

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Phân bổ nguồn lực thực hiện công việc trong dự án
theo chỉ tiêu KPI

CHU MẠNH TIẾN

tien.cm194182@sis.hust.edu.vn

Ngành Khoa học máy tính

Giảng viên hướng dẫn: TS. Vũ Thị Hương Giang _____

TS. Trịnh Tuấn Đạt _____

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 07/2024

LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn sâu sắc tới những người thân trong gia đình của mình. Gia đình luôn là nguồn động lực to lớn nhất, luôn ủng hộ và tạo điều kiện tốt nhất để em có thể tập trung vào việc học tập và thực hiện ĐATN. Những lời khuyên và tình yêu thương của các thành viên trong gia đình đã tiếp thêm sức mạnh để em vượt qua những khó khăn, thử thách trong suốt năm tháng học tập vừa qua.

Em xin chân thành cảm ơn quý thầy cô của Đại học Bách Khoa Hà Nội, đặc biệt là các thầy, cô của Trường Công nghệ Thông tin và Truyền thông. Các thầy cô giảng viên, cán bộ nhà trường luôn nhiệt tình hỗ trợ, tạo điều kiện thuận lợi để em có thể tiếp cận và hoàn thành các nhiệm vụ học tập một cách hiệu quả nhất.

Cảm ơn bạn bè đã luôn ở bên, chia sẻ và giúp đỡ em trong suốt quá trình học tập. Những khoảnh khắc cùng nhau học nhóm, thảo luận và hỗ trợ lẫn nhau đã tạo nên những kỷ niệm đẹp, giúp em vượt qua những khó khăn. Sự gắn bó và tình bạn chân thành từ các bạn là điều quý giá nhất mà em luôn trân trọng.

Đặc biệt, em xin gửi lời tri ân, lời cảm ơn chân thành sâu sắc tới cô TS. Vũ Thị Hương Giang, người đã tận tâm chỉ dẫn, cung cấp những kiến thức và góp ý quý báu để em có thể hoàn thành ĐATN một cách tốt nhất. Sự kiên nhẫn, nhiệt tình và sự hỗ trợ không ngừng nghỉ mọi nơi, mọi lúc của cô là nguồn động lực to lớn giúp em vượt qua mọi khó khăn trong quá trình thực hiện ĐATN. Em cũng xin cảm ơn thầy TS. Trịnh Tuấn Đạt đã luôn nhiệt tình tư vấn, nhận xét để ĐATN của em thêm phần trau chuốt, hoàn thiện hơn.

Em xin cảm ơn quý Công ty Cổ phần công nghệ an toàn thông tin và truyền thông Việt Nam (VNIST) đã tạo điều kiện cho em có thể thực hiện ĐATN tích hợp trên nền tảng ứng dụng của công ty.

Vì thời gian thực hiện cũng như trình độ, năng lực chuyên môn còn hạn chế, ĐATN có thể vẫn còn những thiếu sót, kính mong nhận được đánh giá, nhận xét từ quý thầy cô để đồ án của em được hoàn thiện hơn.

Cảm ơn Bách Khoa, nơi đã rèn luyện bản lĩnh, tinh thần nhẫn nại, quyết tâm vượt khó và nghị lực vươn lên của sinh viên. Cảm ơn Bách Khoa, nơi chứa đựng bao kỷ niệm đẹp nhất thời thanh xuân như đã cố định một phần trong vùng ký ức. Cảm ơn Bách Khoa, nơi cho em biết được em có thể làm những điều mà em từng nghĩ mình không thể. Em xin chân thành cảm ơn!

LỜI CAM KẾT

Họ và tên sinh viên: Chu Mạnh Tiến
Điện thoại liên lạc: 0979932446
Email: tien.cm194182@sis.hust.edu.vn
Lớp: Khoa học máy tính 05 - K64
Hệ đào tạo: Kỹ sư chính quy

Tôi – *Chu Mạnh Tiến* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS. Vũ Thị Hương Giang, TS. Trịnh Tuấn Đạt*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 01 tháng 07 năm 2024

Tác giả ĐATN

Chu Mạnh Tiến

TÓM TẮT NỘI DUNG ĐỒ ÁN

Phân bổ nguồn lực, gồm nhân lực và tài sản, là nghiệp vụ quan trọng trong quản lý dự án. Người quản lý cần phân công nhân lực và tài sản để thực hiện các công việc trong dự án. Để đánh giá hiệu quả thực hiện công việc, các chỉ tiêu KPI được thiết lập. Một phương án phân công hiệu quả giúp dự án hoàn thành và đạt được các mục tiêu, tối ưu về thời gian, chi phí, và KPI để mang lại lợi ích tối đa cho doanh nghiệp.

Tuy nhiên, những thách thức như hạn chế về nhân lực, tài sản, thời gian, yêu cầu năng lực của nhân viên ở mỗi công việc làm cho việc phân bổ trở nên phức tạp và tốn thời gian. Do đó, cần có phương án hỗ trợ phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI một cách tự động để quản lý dự án hiệu quả hơn.

Đồ án thực hiện tìm ra giải pháp cho bài toán phân bổ nguồn lực đã nêu trên. Giải pháp đề xuất một phương án tối ưu cục bộ đáp ứng các tiêu chí về thời gian thực hiện, chi phí thực hiện và các chỉ tiêu KPI.

Giải pháp gồm các bước chính sau. Đầu tiên, sử dụng thuật toán sắp topo (Topological Sort) để sắp xếp thứ tự thực hiện công việc, đồng thời dùng phương pháp CPM (Critical Path Method) để có cái nhìn tổng thể về thời gian thực hiện dự án. Tiếp theo, gán khung thời gian và tài sản thực hiện công việc. Tiếp theo, sử dụng thuật toán tìm kiếm hài hòa (Harmony Search) hoặc cải tiến của nó (DLHS) để phân bổ nhân viên thực hiện các công việc. Cuối cùng, kiểm tra và điều chỉnh lại công việc xung đột nếu có.

Giải pháp đề xuất được so sánh với thuật toán giải quyết bài toán phân bổ nguồn lực thực hiện gói thầu [1], cho thấy hiệu quả vượt trội của thuật toán đề xuất về độ phức tạp tính toán, các giá trị mục tiêu tối ưu về thời gian, chi phí và KPI.

Dựa trên giải pháp đề xuất, ĐATN xây dựng các chức năng chính gồm: (i) quản lý danh sách dự án, (ii) phân bổ nguồn lực thực hiện dự án, (iii) lập lịch thực hiện công việc cho dự án, (iv) quản lý thẻ công việc, (v) quản lý bộ năng lực của nhân viên. Các chức năng này được tích hợp vào hệ thống DXClan [2], để hỗ trợ các nhà quản lý thực hiện phân bổ nguồn lực dự án theo chỉ tiêu KPI một cách tự động.

Keywords: *Resource Allocation, KPI Assignment, Project Management*

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	3
1.3 Định hướng giải pháp.....	4
1.4 Bố cục đồ án	5
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	7
2.1 Khảo sát hiện trạng	7
2.1.1 Khảo sát một số ứng dụng hỗ trợ phân bổ nguồn lực dự án.....	7
2.1.2 Khảo sát một số nghiên cứu, giải pháp cho bài toán phân bổ nguồn lực	9
2.2 Tổng quan chức năng	11
2.2.1 Biểu đồ usecase tổng quan	11
2.2.2 Biểu đồ phân rã usecase quản lý danh sách dự án	11
2.2.3 Biểu đồ phân rã usecase phân bổ nguồn lực dự án	13
2.2.4 Biểu đồ phân rã usecase quản lý thẻ công việc	14
2.2.5 Biểu đồ phân rã usecase quản lý bộ năng lực nhân viên	14
2.2.6 Quy trình nghiệp vụ tạo dự án	15
2.2.7 Quy trình nghiệp vụ phân bổ nguồn lực dự án.....	16
2.3 Đặc tả chức năng	17
2.3.1 Đặc tả usecase tạo dự án	17
2.3.2 Đặc tả usecase phân bổ nguồn lực dự án.....	18
2.3.3 Đặc tả usecase lập lịch thực hiện công việc của dự án.....	19
2.3.4 Đặc tả usecase thêm thẻ công việc	20

2.4 Yêu cầu phi chức năng	21
2.4.1 Yêu cầu chung	21
2.4.2 Yêu cầu về giao diện.....	21
2.4.3 Yêu cầu về bảo mật	21
2.4.4 Yêu cầu về thuật toán, giải pháp đề xuất	22
2.4.5 Yêu cầu về việc tích hợp vào hệ thống DXClan	22
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	23
3.1 Công nghệ sử dụng cho lập trình phía Frontend.....	23
3.1.1 ReactJS	23
3.1.2 Redux	23
3.1.3 Tailwind CSS	24
3.2 Công nghệ sử dụng cho lập trình phía Backend	24
3.2.1 NodeJS	24
3.2.2 ExpressJS.....	25
3.2.3 MongoDB và Mongoose.....	25
3.3 Hệ thống tích hợp DXClan.....	25
3.4 Một số công nghệ, thuật toán khác sử dụng trong DATN.....	26
3.4.1 Thuật toán sắp topo (Topological Sort).....	26
3.4.2 Phương pháp đường găng (Critical Path Method - CPM)	27
3.4.3 Thuật toán phân cụm K-Mean	27
3.4.4 Thuật toán tìm kiếm hài hòa - HS	28
3.4.5 Thuật toán tìm kiếm hài hòa tối ưu cục bộ dựa trên các quần thể con động - DLHS	30
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG	33
4.1 Thiết kế kiến trúc.....	33
4.1.1 Lựa chọn kiến trúc phần mềm	33

4.1.2 Thiết kế tổng quan.....	34
4.1.3 Thiết kế chi tiết gói	36
4.2 Thiết kế chi tiết.....	39
4.2.1 Thiết kế giao diện	39
4.2.2 Thiết kế lớp	43
4.2.3 Thiết kế cơ sở dữ liệu	47
4.3 Xây dựng ứng dụng.....	52
4.3.1 Thư viện và công cụ sử dụng.....	52
4.3.2 Kết quả đạt được	52
4.3.3 Minh họa các chức năng chính	53
4.3.4 Các chức năng liên quan đến phân bổ nguồn lực	58
4.4 Kiểm thử.....	62
4.4.1 Kiểm thử tính tương thích (Compatibility Testing)	62
4.4.2 Kiểm thử hộp đen (Black Box Testing).....	63
4.5 Triển khai	66
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT	67
5.1 Mô hình bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI	67
5.1.1 Inputs (Đầu vào)	67
5.1.2 Outputs (Đầu ra)	70
5.1.3 Constraints (Ràng buộc)	72
5.1.4 Objectives (Hàm mục tiêu)	73
5.2 Thuật toán đề xuất sử dụng để giải quyết bài toán	74
5.2.1 Tiền xử lý: tính toán giá trị hiệu suất thực hiện công việc.....	74
5.2.2 Bước 1: Sắp thứ tự, ước tính thời gian thực hiện công việc	76
5.2.3 Bước 2: Gán tài sản, khung thời gian cho từng công việc đã được sắp thứ tự.....	79

5.2.4 Bước 3: Gán nhân lực, tài sản tùy chọn thực hiện công việc	80
5.2.5 Bước 4: Xử lý các công việc thực hiện chồng chéo nếu có.....	88
5.3 Đánh giá thuật toán.....	89
5.3.1 So sánh thuật toán đề xuất với thuật toán đã có	89
5.3.2 Thực nghiệm với bộ dữ liệu chuẩn	92
5.3.3 Thực nghiệm với bộ dữ liệu thực tế.....	94
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	97
6.1 Kết luận.....	97
6.2 Hướng phát triển.....	97
TÀI LIỆU THAM KHẢO.....	99
PHỤ LỤC.....	101
CHƯƠNG A. BỘ DỮ LIỆU THỬ NGHIỆM THỰC.....	101

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ usecase tổng quan	11
Hình 2.2	Biểu đồ phân rã usecase quản lý danh sách dự án	12
Hình 2.3	Biểu đồ phân rã usecase phân bổ nguồn lực dự án	13
Hình 2.4	Biểu đồ phân rã usecase quản lý thẻ công việc	14
Hình 2.5	Biểu đồ phân rã usecase quản lý bộ năng lực nhân viên	14
Hình 2.6	Biểu đồ hoạt động quá trình tạo dự án	15
Hình 2.7	Biểu đồ hoạt động quá trình phân bổ nguồn lực dự án	16
Hình 3.1	Sơ đồ mô tả thuật toán tìm kiếm hài hòa (Harmony Search)	28
Hình 3.2	Tìm giải pháp cải tiến dựa trên bộ nhớ HM	29
Hình 3.3	Tìm giải pháp cải tiến dựa trên phương án tối ưu cục bộ của bộ nhớ con sub-HM	31
Hình 4.1	Kiến trúc tổng quan của hệ thống	34
Hình 4.2	Kiến trúc tổng quan của ứng dụng phía client	35
Hình 4.3	Kiến trúc tổng quan của ứng dụng phía server	36
Hình 4.4	Thiết kế chi tiết gói phía server	37
Hình 4.5	Thiết kế chi tiết gói phía client cho nhóm chức năng quản lý danh sách dự án	38
Hình 4.6	Thiết kế chi tiết gói phía client cho nhóm chức năng phân bổ nguồn lực	39
Hình 4.7	Cấu trúc giao diện phía DXClan	40
Hình 4.8	Sitemap giao diện của ứng dụng	40
Hình 4.9	Thiết kế giao diện màn danh sách dự án	41
Hình 4.10	Thiết kế giao diện màn tạo mới dự án	41
Hình 4.11	Thiết kế giao diện màn phân bổ nguồn lực dự án	42
Hình 4.12	Biểu đồ trình tự cho usecase tạo dự án	43
Hình 4.13	Thiết kế chi tiết lớp ProjectInfoTab, ProjectResourceTab, Pro- jectTaskTab	44
Hình 4.14	Biểu đồ trình tự cho usecase phân bổ nguồn lực, lập lịch thực hiện dự án	46
Hình 4.15	Thiết kế chi tiết cho lớp ProjectProposalService	47
Hình 4.16	Biểu đồ thực thể liên kết	48
Hình 4.17	Màn hình danh sách dự án	53
Hình 4.18	Màn hình thêm dự án - Tab thông tin chung	54
Hình 4.19	Màn hình thêm dự án - Tab Tài nguyên nhân lực và tài sản	54

Hình 4.20	Màn hình thêm dự án - Tab Thêm công việc	55
Hình 4.21	Màn hình thêm dự án - Tab Thêm công việc	55
Hình 4.22	Màn hình thêm dự án - Tab Thêm công việc, danh sách công việc dạng bảng	56
Hình 4.23	Màn hình thêm dự án - Tab Thêm công việc, danh sách công việc dạng biểu đồ	56
Hình 4.24	Màn hình thông tin dự án	57
Hình 4.25	Màn hình chỉnh sửa dự án	57
Hình 4.26	Màn hình sau khi thêm dự án thành công	58
Hình 4.27	Màn hình phân bổ nguồn lực dự án	58
Hình 4.28	Màn hình phân bổ nguồn lực dự án, biểu mẫu thiết lập tham số	59
Hình 4.29	Màn hình phân bổ nguồn lực dự án, bảng thông tin phân bổ . .	60
Hình 4.30	Màn hình phân bổ nguồn lực dự án, kết quả phân bổ	60
Hình 4.31	Màn hình phân bổ nguồn lực dự án, kết quả phân bổ (2)	61
Hình 4.32	Màn hình phân bổ nguồn lực dự án, lịch công việc	61
Hình 4.33	Màn hình phân bổ nguồn lực dự án, lịch nhân công	62
Hình 4.34	Màn hình phân bổ nguồn lực dự án, lịch tài sản	62
Hình 5.1	Minh họa phương pháp CPM cho các công việc	77

