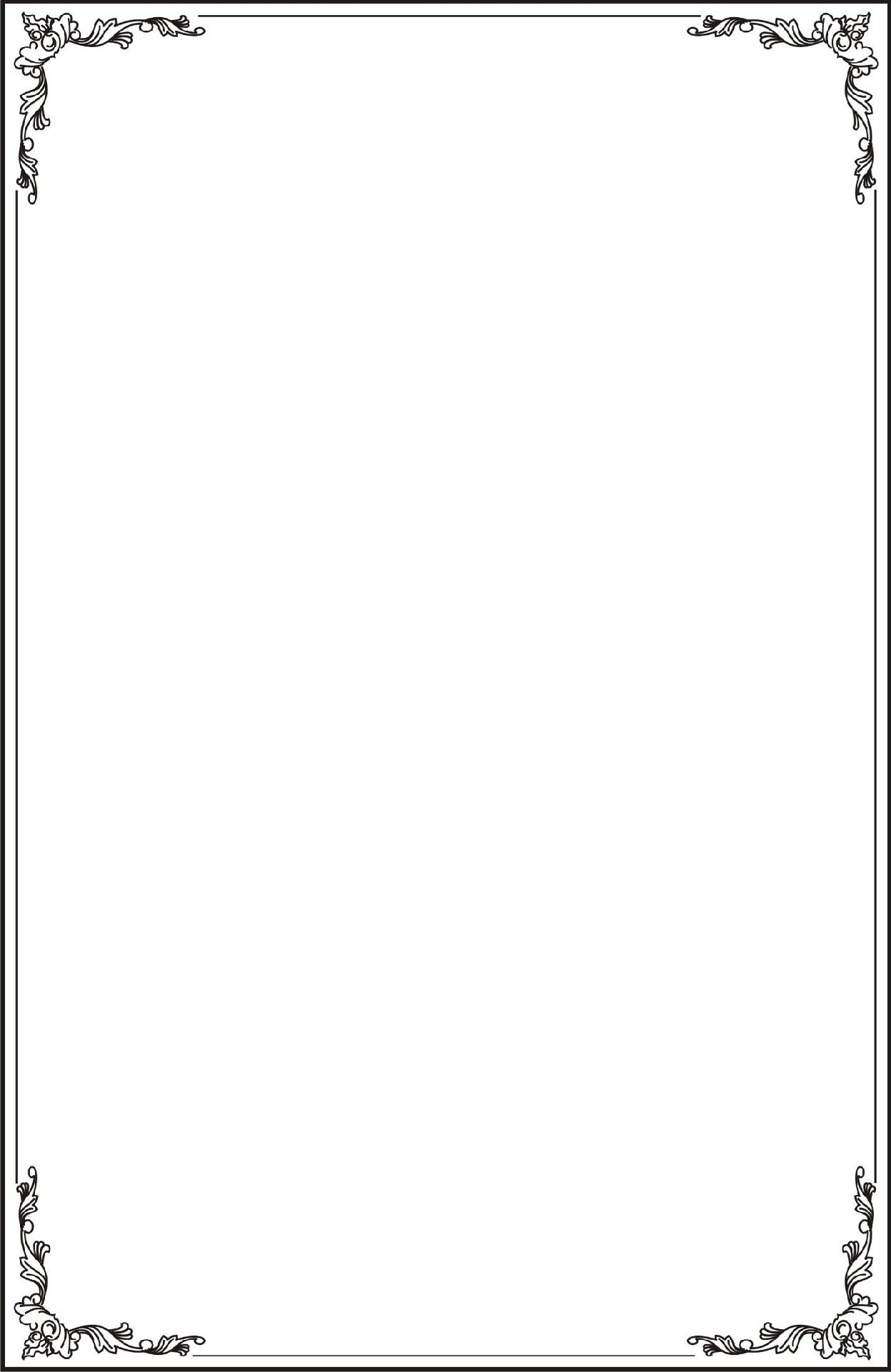
ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**Đề tài: XÂY DỰNG ỨNG DỤNG ANDROID MÔ PHỎNG HỆ MẶT TRỜI**

LỚP: D19CNPM01

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Đinh Xuân Trường |
| **Sinh viên thực hiện:** | Bùi Đức Khanh |
| **Mã sinh viên:** | B19DCCN505 |
| **Lớp:** | D19CNPM1 |
| **Niên khóa:** | 2019-2024 |
| **Hệ đào tạo:** | Đại học chính quy |

**Hà Nội 2023**



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

**KHOA CÔNG NGHỆ THÔNG TIN**

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

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

**Đề tài: XÂY DỰNG ỨNG DỤNG ANDROID MÔ PHỎNG HỆ MẶT TRỜI**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Đinh Xuân Trường |
| **Sinh viên:** | Nguyễn Hồng Phúc |
| **Mã sinh viên:** | B19DCCN505 |
| **Lớp:** | D19CNPM1 |
| **Niên khóa:** | 2019-2024 |
| **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 bài báo cáo nhưng trong quá trình thực hiện không thể tránh được những thiếu sót, em kính mong nhận được những đánh giá , nhận xét của các thầy cô để có thể hoàn thiện sản phẩm của mình hơn nữa trong tương lai.

Em xin chân thành cảm ơn!

**Hà Nội, tháng 12 năm 2023**

**Sinh viên**

**Nguyễn Hồng Phúc**

**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

Ngành thiên văn học bắt đầu với sự quan sát thiên nhiên và vũ trụ từ thời kỳ cổ đại. Mục đích nghiên cứu thiên văn trong thời kỳ này là để dự đoán các hiện tượng thiên văn như chấm dứt của một chuỗi ngày đêm, các mùa trong năm và các sự kiện thiên văn khác. Ngày này ngành thiên văn học sử dụng các kỹ thuật tiên tiến để quan sát, sử dụng máy tính mô phỏng để nghiên cứu về nguồn gốc, cấu trúc của vũ trụ và khả năng tồn tại sự sống ngoài vũ trụ. Thiên văn học là một lĩnh vực khoa học có nhiều ứng dụng quan trọng trong cuộc sống hiện đại:

* Định vị và định hướng:Thiên văn học đã giúp phát triển các hệ thống định vị toàn cầu như GPS (Global Positioning System), cho phép chúng ta xác định vị trí và định hướng với độ chính xác cao.
* Dự báo thời tiết: theo dõi các hiện tượng thiên văn như cấu trúc mây và biểu đồ nhiệt độ của các hành tinh trong hệ Mặt Trời có thể cung cấp thông tin hữu ích cho dự báo thời tiết.
* Công nghiệp hàng không và không gian: hỗ trợ việc thiết kế, dự đoán, và điều khiển các vệ tinh và tàu vũ trụ.
* Giáo dục và tạo đam mê: Thiên văn học là nguồn cảm hứng không giới hạn cho giáo dục và tạo đam mê đối với khoa học trong các thế hệ trẻ. Các chương trình giảng dạy và sự kiện thiên văn học thú vị giúp truyền đạt kiến thức về vũ trụ và khoa học thiên văn cho mọi người.

Tuy nhiên tại Việt Nam, ngành thiên văn học còn chưa được chú trọng và quantâm. Chính vì thế em muốn xây dựng một ứng dụng cung cấp nhưng kiến thức, thông tin cơ bản về thiên văn(cụ thể ở đây là hệ Mặt Trời) nhằm giúp mọi người có cái nhìn mới về một ngành khoa học còn non trẻ tại Việt Nam, đồng thời giúp những người có sở thích về thiên văn có thêm một công cụ để tìm hiểu nhiều hơn về các thiên thể trong hệ Mặt Trời.

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, ngôn ngữ cũng như engine sử dụng: C#, OpenGL, Unity và các frame work liên quan.

* **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 hiện trạng, mức độ quan tâm tới ngành vật lý thiên văn trên thế giới nói chung và tại Việt Nam nói riêng đồng thời giới thiệu tổng quan về đề tài: Xây dựng ứng dụng Android mô phỏng hệ Mặt Trời.

## 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 mô hình hệ Mặt Trời cung cấp một cách trực quan về thông tin thông qua các hình ảnh và model 3D.
* Cung cấp hệ thống giả lập quỹ đạo cho các hành tinh trong hệ Mặt Trời.
* Cung cấp hệ thống bản đồ sao chi tiết.
* Thực hiện bản địa hóa cho các ngôn ngữ khác nhau.
* Đối tượng hướng đến: những người quan tâm và yêu thích tới vật lý thiên văn.

## Mục tiêu nghiên cứu

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

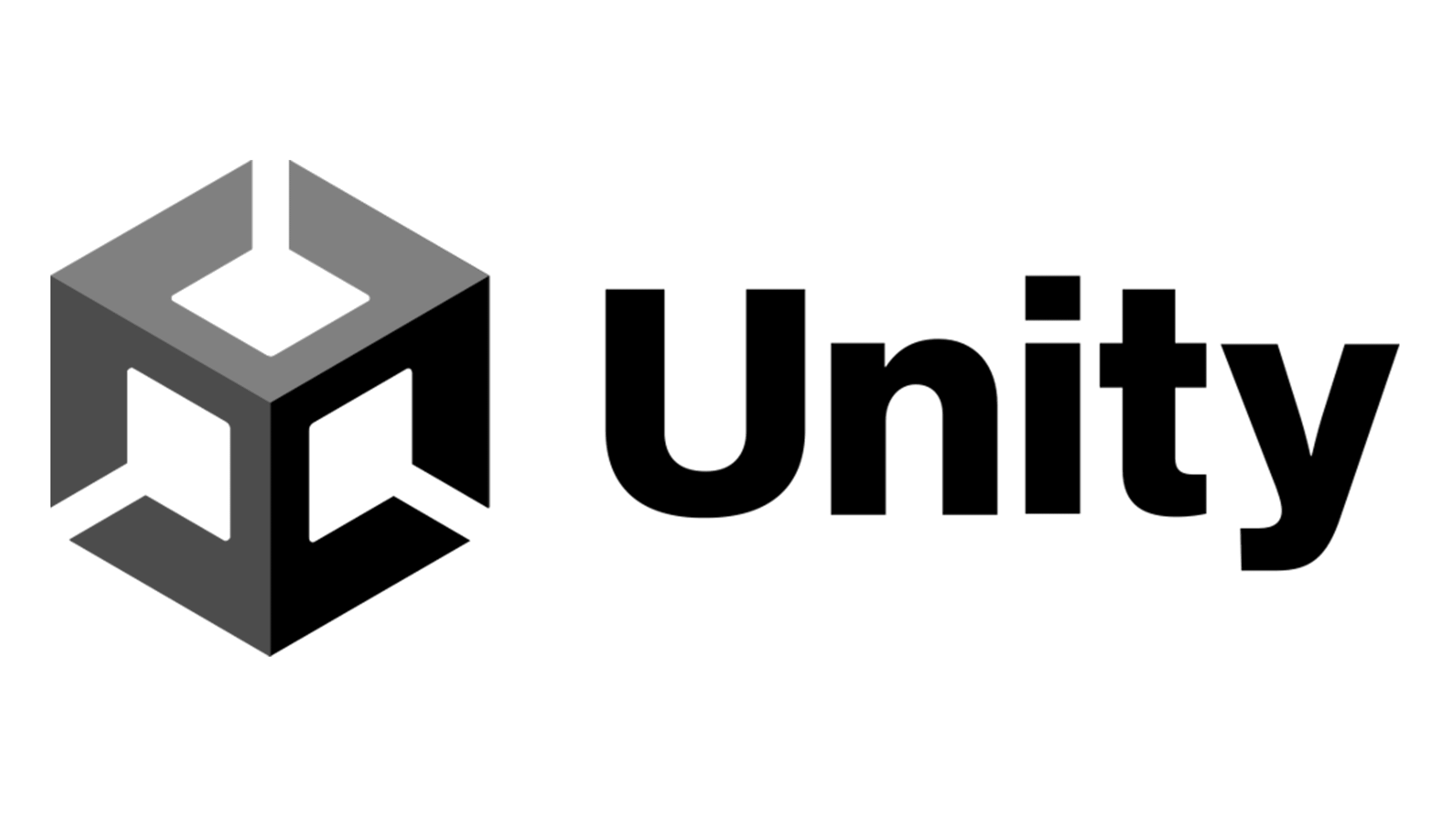
* Tìm hiểu về kiến trúc và các design partem trong việc thiết kế ứng dụng Android.
* Tìm hiểu về ngôn ngữ lập trình C# thông qua việc xây dựng ứng dụng Android.
* Tìm hiểu OpenGL(một API trừu tượng về đồ họa 2D và 3D) thông qua Shader và Shader Graph được tích hợp sẵn trong Unity.
* Áp dụng những kiến thức đã tìm hiểu để xây dựng một hệ thống giúp người sử dụng tìm hiểu về lĩnh vực vật lý thiên văn.

## 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 Unity Engine



Unity là một game engine [đa nền tảng](https://vi.wikipedia.org/wiki/%C4%90a_n%E1%BB%81n_t%E1%BA%A3ng) được phát triển bởi Unity Technologies, mà chủ yếu để phát triển video game cho máy tính, consoles và điện thoại. Lần đầu tiên nó được công bố chạy trên hệ điều hành OS X, tại Apple's Worldwide Developers Conference vào năm 2005, đến nay đã mở rộng 27 nền tảng.

Unity hỗ trợ đồ họa 2D và 3D, các chức năng được viết chủ yếu qua ngôn ngữ [C#](https://vi.wikipedia.org/wiki/C_th%C4%83ng). Hai ngôn ngữ lập trình khác cũng được hỗ trợ: Boo, đã bị loại cùng với việc phát triển Unity 5 và UnityScript bị loại vào tháng 8 năm 2017 sau khi phát hành Unity 2017.1UnityScript là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh_k%E1%BB%8Bch_b%E1%BA%A3n) độc quyền có cú pháp tương tự [JavaScript](https://vi.wikipedia.org/wiki/JavaScript).

Phần mềm nhắm mục tiêu các đồ họa APIs sau: Direct3D trên Windows và Xbox One; [OpenGL](https://vi.wikipedia.org/wiki/OpenGL) trên Linux, macOS, và Windows; OpenGL ES trên Android và iOS; WebGL trên web; và APIs độc quyền trên các [máy chơi video game](https://vi.wikipedia.org/wiki/Video_game_console).

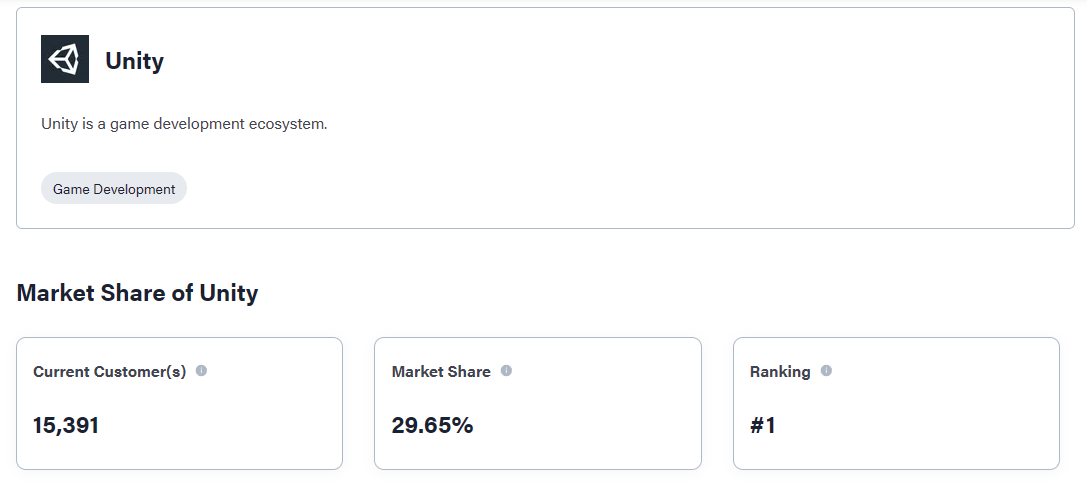
Ngoài ra, Unity hỗ trợ APIs cấp thấp như Metal trên iOS và macOS và Vulkan trên Android, Linux, và Windows, cũng như Direct3D 12 trên Windows và Xbox One.

Trong 2D games, Unity cho phép nhập sprites và một renderer thế giới 2D tiên tiến. Đối với 3D games, Unity cho phép thiết lập các đập điểm kỹ thuật của các kết cấu và độ phân giải mà công cụ trò chơi hỗ trợ, cung cấp các hỗ trợ cho bump mapping, reflection mapping, parallax mapping, cảnh không gian ambient occlusion (SSAO), hiệu ứng bóng đổ bằng cách sử dụng shadow maps, render thiết lập toàn cảnh đến hiệu ứng.

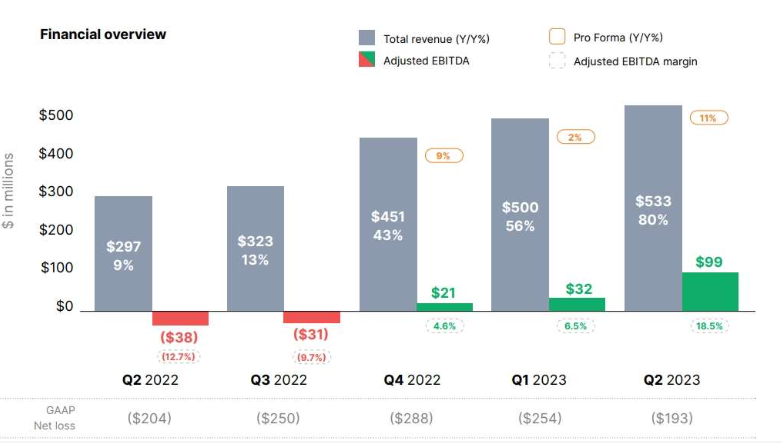
Unity cũng cung cấp các dịch vụ cho nhà phát triển, bao gồm: Unity Ads, Unity Analytics, Unity Certification, Unity Cloud Build, Unity Everyplay, Unity API, Unity Multiplayer, Unity Performance Reporting and Unity Collaborate.

Một số thống kê về Unity

* Tính đến quý 3 năm 2016 đã có hơn 5 tỉ lượt tải các ứng dụng được phát triển bởi Unity.
* 2,4 tỉ thiết bị di động đã từng cài ít nhất một ứng dụng bởi Unity.
* Trong top 1000 các sản phẩm game mobiles miễn phí thì số lượng game được tạo ra bởi Unity chiếm tới 34%.



* Theo báo cáo quý 2, Unity đạt doanh thu 533 triệu đô la trong quý, tăng 80% so với năm trước. Tạo giải pháp (công cụ, đăng ký, v.v.) đạt doanh thu 193 triệu đô la, tăng 17% so với năm trước (do hoạt động kinh doanh đăng ký của Unity thúc đẩy). Grow Solutions (quảng cáo, kiếm tiền) đạt doanh thu 340 triệu USD, tăng 157% so với năm trước.



Ưu điểm của Unity:

* Cộng đồng lớn và hỗ trợ mạnh mẽ: Có một cộng đồng lớn của những người phát triển sử dụng Unity và UnityEngine, cung cấp tài nguyên học tập, diễn đàn, và hỗ trợ.
* Tiết kiệm thời gian và tài nguyên: Unity cung cấp nhiều công cụ và tính năng giúp giảm thời gian phát triển và tối ưu hóa quy trình làm việc.
* Asset Store: Unity Asset Store cung cấp một cách thuận tiện để mua và bán tài nguyên, công cụ, và giao diện người dùng, giúp những người phát triển tiết kiệm thời gian và công sức.
* Đồ họa và hiệu ứng: Unity hỗ trợ đồ họa 2D và 3D mạnh mẽ, cung cấp khả năng tạo ra trải nghiệm đồ họa và hiệu ứng đa dạng.
* Độ Linh Hoạt Cao: Unity hỗ trợ nhiều ngôn ngữ lập trình như C#, JavaScript, và Boo, cho phép lập trình viên lựa chọn ngôn ngữ mà họ thoải mái nhất.
* Hỗ trợ VR/AR: Unity hỗ trợ phát triển trò chơi và ứng dụng thực tế ảo (VR) và thực tế ảo (AR), giúp nhà phát triển tham gia vào lĩnh vực công nghiệp này.

### Tổng quan về C#

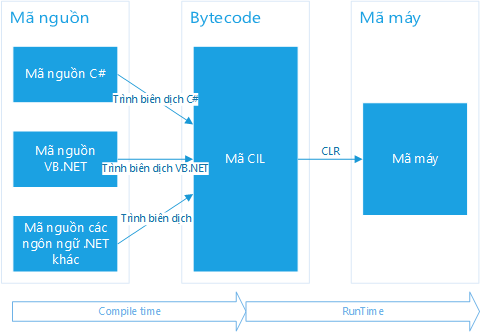
C# (hay còn gọi là C Sharp, là ngôn ngữ lập trình hướng đối tượng, hiện đại do Microsoft phát triển như một phần của nền tảng .NET. Nó được giới thiệu lần đầu tiên vào năm 2000 và kể từ đó đã trở thành một trong những ngôn ngữ lập trình được sử dụng rộng rãi nhất trong ngành.

C# có thiết kế đơn giản, mạnh mẽ và linh hoạt. Sở hữu một cú pháp tương tự như các ngôn ngữ lập trình phổ biến, bao gồm Java và C++, giúp các nhà phát triển đã quen thuộc với các ngôn ngữ này dễ làm quen.

Một trong những tính năng chính của C# là an toàn kiểu của nó, giúp ngăn ngừa các lỗi lập trình có thể xảy ra với các ngôn ngữ khác. Không những vậy, nó còn hỗ trợ quản lý bộ nhớ tự động thông qua trình thu gom rác, giúp dễ dàng viết mã đáng tin cậy mà không làm rò rỉ bộ nhớ hay gây ra bất kỳ sự cố nào.

Nhờ những điều đó, nên C# được sử dụng rộng rãi trong việc phát triển các ứng dụng Windows dành cho máy tính để bàn, game, ứng dụng web và các ứng dụng dành cho thiết bị di động. Ngoài ra, nó còn được sử dụng trong phát triển phần mềm doanh nghiệp, với nhiều công ty lớn sử dụng C# để xây dựng các ứng dụng kinh doanh của họ.

Với tổng quan những điều trên, thì chúng ta dễ dàng nhận thấy C# là một ngôn ngữ mạnh mẽ và linh hoạt, phù hợp với nhiều nhiệm vụ lập trình. Cho dù bạn mới bắt đầu làm quen với lập trình hay là một nhà phát triển có kinh nghiệm, thì C# luôn là một ngôn ngữ tuyệt vời để học và sử dụng khi muốn xây dựng các ứng dụng mạnh mẽ, đáng tin cậy và có thể mở rộng.