DANH MỤC BẢNG BIỂU

Bảng 2.1	Bảng so sánh chức năng của một số phần mềm hiện có	9
Bảng 2.2	Đặc tả usecase tạo dự án	18
Bảng 2.3	Đặc tả usecase phân bổ nguồn lực dự án	19
Bảng 2.4	Đặc tả usecase lập lịch thực hiện công việc của dự án	20
Bảng 2.5	Đặc tả usecase thêm thẻ công việc	21
Bảng 4.1	Thuộc tính và phương thức của lớp ProjectInfoTab	44
Bảng 4.2	Thuộc tính và phương thức của lớp ProjectResourceTab	45
Bảng 4.3	Thuộc tính và phương thức của lớp ProjectTaskTab	45
Bảng 4.4	Thuộc tính và phương thức của lớp ProjectProposalPage	47
Bảng 4.5	Bảng liệt kê các thực thể và ý nghĩa	48
Bảng 4.6	Thiết kế chi tiết cho thực thể Project	49
Bảng 4.7	Thiết kế chi tiết cho thực thể Capacity	49
Bảng 4.8	Thiết kế chi tiết cho thực thể Task	50
Bảng 4.9	Thiết kế chi tiết cho thực thể Tag	50
Bảng 4.10	Thông tin mô tả thực thể Employee	51
Bảng 4.11	Thông tin mô tả thực thể Asset	51
Bảng 4.12	Thông tin mô tả thực thể OrganizationalUnitKPI	51
Bảng 4.13	Danh sách thư viện và công cụ sử dụng	52
Bảng 4.14	Thống kê về ứng dụng	53
Bảng 4.15	Kiểm thử tính tương thích trên các thiết bị	63
Bảng 4.16	Kiểm thử hộp đen cho nhóm chức năng quản lý danh sách dự án	64
Bảng 4.17	Kiểm thử hộp đen cho nhóm chức năng phân bổ nguồn lực dự án	66
Bảng 5.1	Đầu vào thông tin chung về dự án	67
Bảng 5.2	Đầu vào thông tin về KPI chỉ tiêu	67
Bảng 5.3	Đầu vào danh sách công việc trong dự án	68
Bảng 5.4	Đầu vào danh sách nhân viên tham gia dự án	69
Bảng 5.5	Đầu vào danh sách tài sản tham gia dự án	69
Bảng 5.6	Đầu ra kế hoạch thực hiện dự án	70
Bảng 5.7	Đầu ra kế hoạch thực hiện dự án	70
Bảng 5.8	Đầu ra lịch trình thực hiện công việc của dự án	71
Bảng 5.9	Đầu ra KPI sẽ giao cho nhân viên	72
Bảng 5.10	Bảng so sánh về nghiệp vụ thực hiện phân bổ nguồn lực	89

Bảng 5.11	Bảng so sánh độ phức tạp thuật toán	90
Bảng 5.12	Mô tả bộ dữ liệu so sánh 2 thuật toán	90
Bảng 5.13	Bảng so sánh kết quả thuật toán trên 3 bộ dữ liệu	91
Bảng 5.14	Mô tả bộ dữ liệu chuẩn	93
Bảng 5.15	Bảng so sánh kết quả thực nghiệm với bộ dữ liệu chuẩn	93
Bảng 5.16	Tiêu chí đánh giá thuật toán	94
Bảng 5.17	Kết quả thực hiện phân bố	95
Bảng 5.18	Kết quả thực hiện phân bố trên các tiêu chí đánh giá	96
Bảng A.1	Thông tin chung về dự án	101
Bảng A.2	Danh sách chỉ tiêu KPI của dự án	101
Bảng A.3	Danh sách công việc của dự án (01)	101
Bảng A.4	Danh sách công việc của dự án (02)	102
Bảng A.5	Yêu cầu về nhân lực và tài sản tương ứng của các công việc (01)	103
Bảng A.6	Yêu cầu về nhân lực và tài sản tương ứng của các công việc (02)	104
Bảng A.7	Danh sách nhân viên tham gia dự án	105
Bảng A.8	Danh sách tài sản tham gia dự án	106
Bảng A.9	Danh sách công việc mà các nhân viên đang thực hiện ngoài dự án	107
Bảng A.10	Danh sách công việc mà các nhân viên đã thực hiện trong quá khứ (1)	108
Bảng A.11	Danh sách công việc mà các nhân viên đã thực hiện trong quá khứ (2)	109
Bảng A.12	Danh sách công việc mà các nhân viên đã thực hiện trong quá khứ (3)	110
Bảng A.13	Danh sách công việc mà các nhân viên đã thực hiện trong quá khứ (4)	111
Bảng A.14	Danh sách công việc mà các nhân viên đã thực hiện trong quá khứ (5)	112
Bảng A.15	Hiệu suất thực hiện công việc của nhân viên	113

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
ĐATN	Đồ án tốt nghiệp
API	Giao diện lập trình ứng dụng (Application Programming Interface)
bw	Khoảng cách điều chỉnh cao độ
CPM	Phương pháp đường găng
CSDL	Cơ sở dữ liệu
DLHS	Thuật toán tìm kiếm hài hòa bản có cải tiến - DLHS
EF	Thời gian kết thúc sớm nhất
ES	Thời gian bắt đầu sớm nhất
GWT	Công cụ lập trình Javascript bằng Java của Google (Google Web Toolkit)
HM	Bộ nhớ hài hòa
HMCR	Tỷ lệ xem xét bộ nhớ hài hòa
HMS	Kích thước bộ nhớ hài hòa
HS	Thuật toán tìm kiếm hài hòa
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
Key Performance Indicator	Chỉ số đo lường và đánh giá hiệu quả hoạt động của một bộ phận trong một công ty hoặc sự vận hành của cả công ty.
LF	Thời gian kết thúc muộn nhất
LS	Thời gian bắt đầu muộn nhất
PAR	Tỷ lệ tùy chỉnh cao độ
PSL	Bộ danh sách tham số
WPSL	Bộ danh sách tham số chiến thắng

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Phân bổ nguồn lực dự án là quá trình các nhà quản lý, hoạch định dự án xác định, lựa chọn và phân phối các nguồn lực cần thiết, bao gồm nhân lực và tài sản, để thực hiện các công việc trong dự án. Đầu tiên, người quản lý dự án cần liệt kê từng công việc của dự án, xác định thứ tự thực hiện và phân tích nhu cầu nguồn lực. Tiếp theo, kiểm tra và đánh giá nguồn lực hiện có để xác định những gì đã sẵn sàng, những gì cần bổ sung hoặc thuê ngoài. Sau đó, xây dựng kế hoạch phân bổ nguồn lực chi tiết, bao gồm thời gian, nhân lực và tài sản cụ thể cho từng công việc trong dự án. Cuối cùng, phân công thực hiện công việc theo kế hoạch đã đề ra.

Để đánh giá hiệu quả thực hiện của dự án, các chỉ tiêu KPI (Key Performance Indicators - Chỉ số đánh giá hiệu suất) sẽ được thiết lập và áp dụng. KPI là các đơn vị đo lường quan trọng để đánh giá mức độ đạt được của các mục tiêu quan trọng trong dự án. KPI trong dự án được chia ra theo từng lĩnh vực, ví dụ như chất lượng sản phẩm, số mô-đun, chức năng hoàn thành đúng hạn, số tài liệu cần giao nộp ... và các chỉ số khác liên quan đến mục tiêu và yêu cầu của dự án. Mỗi công việc trong dự án sẽ liên quan đến một chỉ tiêu KPI và chiếm trọng số nhất định. Các chỉ tiêu KPI của dự án sẽ được chia cho nhân viên, nhân viên được phân công và thực hiện công việc, góp phần vào việc đạt chỉ tiêu KPI của dự án.

Phân bổ nguồn lực dự án theo chỉ tiêu KPI là phân bổ nguồn lực với mong muốn đáp ứng được các chỉ tiêu KPI, mục tiêu mà dự án đề ra. Trước khi tiến hành phân công thực hiện theo kế hoạch phân bổ, một phương án phân bổ cần đáp ứng các ràng buộc về thời gian, nhân lực, tài sản, thứ tự thực hiện công việc và đảm bảo mục tiêu về thời gian thực hiện, chi phí cũng như các chỉ tiêu KPI đã đề ra.

Khi phân bổ nguồn lực theo chỉ tiêu KPI, dự án có khả năng cao hơn để hoàn thành tốt các mục tiêu quan trọng đã đề ra. Bằng việc đặt ra từng chỉ tiêu KPI và gắn kết chúng với việc phân bổ nguồn lực, mọi cá nhân thuộc dự án sẽ xác định rõ ràng vai trò và nhiệm vụ của mình. Điều này cũng thúc đẩy tinh thần chủ động, nỗ lực từ các cá nhân. Dựa vào chỉ số hiệu suất của nhân viên trong quá khứ để giao việc, người quản lý sẽ có cơ sở để đảm bảo rằng khi giao công việc cho mỗi nhân viên thì nhân viên đó có thể hoàn thành theo yêu cầu mong muốn hay không. Từ đó, kế hoạch thực hiện công việc đề ra thông qua việc phân bổ nguồn lực sẽ đáng tin cậy hơn, khả thi hơn, đem lại tỷ lệ thành công cao hơn.

Tuy nhiên, có nhiều khó khăn khi xây dựng một phương án phân bổ nguồn lực

thực hiện dự án, phổ biến nhất chính là các ràng buộc, hạn chế về thời gian, nhân lực và tài sản, về thứ tự thực hiện khi có sự ưu tiên trước - sau giữa các công việc trong dự án. Thứ hai, mỗi một công việc đòi hỏi yêu cầu về năng lực của nhân lực và tài sản riêng, không phải nguồn lực nào cũng có thể đáp ứng được yêu cầu của công việc, đồng thời, không phải thời điểm nào nguồn lực cũng sẵn có, điều này càng làm cho việc phân bổ trở nên phức tạp hơn. Đặc biệt, khi đi kèm chỉ tiêu KPI đề ra trong dự án, nhà quản lý cần phải tìm ra phương án phân bổ nguồn lực sao cho có thể tối ưu được KPI của dự án đồng thời cân bằng được KPI của nhân viên tham gia dựa trên năng lực của từng người.

Hiện nay, có khá nhiều ứng dụng hỗ trợ phân bổ nguồn lực cho dự án thông qua việc cung cấp các tính năng để phân bổ, quản lý, theo dõi việc sử dụng nguồn lực. Có thể kể đến một vài sản phẩm tiêu biểu như: Microsoft Project¹ trên nền tảng ứng dụng máy tính; Trello², Jira³, Base Wework⁴, LiquidPlanner⁵ trên nền tảng ứng dụng web. Các ứng dụng trên cho phép người dùng có thể tạo ra các thẻ công việc và gán nguồn lực, gồm nhân lực, tài sản thực hiện công việc. Người dùng chỉ cần thực hiện các thao tác kéo thả là có thể phân bổ nguồn lực cho các công việc trong dự án một cách dễ dàng. Một số ứng dụng cho phép hiển thị lịch trình dự án dưới dạng bảng, biểu đồ trình tự, danh sách và đưa ra những thống kê rất trực quan, sinh động, phục vụ cho người dùng khai thác, nắm bắt tối đa thông tin về dự án.

Tuy nhiên, các phần mềm này vẫn cần người dùng thực hiện thao tác thủ công để phân bổ nguồn lực. Mặc dù một số phần mềm đã có những công cụ hỗ trợ tính toán và dự đoán thời gian thực hiện công việc, nhưng việc tự động hóa hoàn toàn quá trình phân bổ nguồn lực vẫn chưa phổ biến hoặc không được thực hiện một cách tự động đầy đủ. Một số phần mềm có tính năng gợi ý hoặc đề xuất phân bổ nguồn lực dựa trên các yếu tố như kỹ năng, kinh nghiệm và khả năng của nhân viên, nhưng người dùng vẫn cần thực hiện việc áp dụng thủ công các gợi ý này.

Tựu trung lại, các ứng dụng hỗ trợ phân bổ nguồn lực hiện tại vẫn còn những tồn tại, vấn đề đang gặp phải, tiêu biểu là: (i) chưa có cơ chế phân bổ nguồn lực thực hiện dự án một cách tự động, (ii) chưa quan tâm đến yêu cầu về năng lực của nhân viên, tài sản trong việc định nghĩa, mô tả công việc, (iii) chưa quan tâm đến yếu tố chỉ tiêu KPI của dự án, KPI của nhân viên thực hiện công việc.

Chỉ tập trung tối ưu hóa thời gian và chi phí mà bỏ qua yếu tố KPI khi phân bổ

¹<https://www.microsoft.com/vi-vn/microsoft-365/project>, truy cập lần cuối ngày 27/04/2024

²<https://trello.com>, truy cập lần cuối ngày 28/04/2024

³<https://www.atlassian.com/software/jira>, truy cập lần cuối ngày 29/04/2024

⁴<https://base.vn/wework>, truy cập lần cuối ngày 29/04/2024

⁵<https://www.liquidplanner.com>, truy cập lần cuối ngày 30/04/2024

nguồn lực sẽ gặp nhiều rủi ro. Điều này thường không đảm bảo nguồn lực được sử dụng hiệu quả nhất và có thể dẫn đến việc phân bổ không hợp lý. Nhân viên không phù hợp có thể được giao những công việc không thuộc thế mạnh của họ, dẫn đến hiệu quả thực hiện công việc kém. Ngay cả khi có năng lực tốt, một người cũng không chắc sẽ đạt hiệu quả cao trong mọi công việc. Hơn nữa, hai người có cùng năng lực nhưng làm cùng một công việc có thể mang lại kết quả khác nhau.

Nếu không căn cứ vào kết quả hiệu suất làm việc trong quá khứ của nhân viên thì rất khó để kết luận một phương án phân bổ là tối ưu và có thể đạt chất lượng đúng theo kế hoạch đề ra. Giả sử như một người làm việc nào đó trước kia đã phải điều chỉnh, mà khi giao nhiệm vụ tương tự, vẫn phân bổ cho người đó mà không quan tâm đến hiệu suất làm việc của họ thực hiện công việc đó trong quá khứ, rất dễ sẽ phải điều chỉnh lại dự án.

Việc giải quyết bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI mang lại nhiều lợi ích. Nó giúp người quản lý, nhà hoạch định dự án phân bổ nguồn lực một cách thuận tiện, hiệu quả và tiết kiệm. Đồng thời, việc này cũng mang lại một chiến lược thực hiện dự án đúng đắn, tối ưu hóa thời gian, chi phí và nguồn lực, cùng với việc hoàn thành các chỉ tiêu KPI đề ra. Qua đó, sự thành công của dự án được đẩy mạnh. Doanh nghiệp có thể khai thác và phát huy tối đa năng lực của nhân viên. Điều này mang lại nhiều lợi ích, tạo nền tảng cho sự thành công của mỗi cá nhân, mỗi dự án và doanh nghiệp nói chung.

1.2 Mục tiêu và phạm vi đề tài

Từ những đánh giá, phân tích về các vấn đề, tồn tại trên, ĐATN hướng đến việc áp dụng các phương pháp toán học, thuật toán tối ưu để xây dựng và tìm kiếm giải pháp cho bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI. Từ đó làm cơ sở xây dựng các chức năng, mô-đun cho ứng dụng hỗ trợ việc phân bổ này một cách tự động.

Các chức năng, mô-đun sẽ được phát triển và tích hợp vào hệ thống DXClan⁶ - ứng dụng giải pháp không gian làm việc số cho doanh nghiệp do Công ty Cổ phần Công nghệ An toàn thông tin và truyền thông Việt Nam (VNIST) cung cấp. DXClan là ứng dụng cung cấp các giải pháp về quản lý trong doanh nghiệp, phục vụ các công ty vừa và nhỏ trong tất cả các ngành liên quan đến công nghệ với các nhóm dịch vụ chất lượng cao bao gồm: quản lý KPI, quản lý công việc, quản lý tài liệu, quản lý nhân sự, quản lý tài sản.

Hiện tại, hệ thống đã có mô-đun quản lý dự án tuy nhiên vẫn chưa hỗ trợ các chức năng: (i) tạo dự án kèm các chỉ tiêu KPI và nguồn lực gồm nhân sự, tài sản;

⁶<https://dxclan.com>, truy cập lần cuối ngày 02/06/2024

(ii) phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI; và (iii) lập lịch thực hiện các công việc cho dự án. Cùng với đó, hệ thống DXClan vẫn tồn tại những hạn chế khác tương tự như đã đề cập ở các phần mềm khác.

Sản phẩm của DATN sẽ kế thừa các chức năng đã có sẵn của các mô-đun quản lý dự án, quản lý KPI, quản lý nhân sự, và phát triển thêm những chức năng mới. Đồng thời tích hợp vào hệ thống DXClan nhằm giải quyết những vấn đề đã nêu ở trên. Các chức năng chính được kế thừa, phát triển trong DATN bao gồm: (i) thêm, sửa, xóa dự án với danh sách công việc, nguồn lực và chỉ tiêu KPI; (ii) phân bổ nguồn lực thực hiện công việc trong dự án; (iii) lập lịch thực hiện công việc trong dự án; (iv) xem lịch nhân công, lịch máy móc tương ứng dữ liệu phân bổ; (v) quản lý thẻ công việc và (vi) quản lý bộ năng lực của nhân viên.

1.3 Định hướng giải pháp

DATN đã khảo sát, phân tích, nêu ra vấn đề của các ứng dụng hỗ trợ phân bổ nguồn lực thực hiện công việc trong dự án hiện nay trên thị trường và các chức năng hiện có phục vụ cho việc phân bổ nguồn lực của hệ thống DXClan.

Để đáp ứng, hoàn thiện, khắc phục những hạn chế, tồn tại đã chỉ ra ở trên, DATN đưa ra phương án giải quyết cho từng vấn đề: (i) mô hình hóa, xây dựng thuật toán giải quyết bài toán tối ưu phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI; (ii) kế thừa, tái cấu trúc, xây dựng các chức năng chính của ứng dụng hỗ trợ việc phân bổ nguồn lực cho bài toán trên và (iii) tích hợp các chức năng xây dựng vào hệ thống DXClan.

Từ đầu vào của một dự án bao gồm: (i) thời gian bắt đầu và kết thúc cho phép, (ii) các chỉ tiêu KPI kèm giá trị ngưỡng yêu cầu, (iii) tập công việc, (iv) tập nguồn lực gồm nhân viên và tài sản, (v) kết quả thực hiện công việc của nhân viên trong quá khứ, thuật toán đề xuất cần đưa ra một phương án phân bổ nhân viên, tài sản cho tập công việc kèm thời gian thực hiện tương ứng. Trước hết, kết quả phân bổ này cần thỏa mãn ràng buộc về thứ tự thực hiện công việc, tính sẵn có của nguồn lực. Thêm vào đó là mong muốn đạt được các mục tiêu về: (i) thời gian thực hiện các công việc của dự án, (ii) chi phí thực hiện các công việc dự án và (iii) đạt được các chỉ tiêu KPI.

Do bài toán thực hiện không thể quan tâm đến các yếu tố ngoại cảnh, kết hợp với hạn chế về mặt thời gian thực hiện DATN, giả sử một số giả thiết và phạm vi sau. Thứ nhất, trong quá trình phân bổ và thời gian dự kiến thực hiện các công việc, giả sử máy không bị hỏng hóc và trục trặc gì. Thứ hai, trong quá trình phân bổ và thời gian dự kiến thực hiện công việc, giả sử rằng không có sự thay đổi, nghỉ việc của nhân viên trong dự án. Thứ ba, mỗi công việc chỉ yêu cầu tối đa một người và

một loại tài sản.

Thuật toán đề xuất bao gồm 4 bước chính, cụ thể thực hiện theo thứ tự như sau. Ban đầu, sử dụng thuật toán sắp topo (Topological Sort) để sắp xếp thứ tự thực hiện công việc, đồng thời dùng phương pháp đường găng (Critical Path Method - CPM) để có cái nhìn tổng thể về thời gian thực hiện dự án. Tiếp theo, gán khung thời gian và tài sản thực hiện công việc. Tiếp theo, sử dụng thuật toán tìm kiếm hài hòa (Harmony Search) hoặc cải tiến của nó (DLHS) để phân bổ nhân viên thực hiện các công việc. Cuối cùng, thực hiện kiểm tra và điều chỉnh lại công việc xung đột nếu có.

Công nghệ được sử dụng cho ĐATN bao gồm nhiều thành phần khác nhau. Thư viện ReactJS [3] và Redux [4] hỗ trợ việc lập trình phía Frontend. ReactJS cung cấp cơ chế Single Page Application giúp tải trang nhanh hơn, còn Redux là thư viện hỗ trợ quản lý các trạng thái (state) của ứng dụng. Thư viện NodeJS [5] và ExpressJS [6] được sử dụng để lập trình phía Backend. Hệ quản trị cơ sở dữ liệu MongoDB [7] cho phép định nghĩa và sử dụng kiểu dữ liệu với cấu trúc phức tạp, linh hoạt và có khả năng mở rộng cao.

Các đóng góp chính của đề án bao gồm: (i) xây dựng giải thuật kết hợp sắp topo (Topological Sort), phương pháp đường găng (Critical Path Method - CPM) và tìm kiếm hài hòa (Harmony Search) hoặc cải tiến của nó (DLHS) cho bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI; (ii) so sánh hiệu quả thuật toán đề xuất với thuật toán giải quyết bài toán phân bổ nguồn lực thực hiện gói thầu, không quan tâm đến yếu tố KPI [1]; (iii) xây dựng, tái cấu trúc lại chức năng liên quan đến tạo, sửa, xóa dự án với chỉ tiêu KPI; (iv) xây dựng chức năng phân bổ nguồn lực thực hiện dự án; (v) xây dựng chức năng lập lịch thực hiện công việc cho dự án; (vi) xây dựng chức năng quản lý thẻ (tag) công việc; (vii) xây dựng chức năng quản lý bộ năng lực của nhân viên; (viii) tích hợp các chức năng xây dựng vào hệ thống DXClan.

1.4 Bố cục đề án

Phần còn lại của báo cáo ĐATN này được tổ chức như sau.

Chương 2 trình bày về khảo sát một số sản phẩm phần mềm, ứng dụng hỗ trợ phân bổ nguồn lực cho dự án trên thị trường hiện nay. Đồng thời, chương này cũng trình bày một số nghiên cứu, thuật toán đề xuất giải quyết cho bài toán phân bổ nguồn lực. Từ đó đưa ra những vấn đề còn tồn tại ở các ứng dụng, phương pháp thực hiện làm tiền đề để xác định rõ cải thiện và hướng giải pháp cho ĐATN.

Chương 3 trình bày về các công nghệ, kiến thức nền tảng, cơ sở lý thuyết được

sử dụng cho việc thực hiện ĐATN, những điểm nổi bật và lý do tại sao chọn những công nghệ, kiến thức đó. Mục tiêu của chương này là định hình cho người đọc về cơ sở lý luận và công nghệ cơ bản của hệ thống.

Chương 4 của đồ án sẽ tập trung vào việc thiết kế, phát triển và triển khai ứng dụng dựa trên những yêu cầu, chức năng đã phân tích. Cụ thể, chương này sẽ gồm các phần sau: (i) thiết kế kiến trúc, (ii) thiết kế chi tiết, (iii) xây dựng ứng dụng, (iv) kiểm thử và (v) triển khai. Phần thiết kế kiến trúc sẽ trình bày tổng quan về kiến trúc phần mềm và biểu đồ thiết kế các gói. Phần thiết kế chi tiết trình bày về thiết kế giao diện, thiết kế lớp và thiết kế cơ sở dữ liệu cho phần mềm. Tiếp theo, phần xây dựng ứng dụng trình bày về các công cụ sử dụng cho đồ án và kết quả đạt được của ĐATN. Phần kiểm thử sẽ trình bày về các kịch bản kiểm thử kèm kết quả thực hiện kiểm thử. Phần triển khai trình bày về cách cài đặt, triển khai ứng dụng.

Chương 5 trình bày về những đóng góp nổi bật trong ĐATN. Cụ thể, chương này sẽ trình bày về mô hình bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI. Thêm vào đó, chi tiết các bước của thuật toán đề xuất được xây dựng để giải quyết bài toán cũng được mô tả. Chương cũng nêu ra các thử nghiệm của thuật toán đề xuất, bao gồm cả bộ dữ liệu chuẩn và bộ dữ liệu thực tế, những so sánh giữa thuật toán đề xuất so với thuật toán phục vụ bài toán phân bổ nguồn lực đã có. Từ đó đưa ra và làm rõ những điểm nổi bật của thuật toán mà ĐATN đã xây dựng để giải quyết bài toán trên.

Chương 6 sẽ thực hiện đánh giá những kết quả đạt được cùng những điểm còn hạn chế, chưa thực hiện được của ĐATN. Từ đó đưa ra giải pháp, hướng phát triển trong tương lai.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

2.1.1 Khảo sát một số ứng dụng hỗ trợ phân bổ nguồn lực dự án

Trong phần này, em xin trình bày việc khảo sát về một số ứng dụng tương tự trên thị trường cung cấp chức năng phân bổ nguồn lực cho dự án, cụ thể là các phần mềm sau: Trello¹, Jira², Microsoft Project³, LiquidPlanner⁴ và hệ thống DXClan⁵.

Trello hỗ trợ quản lý dự án theo không gian làm việc (workspace), ta có thể khởi tạo dự án kèm theo các công việc của nó trong một bảng (board) của workspace. Ứng dụng này giúp ta quản lý dự án dựa trên hệ thống thẻ, cung cấp một giao diện trực quan để theo dõi các nhiệm vụ và phân bổ nguồn lực. Người dùng có thể tạo danh sách công việc, mỗi công việc dưới dạng thẻ, và phân bổ người thực hiện công việc bằng cách chọn thành viên tương ứng. Chúng ta cũng có thể gắn kèm tệp đính kèm, bình luận, và thêm nhãn để mô tả thêm chi tiết về công việc cũng như dễ dàng xác định các yếu tố quan trọng của mỗi công việc. Ngoài ra, Trello cũng cho phép người sử dụng hiển thị các công việc của dự án theo nhiều kiểu khác nhau (bảng, lịch, bảng điều khiển, bản đồ, lịch trình ...) và có những bảng thống kê để dễ dàng đối chiếu. Tuy nhiên, vấn đề gặp phải ở đây là Trello chưa hỗ trợ việc phân bổ nguồn lực một cách tự động và tối ưu, người dùng hoàn toàn phải thực hiện các thao tác thủ công để gán nhân lực thực hiện công việc. Hơn thế, ứng dụng chỉ hỗ trợ gán nguồn lực là nhân sự, đồng thời chưa hỗ trợ tốt về việc mô tả mối quan hệ trước - sau giữa các công việc. Theo như khảo sát, giả sử muốn mô tả một công việc bắt đầu khi những công việc tiên quyết của nó cần thực hiện trước, Trello chưa hỗ trợ thực hiện được.

Jira là một công cụ quản lý dự án mạnh mẽ và linh hoạt, phổ biến trong cộng đồng phát triển phần mềm và quản lý dự án. Jira cho phép người dùng tạo dự án (project) và quản lý nó, thực hiện thêm các công việc trong dự án dưới dạng các thẻ, gọi là Issues. Ta cũng có thể thực hiện thêm các file đính kèm, mô tả về công việc đã làm, gán người thực hiện công việc đó. Jira khắc phục được vấn đề công việc tiên quyết của Trello là khi tạo công việc, người dùng có thể định nghĩa quan hệ cha - con giữa các công việc để mô tả rằng công việc này cần có công việc khác hoàn thành trước khi nó thực hiện. Jira cũng cho phép người dùng có thể xem

¹<https://trello.com>, truy cập lần cuối ngày 28/04/2024

²<https://www.atlassian.com/software/jira>, truy cập lần cuối ngày 29/04/2024

³<https://www.microsoft.com/vi-vn/microsoft-365/project>, truy cập lần cuối ngày 27/04/2024

⁴<https://www.liquidplanner.com>, truy cập lần cuối ngày 30/04/2024

⁵<https://dxclan.com>, truy cập lần cuối ngày 02/06/2024

danh sách, trạng thái các công việc cũng như dự án thông qua nhiều kiểu hiển thị khác nhau như sơ đồ diễn biến theo thời gian, bảng biểu, danh sách, một số biểu đồ thống kê, ... cho người dùng. Tuy nhiên, vấn đề là Jira cũng chưa hỗ trợ việc phân bổ nguồn lực thực hiện công việc một cách tự động, người quản lý dự án vẫn cần thực một cách thủ công.

Microsoft Project là một ứng dụng quản lý dự án chuyên nghiệp cung cấp các tính năng mạnh mẽ để lập kế hoạch, theo dõi, và phân bổ nguồn lực. Ứng dụng cho phép người dùng tạo lập lịch trình dự án chi tiết, phân bổ nguồn lực, và tạo báo cáo tiến độ dự án. Microsoft Project cũng tích hợp với các ứng dụng khác của Microsoft như Microsoft Teams và Microsoft Excel để tối ưu hóa quá trình làm việc nhóm và trao đổi thông tin. Thế nhưng, nhược điểm của Microsoft Project là việc sử dụng khá khó khăn, đòi hỏi tính chuyên nghiệp, ít hỗ trợ các thao tác kéo thả, trực quan so với các ứng dụng khác. Bên cạnh đó, ứng dụng này cũng chưa có tính năng thực hiện phân bổ nguồn lực tối ưu theo chỉ tiêu KPI một cách tự động, người sử dụng vẫn phải gán thủ công cho từng công việc.

LiquidPlanner chủ yếu tập trung vào quản lý dự án, dự báo thời gian, và theo dõi tiến độ. Các công cụ và tính năng trong LiquidPlanner giúp người dùng quản lý các công việc, ước lượng thời gian, và gán nguồn lực, nhưng quá trình này thường phụ thuộc vào sự can thiệp và quyết định của người quản lý dự án. Người quản lý có thể theo dõi và xem xét thông tin dự án, các công việc cùng người thực hiện thông qua nhiều góc nhìn khác nhau dưới dạng các kiểu hiển thị, thống kê theo biểu đồ, bảng biểu. Người quản lý cũng có thể tạo, chỉnh sửa công việc trong dự án, phân bổ công việc cho thành viên thực hiện. Cái hay của ứng dụng ở chỗ khi thực hiện lập lịch cho công việc, sau khi thêm các công việc, phần mềm sẽ tự động tính toán và đề xuất được thời gian bắt đầu kèm dự đoán thời gian kết thúc của công việc và hiển thị cho người dùng một cách trực quan.

Đối với hệ thống phần mềm doanh nghiệp tích hợp DXClan, hiện tại, ở mô-đun quản lý thầu công, người dùng có thể thực hiện phân bổ nguồn lực thực hiện gói thầu gồm (thiết bị, nhân sự) một cách tự động, tuy nhiên chưa hề quan tâm đến yếu tố KPI, điều này có nhiều vấn đề xảy ra và đã được đề cập ở **Chương 1**.

Tất cả các ứng dụng trên đều giúp các nhà quản lý dự án dễ dàng quản lý và phân bổ nguồn lực, từ việc gán công việc cho nhân viên đến theo dõi tiến độ và điều chỉnh kế hoạch khi cần thiết. Tuy nhiên, tựu trung lại các ứng dụng đều có những nhược điểm, vấn đề tồn đọng cần được khắc phục. **Bảng 2.1** dưới đây tổng hợp những điểm, vấn đề còn tồn đọng của các ứng dụng đã khảo sát.