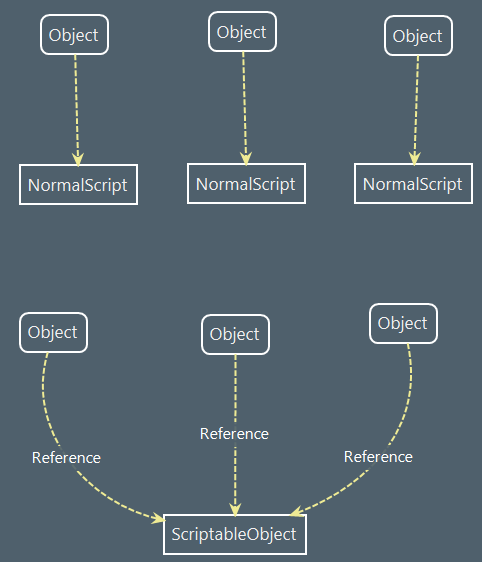
C# có những đặc điểm nổi bật như sau:

* Ngôn ngữ lập trình C# (C Sharp) là một ngôn ngữ lập trình đa mô hình (multi-paradigm) được phát triển bởi Microsoft. Dưới đây là một số đặc điểm nổi bật của ngôn ngữ C#:
* Tích hợp với Hệ điều hành Windows và Microsoft.NET: C# được thiết kế chủ yếu để phát triển ứng dụng chạy trên nền tảng Windows và tích hợp tốt với .NET Framework và .NET Core.
* Ngôn ngữ lập trình hướng đối tượng (OOP): C# hỗ trợ các nguyên tắc lập trình hướng đối tượng như kế thừa, đa hình, và đóng gói, giúp tổ chức mã nguồn và tăng tính tái sử dụng.
* Cú pháp rõ ràng và đơn giản: Cú pháp của C# được thiết kế để dễ đọc, dễ hiểu và giảm thiểu lỗi.
* Quản lý bộ nhớ tự động: C# sử dụng mô hình quản lý bộ nhớ tự động thông qua Garbage Collector, giảm gánh nặng cho lập trình viên khi phải quản lý bộ nhớ thủ công.
* Thư viện chuẩn rộng lớn: C# đi kèm với một số lượng lớn các thư viện chuẩn, cùng với sự hỗ trợ mạnh mẽ từ .NET Framework, giúp lập trình viên phát triển ứng dụng nhanh chóng và hiệu quả.
* Hỗ trợ đa nền tảng: C# đã phát triển để hỗ trợ đa nền tảng thông qua .NET Core và sau này là .NET 5 và các phiên bản mới hơn, cho phép việc phát triển ứng dụng chạy trên nhiều hệ điều hành như Windows, Linux, và macOS.
* Kiểu dữ liệu an toàn: C# hỗ trợ kiểu dữ liệu an toàn, giúp giảm thiểu lỗi liên quan đến kiểu dữ liệu trong quá trình thực thi.
* Delegates và Events: Delegates và Events là tính năng quan trọng của C#, giúp xử lý sự kiện và triển khai các mô hình lập trình không đồng bộ (asynchronous programming).
* LINQ (Language Integrated Query): LINQ là một tính năng quan trọng giúp thực hiện các truy vấn trực tiếp trong mã nguồn C#, giúp giảm độ phức tạp và tăng tính tổ chức của mã nguồn.
* ASP.NET: C# được sử dụng chủ yếu để phát triển ứng dụng web thông qua ASP.NET, một framework mạnh mẽ cho việc xây dựng các ứng dụng web và dịch vụ web.

Những đặc điểm này làm cho C# trở thành một ngôn ngữ lập trình phổ biến trong cộng đồng phần mềm và phát triển ứng dụng.

### Tổng quan về Scriptable Object

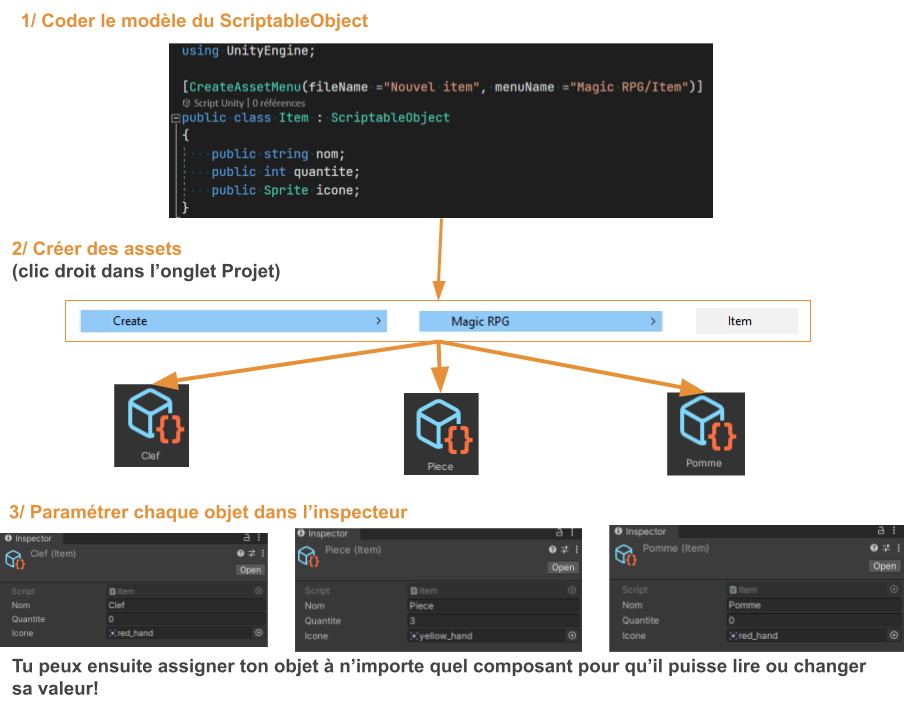
ScriptableObject là một bộ chứa dữ liệu có thể được sử dụng để lưu trữ lượng lớn dữ liệu, độc lập với các class. Một trong những trường hợp sử dụng chính của ScriptableObjects là giảm bớt việc sử dụng bộ nhớ trong Dự án bằng cách tránh sao chép các giá trị. Điều này hữu ích nếu Dự án có một Prefab lưu trữ dữ liệu không thay đổi trong các script MonoBehaviour đính kèm.



Mỗi khi thể hiện Prefab đó, nó sẽ có bản sao riêng của dữ liệu đó. Thay vì sử dụng phương thức này và lưu trữ dữ liệu trùng lặp, có thể sử dụng một ScriptableObject để lưu trữ dữ liệu và sau đó truy cập nó theo tham chiếu từ tất cả các Prefabs. Điều này có nghĩa là có một bản sao duy nhất của dữ liệu trong bộ nhớ

Tương tự như MonoBehaviours, ScriptableObjects được dẫn xuất từ đối tượng cơ sở Unity, nhưng khác với MonoBehaviours, bạn không thể đính kèm một ScriptableObject vào một GameObject. Thay vào đó, bạn cần lưu chúng dưới dạng Tài nguyên trong Dự án của bạn.

Dữ liệu lưu từ Unity engine vào ScriptableObjects như một tài nguyên được ghi vào đĩa và nó bền vững giữa các phiên làm việc.



Có nhiều cách để lưu trữ dữ liệu trong xây dựng ứng dụng Android, nhưng với Unity, Scriptable Object là một phương thức dữ liệu offine tối ưu.Việc lưu trữ bằng ScriptableObject trong Unity có nhiều ưu điểm so với việc lưu trữ dưới dạng các định dạng khác như JSON. Dưới đây là một số lợi ích chính:

* Tích hợp Tốt với Unity: ScriptableObject được tích hợp chặt chẽ với hệ thống Asset và Editor của Unity. Điều này giúp dễ dàng tạo, chỉnh sửa và quản lý các đối tượng ScriptableObject trong môi trường phát triển của Unity.
* Dễ Dàng Sử Dụng Trong Editor: Các đối tượng ScriptableObject có thể được tạo và chỉnh sửa trong Editor, giúp nhà phát triển thực hiện điều chỉnh nhanh chóng mà không cần chạy trò chơi.
* Khả Năng Quản Lý Mã Nguồn: Dữ liệu trong ScriptableObject được tổ chức tốt và có thể dễ dàng quản lý từ mã nguồn. Điều này làm cho mã nguồn của bạn trở nên dễ đọc và bảo trì.
* Tái Sử Dụng Dữ Liệu: Dữ liệu trong ScriptableObject có thể được chia sẻ và tái sử dụng giữa nhiều đối tượng khác nhau mà không cần tạo bản sao của dữ liệu. Điều này giúp giảm bớt lượng bộ nhớ được sử dụng.
* Khả Năng Kế Thừa và Mở Rộng: Bạn có thể kế thừa từ ScriptableObject để tạo các lớp con mới, giúp bạn mở rộng chức năng và cấu trúc của dữ liệu dễ dàng hơn.
* Hiệu Suất: Truy cập dữ liệu từ ScriptableObject có thể nhanh chóng hơn so với việc đọc và phân tích cú pháp các định dạng như JSON hoặc XML. ScriptableObject được tối ưu hóa cho việc sử dụng trong Unity.
* Hỗ Trợ Tốt Cho Prefab và Serialization: ScriptableObject rất phù hợp với việc lưu trữ dữ liệu cho Prefabs và hỗ trợ tốt cho quá trình serialization của Unity.

## Các công cụ

### JetBrains Rider

JetBrains Rider là một môi trường phát triển tích hợp (IDE) do JetBrains phát triển, chuyên dành cho lập trình viên sử dụng các ngôn ngữ lập trình liên quan đến Microsoft .NET. Được phát hành lần đầu vào năm 2016, Rider nhanh chóng trở thành một lựa chọn phổ biến cho các nhà phát triển .NET. Dưới đây là một số điểm đặc sắc của JetBrains Rider:



* Hỗ Trợ Nhiều Ngôn Ngữ: Rider không chỉ hỗ trợ C# mà còn hỗ trợ một loạt các ngôn ngữ lập trình khác như F#, VB.NET, HTML, CSS, JavaScript, TypeScript, và SQL.
* Tích Hợp Với .NET và ASP.NET: Rider được thiết kế để hoạt động chặt chẽ với .NET Framework và .NET Core, giúp bạn phát triển ứng dụng Windows, web, và dịch vụ điện toán đám mây dễ dàng. Cross-Platform:
* Rider có khả năng chạy trên nhiều hệ điều hành, bao gồm Windows, macOS và Linux, giúp nhà phát triển linh hoạt trong việc chọn lựa nền tảng phát triển.
* Đầy Đủ Tính Năng: Rider cung cấp một loạt các tính năng hỗ trợ mã nguồn, gỡ lỗi, kiểm thử, quản lý phiên bản, và nhiều tính năng khác để tối ưu hóa quy trình phát triển.
* Code Analysis và Refactoring: Rider tích hợp các công cụ phân tích mã nguồn mạnh mẽ, cung cấp gợi ý và khuyến nghị để cải thiện chất lượng mã nguồn. Nó cũng hỗ trợ nhiều thao tác tái cấu trúc mã nguồn (refactoring).
* Hỗ Trợ Unity: JetBrains Rider hỗ trợ phát triển trò chơi và ứng dụng sử dụng Unity Game Engine, cung cấp tính năng làm việc với C# trong môi trường phát triển Unity.
* Tích Hợp Git và VCS: Rider tích hợp tốt với các hệ thống quản lý mã nguồn như Git và hệ thống kiểm soát phiên bản khác, giúp quản lý và theo dõi sự thay đổi mã nguồn dễ dàng.
* UI Thân Thiện và Tùy Chỉnh: Giao diện người dùng của Rider được thiết kế để tối ưu hóa trải nghiệm người dùng, và người dùng có thể tùy chỉnh môi trường làm việc theo ý muốn.
* Cập Nhật Thường Xuyên: JetBrains thường xuyên cập nhật Rider để hỗ trợ các tính năng mới của .NET và cải thiện hiệu suất và trải nghiệm người dùng.
* Đồng Tích Hợp với JetBrains Toolbox: Rider có thể tích hợp chặt chẽ với JetBrains Toolbox, giúp quản lý cài đặt và cập nhật IDE cũng như các công cụ khác của JetBrains.

## 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 thực tế của ngành thiên văn học tại Việt Nam nói riêng và thế giới nói chung
* 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

Ứng dụng Android mô phỏng hệ mặt trời.

### Mục tiêu của hệ thống:

Ứng dụng được xây dựng với mục tiêu cung cấp những kiến thức căn bản, cung cấp hệ thống hình ảnh và modle cho những người quan tâm và yêu thích với thiên văn học. Hệ thống giúp người dùng có cái nhìn trực quan hơn về hệ Mặt Trời cũng như các hành tinh trong hệ Mặt Trời.

### Tác nhân của hệ thống

Tác nhân chính của hệ thống là người dùng hệ thống.

### Yêu cầu của hệ thống:

Hệ thông đảm bảo chất lượng của những chức năng cung cấp tới người dùng, tương thích được với nhiều phiên bản máy của người dùng, tối ưu được dung lượng lưu trữ, hoạt động trơn tru trên các thế hệ máy cũ.

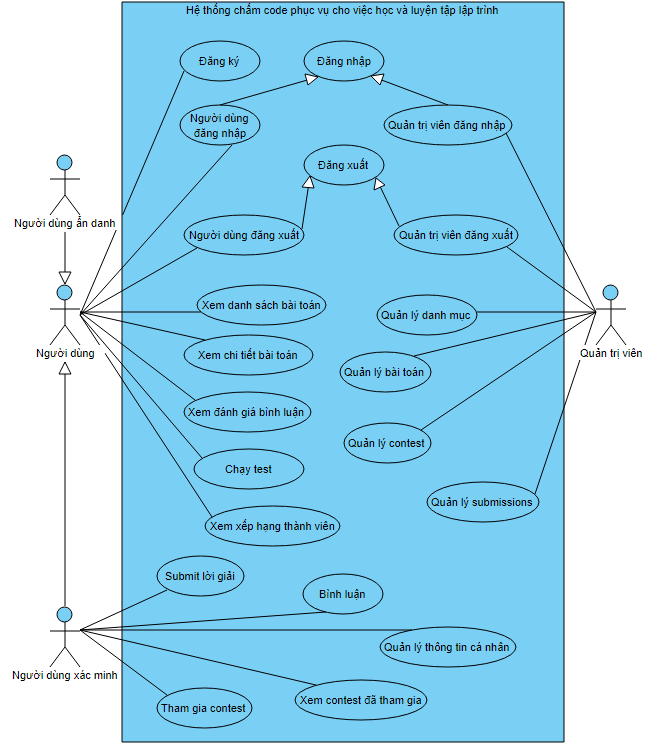
### Phạm vi của hệ thống

Người dùng của hệ thống có thể:

* Xem bản đồ sao.
* Chuyển đổi giữa các bản đồ.
* Xem chuyển động xoay quanh quỹ đạo của hành tinh.
* Thực hiện giả lập chuyển động quanh quỹ đạo của hành tinh.
* Điều chỉnh, giả lập thời gian trong chuyển động quỹ đạo.
* Xem chuyển động của các hành tinh trong hệ Mặt Trời.
* Điều chỉnh vị trí, góc quay của camera.
* Hiển thị thông tin của các hành tinh trong hệ Mặt Trời.
* Xem chi tiết model của các hành tinh trong hệ Mặt Trời.
* Thực hiện chuyển động quay quanh trục cho model của các hành tinh.
* Điều chỉnh hướng cho các model
* Điều chỉnh âm lượng cho toàn bộ hệ thống.
* Điều chỉnh đồ họa cho toàn bộ hệ thống.
* Điều chỉnh ngôn ngữ cho toàn bộ hệ thống.
* Chạy test: thực thi chương trình với đầu vào mong muốn

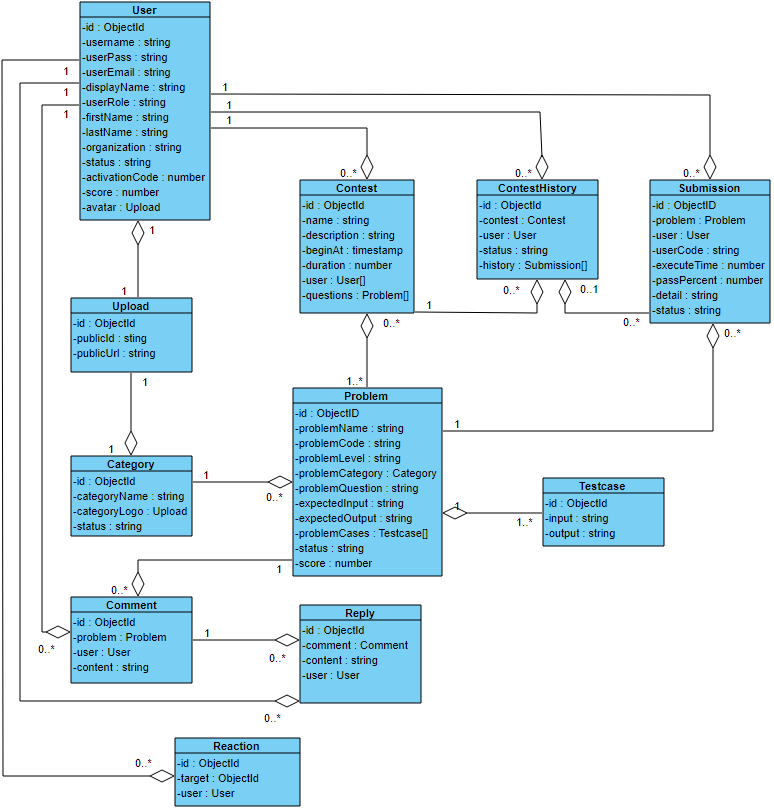
## Thiết kế hệ thống

### 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 xem bản đồ sao

#### Usecase chi tiết

#### Kịch bản

Bảng . Kịch bản xem bản đồ sao

|  |  |
| --- | --- |
| Tên usecase | Xem bản đồ sao |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng lựa chọn chức năng xem bản đồ sao |
| Hậu điều kiện | Người dùng có thể sử dụng bản đồ sao |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem bản đồ sao.  6. Giao diện xem bản đồ sao hiện lên với bản đồ sao dưới dạng 3D cube map, button chuyển đổi bản đồ sao.  7.Người dùng dùng thao tác vuốt để di chuyển tầm nhìn và góc nhìn cho phù hợp để xem được các chòm sao. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự

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

### Module chuyển đồi bản đồ sao

#### Usecase chi tiết

#### Kịch bản

Bảng . Kịch bản module đăng nhập

|  |  |
| --- | --- |
| Tên usecase | Đăng nhập |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sủ dụng chức năng xem bản đồ sao |
| Hậu điều kiện | Người dùng chuyển đổi bản đồ sao thành công |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem bản đồ sao.  6. Giao diện xem bản đồ sao hiện lên với bản đồ sao dưới dạng 3D cube map, button chuyển đổi bản đồ sao.  7. Người dùng chọn vào button thay đổi bản đồ.  8. Hệ thống hiển thị lại bản đồ mới thay thế cho bản đồ trước đó. |
| Ngoại lệ |  |

#### Biểu đồ tuần tự

### Module giả lập quỹ đạo cho hành tinh

#### Usecase chi tiết

#### Kịch bản

Bảng . Kịch bản module giả lập quỹ đạo cho hành tinh

|  |  |
| --- | --- |
| Tên usecase | Giả lập quỹ đạo cho hành tinh |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng lựa chọn chức năng giả lập quỹ đạo cho hành tinh |
| Hậu điều kiện | Quỹ đạo của hành tinh được thay đổi |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng giả lập quỹ đạo cho hành tinh.  6. Giao diện giả lập quỹ đạo cho hành tinh hiện lên với Mặt Trời và Trái Đất(chủ thể quỹ đạo), button hệ thống button điểu chỉnh, slider thay đổi thuộc tính cho camera.  7. Người dùng sử dụng thao tác kéo thả để thay đổi quỹ đạo cho hành tinh.  8. Hệ thống liên tục hiển thị quỹ đạo mới cho hành tinh dựa trên hướng kéo và khoảng cách kéo mà người dùng tác độn lên hành tinh.  9. Người dùng kết thúc thao tác kéo thả để cố định quỹ đạo cho hành tinh.  10. Hệ thống hiển thị quỹ đao cho hành tinh dưới dạng lineRenderer.  11. Người dùng nhấn vào button bắt đầu để thực hiện chuyển độn quay quanh quỹ đạo cho hành tinh.  12. Hành tinh di chuyển trên quỹ đạo mới được tạo  13. Người dùng nhấn button đặt lại để đưa hành tinh về trạng thái ban đầu.  14. Thực hiện lặp lại các bước 8 đến 14 để điều chỉnh lại quỹ đạo cho hành tinh. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự

### Module giả lập thời gian

#### Usecase chi tiết

#### Kịch bản

Bảng . Kịch bản giả lập thời gian

|  |  |
| --- | --- |
| Tên usecase | Giả lập thời gian |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng giả lập quỹ đạo |
| Hậu điều kiện | Thay đổi thành công thời gian tương đối trong hệ thống |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”..  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng giả lập quỹ đạo cho hành tinh.  6. Giao diện giả lập quỹ đạo cho hành tinh hiện lên với Mặt Trời và Trái Đất(chủ thể quỹ đạo), button hệ thống button điểu chỉnh, slider thay đổi thuộc tính cho camera.  7. Người dùng sử dụng thao tác kéo thả để thay đổi quỹ đạo cho hành tinh.  8. Hệ thống liên tục hiển thị quỹ đạo mới cho hành tinh dựa trên hướng kéo và khoảng cách kéo mà người dùng tác độn lên hành tinh.  9. Người dùng kết thúc thao tác kéo thả để cố định quỹ đạo cho hành tinh.  10. Hệ thống hiển thị quỹ đao cho hành tinh dưới dạng lineRenderer.  11. Người dùng nhấn vào button bắt đầu để thực hiện chuyển độn quay quanh quỹ đạo cho hành tinh.  12. Hành tinh di chuyển trên quỹ đạo mới được tạo  13. Người dùng nhấn vào button thay đổi thời gian cho tới khi đáp ứng được thời gian mong muốn.  14. Hành tinh chuyển động trên quỹ đạo với tốc đô tương ứng với gia tốc thời gian. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự

### Module thay đổi trường nhìn

#### Usecase chi tiết

#### Kịch bản

Bảng . Kịch bản module thay đổi trường nhìn

|  |  |
| --- | --- |
| Tên usecase | Thay đổi trường nhìn |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng mô phỏng quỹ đạo |
| Hậu điều kiện | Trường nhìn của camera được thay đổi |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng giả lập quỹ đạo cho hành tinh.  6. Giao diện giả lập quỹ đạo cho hành tinh hiện lên với Mặt Trời và Trái Đất(chủ thể quỹ đạo), button hệ thống button điểu chỉnh, slider thay đổi thuộc tính cho camera.  7. Người dùng sử dụng thao tác kéo để thay đổi giá trị của slider “Fiew Of View” cho đến khi có trường nhìn phù hợp.  8. Hệ thống thay đổi trường nhìn liên tục theo giá trị của slider mà người dùng chọn. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự

### Module xem hệ Mặt Trời

#### Usecase chi tiết

#### Kịch bản

Bảng . Kịch bản module xem hệ Mặt Trời

|  |  |
| --- | --- |
| Tên usecase | Xem hệ Mặt Trời |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng xem hệ Mặt Trời |
| Hậu điều kiện | Người dùng xem được các hành tinh trong hệ Mặt Trời |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem hệ Mặt Trời  6. Hệ thống hiển thị giao diện hệ Mặt Trời với các thiên thể quay trên quỹ đạo quanh Mặt Trời. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự

### Module xem thông tin các hành tinh

#### Usecase chi tiết

#### Kịch bản

Bảng . Kịch bản module xem thông tin các hành tinh

|  |  |
| --- | --- |
| Tên usecase | Xem thông tin các hành tinh |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng xem hệ Mặt Trời |
| Hậu điều kiện | Người dùng xem được thông tin của các hành tinh |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem hệ Mặt Trời  6. Hệ thống hiển thị giao diện hệ Mặt Trời với các thiên thể quay trên quỹ đạo quanh Mặt Trời.  7. Người dùng chọn button tên một hành tinh đang được hiển thị.  8. Hệ thống tự động di chuyển camera đến hành tinh đó và di chuyển bám theo chuyển động của hành tinh đồng thời hiển thị lên màn hình các thông tin cơ bản của hành tinh.  9. Người dùng chọn thoát để trở về trạng thái ban đầu.  10. Lặp lại các bước từ 7 đến 9 để thay đổi thông tin của các hành tinh. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự

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

### Module thay đổi góc nhìn camera

#### Usecase chi tiết

#### Kịch bản

Bảng . Kịch bản module

|  |  |
| --- | --- |
| 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 ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem hệ Mặt Trời  6. Hệ thống hiển thị giao diện hệ Mặt Trời với các thiên thể quay trên quỹ đạo quanh Mặt Trời.  7. Người dùng sử dụng thao tác zoom in, zoom out để thay đổi khoảng cách tương đối của camera với Mặt Trời.  8. Hệ thống thực hiện di chuyển camera tương ứng với độ zoom mà người dùng thực hiện.  9. Người dùng thực hiện thao tác vuốt để thay đổi hướng nhìn của camera.  10. Hệ thống thực hiện thay đổi góc nhìn của camera dựa trên khoảng cách và hướng vuốt của người dùng. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự

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

### Module xem chi tiết model

#### Usecase chi tiết

#### Kịch bản

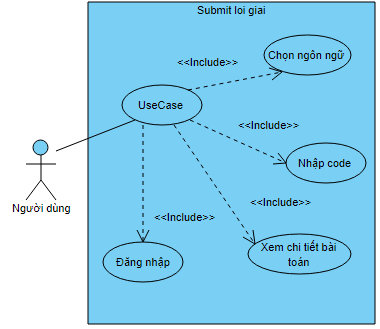
Bảng . Kịch bản module xem chi tiết model

|  |  |
| --- | --- |
| Tên usecase | Xem chi tiết model |
| Actor | Người dùng |
| Tiền điều kiện | Người dùng sử dụng chức năng xem chi tiết model |
| Hậu điều kiện | Người dùng xem được model của các hành tinh trong hệ Mặt Trời |
| Kịch bản | 1. Người dùng truy cập vào ứng dụng  2. Giao diện loading hiện lên với button “click to open”.  3. Người dùng click vào button “click to open” để truy cập giao diện chọn chức năng.  4. Giao diện chọn chức năng hiện lên với các button dưới dạng scroll view để người dùng có thể chọn chức năng.  5. Người dùng chọn chức năng xem chi tiết model.  6. Hệ thống hiển thị giao diện xem chi tiết model với model mặc định là model của Mặt Trời.  7. Người dùng chọn các buttton trái phải để thay đổi giữa các model.  8. Hệ thống thực hiện thay đổi các model tương ứng dựa trên danh sách các hành tinh được cài đặt trước. |
| Ngoại lệ | Không có ngoại lệ |

#### Biểu đồ tuần tự

### Module submit lời giải

#### Usecase chi tiết



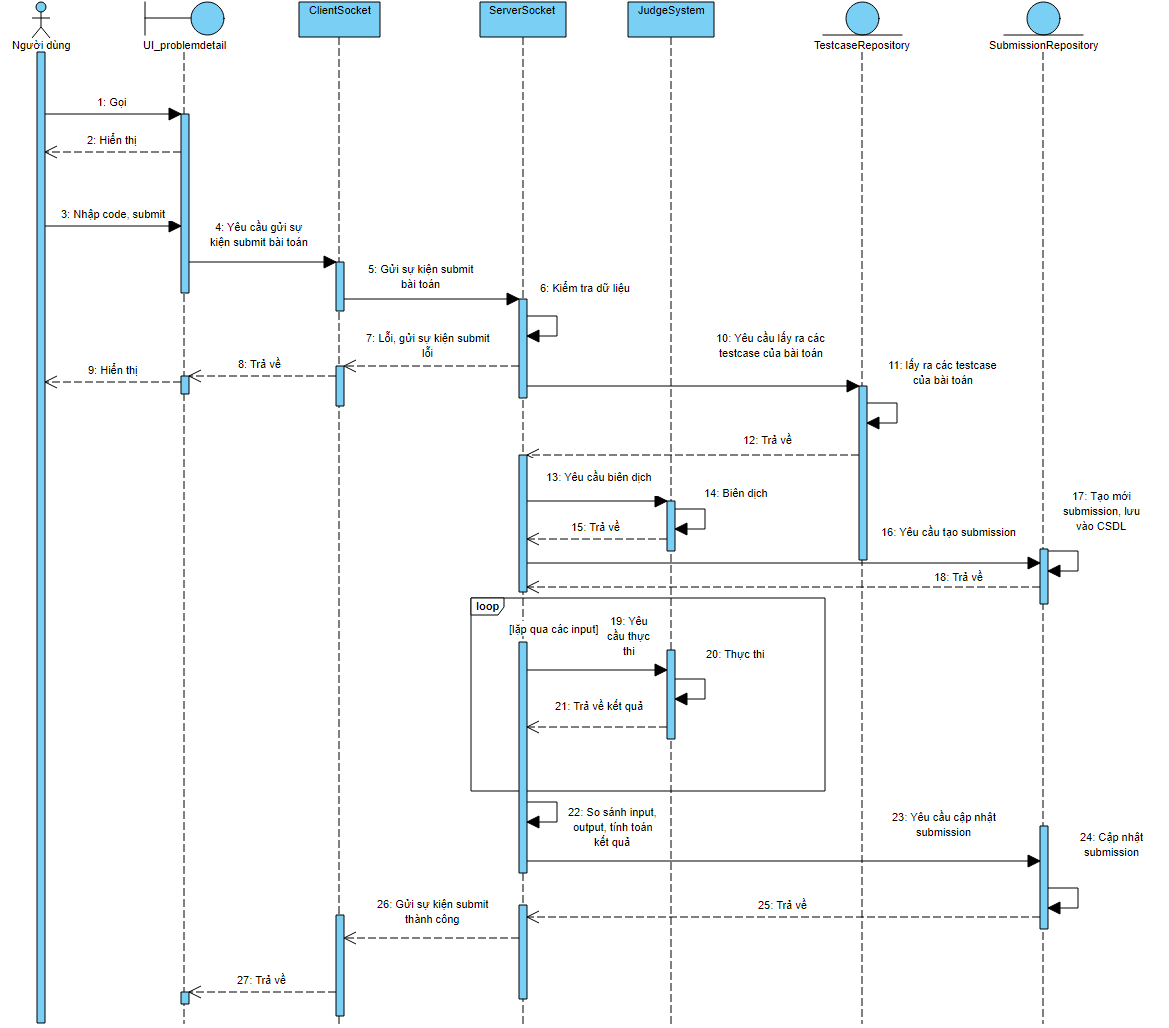
Hình . Usecase chi tiết module submit lời giải

#### 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

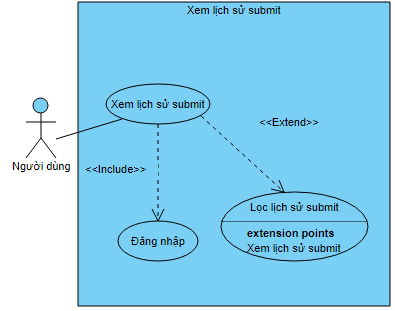


Hình . Biểu đồ tuần tự module submit lời giải (2)

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

### Module xem lịch sử submit

#### Usecase chi tiết



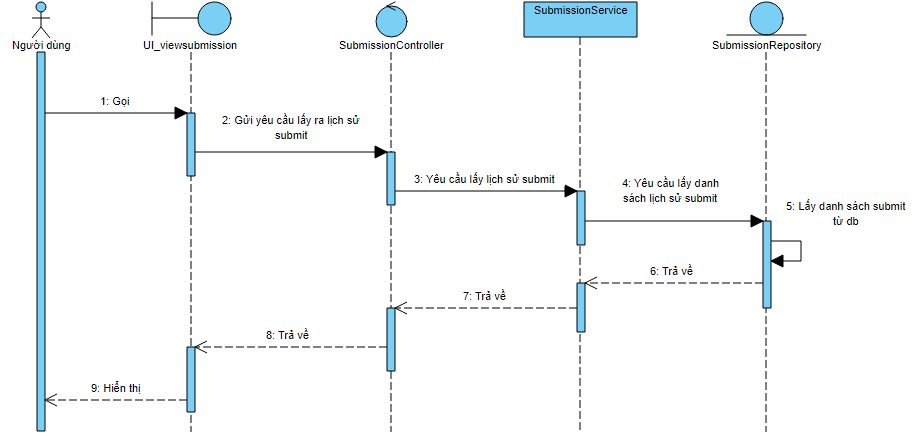
Hình . Usecase chi tiết module xem lịch sử submit

#### 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. |

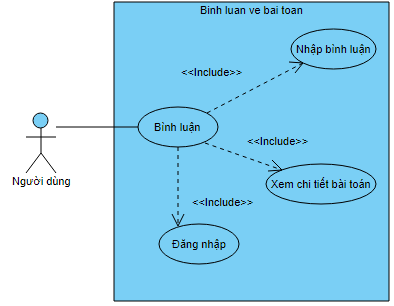
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module xem lịch sử submit

### Bình luận về bài toán

#### Usecase chi tiết



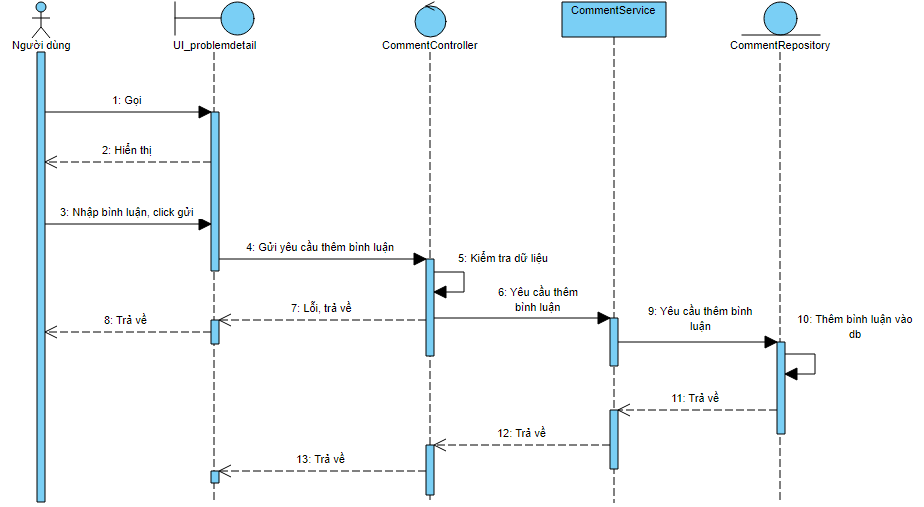
Hình . Usecase chi tiết module bình luận về bài toán

#### 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



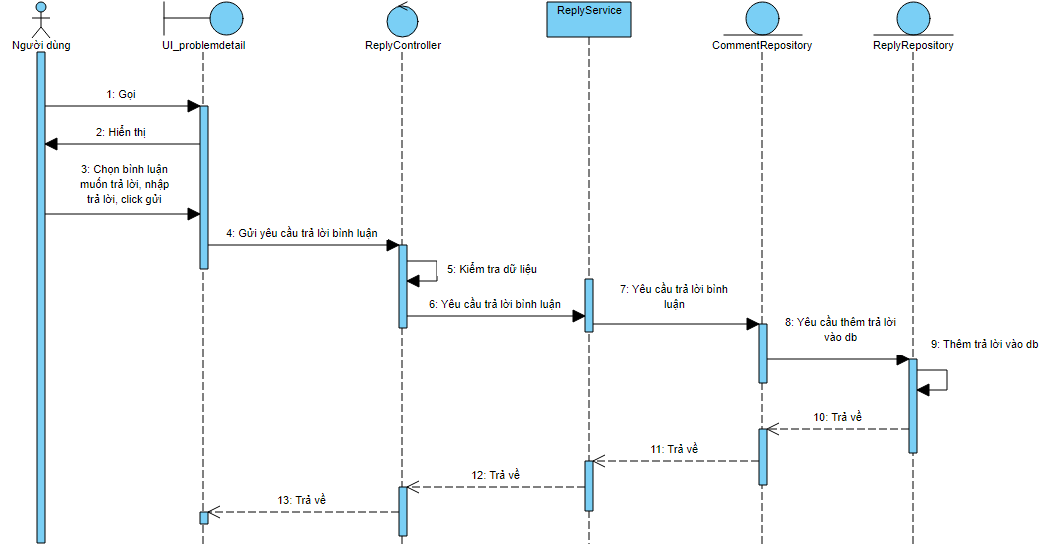
Hình . Usecase chi tiết module trả lời bình luận

#### 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 |

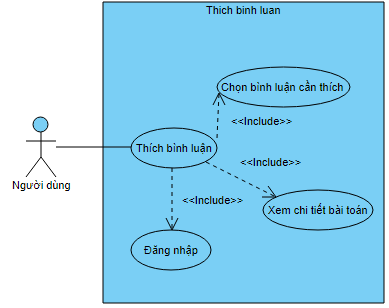
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module trả lời bình luận

### Module thích bình luận

#### Usecase chi tiết



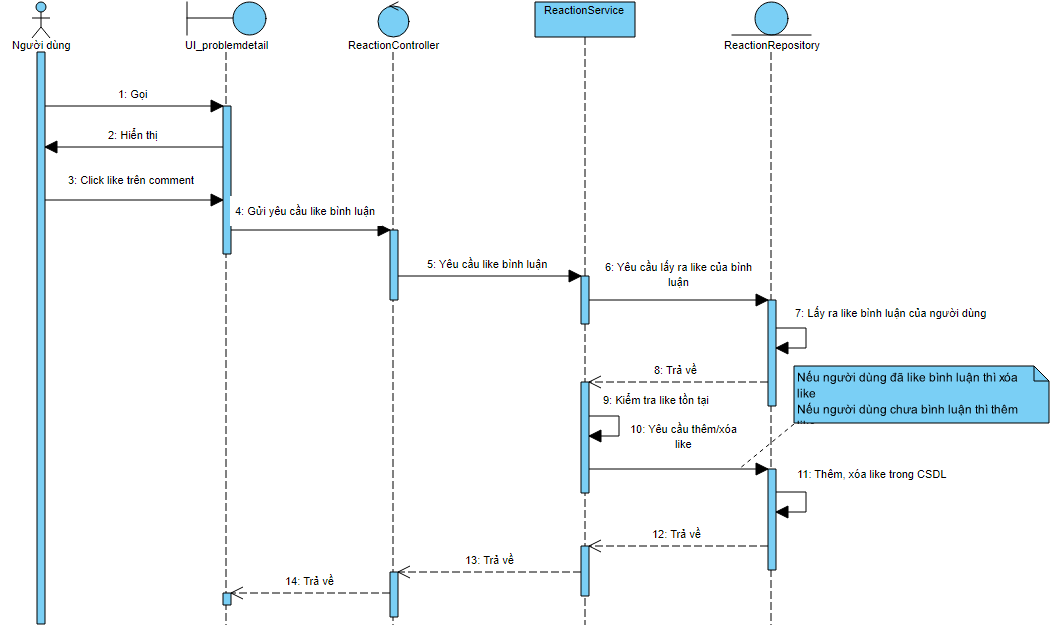
Hình . Usecase chi tiết module thích bình luận

#### 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 |

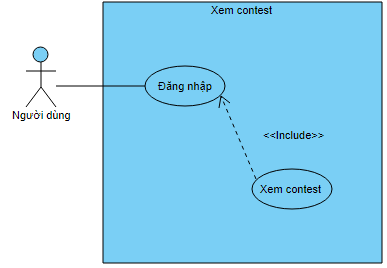
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module thích bình luận

### Module xem contest

#### Usecase tổng quan



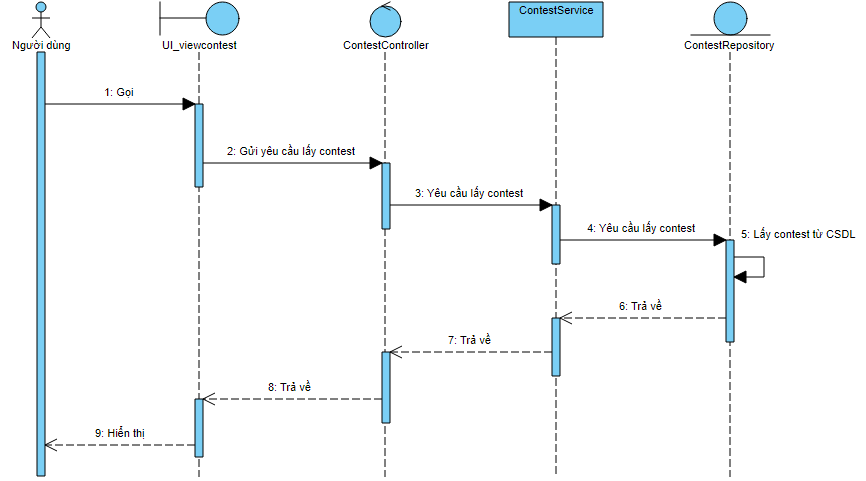
Hình . Usecase chi tiết module xem contest

#### 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. |

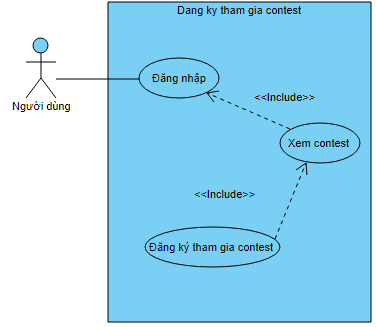
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module xem contest

### Đăng ký tham gia contest

#### Usecase chi tiết



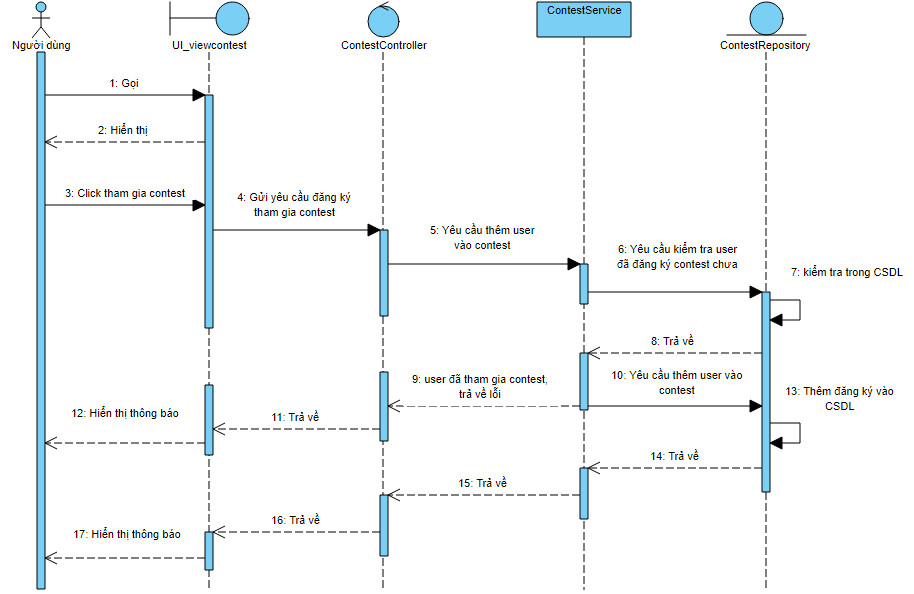
Hình . Usecase chi tiết module đăng ký tham gia contest

#### 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 |

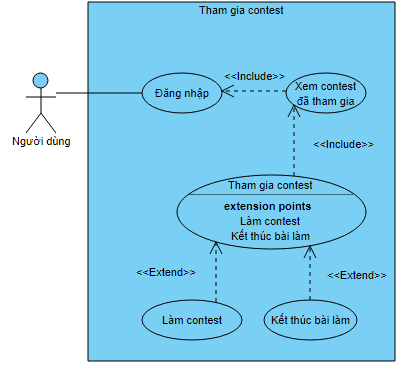
#### Biểu đồ tuần tự



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

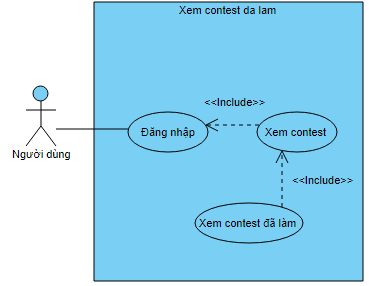


Hình . Biểu đồ tuần tự module tham gia contest (3)