	Trello	Jira	Miscrosoft Project	Liquid Planner	DXClan
Mô hình về thứ tự trước - sau giữa các công việc trong dự án	Chưa có	Đã có	Đã có	Đã có	Đã có
Mô hình yêu cầu tài sản với công việc trong dự án	Chưa có	Chưa có	Chưa có	Chưa có	Đã có
Mô hình yêu cầu năng lực nhân viên, tài sản với công việc trong dự án	Chưa có	Chưa có	Chưa có	Chưa có	Chưa có
Phân bổ nguồn lực thực hiện dự án tự động	Chưa có	Chưa có	Chưa có	Chưa có	Đã có ở mô-đun quản lý thầu
Phân bổ nguồn lực có quan tâm yếu tố KPI	Chưa có	Chưa có	Chưa có	Chưa có	Chưa có

Bảng 2.1: Bảng so sánh chức năng của một số phần mềm hiện có

2.1.2 Khảo sát một số nghiên cứu, giải pháp cho bài toán phân bổ nguồn lực

Về bài toán phân bổ nguồn lực, đã có nhiều nghiên cứu, phương án đề xuất giải quyết bài toán. Tuy nhiên, xét trên ngữ cảnh nghiệp vụ, các hướng tiếp cận ấy không đủ để phân bổ nguồn lực, bao gồm nhân lực và tài sản thực hiện công việc trong dự án theo chỉ tiêu KPI đề ra. Dưới đây là một số giải thuật, hướng giải quyết đề xuất đã được công bố.

1. Giải thuật di truyền (Genetic Algorithm)

Giải thuật được đề xuất trong bài báo *Genetic Algorithms for Optimization of Resource Allocation in Large Scale Construction Project Management* [8]. Giải thuật giải quyết cho bài toán phân bổ nguồn lực tối ưu cho một dự án xây dựng con đập. Với đầu vào là danh sách công việc cần làm kèm thời gian thực hiện, nguồn lực bao gồm các máy móc phục vụ cho xây dựng như máy phá đá, máy đổ bê tông, máy đầm. Kết quả đầu ra là một phương án phân bổ tối ưu cục bộ, với mục tiêu tối ưu là cực tiểu hóa về thời gian thực hiện dự án. Giải pháp trên có hạn chế là chỉ có ràng buộc về tài sản, không quan tâm đến yếu tố nhân lực thực hiện dự án, đồng thời cũng không có ràng buộc về chia sẻ tài nguyên với các công việc khác ngoài dự án.

2. Giải thuật tối ưu hóa đàn kiến (ACO - Ant Colony Optimization)

Giải thuật trên được đề xuất trong bài báo *An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP)* [9]. Giải thuật giải quyết bài toán phân bổ nguồn lực tối ưu đa mục tiêu, với đầu vào là tập các công việc và tập nguồn nhân lực. Kết quả đầu ra là một phương án tối ưu cục bộ thỏa mãn mục tiêu tối ưu về hiệu suất và chi phí thực hiện. Giải pháp trên có hạn chế là chưa quan tâm đến các ràng buộc, yêu cầu về năng lực của nhân viên khi định nghĩa công việc cần thực hiện. Thứ hai, giải pháp đề xuất trên chưa có nguồn lực tài sản, cũng như chưa quan tâm đến ràng buộc thời gian nhân lực thực hiện các công việc khác. Hơn nữa, nhược điểm của thuật toán đàn kiến là hiệu suất có thể giảm đáng kể khi kích thước của bài toán tăng lên. Độ phức tạp tính toán và yêu cầu bộ nhớ sẽ tăng lên theo cấp số nhân dựa trên tham số đầu vào.

3. Giải thuật kết hợp sắp topo (Topological Sort) và thuật toán nhánh cận (Branch and bound)

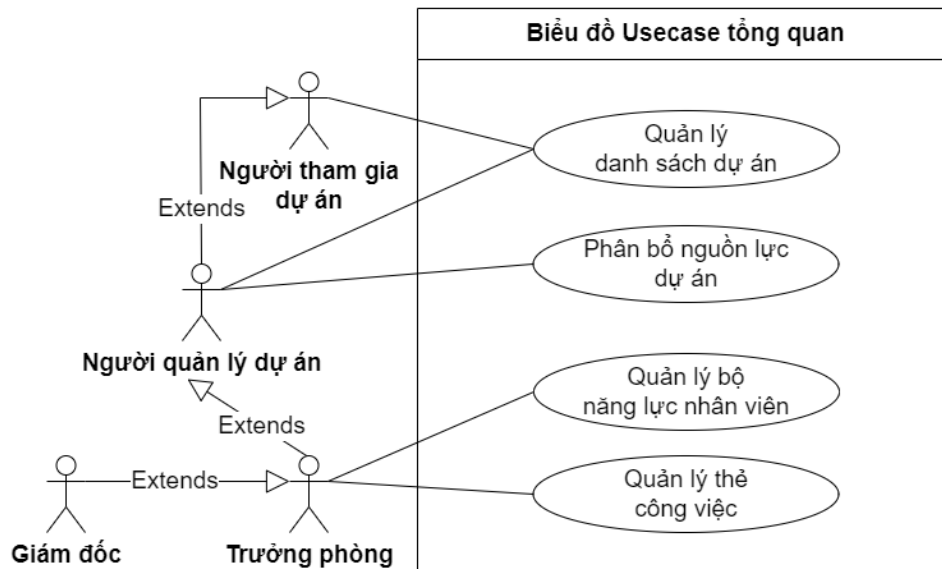
Giải thuật được đề xuất trong bài báo *Efficient Resource Allocation for Multiple Bid Packages and Projects in Enterprise* [1]. Giải thuật giải quyết bài toán phân bổ nguồn lực (gồm nhân lực, tài sản) thực hiện gói thầu. Đầu vào là danh sách nhân lực, tài sản kèm theo các công việc trong gói thầu. Đầu ra mong đợi là một phương án phân bổ tối ưu cục bộ thỏa mãn mục tiêu tối thiểu hóa về thời gian và chi phí thực hiện gói thầu. Giải pháp trên có ưu điểm hơn so với các giải pháp khác đã đề cập ở trên là đã quan tâm đến cả ràng buộc về thời gian thực hiện các công việc khác của tài nguyên. Song, phương pháp đề xuất này vẫn còn một số hạn chế. Đầu tiên, thuật toán sử dụng đệ quy, nên thời độ phức tạp tính toán trong trường hợp tồi nhất có thể lên đến cấp số mũ. Thứ hai, giải pháp chưa quan tâm đến yếu tố KPI khi phân bổ, yêu cầu về năng lực của nhân viên trong định nghĩa mô hình công việc.

Tóm lại, các thuật toán, phương pháp đề xuất cho bài toán phân bổ nguồn lực trên đều tồn tại chung một vấn đề là chưa quan tâm đến yếu tố KPI trong quá trình phân bổ và hiệu suất, năng lực của nhân viên khi thực hiện công việc.

Từ những phân tích đã nêu ở trên, phần tiếp theo của ĐATN sẽ trình bày về tổng quan các chức năng sẽ được xây dựng và tích hợp.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ usecase tổng quan



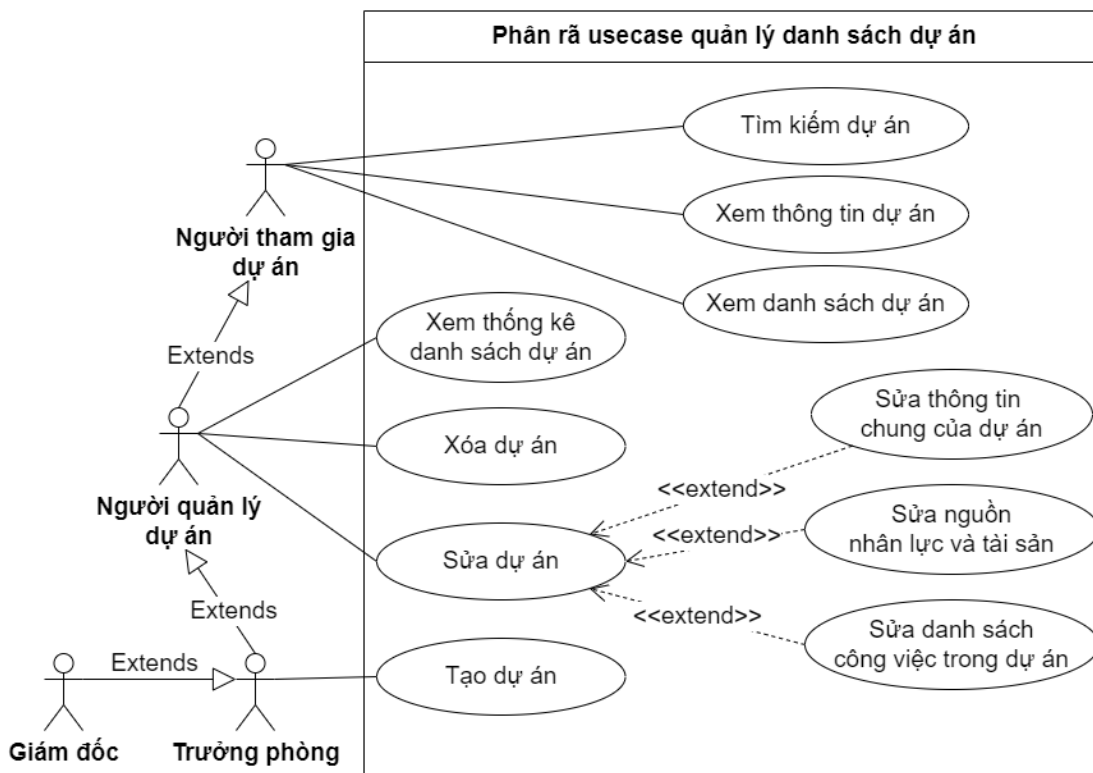
Hình 2.1: Biểu đồ usecase tổng quan

Hình 2.1 minh họa biểu đồ usecase tổng quan gồm các chức năng mà ĐATN phát triển, tích hợp vào DXClan. Cụ thể, các tác nhân gồm có **Người tham gia dự án**, **Người quản lý dự án**, **Trưởng phòng** và **Giám đốc**. Người quản lý dự án có thể thực hiện nhóm các chứng năng gồm: (i) một số chức năng liên quan đến quản lý danh sách dự án, (ii) phân bổ nguồn lực dự án. Người tham gia dự án hay một trong các thành viên của dự án sẽ có khả năng thực hiện một số chức năng liên quan đến quản lý danh sách dự án. Trưởng phòng và Giám đốc ngoài tất cả các chức năng có thể thực hiện với vai trò của người quản lý dự án, còn có thể thực hiện các nhóm chức năng sau: (i) quản lý bộ năng lực nhân viên, (ii) quản lý thể công việc. Tùy theo vai trò người dùng khác nhau được phân quyền trong hệ thống DXClan, dữ liệu hiển thị tương ứng các chức năng sẽ khác nhau.

2.2.2 Biểu đồ phân rã usecase quản lý danh sách dự án

Hình 2.2 mô tả biểu đồ phân rã usecase quản lý danh sách dự án. Thành viên tham gia dự án có thể xem danh sách dự án cũng như thông tin dự án mà mình được sắp xếp tham gia, đồng thời có thể tìm kiếm dự án theo: (i) tên dự án, (ii) tên, email của thành viên tham gia, (iii) tên, email của người quản lý. Thành viên tham gia dự án cũng có thể xem thông tin chi tiết về dự án mà mình tham gia.

Ngoài những chức năng của người tham gia dự án, người quản lý dự án có thể thực hiện các chức năng sau: (i) sửa dự án và (ii) xóa dự án và (iii) xem thống kê danh sách dự án.



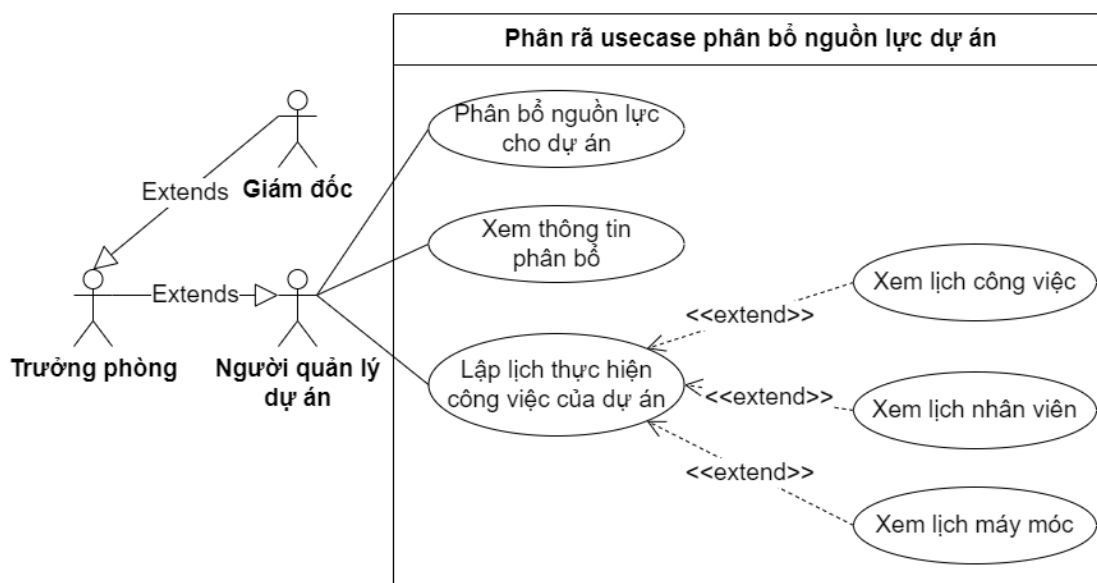
Hình 2.2: Biểu đồ phân rã usecase quản lý danh sách dự án

Với chức năng sửa dự án, người quản lý có thể chỉnh sửa thông tin chung của dự án, điều chỉnh nguồn nhân lực và tài sản, danh sách công việc trong dự án. Người quản lý dự án cũng có thể thực hiện chức năng xóa dự án để bỏ đi những dự án không cần thiết. Về chức năng xem thống kê dự án, người quản lý có thể xem xét một cách tổng quan về các dự án mà mình quản lý kèm theo trạng thái, ví dụ như: đã phân bổ, chưa phân bổ ..., cũng như thông tin về nguồn lực (nhân lực, tài sản) hiện có, chúng đang được sử dụng như thế nào.

Trưởng phòng và Giám đốc có thể thực hiện tất cả các chức năng quản lý danh sách dự án. Chức năng tạo dự án cho phép Trưởng phòng hay Giám đốc tạo một dự án, bao gồm việc thêm thông tin chung về dự án kèm theo các chỉ tiêu KPI đề ra, thêm nguồn lực (nhân lực, tài sản) thực hiện dự án và thêm danh sách công việc trong dự án.

Theo trên, dữ liệu, thông tin về danh sách dự án sẽ hiển thị và được quản lý theo vai trò người dùng trong DXClan. Đối với vai trò người dùng là Trưởng phòng, người quản lý có thể quản lý những dự án mà họ được giao phó, kèm theo những dự án mà nhân viên trong phòng ban đó tham gia thực hiện. Đối với vai trò người dùng là Giám đốc, người quản lý được quyền truy cập và quản lý tất cả các dự án của công ty.