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

### Xem contest đã làm

#### Usecase chi tiết



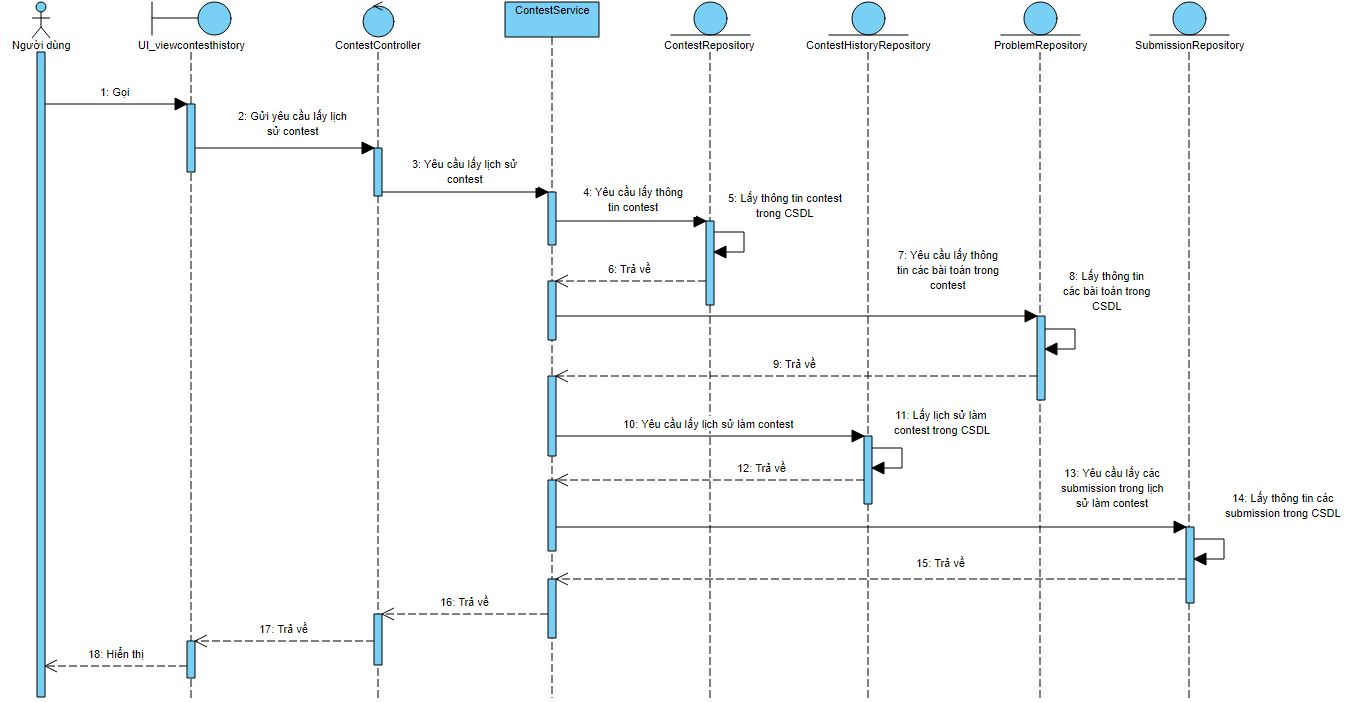
Hình . Usecase chi tiết module xem contest đã làm

#### 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 |

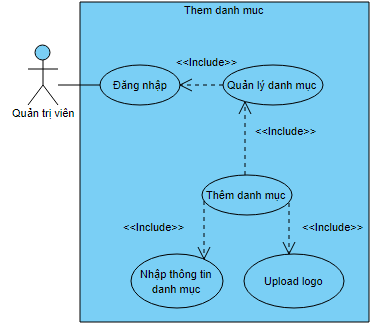
#### Biểu đồ tuần tự



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

### Module thêm danh mục

#### Usecase chi tiết



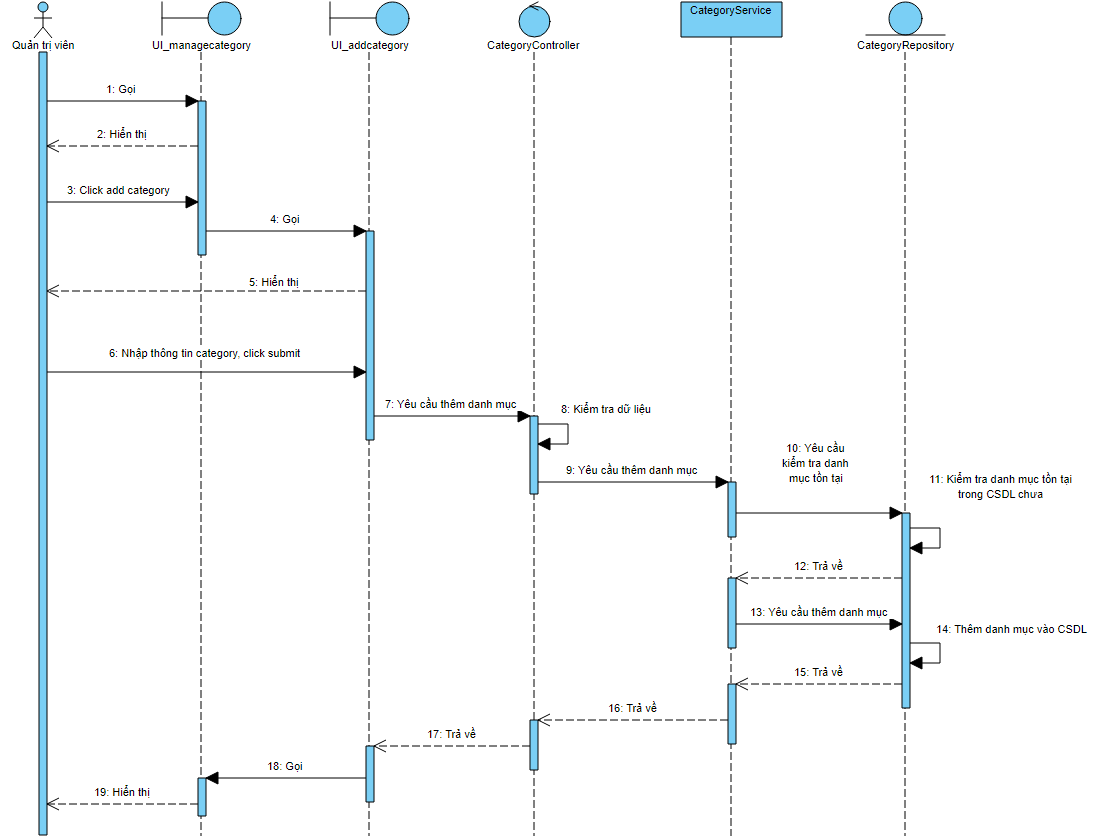
Hình . Usecase chi tiết module thêm danh mục

#### 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 |

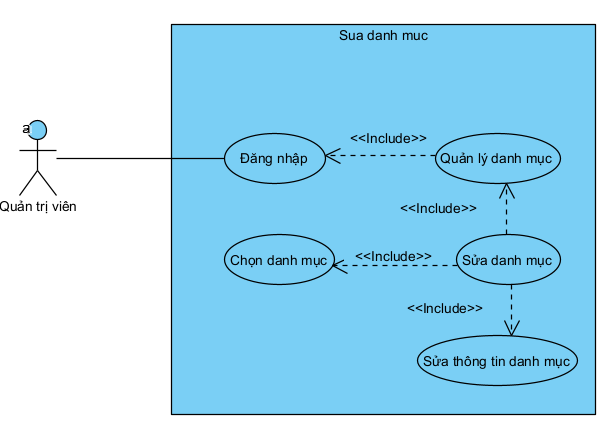
#### Biểu đồ tuần tự



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

### Module sửa danh mục

#### Usecase chi tiết



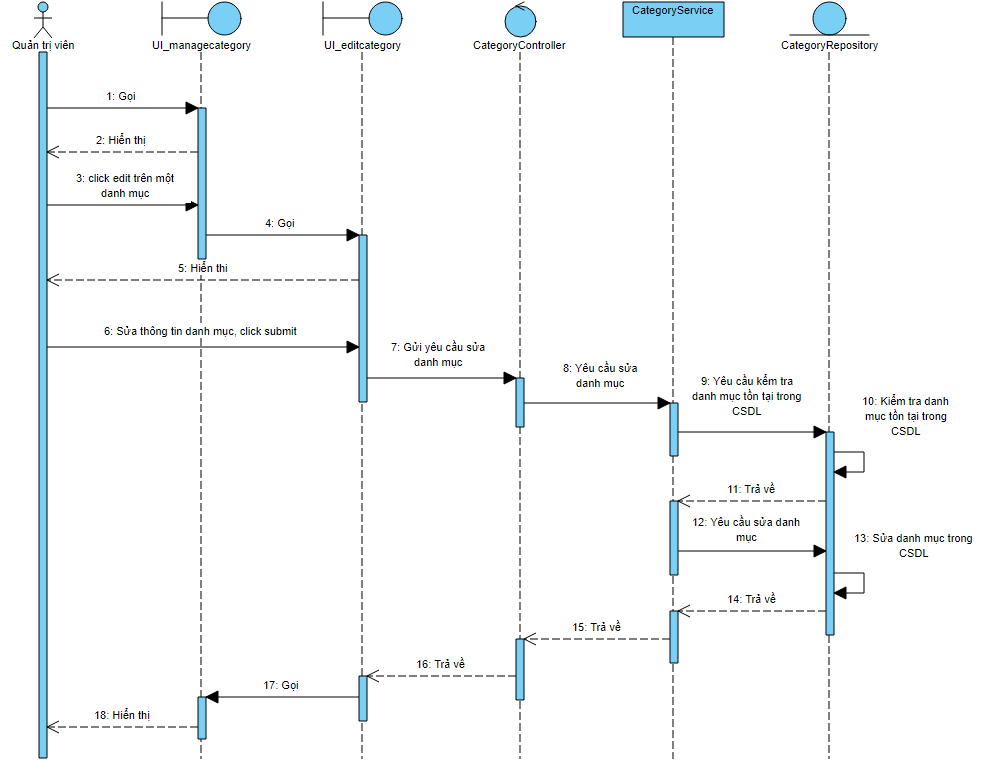
Hình . Usecase chi tiết module sửa danh mục

#### 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



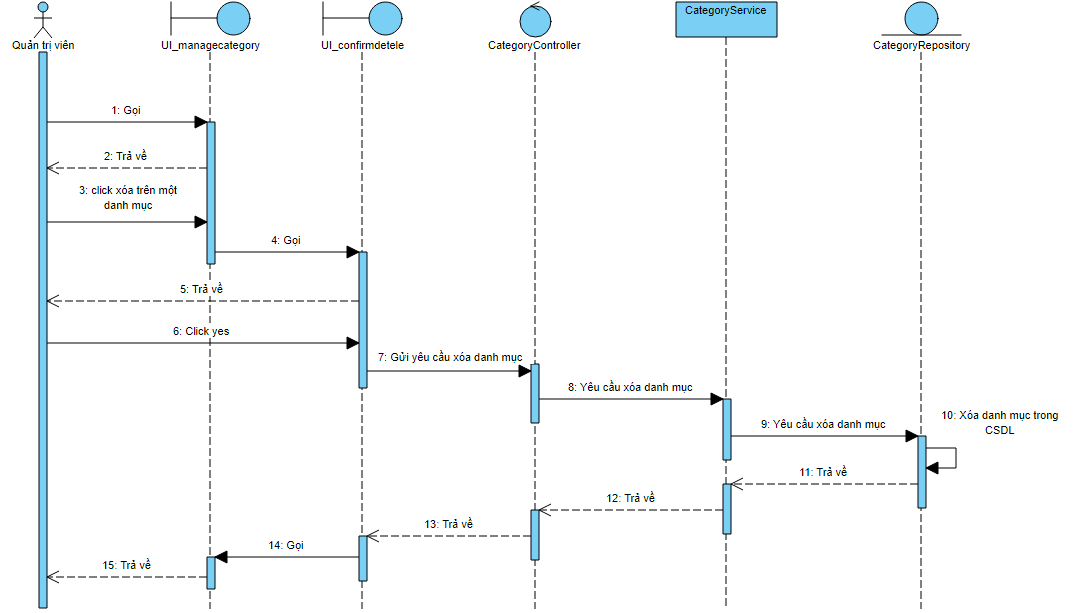
Hình . Usecase chi tiết module xóa danh mục

#### 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 |

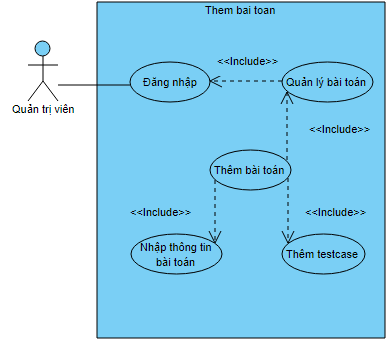
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module xóa danh mục

### Module thêm bài toán

#### Usecase chi tiết



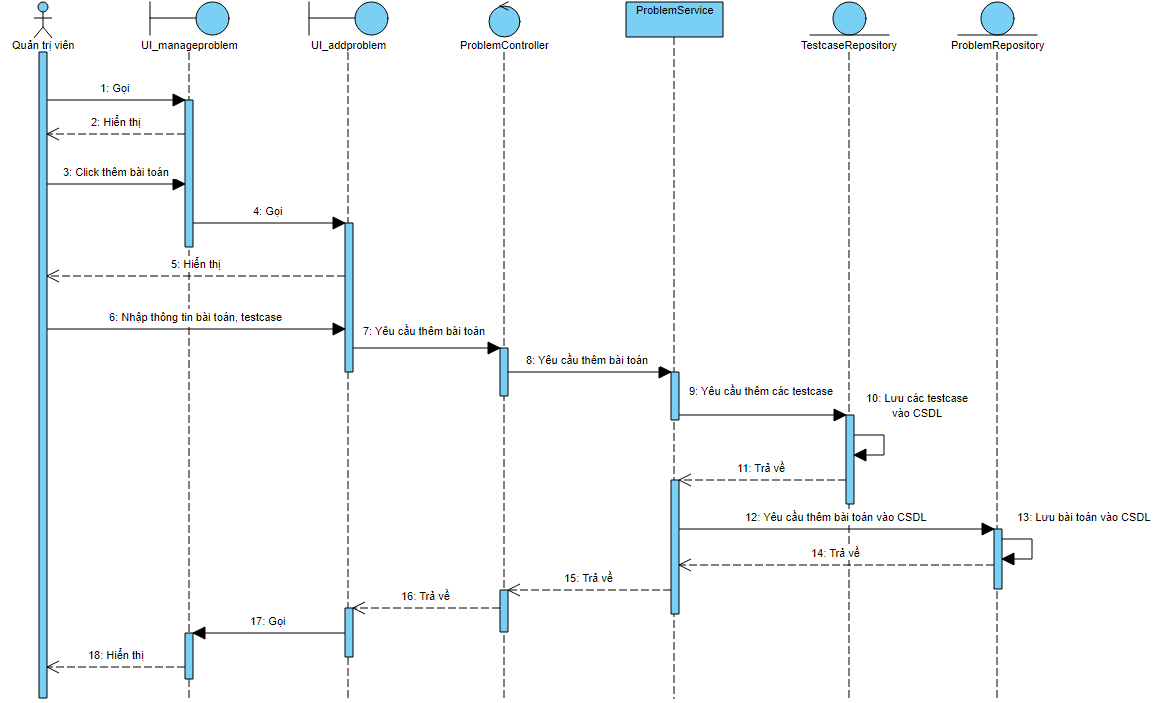
Hình . Usecase chi tiết module thêm bài toán

#### 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 |

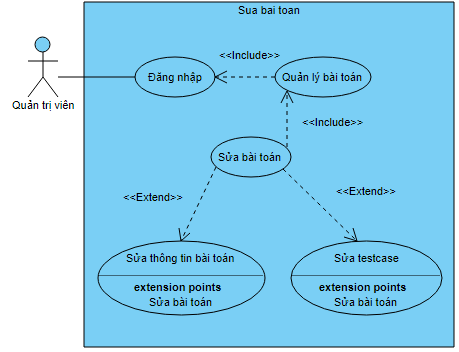
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module thêm bài toán

### Module sửa bài toán

#### Usecase chi tiết



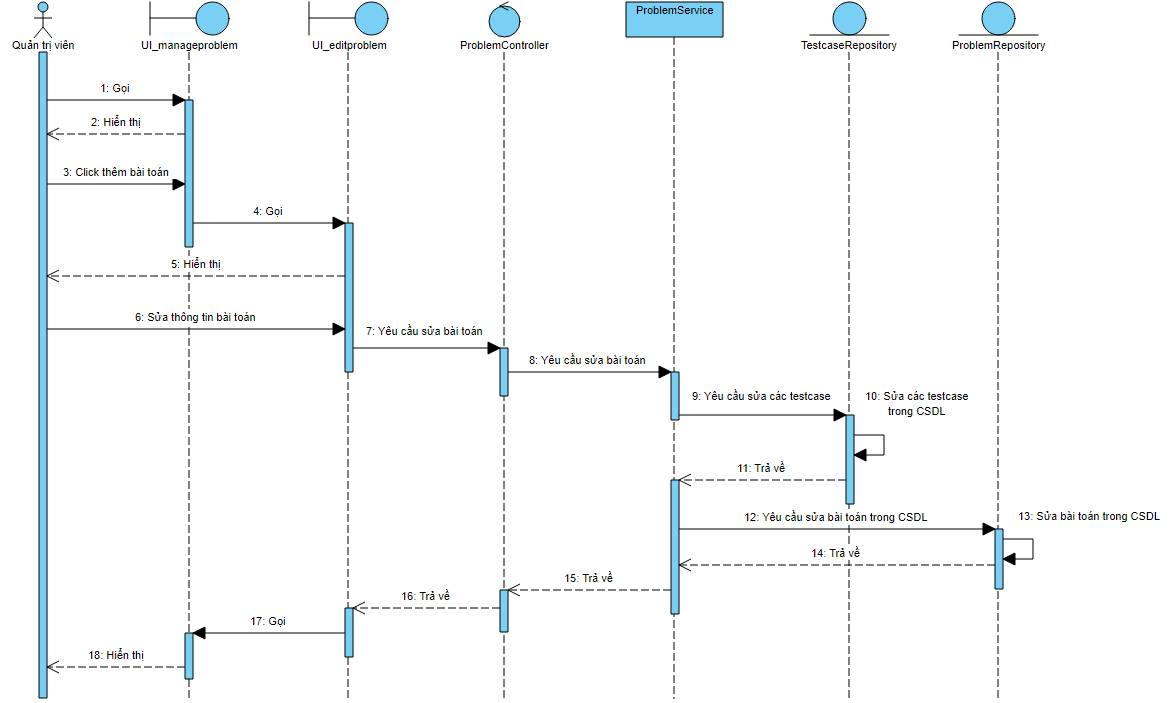
Hình . Usecase chi tiết module sửa bài toán

#### 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 |

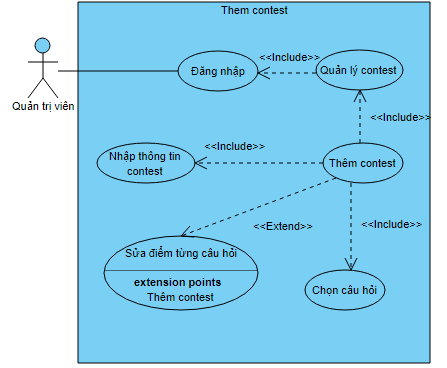
#### Biểu đồ tuần tự



Hình . Biểu đồ tuần tự module sửa bài toán

### Module thêm contest

#### Usecase chi tiết



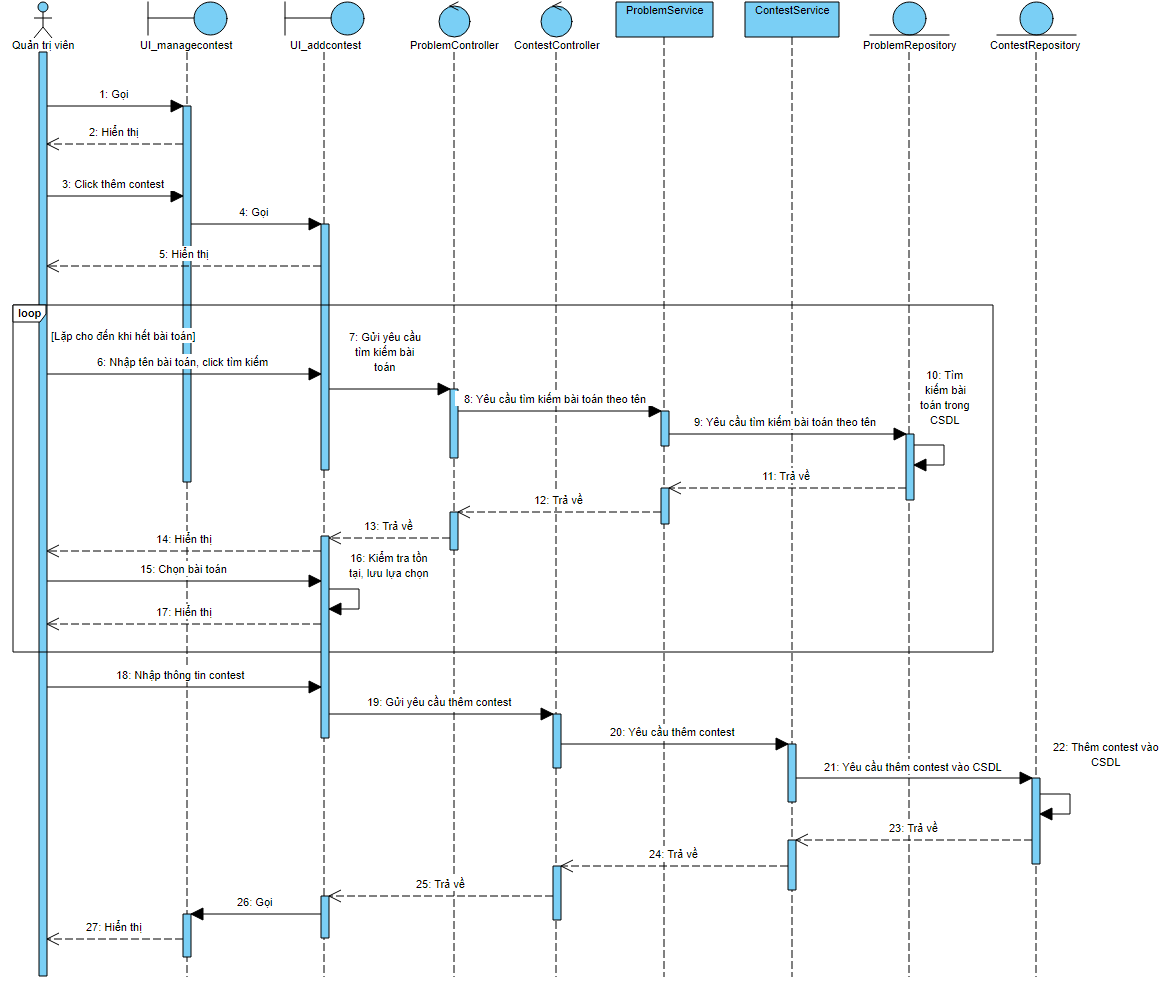
Hình . Usecase chi tiết module thêm contest

#### 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

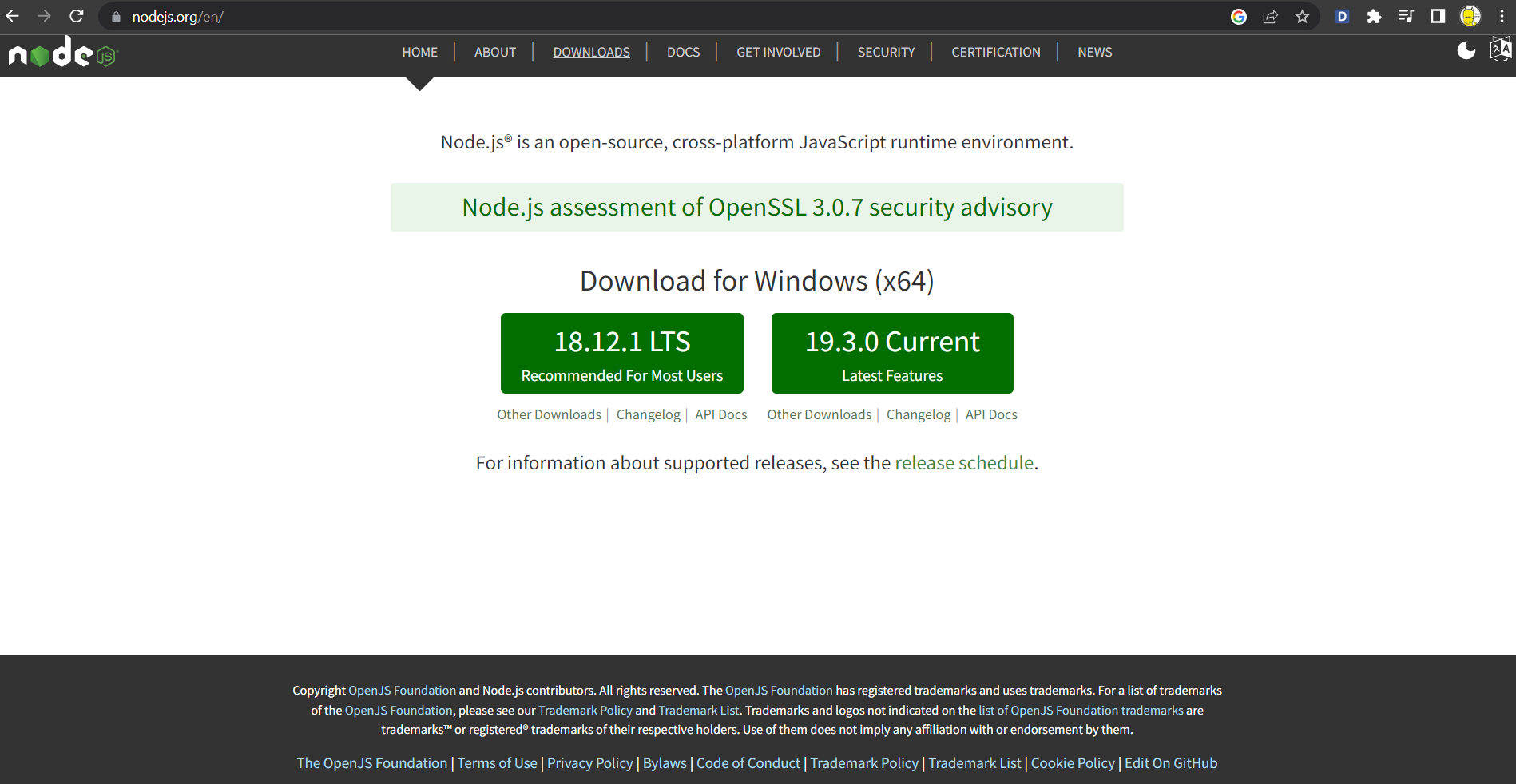
## Một số công cụ, thư viện hỗ trợ

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

## Cài đặt

### 3.3.1 Cài đặt Nodejs

Bước 1: Tải NodeJS



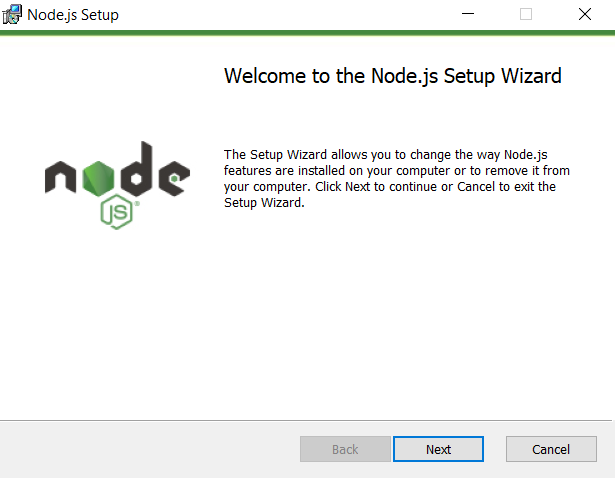
Hình . Hướng dẫn tải Nodejs

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

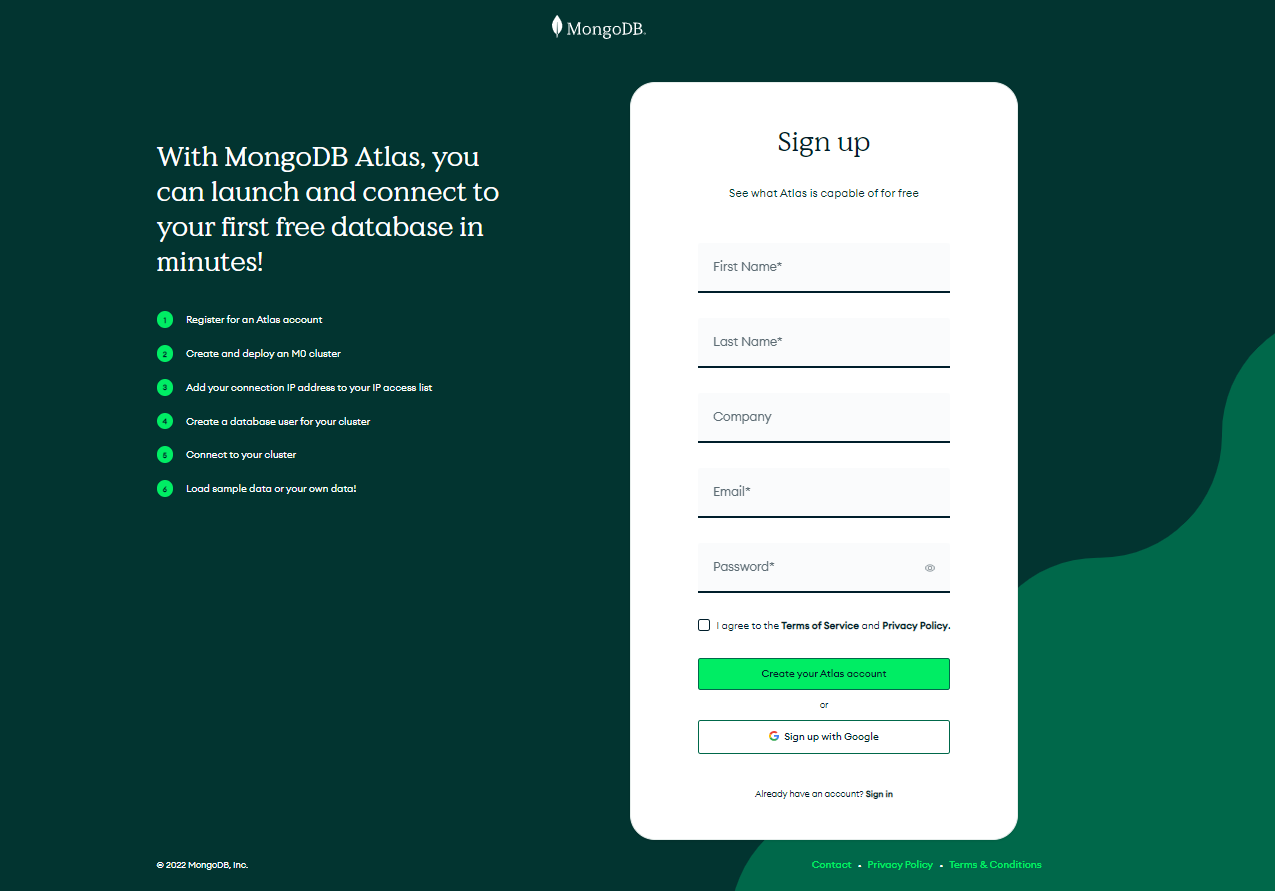
Sau khi đã tải về, hãy tiến hành cài đặt Nodejs (có thể để các options mặc định trong khi cài đặt).



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

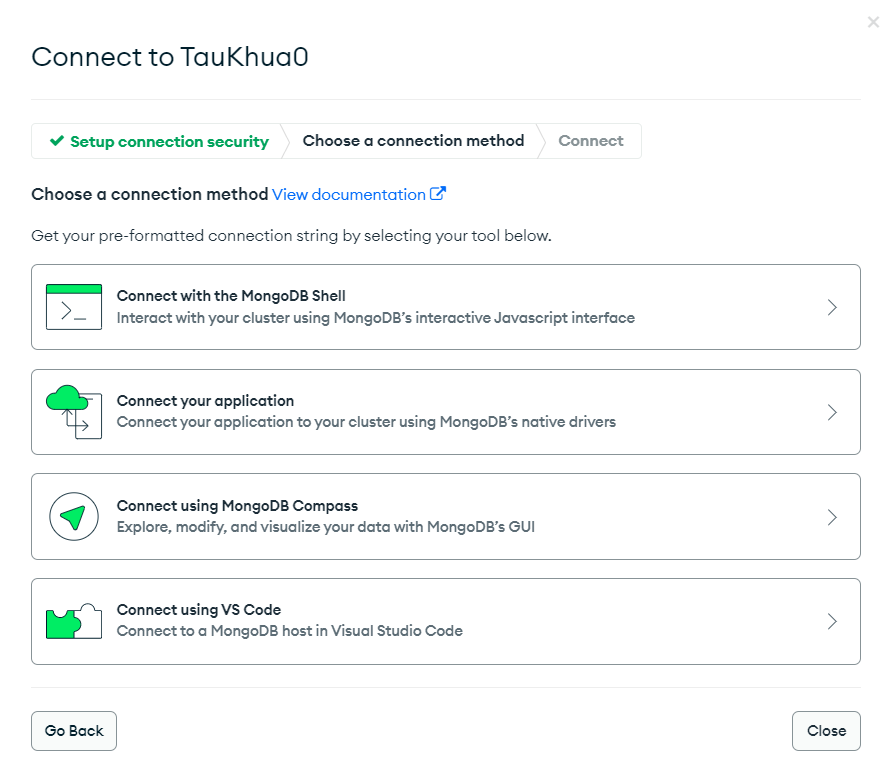
Bước 2: Điền đầy đủ thông tin rồi click create your Atlas Account, hoặc đơn giản hơn click nút sign up with Google để đăng ký bằng tài khoản google.

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

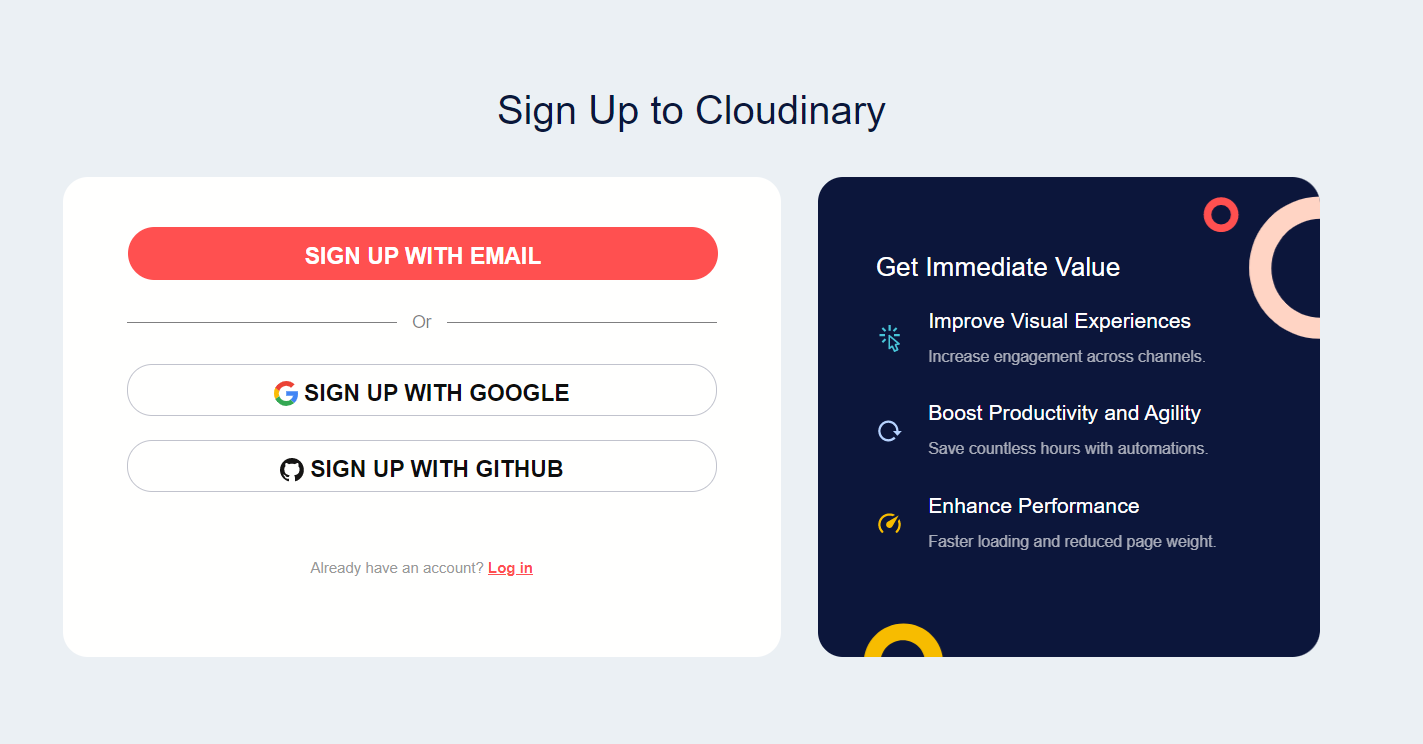


Hình . Giao diện trang hướng dẫn kết nối

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:

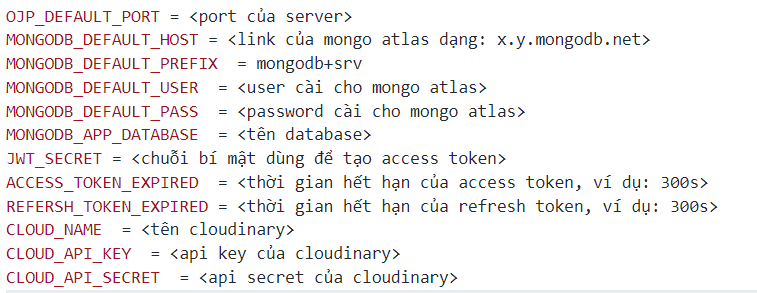
Sử dụng câu lệnh: cd <directory path>

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:

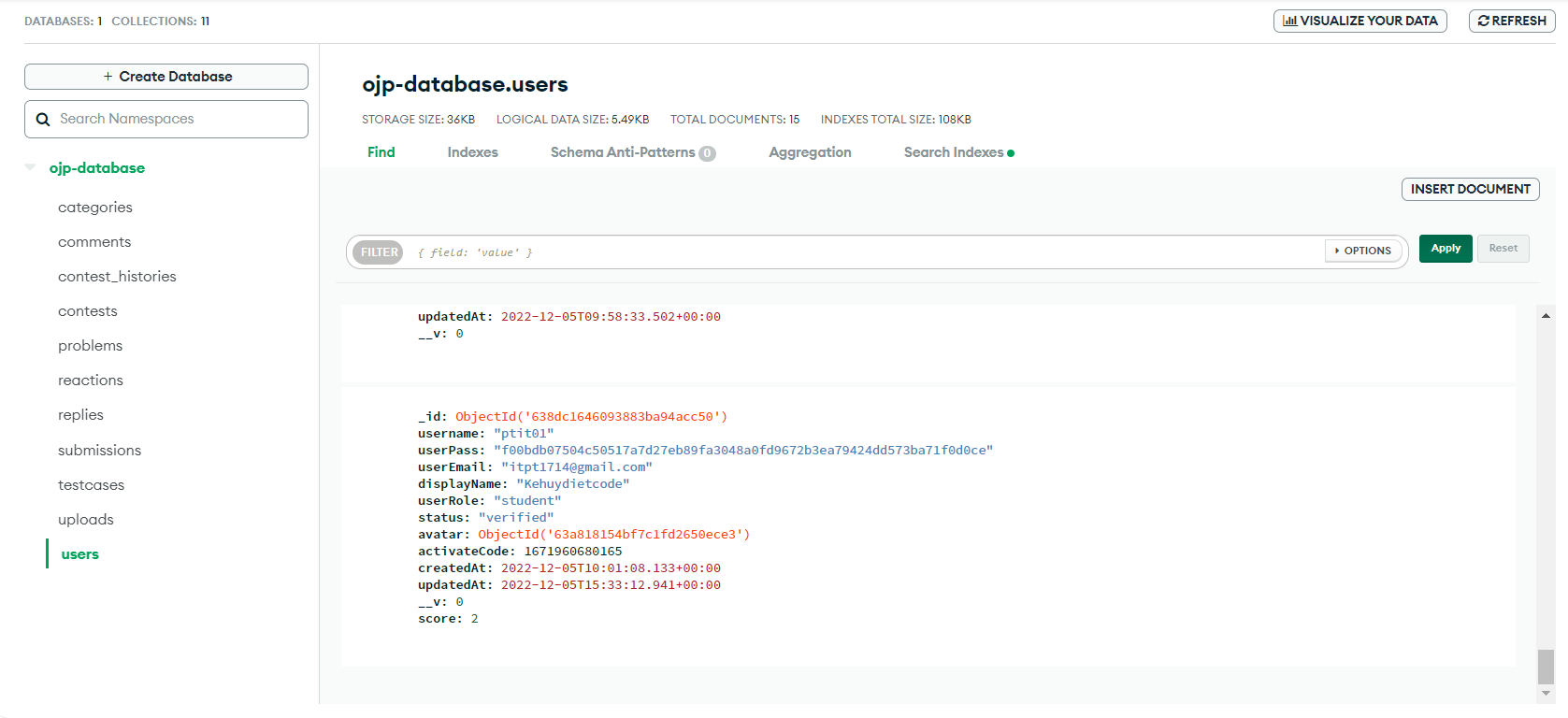
Sử dụng lệnh: cd <directory path>

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

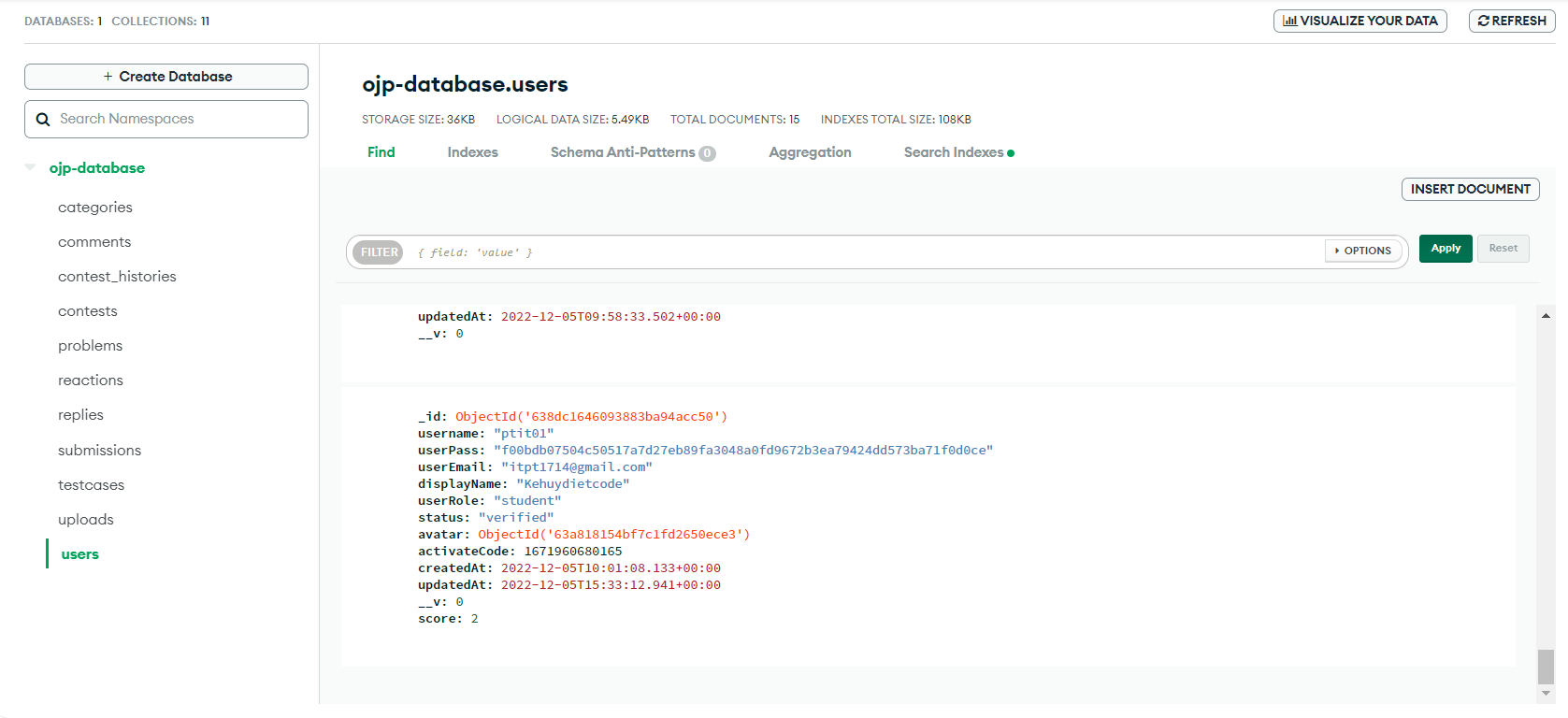
### Kết quả cài đặt database



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

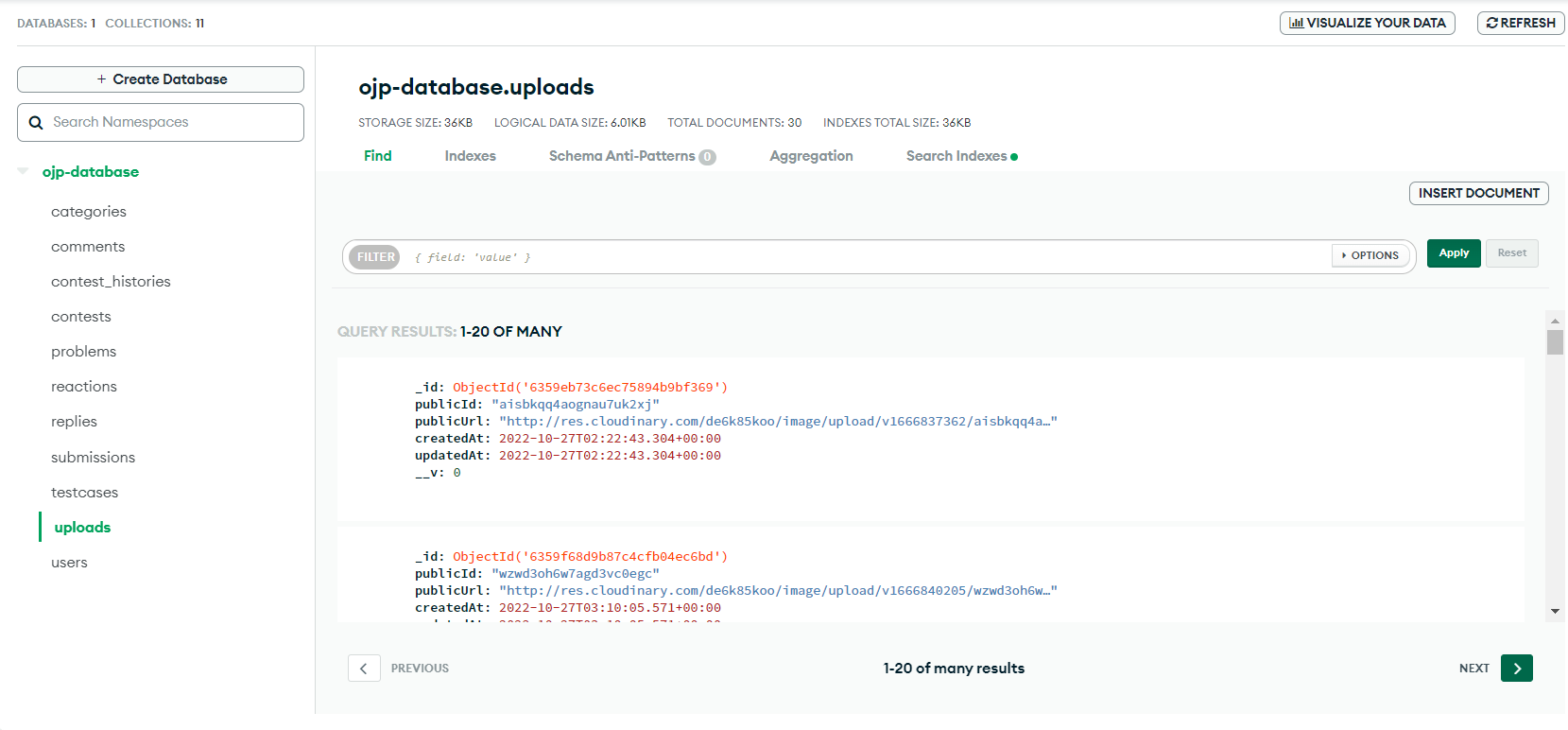


Hình . Kết quả cài đặt bảng users

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.