2.2.3 Biểu đồ phân rã usecase phân bổ nguồn lực dự án



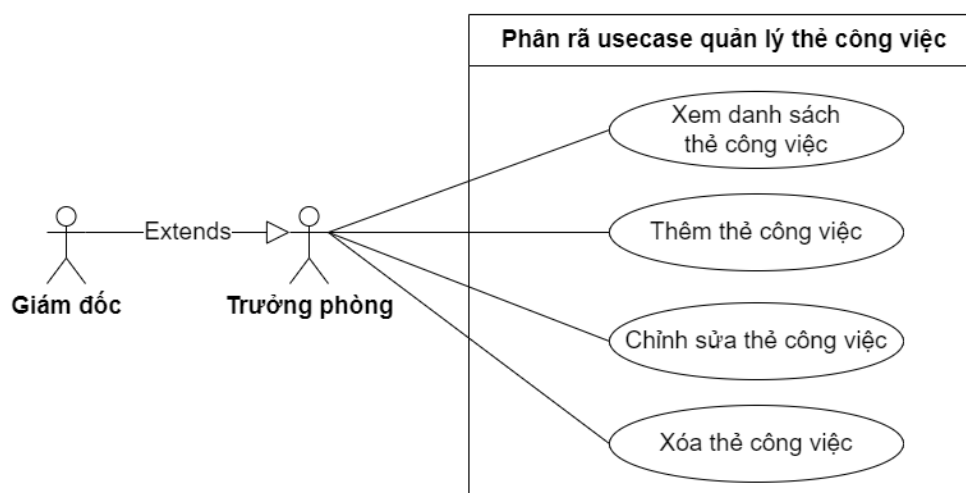
Hình 2.3: Biểu đồ phân rã usecase phân bổ nguồn lực dự án

Hình 2.3 mô tả biểu đồ phân rã usecase phân bổ nguồn lực dự án. Người quản lý dự án có thể thực hiện chức năng phân bổ nguồn lực cho dự án. Từ những thông tin về dự án như thời gian, chỉ tiêu KPI, nguồn lực, các công việc, ứng dụng cung cấp cho người dùng một phương án tối ưu cục bộ phân bổ nguồn lực hiện có để thực hiện các công việc đề ra trong dự án. Đối với những dự án đã thực hiện phân bổ, người quản lý có thể xem thông tin phân bổ. Đồng thời, họ có thể thực hiện việc lập lịch thực hiện công việc trong dự án. Kết quả đạt được bao gồm tập công việc của dự án kèm theo thời gian thực hiện, nhân công thực hiện và máy móc thực hiện những công việc đó.

Người quản lý dự án cũng có thể quan sát việc phân bổ dưới góc nhìn đa chiều hơn thông qua việc lựa chọn một trong ba loại lịch: xem lịch công việc, xem lịch nhân nhiên và xem lịch máy móc tương ứng với kết quả phân bổ đã có.

Tương tự như trên, xét theo vai trò người dùng, Trưởng phòng có thể thực hiện nhóm chức năng phân bổ nguồn lực với những dự án mà họ trực tiếp quản lý, hoặc dự án mà nhân viên trong phòng ban đó tham gia. Giám đốc có thể thực hiện nhóm chức năng trên với tất cả các dự án trong công ty.

2.2.4 Biểu đồ phân rã usecase quản lý thẻ công việc

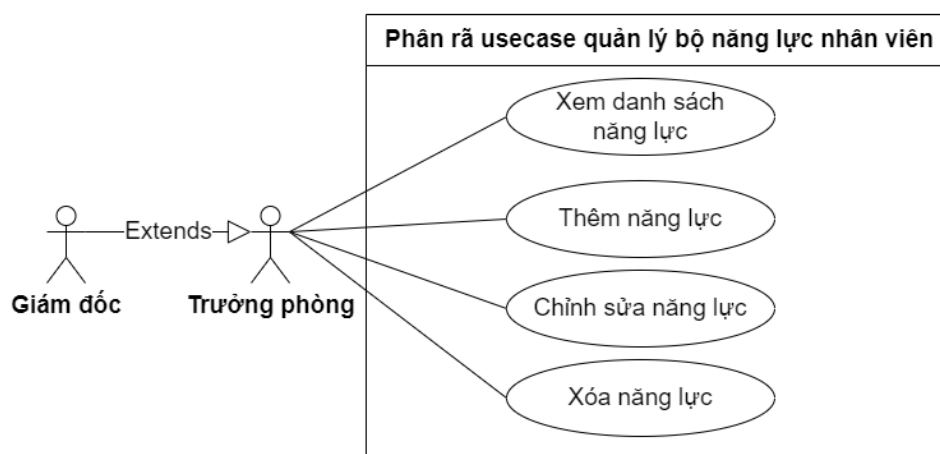


Hình 2.4: Biểu đồ phân rã usecase quản lý thẻ công việc

Hình 2.4 mô tả biểu đồ phân rã usecase quản lý thẻ công việc. Mỗi công việc sẽ thuộc về một lĩnh vực nhất định (ví dụ: phát triển phía backend, phát triển phía frontend, ...) và được gán vào một nhóm nghiệp vụ cụ thể, thông qua thẻ công việc. Trưởng phòng hoặc Giám đốc có thể xem xét danh sách thẻ công việc, thêm mới, chỉnh sửa hoặc xóa các thẻ này để định nghĩa và phân nhóm cho các công việc trong dự án.

2.2.5 Biểu đồ phân rã usecase quản lý bộ năng lực nhân viên

Hình 2.5 mô tả biểu đồ phân rã usecase quản lý bộ năng lực của nhân viên.

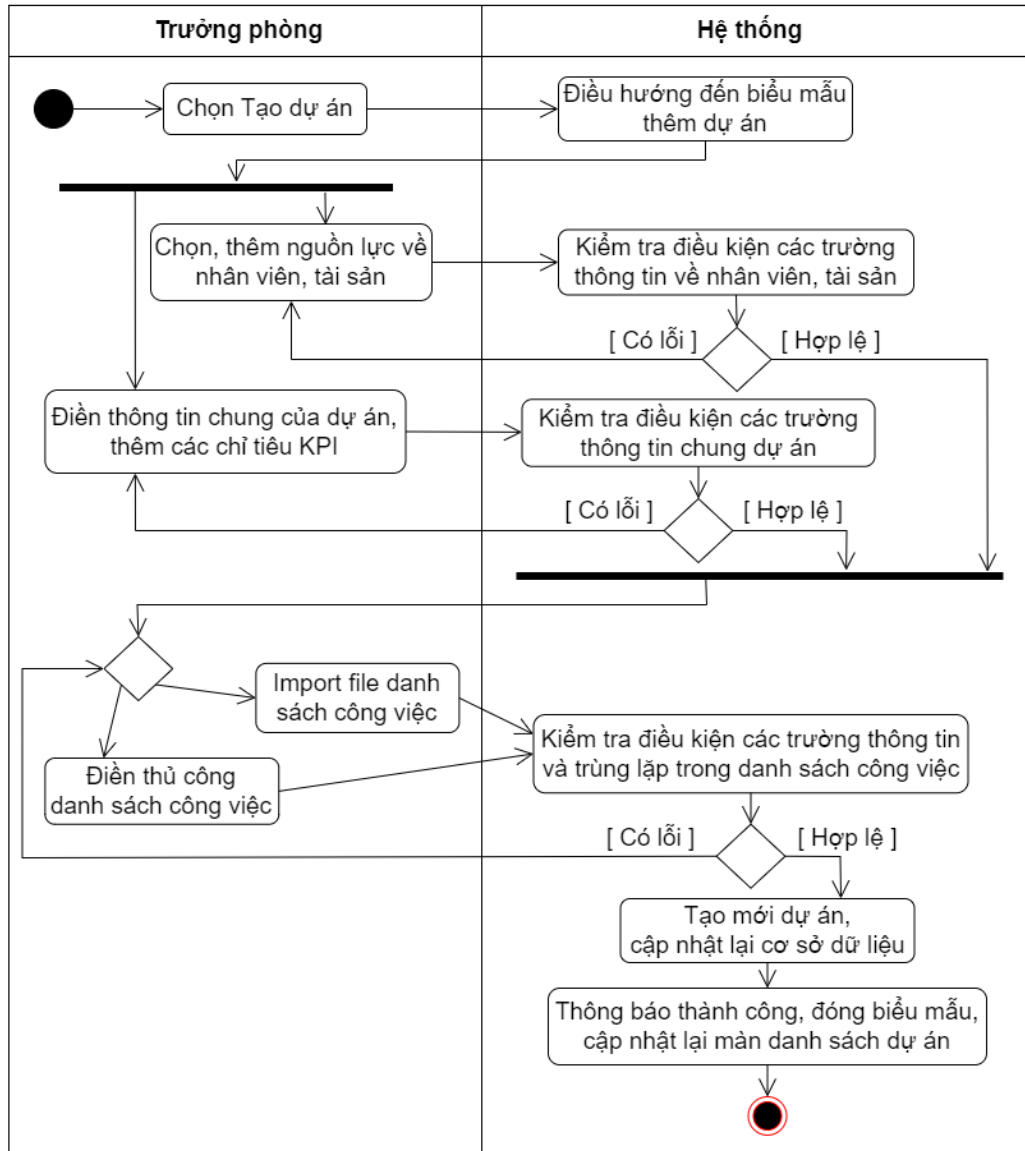


Hình 2.5: Biểu đồ phân rã usecase quản lý bộ năng lực nhân viên

Mỗi nhân viên sẽ có một bộ năng lực kèm theo các giá trị liên quan đến năng lực đó, ví dụ: nhân viên A có năng lực tiếng Anh ở mức B2. Trưởng phòng, Giám đốc có thể xem xét danh sách năng lực, thêm mới, chỉnh sửa hoặc xóa các năng lực

để gán năng lực tương ứng cho nhân viên. Bộ năng lực này không chỉ giúp định nghĩa và đánh giá năng lực của mỗi nhân viên mà còn giúp xác định xem một nhân viên có phù hợp với yêu cầu năng lực để thực hiện một công việc cụ thể hay không.

2.2.6 Quy trình nghiệp vụ tạo dự án



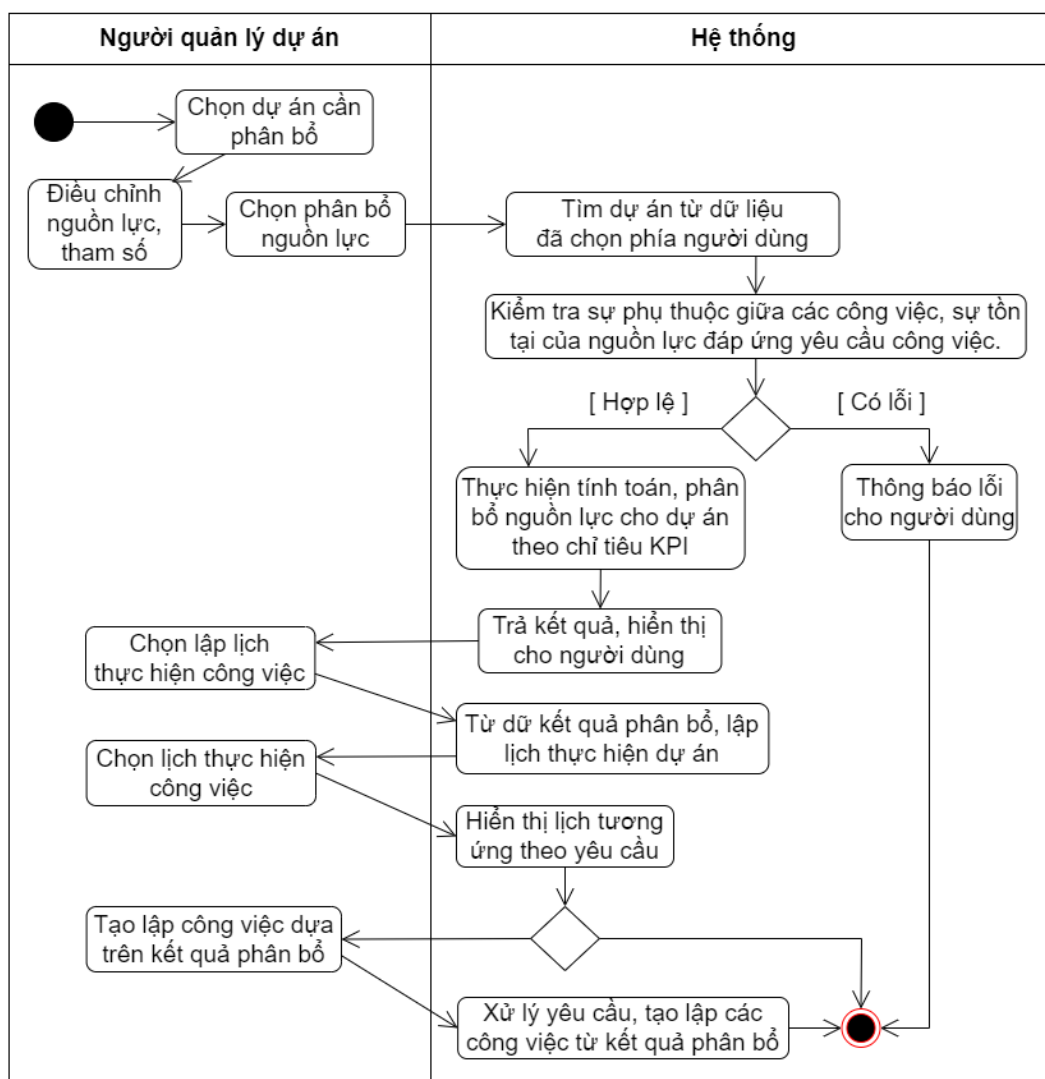
Hình 2.6: Biểu đồ hoạt động quá trình tạo dự án

Hình 2.6 mô tả về quy trình nghiệp vụ cho chức năng tạo dự án. Quy trình này thực hiện dựa trên việc tái cấu trúc chức năng tạo dự án. Mục đích là có thể tạo và thiết lập một dự án mang thông tin phù hợp với yêu cầu đầu vào của bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI mà ĐATN thực hiện.

Theo đó, Trưởng phòng (hoặc Giám đốc) sẽ thực hiện tạo một dự án cho công ty. Bỏ qua các thao tác chọn, điều hướng phía giao diện người dùng, việc tạo lập một dự án bao gồm 3 bước chính. Ban đầu, người dùng sẽ thực hiện điền thông tin

chung của dự án, kèm các chỉ tiêu KPI đề ra vào biểu mẫu. Tiếp đến, người dùng thêm nguồn lực gồm nhân lực, tài sản thực hiện dự án. Cuối cùng, Trưởng phòng (hoặc Giám đốc) thêm danh sách công việc trong dự án. Bước này có thể thực hiện bằng cách điền thủ công hoặc tải lên một tệp chứa danh sách công việc đã xây dựng sẵn. Ở mỗi bước, hệ thống sẽ kiểm tra điều kiện của các trường thông tin trong biểu mẫu và hiển thị lỗi nếu có. Chỉ khi đảm bảo không còn lỗi nào xảy ra, người dùng mới có thể thực hiện tạo dự án. Kết quả của quy trình nghiệp vụ trên là tạo và thiết lập được một dự án, làm đầu vào cho quá trình phân bổ nguồn lực dự án phía sau.

2.2.7 Quy trình nghiệp vụ phân bổ nguồn lực dự án



Hình 2.7: Biểu đồ hoạt động quá trình phân bổ nguồn lực dự án

Hình 2.7 mô tả về quy trình nghiệp vụ thực hiện chức năng phân bổ nguồn lực dự án. Từ thông tin dự án đã thiết lập, người quản lý dự án có thể điều chỉnh nguồn lực và thiết lập chiến lược phân bổ kèm tham số thuật toán phù hợp. Sau khi hoàn tất, họ chọn phân bổ nguồn lực, hệ thống sẽ thực hiện phân bổ nguồn lực tự động và

trả kết quả. Kết quả phân bổ trả về cho người dùng gồm: (i) lịch trình thực hiện dự án với thời gian bắt đầu, thời gian kết thúc, chi phí thực hiện; (ii) danh sách các chỉ tiêu KPI đạt được theo phân bổ; (iii) tập công việc kèm nhân viên, tài sản và thời gian thực hiện của từng công việc đó; (iv) KPI dự kiến của từng nhân viên tương ứng các chỉ tiêu KPI của dự án theo kết quả phân bổ. Từ kết quả phân bổ nguồn lực, người quản lý dự án có thể lập lịch thực hiện các công việc trong dự án. Sau khi đã xem xét, lựa chọn được phương án phân bổ, lịch thực hiện công việc phù hợp cho dự án, người quản lý có thể tạo lập công việc dựa trên kết quả phân bổ, phân công cho nhân viên, tài sản thực hiện tương ứng.

2.3 Đặc tả chức năng

2.3.1 Đặc tả usecase tạo dự án

Bảng 2.2 là bảng đặc tả usecase tạo dự án.