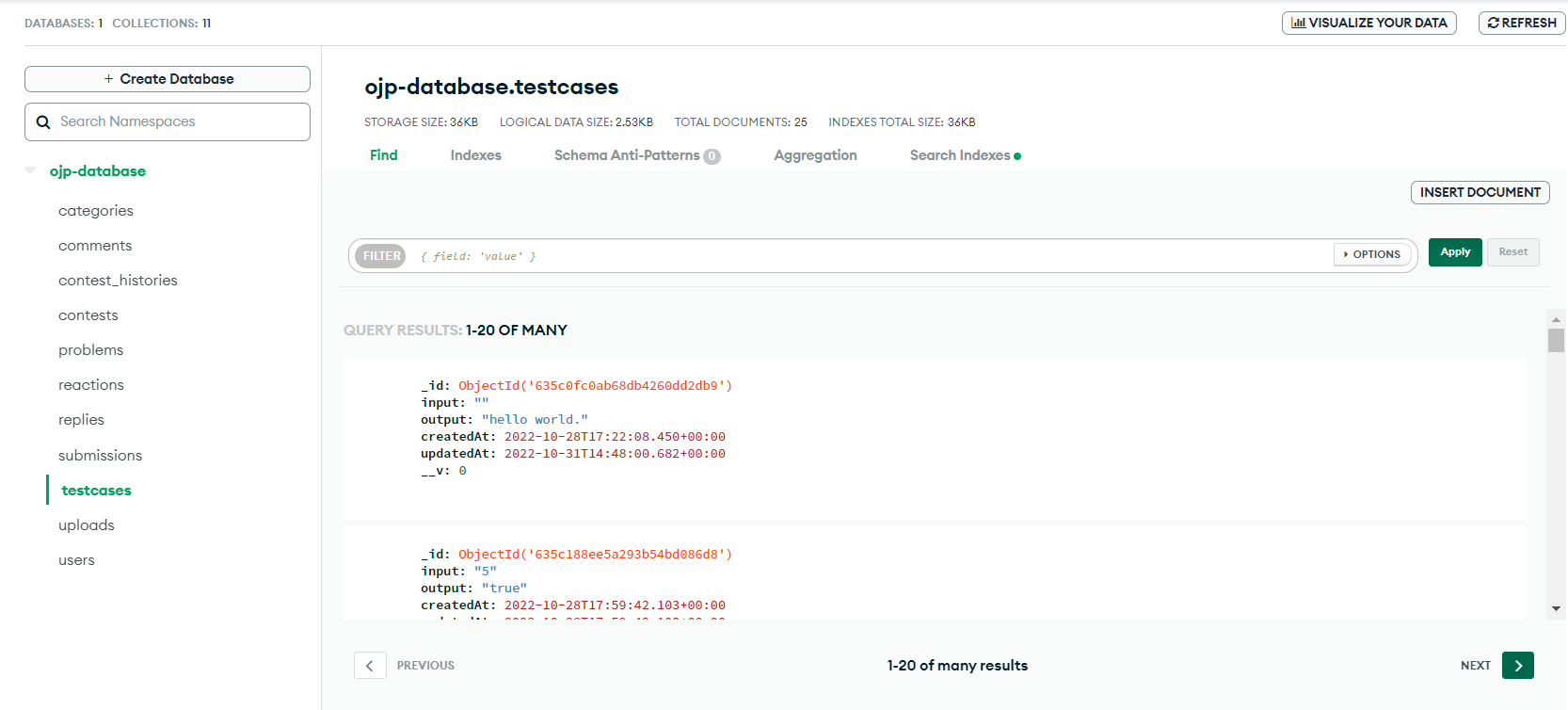
Bảng uploads



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.

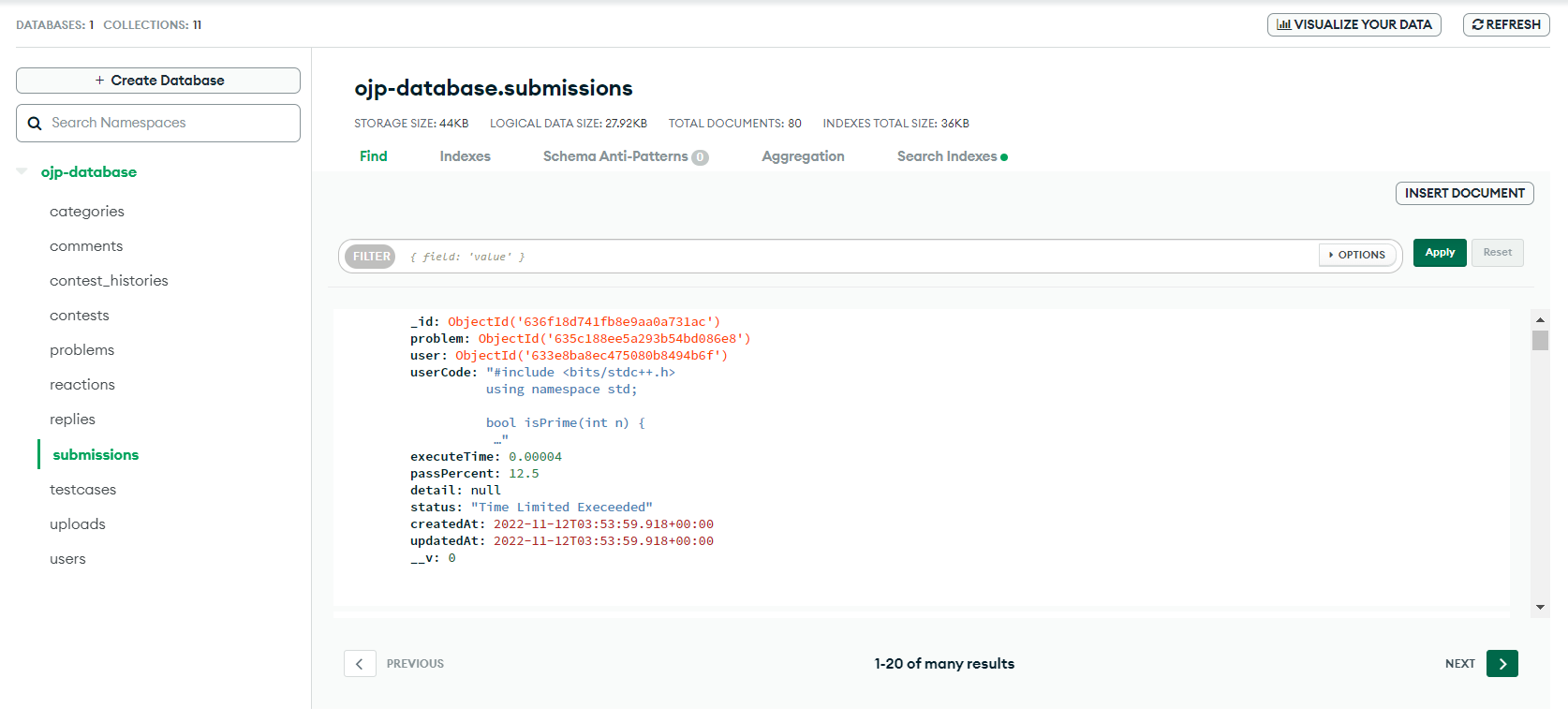
Bảng testcases:



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

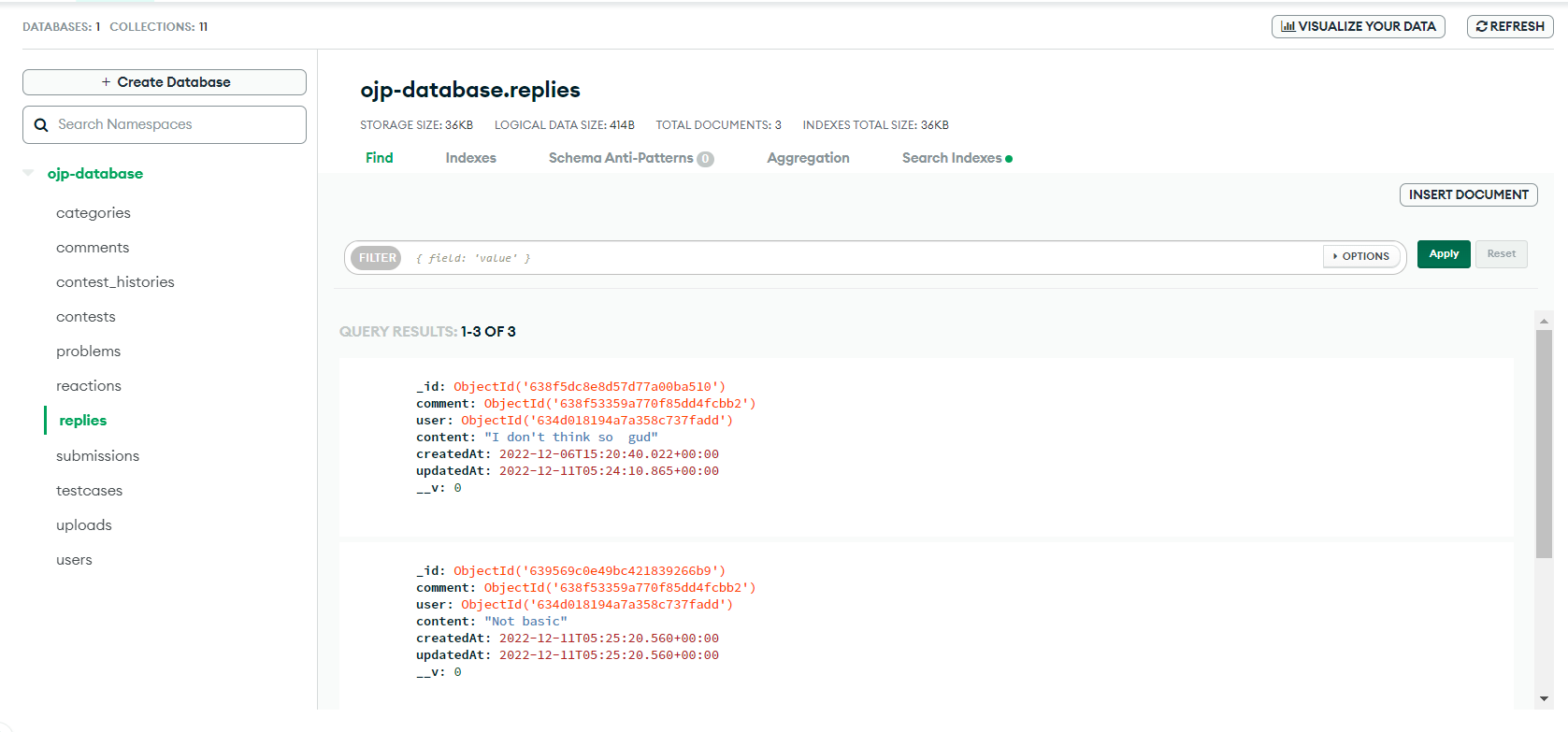
Bảng submissions:



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

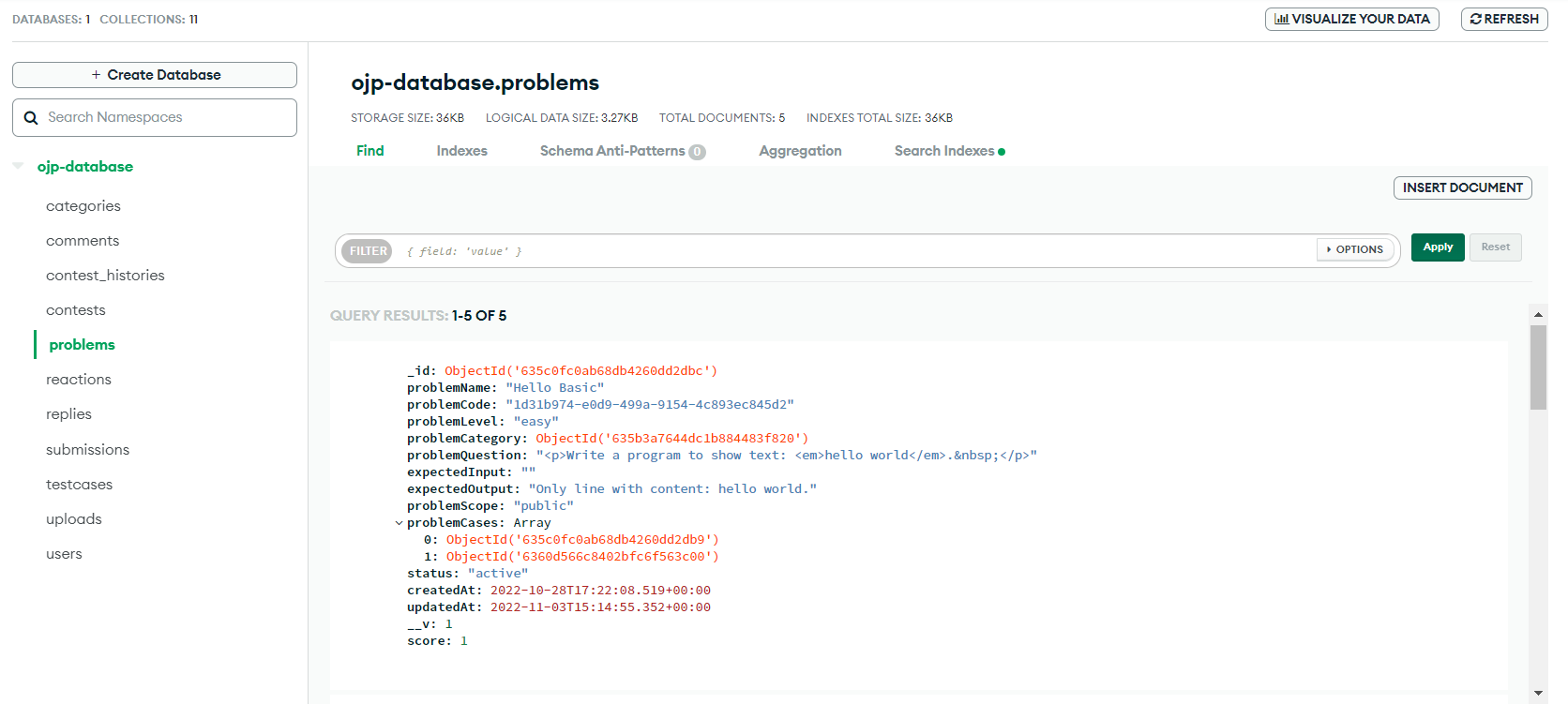
Bảng replies



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

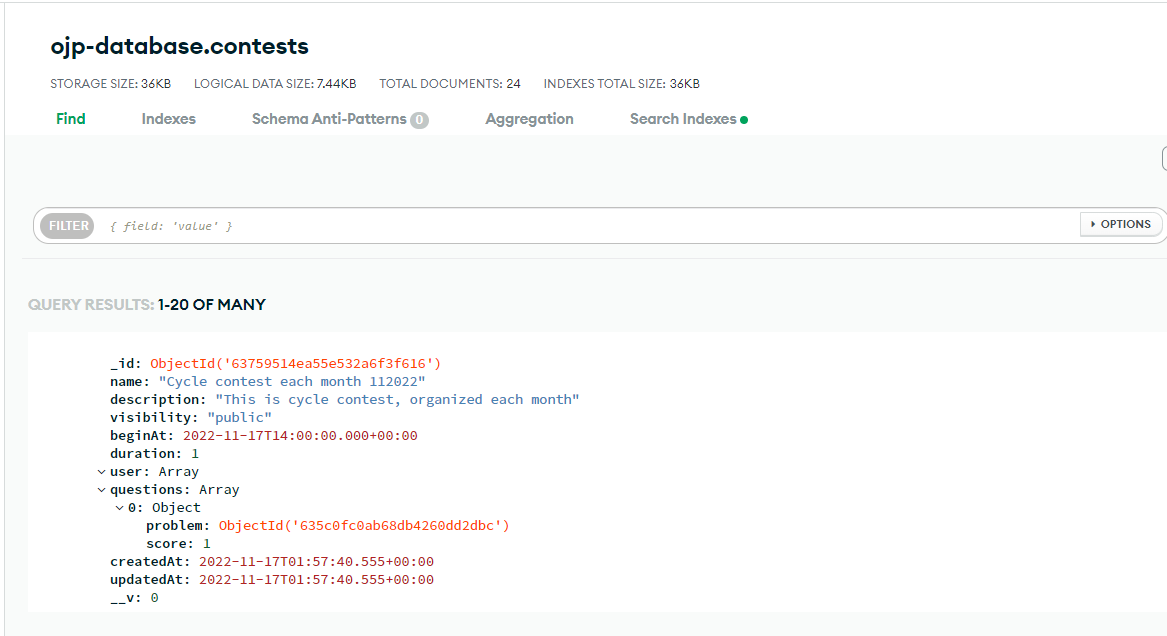
Bảng problems:



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

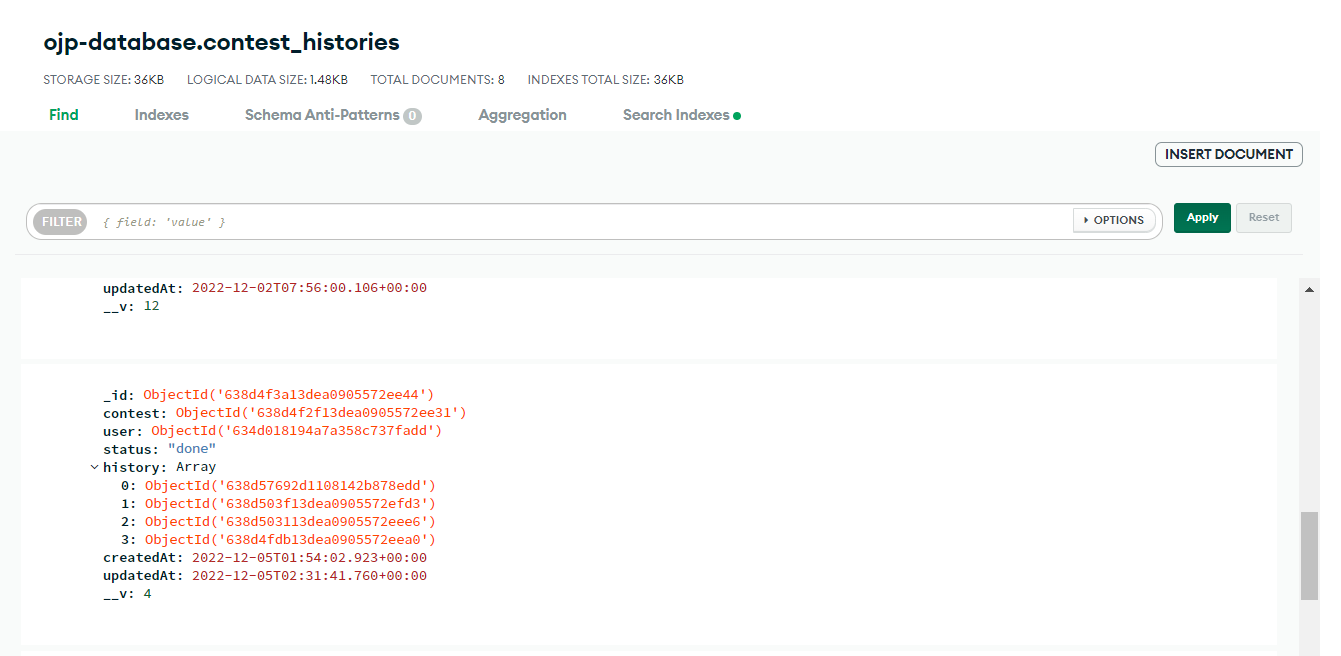
Bảng contests



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

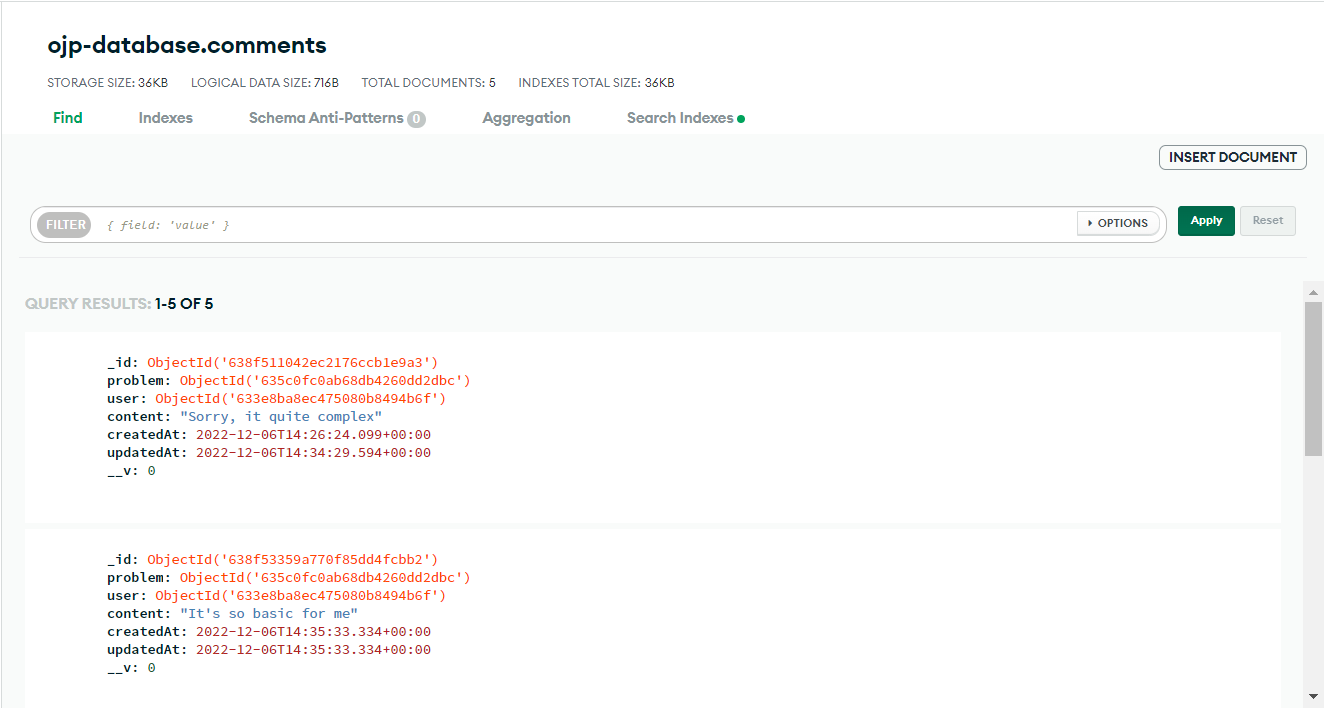
Bảng contest\_histories



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

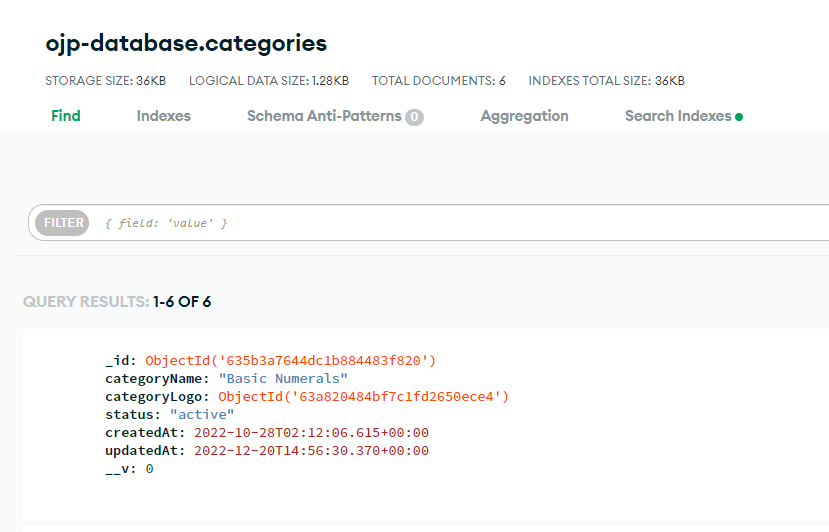
Bảng comments



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



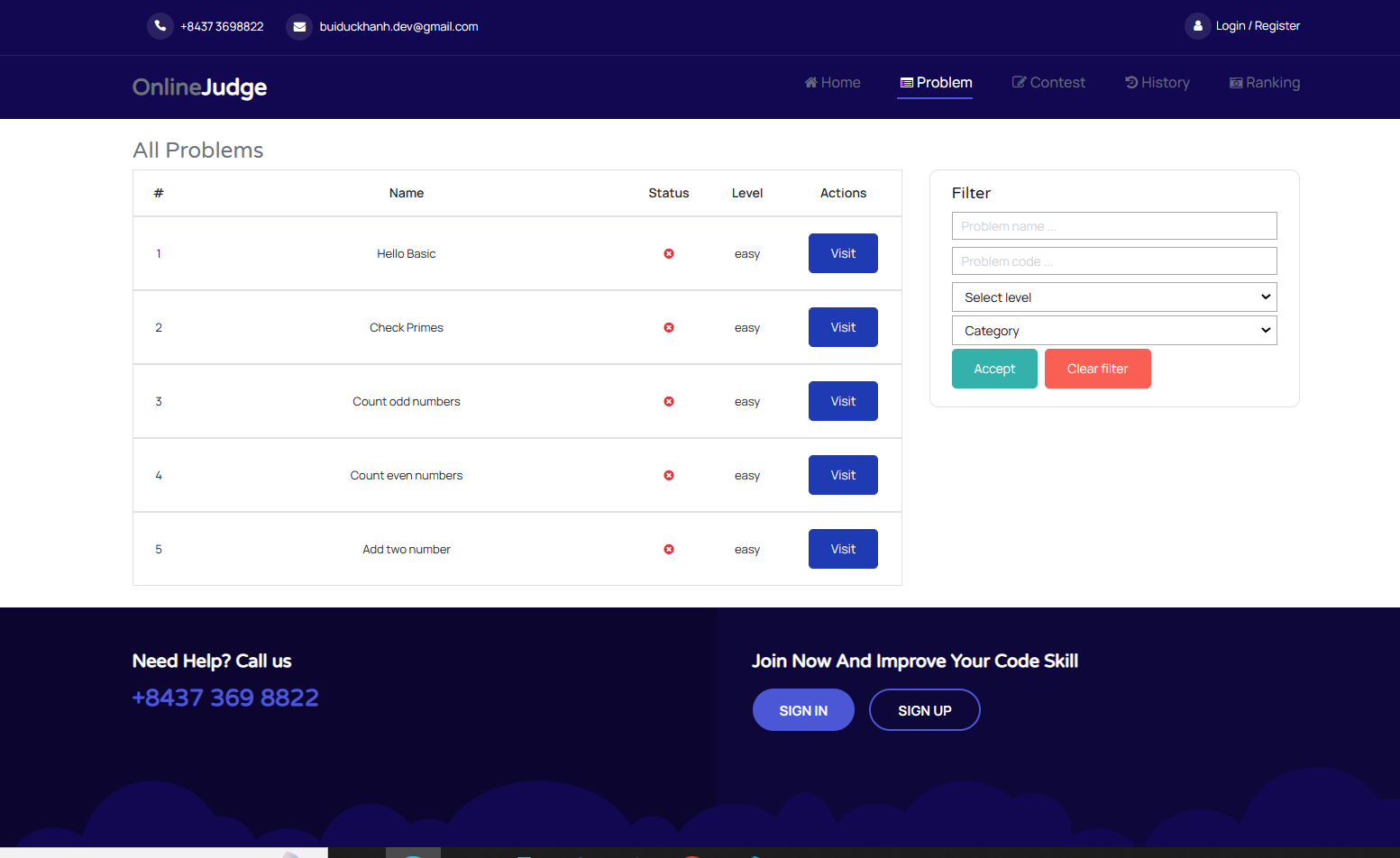
Hình . Kết quả cài đặt bảng categories

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:

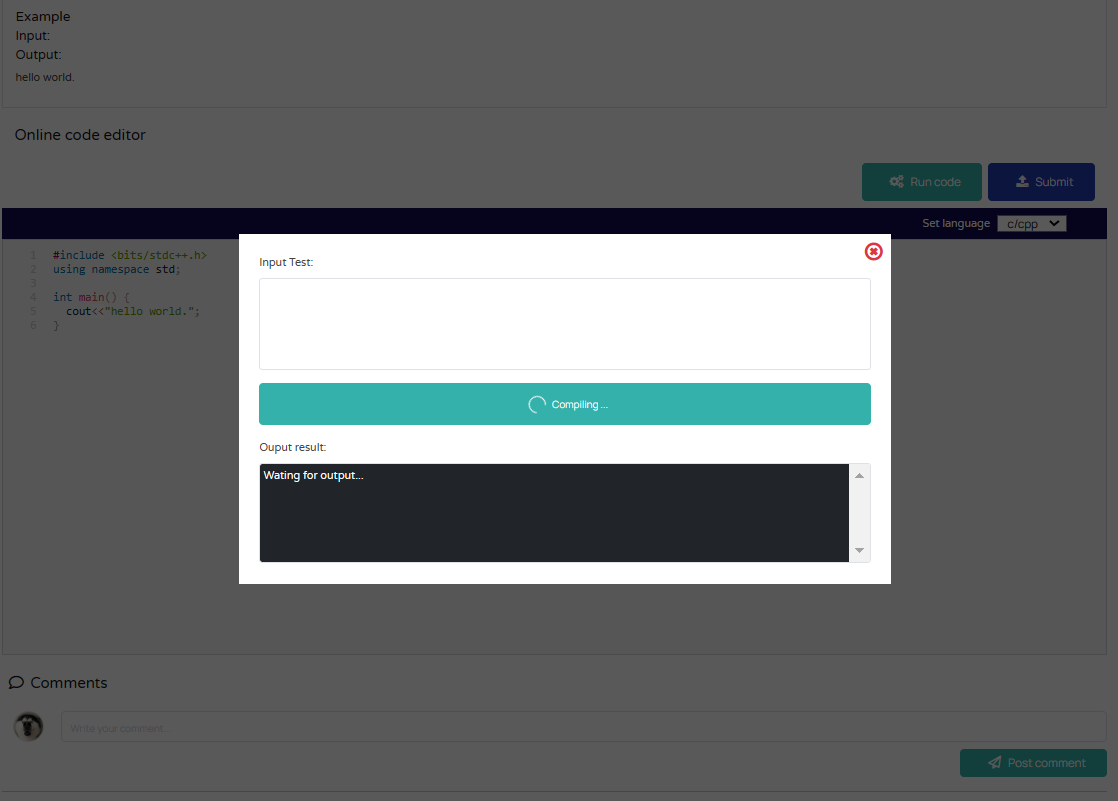
+ Giao diện xem bài toán



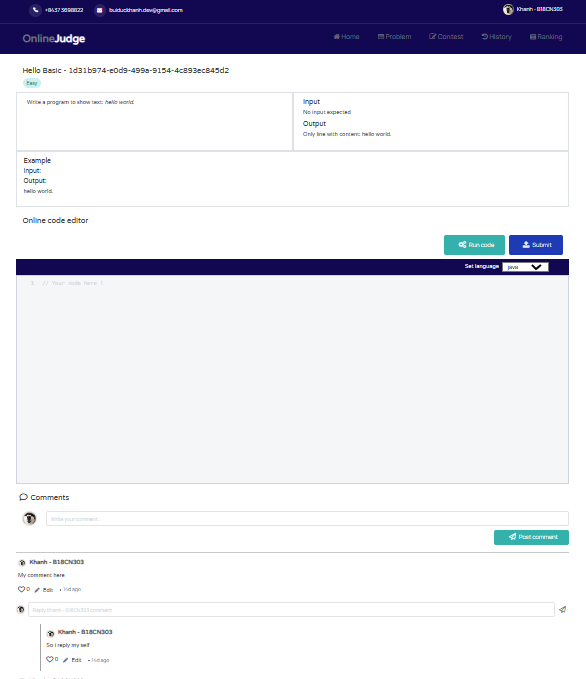
Hình . Giao diện trang xem bài toán

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.

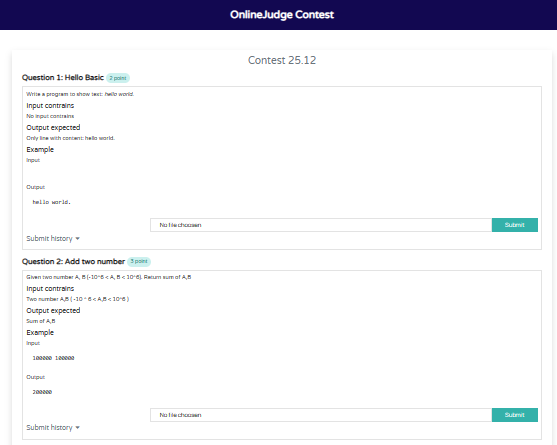
+ Giao diện xem contest



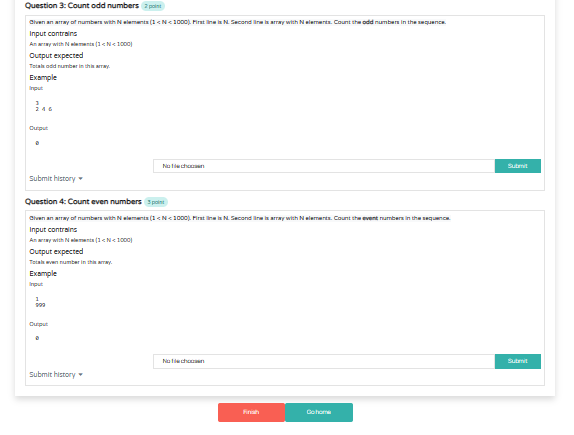
Hình . Giao diện xem contest

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:



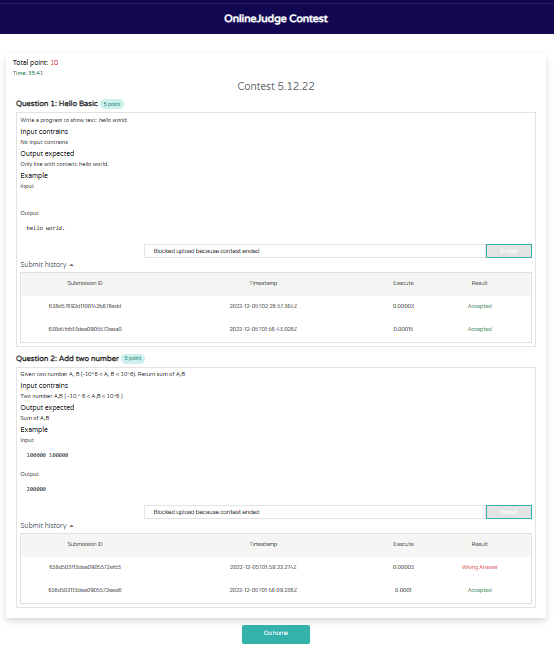
Hình . Giao diện làm contest (1)



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.

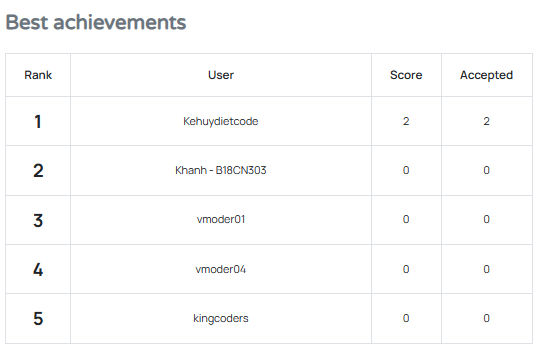
+ Giao diện xem lịch sử submit



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.

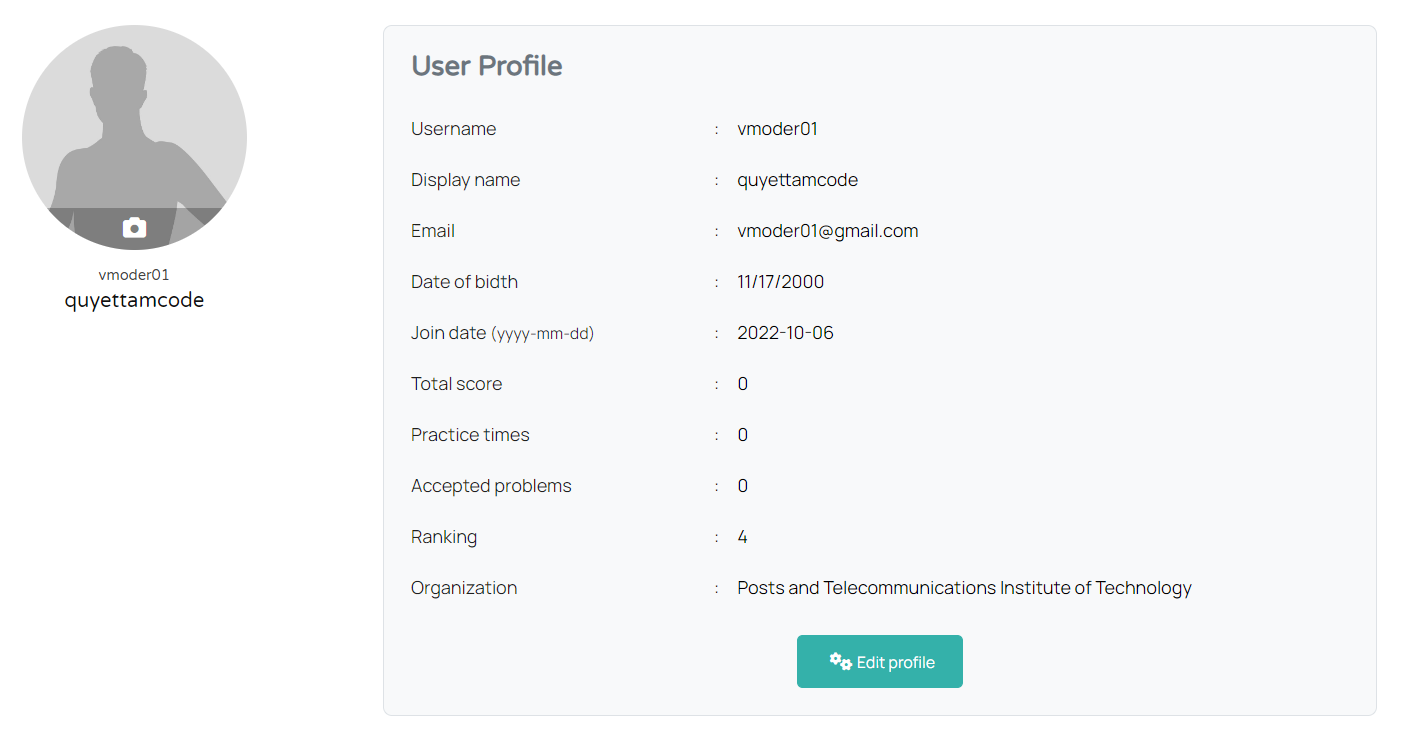
+ Giao diện xem xếp hạng



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.

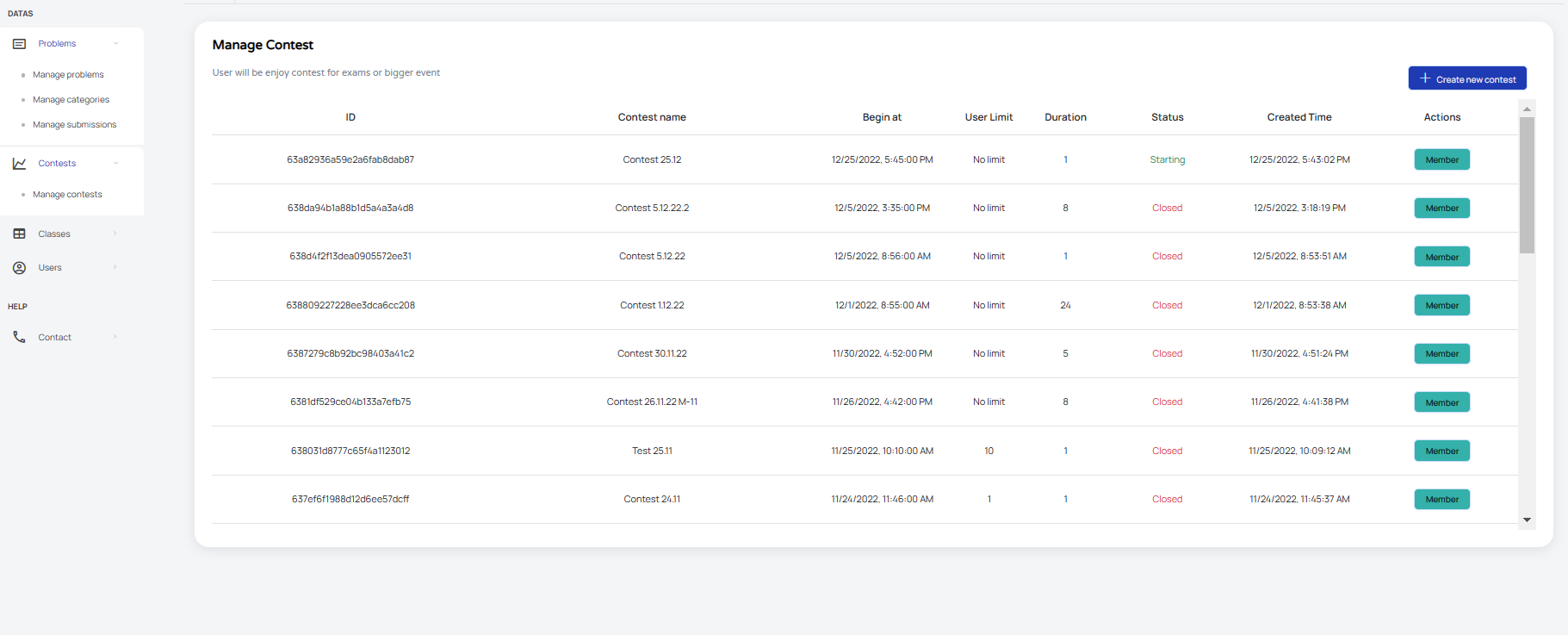
+ Giao diện quản lý thông tin cá nhân:



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.