Mã usecase: UC01	Tên usecase: Tạo dự án
Tác nhân	Trưởng phòng hoặc Giám đốc
Mô tả vắn tắt	Cho phép người dùng có thể tạo một dự án
Tiền điều kiện	Người dùng đăng nhập thành công và đang ở trang Danh sách dự án
Luồng chính	<ol style="list-style-type: none"> 1. Người dùng chọn Thêm dự án 2. Hệ thống hiển thị biểu mẫu thêm dự án 3. Người dùng điền thông tin vào Tab thông tin chung của dự án 4. Hệ thống kiểm tra các trường thông tin liên quan đến thông tin chung của dự án 5. Người dùng chuyển sang Tab nguồn lực, chọn nguồn lực nhân viên, tài sản cho dự án 6. Hệ thống kiểm tra các trường thông tin liên quan đến nguồn lực 7. Người dùng chuyển sang Tab thêm công việc, tiến hành điền và thêm các công việc trong dự án 8. Hệ thống kiểm tra các trường thông tin liên quan đến công việc, kiểm tra điều kiện giữa các công việc 9. Người dùng ấn nút Lưu

Luồng chính	<p>10. Hệ thống tiếp nhận biểu mẫu, xử lý việc thêm dự án, lưu vào cơ sở dữ liệu và trả về thông báo cho người dùng</p> <p>11. Hệ thống đóng biểu mẫu, trở lại màn hình danh sách dự án và cập nhật lại danh sách dự án hiển thị cho người dùng</p>
Luồng thay thế 1	<p>9.1. Người dùng chọn Hủy</p> <p>10.1. Hệ thống đóng biểu mẫu thêm dự án, quay trở lại trang danh sách dự án</p>
Luồng thay thế 2	<p>4.2. Nếu người dùng nhập sai, thiếu một số trường thông tin, báo lỗi cho người dùng</p> <p>5.2. Quay lại bước 3</p>
Luồng thay thế 3	<p>6.3. Nếu người dùng nhập sai, thiếu một số trường thông tin, báo lỗi cho người dùng</p> <p>7.3. Quay lại bước 5</p>
Luồng thay thế 4	<p>8.4. Nếu người dùng nhập sai, thiếu một số trường thông tin, báo lỗi cho người dùng</p> <p>9.4. Quay lại bước 5</p>
Luồng thay thế 5	<p>7.5. Người dùng chuyển sang Tab thêm công việc, import danh sách công việc từ file</p> <p>8.5. Hệ thống kiểm tra danh sách công việc từ file import</p> <p>9.5. Tiếp tục bước 8 ở luồng chính</p>
Hậu điều kiện	Cập nhật thông tin dự án mới tạo vào cơ sở dữ liệu, đóng biểu mẫu và cập nhật lại màn hình danh sách dự án

Bảng 2.2: Đặc tả usecase tạo dự án

2.3.2 Đặc tả usecase phân bổ nguồn lực dự án

Bảng 2.3 đặc tả cho chức năng phân bổ nguồn lực dự án.

Mã usecase: UC02	Tên usecase: Phân bổ nguồn lực dự án
Tác nhân	Người quản lý dự án
Mô tả vắn tắt	Cho phép người quản lý thực hiện phân bổ nguồn lực thực hiện các công việc của dự án

Tiền điều kiện	Người dùng đăng nhập thành công và đang ở trang Phân bổ nguồn lực
Luồng chính	<ol style="list-style-type: none"> 1. Người dùng chọn dự án cần phân bổ, chọn chiến lược phân bổ 2. Người dùng xem, điều chỉnh nguồn lực, tham số đầu vào 3. Người dùng ấn Phân bổ 4. Hệ thống tìm dự án theo yêu cầu của người dùng, kiểm tra các điều kiện về quan hệ vòng tròn, về việc đáp ứng yêu cầu của các công việc 5. Hệ thống xử lý phân bổ nguồn lực dựa trên những thông tin của dự án, trả về kết quả phân bổ cho người dùng
Luồng thay thế 1	4.1. Nếu các công việc trong dự án có quan hệ theo vòng tròn khép kín, hoặc không có tài sản hoặc nhân viên đáp ứng được yêu cầu của công việc nào đó trong dự án hệ thống báo lỗi cho người dùng
Luồng thay thế 2	4.2. Nếu không có phương án phân bổ nào phù hợp với thời gian yêu cầu của dự án, hệ thống thông báo lỗi cho người dùng
Hậu điều kiện	Cập nhật lại thông tin dự án, hiển thị kết quả phân bổ cho người dùng

Bảng 2.3: Đặc tả usecase phân bổ nguồn lực dự án

2.3.3 Đặc tả usecase lập lịch thực hiện công việc của dự án

Bảng 2.4 đặc tả cho usecase lập lịch thực hiện công việc của dự án.

Mã usecase: UC03	Tên usecase: Lập lịch thực hiện công việc của dự án
Tác nhân	Người quản lý dự án
Mô tả vắn tắt	Cho phép người quản lý lập lịch thực hiện công việc của dự án từ kết quả phân bổ
Tiền điều kiện	Người dùng đăng nhập thành công và đang ở trang Phân bổ nguồn lực, đã thực hiện phân bổ nguồn lực cho dự án

Luồng chính	<ol style="list-style-type: none"> 1. Người dùng chọn Lập lịch thực hiện 2. Hệ thống xử lý từ dữ liệu phân bổ đã có, tạo ra lịch trình thực hiện dự án, gồm lịch trình thực hiện công việc kèm theo thời gian, nhân công và tài sản tham gia, hiển thị cho người dùng
Luồng thay thế 1	<ol style="list-style-type: none"> 1.1. Người dùng chọn xem lịch nhân công 2.1. Hệ thống xử lý từ dữ liệu phân bổ đã có, tạo ra nhân công, gồm thời gian thực hiện công việc của từng nhân công trong dự án
Luồng thay thế 2	<ol style="list-style-type: none"> 1.2. Người dùng chọn xem lịch tài sản 2.2. Hệ thống xử lý từ dữ liệu phân bổ đã có, tạo ra tài sản, gồm thời gian thực hiện công việc của các tài sản trong dự án
Hậu điều kiện	Hệ thống hiển thị lịch trình thực hiện công việc của dự án cho người dùng

Bảng 2.4: Đặc tả usecase lập lịch thực hiện công việc của dự án

2.3.4 Đặc tả usecase thêm thẻ công việc

Bảng 2.5 đặc tả cho chức năng thêm thẻ công việc.

Mã usecase: UC04	Tên usecase: Thêm thẻ công việc
Tác nhân	Trưởng phòng hoặc Giám đốc
Mô tả vắn tắt	Cho phép người dùng thực hiện thêm thẻ công việc để định nghĩa một nhóm, lĩnh vực mà công việc thuộc về
Tiền điều kiện	Người dùng đăng nhập thành công và đang ở trang danh sách thẻ công việc
Luồng chính	<ol style="list-style-type: none"> 1. Người dùng chọn Thêm thẻ công việc 2. Hệ thống hiển thị biểu mẫu điền thông tin thêm thẻ công việc 3. Người dùng điền thông tin cần thiết vào biểu mẫu 4. Hệ thống tiếp nhận biểu mẫu, kiểm tra các trường thông tin

Luồng chính	<p>5. Người dùng chọn Thêm</p> <p>6. Hệ thống xử lý thêm thẻ công việc, lưu dữ liệu vào cơ sở dữ liệu, thông báo thành công cho người dùng, đóng biểu mẫu và trở lại trang danh sách thẻ</p>
Luồng thay thế 1	<p>4.1. Nếu tồn tại trường thông tin không hợp lệ, hiển thị lỗi ra màn hình</p> <p>5.1. Quay lại bước 3</p>
Luồng thay thế 2	<p>5.2. Người dùng chọn Hủy</p> <p>6.2. Đóng biểu mẫu, trở lại trang danh sách thẻ công việc</p>
Hậu điều kiện	Cập nhật thông tin thẻ công việc mới tạo vào cơ sở dữ liệu, đóng biểu mẫu và cập nhật lại màn hình danh sách thẻ công việc

Bảng 2.5: Đặc tả usecase thêm thẻ công việc

2.4 Yêu cầu phi chức năng

2.4.1 Yêu cầu chung

Ứng dụng đảm bảo quyền truy cập các tài nguyên trên trang web, hiển thị đúng dữ liệu theo vai trò, nhiệm vụ, chức năng của người dùng đã được phân quyền tương ứng trên hệ thống DXClan. Đồng thời, cần có tính mở rộng để việc sửa đổi các mô-đun là ít nhất có thể khi yêu cầu người dùng tăng lên. Ngoài ra, ứng dụng cho phép hỗ trợ chuyển đổi song ngữ Anh - Việt để phù hợp cho nhiều đối tượng người dùng hơn.

2.4.2 Yêu cầu về giao diện

Ngoài việc cần tuân theo khung giao diện đã có của phần mềm DXClan, giao diện của ứng dụng cần phải thân thiện, trực quan và dễ dàng điều hướng. Màu sắc và bố cục cần nhất quán trên toàn bộ ứng dụng. Các nút điều hướng, các dòng thông báo nhanh thông thường sẽ ở phía trên và bên phải của bố cục trang. Với các biểu mẫu, các nút sẽ ở phía dưới, bên phải. Quy định màu sắc giữa các nút phải thống nhất với nhau, cụ thể với nút xóa, hủy sẽ có màu đỏ, các nút liên quan đến thêm, lưu sẽ có màu xanh lá. Đồng thời, giao diện cần đảm bảo phù hợp và tương thích với nhiều loại thiết bị, kích thước màn hình khác nhau.

2.4.3 Yêu cầu về bảo mật

Ứng dụng đảm bảo những thông tin của người dùng tránh bị truy cập và sử dụng trái phép. Thông tin nhạy cảm của tài khoản người dùng như mật khẩu cần được

mã hóa và không lưu trên trình duyệt.

Như đã đề cập, ứng dụng cần bảo mật dữ liệu theo vai trò người dùng trong DXClan. Người tham gia dự án chỉ có thể tìm kiếm và xem thông tin dự án mà họ tham gia. Người quản lý dự án chỉ được truy xuất và thực hiện các chức năng quản lý, phân bổ nguồn lực cho các dự án mà họ được phân công. Trưởng phòng có thể quản lý các dự án của họ và của nhân viên trong phòng ban. Giám đốc có thể truy xuất tất cả các dự án của công ty và thực hiện mọi chức năng mà các vai trò khác có thể thực hiện.

2.4.4 Yêu cầu về thuật toán, giải pháp đề xuất

Thuật toán đề xuất trong DATN nhằm giải quyết bài toán phân bổ nguồn lực theo chỉ tiêu KPI cần đáp ứng các yêu cầu và mục tiêu đề ra. So với các thuật toán và phương pháp hiện có [1], [10], [11], [12], thuật toán này xử lý nhiều yêu cầu và ràng buộc hơn, bao gồm: (i) bổ sung về yêu cầu năng lực nhân viên, yêu cầu tài sản khi định nghĩa công việc; (ii) xem xét yếu tố KPI khi phân bổ nguồn lực và (iii) quan tâm đến hiệu suất làm việc, năng lực của nhân viên đối với từng công việc. Thuật toán đề xuất cũng cần cải thiện độ phức tạp tính toán so với thuật toán [1]. Kết quả của thuật toán không chỉ đạt được nhiều mục tiêu tối ưu hơn mà còn phải có hiệu quả tốt hơn hoặc tương đương với các tiêu chí của thuật toán [1].

2.4.5 Yêu cầu về việc tích hợp vào hệ thống DXClan

Các chức năng được xây dựng trong DATN phải tương thích với cấu trúc dữ liệu hiện tại của DXClan, đảm bảo không gây xung đột hoặc mất mát dữ liệu. Nhóm chức năng quản lý danh sách dự án và phân bổ nguồn lực dự án cần liên kết dữ liệu với mô-đun quản lý nhân sự, mô-đun quản lý tài sản hiện có của DXClan. Điều này đảm bảo việc thiết lập và phân bổ nguồn lực cho dự án được thực hiện một cách chính xác. Ngoài ra, các chức năng liên quan đến quản lý bộ năng lực nhân viên cần kết nối với mô-đun quản lý nhân sự của DXClan, vì bộ năng lực liên quan trực tiếp đến các nhân viên.

Ngoài những yêu cầu đã đề cập phía trên, ứng dụng tích hợp vào hệ thống DXClan cần đảm bảo chạy ổn định, không bị gián đoạn các chức năng. Ngoài ra, việc xây dựng ứng dụng cần phải đảm bảo tuân theo các quy định, chuẩn tắc về cách thiết kế, cách viết mã của hệ thống tích hợp DXClan.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Chương này trình bày các công nghệ, thuật toán và cơ sở lý thuyết mà em đã học hỏi và sử dụng trong quá trình làm ĐATN. Đồng thời, chương cũng giới thiệu về hệ thống DXClan, nền tảng để tích hợp các chức năng mà ĐATN xây dựng.

3.1 Công nghệ sử dụng cho lập trình phía Frontend

3.1.1 ReactJS

ReactJS [3] là một thư viện JavaScript được phát triển bởi Facebook, giúp xây dựng giao diện người dùng một cách hiệu quả và thuận tiện.

Điểm nổi bật của ReactJS là kiến trúc dựa trên các thành phần, được gọi là component, cho phép chia giao diện thành các thành phần độc lập, dễ dàng tái sử dụng và quản lý. Tính năng Virtual DOM giúp tối ưu hóa việc cập nhật giao diện, nâng cao hiệu suất ứng dụng. JSX, một cú pháp mở rộng của JavaScript, cho phép viết mã HTML trực tiếp trong JavaScript, giúp mã nguồn dễ bảo trì và đọc hiểu. React áp dụng cơ chế one-way data binding, giúp việc kiểm soát và gỡ lỗi trở nên dễ dàng hơn.

Hệ sinh thái phong phú của React bao gồm nhiều thư viện hỗ trợ như React Router và Redux, cùng với cộng đồng lớn mạnh luôn sẵn sàng hỗ trợ qua các diễn đàn và blog. Với sự linh hoạt, hiệu suất cao và sự hỗ trợ mạnh mẽ từ cộng đồng, ReactJS đã trở thành lựa chọn ưu tiên cho việc phát triển các ứng dụng web hiện đại, được nhiều công ty lớn sử dụng.

ĐATN sử dụng ReactJS làm thư viện để xây dựng chương trình phía giao diện người dùng.

3.1.2 Redux

Redux [4] là một thư viện JavaScript mã nguồn mở được sử dụng để quản lý trạng thái (state) trong các ứng dụng JavaScript, phổ biến nhất là với React. Redux giúp đơn giản hóa việc quản lý state của ứng dụng thông qua một kiến trúc nhất quán và dễ dự đoán. Redux hoạt động dựa trên ba nguyên tắc chính: (i) toàn bộ state của ứng dụng được lưu trữ trong một cây state duy nhất, giúp dễ dàng theo dõi và quản lý; (ii) state là chỉ đọc và chỉ có thể được thay đổi bằng cách gửi các hành động (actions), là các đối tượng mô tả những gì đã xảy ra; (iii) thay đổi state được thực hiện bằng các hàm thuần túy gọi là reducers, nhận state hiện tại cùng với action làm đầu vào và trả về state mới.

Với Redux, các nhà phát triển có thể xây dựng các ứng dụng dễ dàng bảo trì, kiểm thử và mở rộng. Redux đã trở thành một công cụ không thể thiếu cho nhiều

dự án lớn và phức tạp, giúp các nhà phát triển duy trì sự nhất quán và dễ dự đoán trong quá trình phát triển ứng dụng.

Redux cung cấp một luồng dữ liệu nhất quán, giúp tránh các vấn đề phức tạp liên quan đến quản lý state và đồng bộ hóa state trong các thành phần của ứng dụng. Điều này giúp khắc phục được khó khăn trong việc quản lý các state khi mỗi component của React đều có state riêng và quản lý riêng rẽ với nhau. Vì những ưu điểm trên, hệ thống DXClan sử dụng Redux để thực hiện quản lý trạng thái (state) hiệu quả.