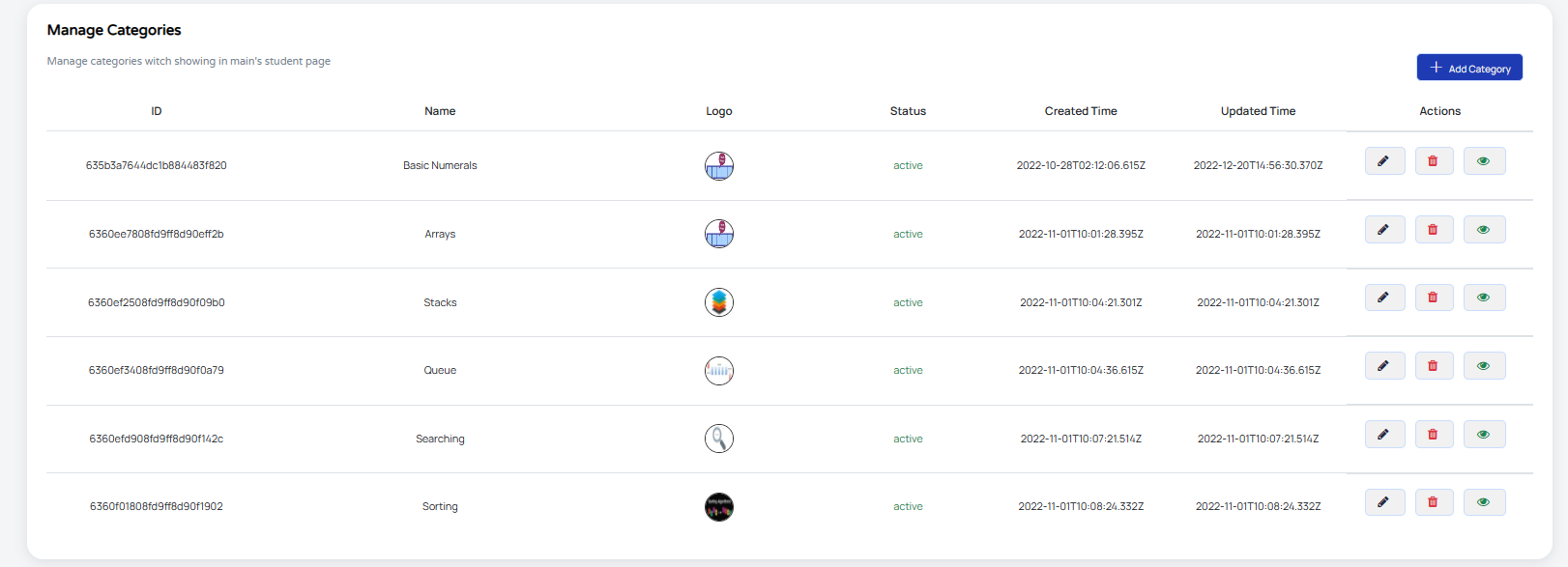
+ Giao diện quản lý contest



Hình . Giao diện quản lý contest

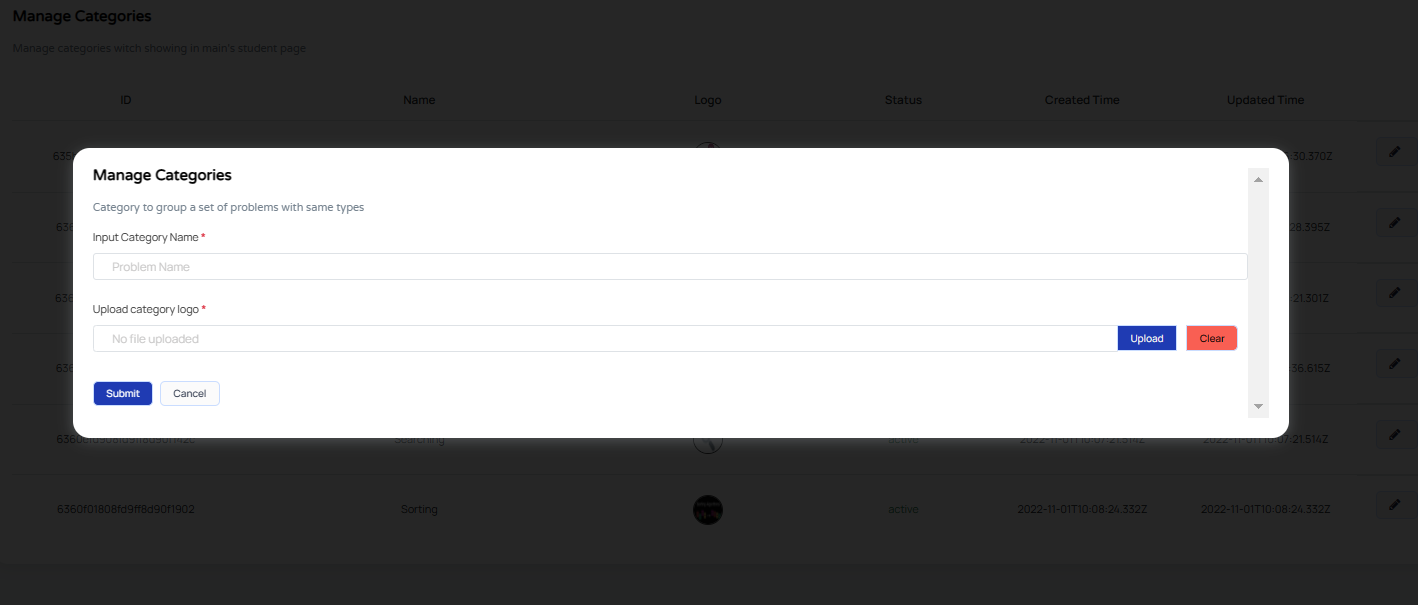
Giao diện quản lý contest bao gồm các thông tin về contest gồm: mã định danh, tên contets, thời gian bắt đầu, số người, thời lượng, trạng thái contest, thời gian tạo và nút thêm contest.

+ Giao diện quản lý category:



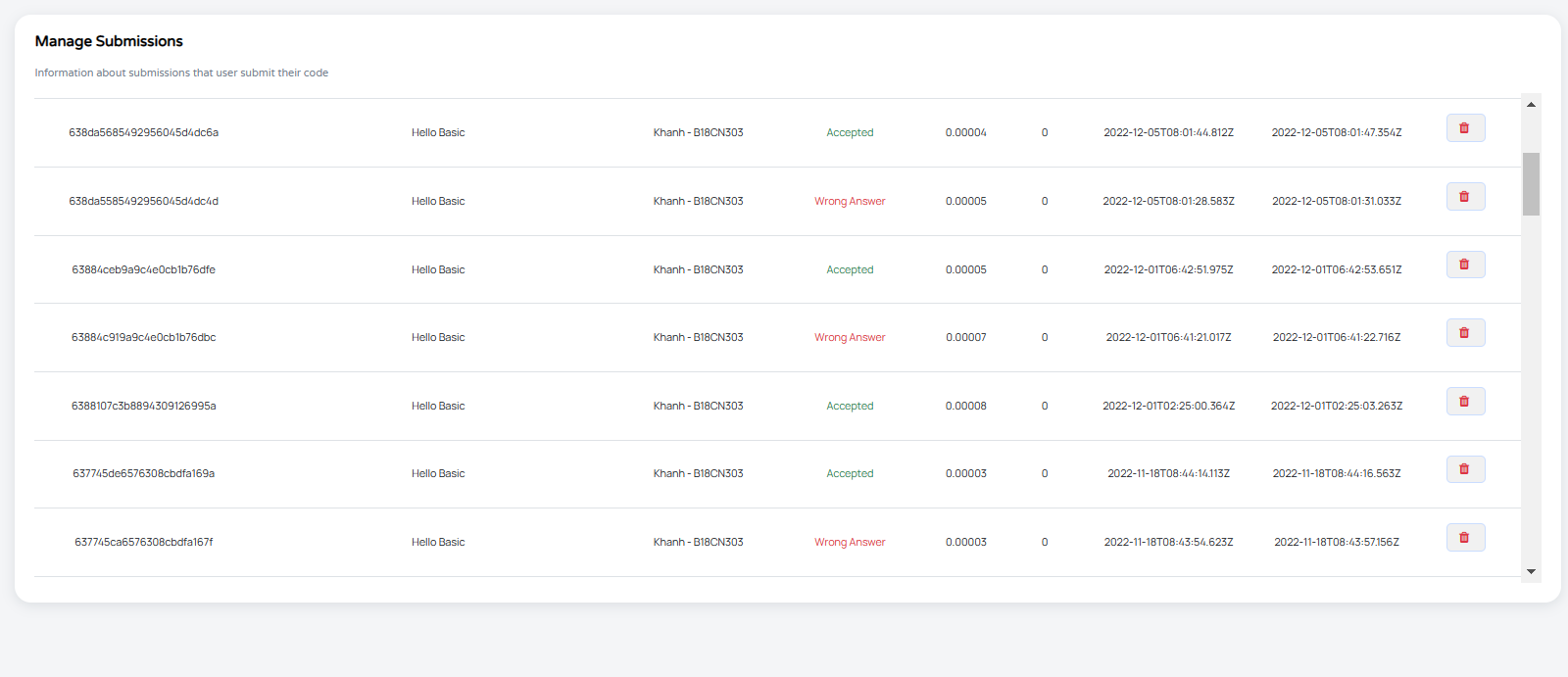
Hình . Giao diện quản lý danh mục

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.

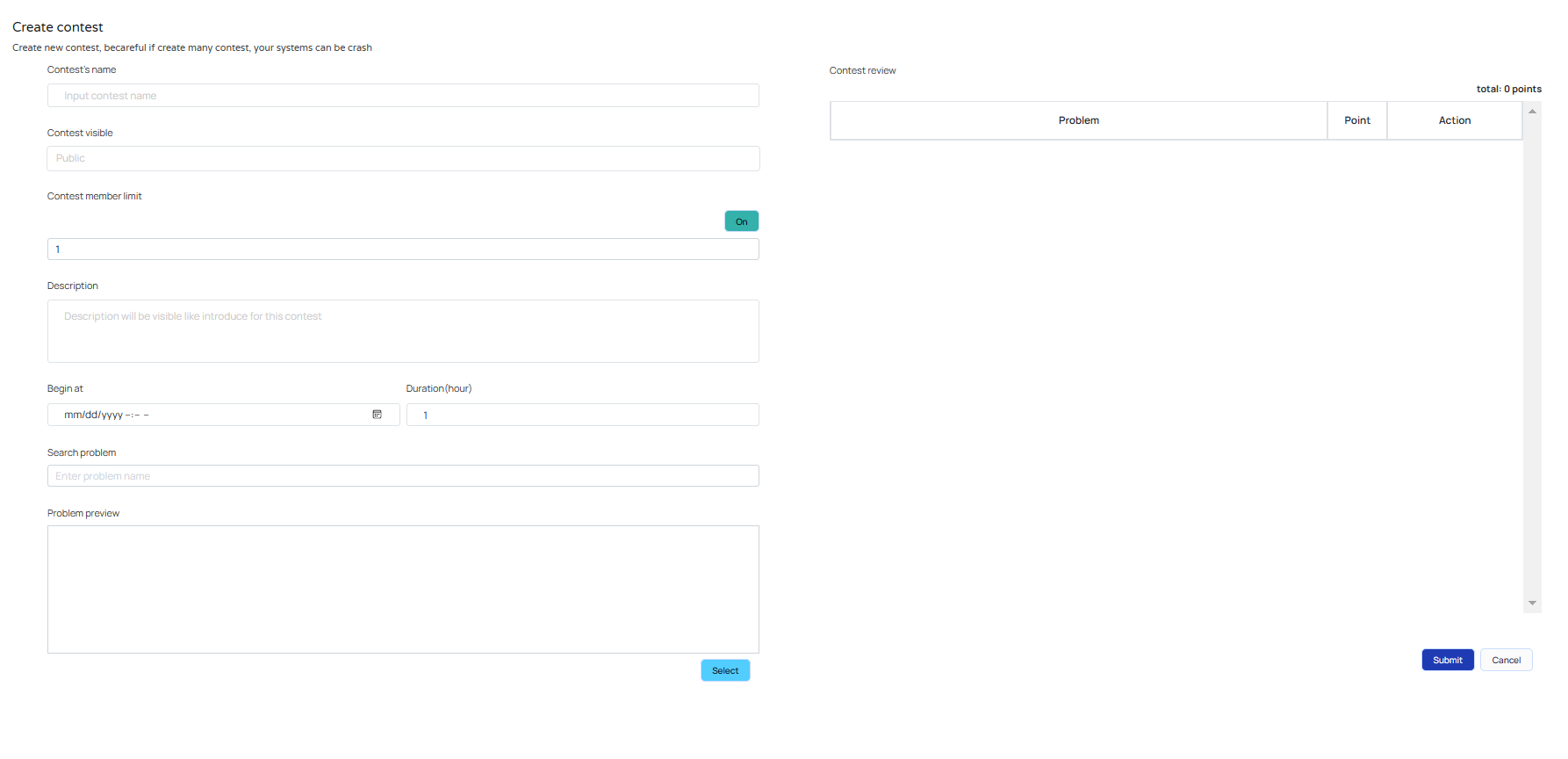


Hình . Giao diện thêm danh mục

+ 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.

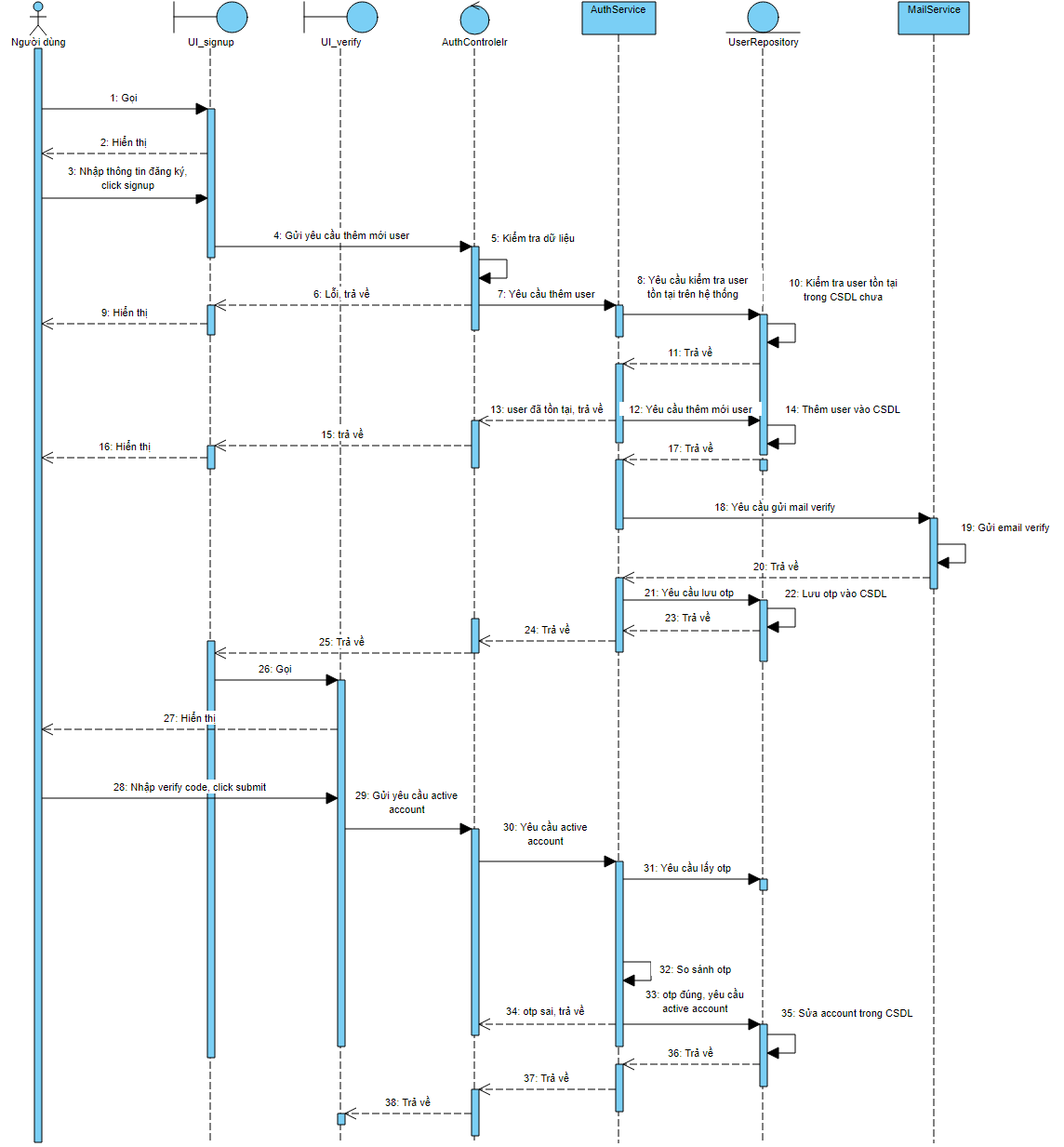
# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [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. |

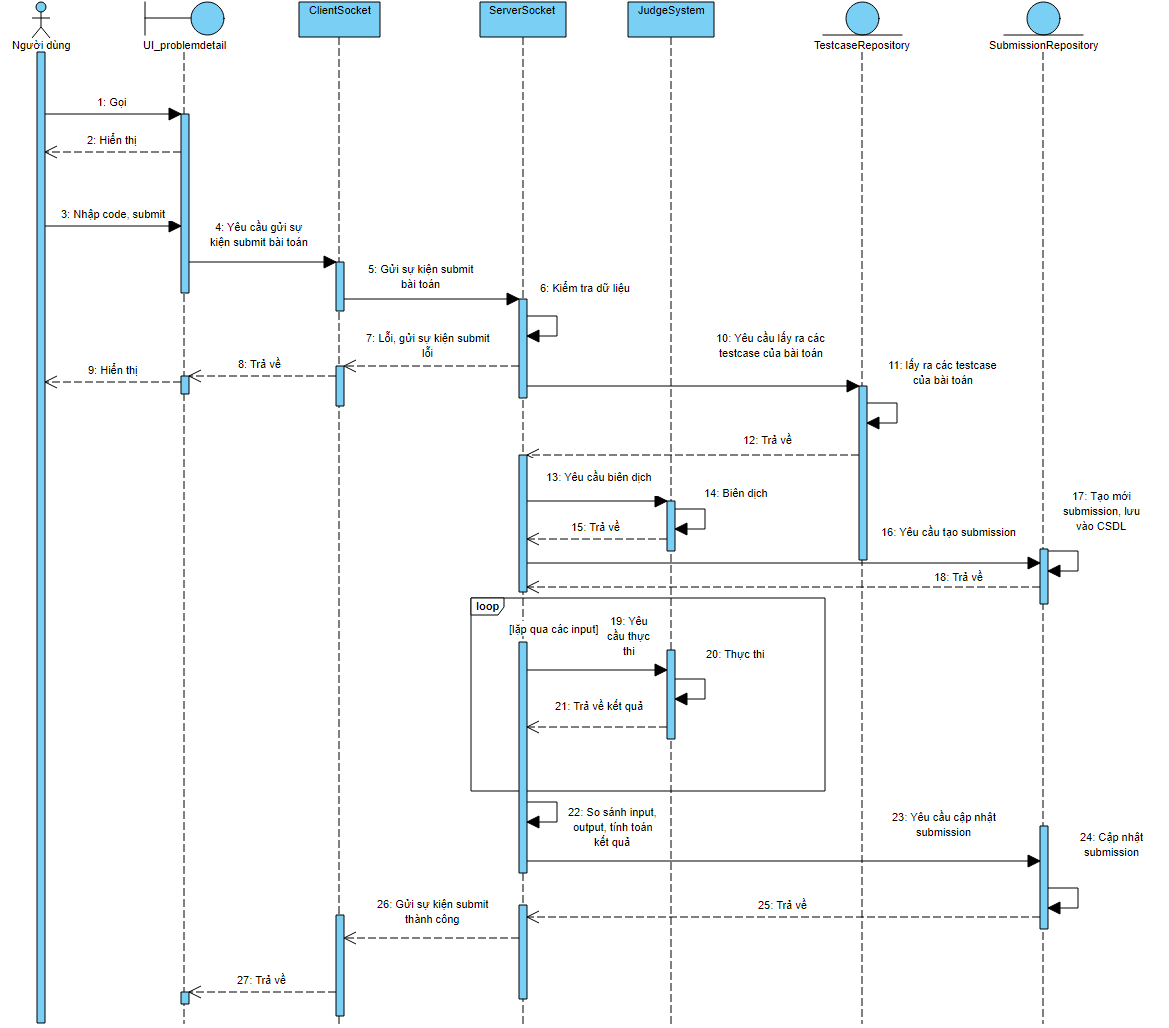
# PHỤ LỤC

## Chi tiết sơ đồ mô phỏng

*Biểu đồ tuần tự module đăng ký*



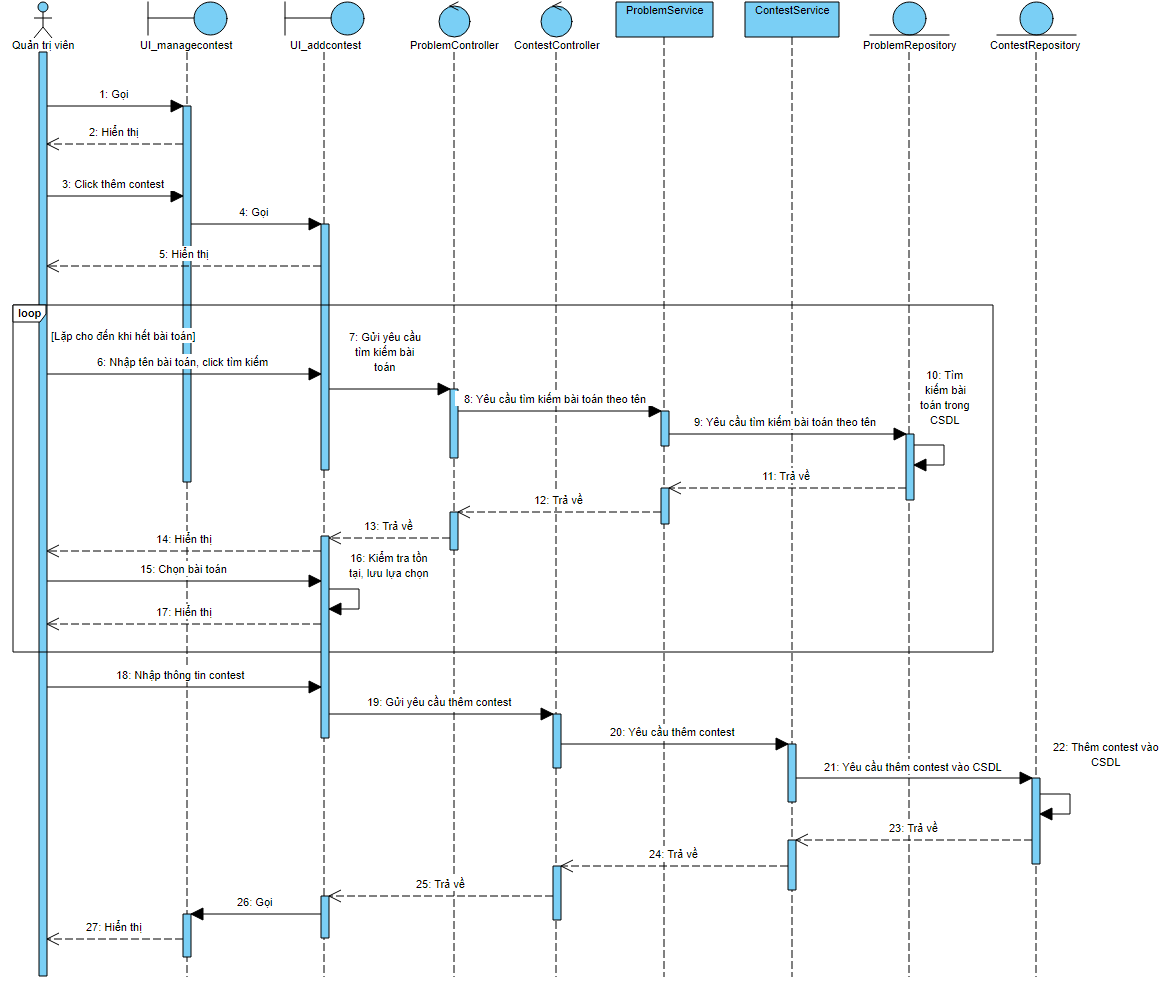
*Biểu đồ tuần tự module submit lời giải*

**

*Biểu đồ tuần tự module tham gia contest*

**

*Biểu đồ tuần tự module thêm contest*

**