3.1.3 Tailwind CSS

Tailwind CSS [13] là một framework CSS tiện ích, được thiết kế để tăng tốc quá trình phát triển giao diện người dùng và duy trì tính nhất quán trong thiết kế. Tailwind CSS cung cấp một tập hợp các lớp tiện ích nhỏ gọn giúp xây dựng các giao diện phức tạp nhanh chóng và dễ dàng mà không cần viết nhiều mã CSS tùy chỉnh.

Framework này rất dễ tùy chỉnh thông qua tệp cấu hình, cho phép điều chỉnh các giá trị màu sắc, khoảng cách, kích thước font, và nhiều thuộc tính khác. Tailwind CSS hỗ trợ thiết kế đáp ứng (responsive design) với các lớp tiện ích, giúp thay đổi kiểu dáng trên các kích thước màn hình khác nhau.

Công cụ PurgeCSS tích hợp giúp loại bỏ các lớp CSS không sử dụng, giảm kích thước tệp và cải thiện hiệu suất tải trang. Tailwind CSS kết hợp tốt với các framework JavaScript hiện đại và tiêu biểu là React, hỗ trợ tạo ra các thành phần UI có thể tái sử dụng mà vẫn duy trì phong cách nhất quán.

3.2 Công nghệ sử dụng cho lập trình phía Backend

3.2.1 NodeJS

NodeJS [5] (hay Node.js) là một nền tảng chạy JavaScript phía máy chủ, được xây dựng trên trình thông dịch V8 JavaScript của Google Chrome. NodeJS cho phép các nhà phát triển xây dựng các ứng dụng mạng có hiệu suất cao và khả năng mở rộng tốt nhờ vào mô hình vào - ra không đồng bộ và hướng sự kiện.

NodeJS rất phù hợp cho các ứng dụng thời gian thực như chat, game online và các API RESTful. Với hệ sinh thái phong phú và cộng đồng phát triển lớn mạnh, NodeJS đã trở thành một công nghệ quan trọng trong phát triển web hiện đại. NodeJS kế thừa các lợi thế của JavaScript, chẳng hạn như hiệu suất ấn tượng và khả năng xử lý dữ liệu nhanh. Nó hỗ trợ tái sử dụng mã nguồn và mã dễ đọc, dễ viết, dễ bảo trì. Hơn nữa, NodeJS còn có một kho thư viện phong phú và cộng đồng hỗ trợ rộng lớn, cung cấp nhiều tài liệu và hướng dẫn miễn phí.

3.2.2 ExpressJS

ExpressJS [6] là một framework web Node.js mạnh mẽ và phổ biến, được biết đến với sự linh hoạt và hiệu suất cao trong việc xây dựng các ứng dụng web và API.

Một trong những ưu điểm chính của ExpressJS là sự đơn giản và dễ sử dụng. Với cú pháp dễ hiểu và các tính năng mạnh mẽ như middleware và routing, việc xây dựng các ứng dụng web với ExpressJS trở nên dễ dàng hơn bao giờ hết. Nó cung cấp một cách tiếp cận tự nhiên hơn trong việc xử lý yêu cầu HTTP, cho phép nhà phát triển tập trung vào logic của ứng dụng mà không phải lo lắng về các chi tiết kỹ thuật phức tạp.

ExpressJS cũng nổi tiếng với sự linh hoạt. Nó không áp đặt bất kỳ quy tắc hay cấu trúc cụ thể nào, cho phép bạn tự do tổ chức mã nguồn theo cách tốt nhất cho dự án của mình. Bạn có thể chọn sử dụng các thư viện và công nghệ phù hợp với yêu cầu cụ thể của ứng dụng mà không gặp rào cản nào từ ExpressJS.

3.2.3 MongoDB và Mongoose

MongoDB [7] là một hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) phổ biến, nổi tiếng với tính linh hoạt và hiệu suất cao. Với cấu trúc lưu trữ dữ liệu dưới dạng tài liệu (documents) và tập hợp (collections), MongoDB cho phép nhà phát triển lưu trữ và truy xuất dữ liệu một cách linh hoạt và hiệu quả. Điều này giúp cho việc phát triển các ứng dụng web, di động và Big Data trở nên dễ dàng hơn bao giờ hết.

MongoDB cung cấp một ngôn ngữ truy vấn mạnh mẽ và linh hoạt, cùng với các tính năng như replica set và sharding để đảm bảo độ tin cậy và hiệu suất cao. Với một cộng đồng phát triển lớn mạnh và nhiều tài liệu hỗ trợ, MongoDB là một lựa chọn ưa thích cho các dự án công nghệ lớn và phức tạp.

Do tính phức tạp về cấu trúc, quy mô lớn về cơ sở dữ liệu phía hệ thống DXClan, việc lựa chọn hệ quản trị cơ sở MongoDB cho ĐATN là hoàn toàn hợp lý.

ĐATN sử dụng Mongoose [14] để có thể truy vấn, tương tác với dữ liệu trong cơ sở dữ liệu của ứng dụng. Mongoose là một thư viện, một công cụ mạnh mẽ và linh hoạt được thiết kế để làm việc với MongoDB ứng dụng Node.js.

3.3 Hệ thống tích hợp DXClan

DXClan là một ứng dụng hỗ trợ giải pháp toàn diện giúp nâng cao hiệu quả hoạt động và quản lý doanh nghiệp, phục vụ các công ty vừa và nhỏ trong các ngành liên quan đến công nghệ. Là một nền tảng hiện đại, được thiết kế để hỗ trợ nhiều doanh nghiệp cùng lúc, DXClan cung cấp nhóm dịch vụ chất lượng cao bao gồm: quản lý KPI, quản lý công việc, quản lý tài liệu, quản lý nhân sự, quản lý tài sản, ..., v.v. DXClan mang lại một giải pháp mạnh mẽ, một môi trường làm việc số

hoàn chỉnh với các chức năng quản lý toàn diện.

Hệ thống DXClan triển khai mô hình phân quyền theo vai trò (RBAC), cho phép người dùng đảm nhận một hoặc nhiều vai trò với quyền hạn khác nhau trên các tài nguyên và giao diện. Mô hình RBAC của DXClan định cấu hình quyền truy cập dựa trên vai trò cụ thể của từng người dùng, từ quản trị viên hệ thống đến nhân viên thông thường. Điều này đảm bảo mỗi người chỉ truy cập được các tài nguyên và chức năng phù hợp với nhiệm vụ của mình, giúp tăng cường tính bảo mật và hiệu quả trong quản lý.

DXClan có nhiều mô-đun được thiết kế, như mô-đun quản lý KPI, quản lý nhân sự, và quản lý tài sản . . . , v.v. Mỗi mô-đun cung cấp các API phong phú, cho phép tích hợp và mở rộng chức năng của hệ thống một cách linh hoạt. Các API này cung cấp các chức năng như thêm mới, cập nhật, xóa và truy vấn thông tin chỉ số KPI, nhân sự, tài sản. Sản phẩm của ĐATN sẽ tái sử dụng các mô-đun này để truy vấn thông tin về chỉ tiêu KPI, nhân sự và tài sản, phục vụ cho các chức năng mà ĐATN sẽ xây dựng.

3.4 Một số công nghệ, thuật toán khác sử dụng trong ĐATN

Trong phần này, em xin trình bày về một số thuật toán, cơ sở lý thuyết cốt lõi mà ĐATN sử dụng trong quá trình xây dựng thuật toán đề xuất cho bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI.

3.4.1 Thuật toán sắp xếp topo (Topological Sort)

Thuật toán sắp xếp topo (Topological Sort) [15] là một thuật toán trong lĩnh vực lý thuyết đồ thị, được sử dụng để sắp xếp các đỉnh của một đồ thị có hướng sao cho mọi cạnh chỉ đi từ đỉnh trước đến đỉnh sau. Thuật toán này chú trọng vào việc tìm ra một thứ tự sắp xếp của các đỉnh sao cho không có chu trình trong đồ thị.

Thuật toán sắp xếp topo thường được áp dụng trong các bài toán như xây dựng ứng dụng phân tầng, quy hoạch công việc, và phân tích dữ liệu. Thuật toán hoạt động bắt đầu bằng việc tìm một đỉnh không có cạnh vào (in-degree) và thêm nó vào kết quả sắp xếp. Sau đó, loại bỏ đỉnh này cùng với tất cả các cạnh xuất phát từ nó và lặp lại quá trình cho đến khi không còn đỉnh nào có in-degree bằng 0.

Thuật toán sắp xếp topo thường được triển khai thông qua các phương pháp như DFS (Depth-First Search) hoặc BFS (Breadth-First Search) trên đồ thị. Độ phức tạp thời gian của thuật toán là $O(V + E)$, trong đó V là số lượng đỉnh và E là số lượng cạnh trong đồ thị.

Trong phạm vi của ĐATN, em sử dụng thuật toán sắp xếp topo để sắp xếp thứ tự thực hiện công việc trong một dự án, đảm bảo các công việc tiên nhiệm của công

việc bất kỳ đều thực hiện trước nó.

3.4.2 Phương pháp đường găng (Critical Path Method - CPM)

Phương pháp đường găng - Critical Path Method (CPM) [16] là một kỹ thuật quản lý dự án hiệu quả, được sử dụng để hoạch định các công việc, từ đó đánh giá thời gian hoàn thành tổng thể của dự án. CPM bắt đầu bằng việc liệt kê tất cả các công việc cần thiết và xác định mối quan hệ phụ thuộc giữa chúng, sau đó vẽ một sơ đồ mạng lưới để biểu diễn các công việc và mối quan hệ của chúng. Mỗi công việc được gán thời gian ước tính để hoàn thành, từ đó giúp xây dựng cơ sở cho các tính toán tiếp theo.

Điểm đặc biệt của CPM là việc tính toán thời gian bắt đầu và kết thúc sớm nhất (Earliest Start - ES and Earliest Finish - EF) cũng như thời gian bắt đầu và kết thúc muộn nhất (Latest Start - LS and Latest Finish - LF) cho mỗi công việc. Thời gian bắt đầu sớm nhất là thời điểm sớm nhất mà một công việc có thể bắt đầu, dựa trên việc hoàn thành các công việc trước đó. Tương tự, thời gian kết thúc sớm nhất là thời điểm sớm nhất mà một công việc có thể hoàn thành. Ngược lại, thời gian bắt đầu muộn nhất là thời điểm muộn nhất mà một công việc có thể bắt đầu mà không làm trì hoãn dự án, và thời gian kết thúc muộn nhất là thời điểm muộn nhất mà một công việc có thể hoàn thành mà không ảnh hưởng đến tiến độ dự án.

Phương pháp CPM được sử dụng như một bước trong thuật toán đề xuất để tính toán các thời điểm bắt đầu, kết thúc sớm nhất (ES, EF) và muộn nhất (LS, LF) cho từng công việc trong dự án sau khi chúng được sắp xếp thứ tự thực hiện. Điều này giúp hoạch định thời gian thực hiện từng công việc, xác định các luồng song song có thể cần thực hiện, và đánh giá xem nguồn lực hiện tại có sẵn sàng đáp ứng yêu cầu về thời gian của dự án hay không.

3.4.3 Thuật toán phân cụm K-Mean

Thuật toán phân cụm K-Mean là một trong những thuật toán phân cụm (clustering) phổ biến trong lĩnh vực học máy và khai thác dữ liệu. Mục tiêu của thuật toán là chia dữ liệu thành các cụm (clusters) sao cho các điểm trong một cụm có sự tương đồng cao với nhau và khác biệt đáng kể so với các điểm trong các cụm khác.

Cách hoạt động của thuật toán bắt đầu bằng việc chọn ngẫu nhiên k điểm làm trung tâm của các cụm ban đầu. Tiếp theo, mỗi điểm dữ liệu được gán vào cụm có trung tâm gần nhất dựa trên khoảng cách Euclid giữa điểm và trung tâm cụm. Sau đó, trung tâm của mỗi cụm được cập nhật bằng cách tính trung bình của tất cả các điểm trong cụm đó. Quá trình này lặp lại cho đến khi không có sự thay đổi nào trong việc gán điểm vào các cụm hoặc đạt được số lần lặp cần thiết.

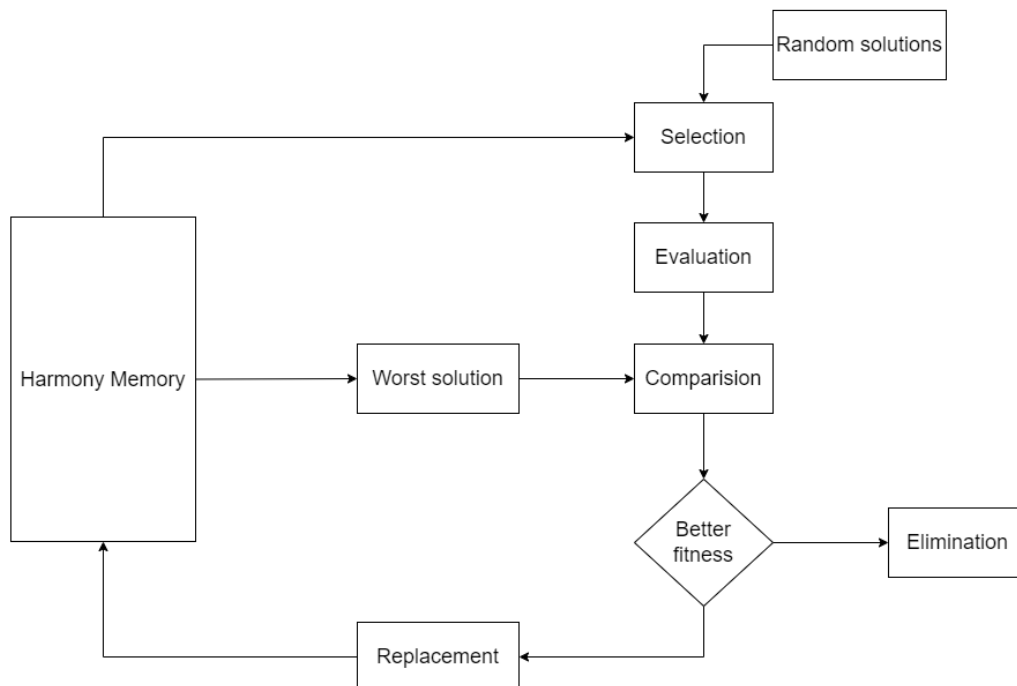
Thuật toán K-Mean có một số ưu điểm như đơn giản và dễ triển khai, thích hợp cho các tập dữ liệu lớn. Tuy nhiên, nó cũng có nhược điểm là cần phải biết trước số lượng cụm k và dễ bị ảnh hưởng bởi sự lựa chọn ban đầu của các trung tâm cụm. Đồng thời, nó cũng không hoạt động tốt trên các dữ liệu có các cụm có kích thước và hình dạng không đồng đều.

Trong thuật toán đề xuất của ĐATN, thuật toán K-Mean được sử dụng để phân chia nhóm nhân viên trong dự án theo năng lực.

3.4.4 Thuật toán tìm kiếm hài hòa - HS

Thuật toán tìm kiếm hài hòa - Harmony Search (HS) [17] là một thuật toán tìm kiếm được lấy cảm hứng từ quá trình hòa âm (harmonizing) trong âm nhạc. Nó là một trong những phương pháp tìm kiếm cực tiểu hoặc cực đại trong không gian nhiều chiều, thường được áp dụng trong các bài toán tối ưu hóa.

HS là một phương pháp tìm kiếm dựa trên việc tạo ra và kết hợp các giải pháp mới từ các giải pháp hiện có, giống như cách một nhóm nhạc tạo ra những giai điệu mới bằng cách kết hợp các nốt nhạc từ các bản nhạc khác nhau. **Hình 3.1** [17] mô tả quá trình thực hiện của thuật toán tìm kiếm hài hòa.



Hình 3.1: Sơ đồ mô tả thuật toán tìm kiếm hài hòa (Harmony Search)

Chi tiết các bước của thuật toán được thực hiện như sau:

1. Bước 1: Khởi tạo bộ nhớ hài hòa - Harmony Memory (HM)

Ở bước này, ta sẽ thực hiện khởi tạo một số lượng nhất định các giải pháp ngẫu nhiên cho bài toán tối ưu đang được xem xét, cụ thể số giải pháp cần

khởi tạo ngẫu nhiên được xác định bằng giá trị của kích thước bộ nhớ hài hòa - Harmony Memory Size (HMS). Cụ thể, với bài toán tối ưu có n chiều và biến x đang xét với $x = (x_1, x_2, \dots, x_n)$ là một phương án giải pháp hay một phương án (hoặc một véc-tơ hài hòa), x_i là một phần tử hay một ứng cử viên của giải pháp. Ta sẽ khởi tạo HMS véc-tơ giải pháp $x^i = (x_1^i, x_2^i, \dots, x_n^i)$, trong đó $i = 1, 2, \dots, HMS$. Các giải pháp khởi tạo này sẽ được đưa vào HM.

2. Bước 2: Cải tiến giải pháp mới từ HM

Tại bước này, thuật toán sẽ thực hiện tìm ra một giải pháp cải tiến $x^{new} = (x_1^{new}, x_2^{new}, \dots, x_n^{new})$ từ các giải pháp đã khởi tạo trong HM. Theo đó, mỗi ứng cử viên x_j^{new} giải pháp cải tiến được xác định dựa trên các tham số chính liên quan đến thuật toán sau:

- Tỷ lệ xem xét bộ nhớ hài hòa - Harmony Memory Considering Rate (HMCR): đây là xác suất để x_j^{new} được gán giá trị ứng cử viên x_j^i của một giải pháp x^i đã lưu trong HM. Ngược lại, x_j^{new} sẽ được khởi tạo một giá trị ngẫu nhiên.
- Tỷ lệ điều chỉnh cao độ - Pitching Adjust Rate (PAR), khoảng cách thay đổi cao độ - Pitch Adjustment Distance bandwidth (bw): khi x_j^{new} được gán giá trị ứng cử viên x_j^i của một giải pháp x^i đã lưu trong HM, PAR là xác suất để quyết định xem có thay đổi ứng cử viên đó hay không. Cụ thể x_j^{new} sẽ được tính toán dựa trên x_j^i và thay đổi nó với khoảng cách bw.

Hình 3.2 [17] mô tả cách tìm phương pháp cải tiến từ bộ nhớ HM.

```

for (j = 1 to n) do
  if (rand() < HMCR) then                                     // memory consideration
     $x_{new}(j) = x_a(j)$  where  $a \in (1, 2, \dots, HMS)$ 
    if (rand() < PAR) then                                     // pitch adjustment
       $x_{new}(j) = x_{new}(j) \pm rand() \times BW$ 
    endif
  else                                                         // random selection
     $x_{new}(j) = LB_j + rand() \times (UB_j - LB_j)$ 
  endif
endfor

```

Hình 3.2: Tìm giải pháp cải tiến dựa trên bộ nhớ HM

3. Bước 3: Thay thế giải pháp cải tiến

Tại bước này, ta sẽ thực hiện so sánh giải pháp cải tiến x^{new} so với phương án tồi tệ nhất x^{worst} có trong bộ nhớ HM. Trong trường hợp x^{new} đem lại kết quả mục tiêu tốt hơn so với x^{worst} , thực hiện thay thế x^{worst} bằng x^{new} trong bộ nhớ HM. Trong giải thuật này, các phương án giải pháp sẽ được so sánh

thông qua "độ hài hòa" được tính bởi "hàm hài hòa". Hàm hài hòa sẽ được định nghĩa phụ thuộc vào ngữ cảnh cũng như các mục tiêu của bài toán.

4. **Bước 4:** quá trình sẽ được lặp lại các bước 2, 3 cho đến khi đạt được một trong các điều kiện dừng (ví dụ như đã vượt qua số vòng lặp của thuật toán hay tìm được một phương án thực sự tốt).

Thuật toán tìm kiếm hài hòa có ưu điểm là có khả năng tìm ra lời giải tối ưu trong không gian nhiều chiều và không yêu cầu tính toán phức tạp.

3.4.5 Thuật toán tìm kiếm hài hòa tối ưu cục bộ dựa trên các quần thể con động - DLHS

Thuật toán tìm kiếm hài hòa tối ưu cục bộ dựa trên các quần thể con động - Local-best harmony search with dynamic subpopulations (DLHS) [18] là một biến thể của thuật toán tìm kiếm hài hòa đã nêu ở trên. Theo đó, trong thuật toán DLHS, các điểm cải tiến hơn so với thuật toán gốc bao gồm: (i) tìm kiếm hài hòa trên các quần thể con, (ii) việc tìm giải pháp cải tiến trong mỗi bước tận dụng tối đa kết quả của phương án cục bộ tốt nhất, (iii) lựa chọn và tự điều chỉnh các tham số linh hoạt qua từng giai đoạn của thuật toán thay vì thiết lập cố định ngay từ ban đầu.

1. Tìm kiếm hài hòa dựa trên các quần thể con

Bộ nhớ hài hòa (HM) của thuật toán DLHS ban đầu được chia ngẫu nhiên thành các tiểu quần thể (sub-HM). Sau đó, thuật toán thực hiện tìm kiếm hài hòa trên từng sub-HM một cách độc lập. Việc tìm kiếm trên các tiểu quần thể nhỏ giúp tăng tốc độ và hiệu quả. Tuy nhiên, phát triển độc lập trên các sub-HM nhỏ có thể dẫn đến tối ưu cục bộ và mất tính đa dạng. Để khắc phục vấn đề này, sau mỗi chu kỳ, các tiểu quần thể được gom lại và chia nhóm lại, rồi tiếp tục tìm kiếm hài hòa trên các nhóm mới. Quá trình này giúp trao đổi thông tin giữa các quần thể và tăng độ đa dạng của từng tiểu quần thể. Khi đạt đến giới hạn vòng lặp nhất định, thuật toán sẽ gom nhóm các tiểu quần thể lại để tạo thành bộ nhớ HM mới và thực hiện tìm kiếm hài hòa trên bộ nhớ này, nhằm tối đa hóa việc khai thác cục bộ.

2. Tìm giải pháp cải tiến trong mỗi vòng lặp

Khác với thuật toán HS gốc, tại mỗi vòng lặp trong một quần thể con, việc thực hiện tạo một giải pháp cải tiến x^{new} của DLHS sẽ ưu tiên lựa chọn và điều chỉnh từ phương án cục bộ tốt nhất của từng bộ nhớ hài hòa con sub-HM - x^{LB} (local-best) thay vì lựa chọn và điều chỉnh từ một phương án ngẫu nhiên trong bộ nhớ hài hòa như thuật toán HS. Điều này giúp khai thác tối đa được các phương án tối ưu cục bộ hiện có, từ đó cải tiến dần lên để tìm phương án tốt hơn. **Hình 3.3** [18] mô tả cách tạo ra một giải pháp cải tiến của thuật toán

DLHS trên bộ nhớ con sub-HM.

```

for ( $j = 1$  to  $n$ ) do
    if ( $rand() < HMCR$ ) then                                     // memory consideration
         $x_{new}(j) = x_{lB}(j)$ 
        if ( $rand() < PAR$ ) then                                     // pitch adjustment
             $x_{new}(j) = x_a(j) \pm rand() \times BW$ 
        endif
    else                                                         // random selection
         $x_{new}(j) = LB_j + rand() \times (UB_j - LB_j)$ 
    endif
endfor

```

Hình 3.3: Tìm giải pháp cải tiến dựa trên phương án tối ưu cục bộ của bộ nhớ con sub-HM

3. Lựa chọn và tự điều chỉnh các tham số linh hoạt

Nếu như ở thuật toán HS thuần túy, các tham số như: HMCR, PAR, bw đều cố định xuyên suốt quá trình thực hiện, thì với DLHS, các tham số này sẽ được thay đổi linh hoạt hơn trong quá trình tìm kiếm hài hòa. Đầu tiên, một bộ danh sách tham số (Parameter Set List - PSL) với độ dài xác định được tạo ra bằng cách điền vào các bộ tham số HMCR và PAR, trong đó các bộ tham số này được tạo ngẫu nhiên với sự phân bố đồng đều trong các khoảng $HMCR \in [0.9, 1]$ và $PAR \in [0, 1]$. Thuật toán DLHS được bắt đầu với tham số được đặt trong PSL. Ở mỗi thế hệ, một bộ tham số (HMCR, PAR) được lấy ra khỏi PSL và được sử dụng để tạo ra một giải pháp mới. Nếu giải pháp mới x^{new} thay thế thành công so với giải pháp kém nhất x^{worst} trong sub-HM của nó, thì bộ tham số (HMCR, PAR) vừa này được gọi là bộ tham số chiến thắng, lưu vào WPSL (winning parameter set list). Khi PSL trống, nó sẽ được điền lại bằng cách chọn 75% phần tử của PSL sẽ được lấy bằng cách chọn ngẫu nhiên từ tập WPSL, 25% còn lại lấy ngẫu nhiên như lúc trước $HMCR \in [0.9, 1]$ và $PAR \in [0, 1]$. Nếu như WPSL trống (điều này có thể xảy ra khi việc tìm kiếm gần đạt tới mục tối ưu với độ đa dạng của quần thể là không đáng kể), PSL cũ sẽ được dùng lại. Việc làm như vậy giúp ta có căn cứ và xác suất lớn hơn để tìm được một giải pháp cải tiến thực sự tốt.

Ngoài ra, tham số bw cũng không cố định, theo đó, khoảng cách thay đổi cao độ bw sẽ giảm linh hoạt dựa trên số lần tính toán. Giá trị bw lớn sẽ có lợi cho thuật toán thực hiện tìm kiếm toàn cục, trong khi giá trị bw nhỏ phù hợp để tìm kiếm xung quanh các vector giải pháp tốt nhất [18]. Công thức tính bw

theo từng vòng lặp được xác định như sau:

$$bw = \begin{cases} bwMax - \frac{2 \cdot (bwMax - bwMin) \cdot FEs}{maxFEs}, & \text{nếu } 2 \cdot FEs < maxFEs \\ bwMin, & \text{ngược lại} \end{cases} \quad (3.1)$$

Trong đó:

- $bwMin, bwMax$ là giá trị nhỏ nhất, lớn nhất của khoảng cách điều chỉnh cao độ bw, là tham số đầu vào của DLHS.
- $FEs, maxFEs$ lần lượt là vòng lặp hiện xét, tổng số vòng lặp thực hiện.

Thuật toán DLHS cải tiến hơn so với thuật toán HS ở nhiều khía cạnh, bao gồm tốc độ và khả năng tránh tối ưu cục bộ. DLHS tăng tốc độ thực hiện và cân bằng tốt hơn giữa tốc độ hội tụ sớm và độ đa dạng của quần thể. Việc thay đổi linh hoạt các tham số trong quá trình thực hiện giúp tìm ra các giải pháp tốt hơn. Mặc dù HS thuần túy theo lý thuyết có thể tìm ra giải pháp tối ưu hơn trong cùng một số vòng lặp giới hạn do thực hiện tìm kiếm hài hòa trên không gian với số lần tính toán lớn hơn, nhưng với những bài toán có không gian tìm kiếm đủ lớn, tốc độ tính toán của HS trở nên không hiệu quả. Trong những trường hợp này, việc áp dụng ý tưởng của thuật toán DLHS là hoàn toàn hợp lý, vì nó đưa ra các giải pháp tối ưu không kém HS nhưng với thời gian tính toán nhanh hơn nhiều.

Thuật toán HS và DLHS được áp dụng tại bước gán nhân lực thực hiện công việc trong dự án trong thuật toán đề xuất của DATN. Để phù hợp với ngữ cảnh bài toán và cải thiện hiệu quả cũng như thời gian thực hiện thuật toán, bước khởi tạo ngẫu nhiên các phương án lời giải vào bộ nhớ hài hòa HM sẽ được điều chỉnh. Cụ thể, các phương án này sẽ được khởi tạo ngẫu nhiên nhưng phải thỏa mãn ràng buộc, tức là ngẫu nhiên trong một không gian tìm kiếm giới hạn, không phải ngẫu nhiên hoàn toàn. Chi tiết sẽ được trình bày ở **Chương 5**.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

Theo như hệ thống DXClan, phần mềm được phát triển trên mô hình kiến trúc khách - chủ (client - server). Trong đó, các máy khách (client) sẽ gửi yêu cầu (request) tới máy chủ (server) thông qua một giao thức mạng, phổ biến là HTTP/HTTPS. Sau khi tiếp nhận các yêu cầu từ máy khách, máy chủ sẽ thực hiện xử lý các yêu cầu (thực hiện các tác vụ, truy xuất dữ liệu, tính toán...) rồi trả về kết quả theo yêu cầu của phía máy khách.

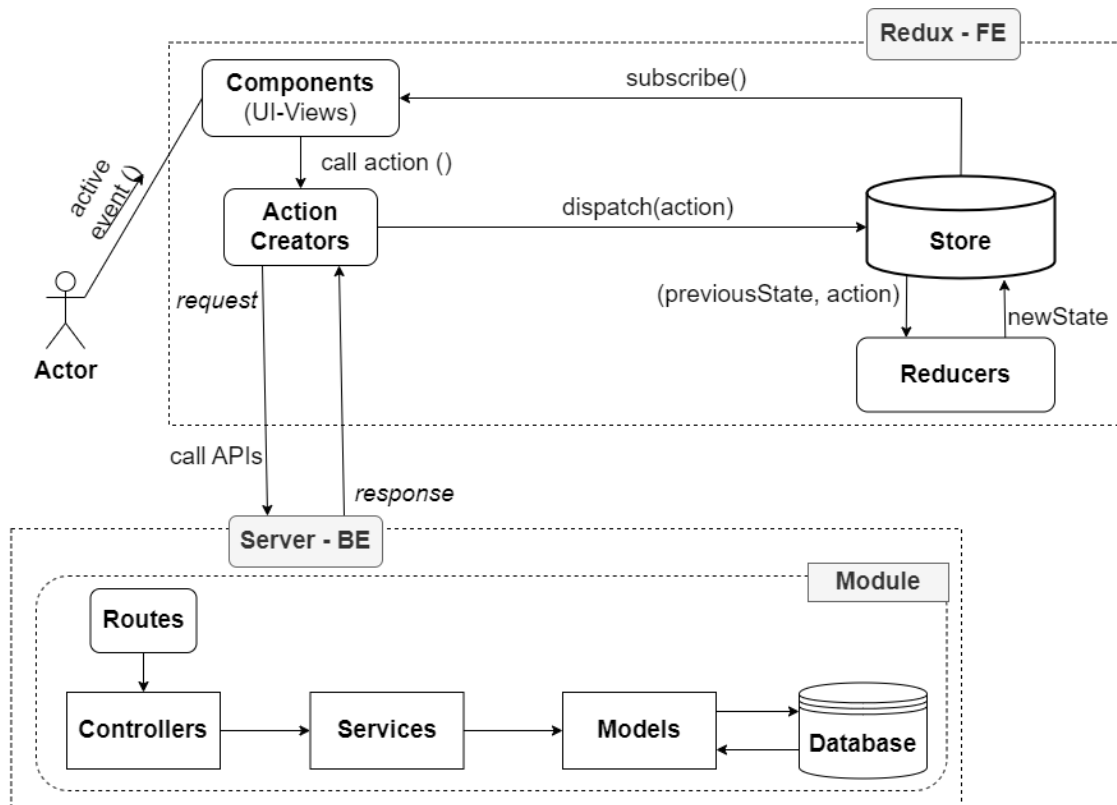
Trong kiến trúc này, vai trò của client và server được phân tách rõ ràng, giúp dễ bảo trì và phát triển. Client chịu trách nhiệm gửi yêu cầu và hiển thị dữ liệu, trong khi server xử lý yêu cầu và quản lý dữ liệu. Việc tách biệt nhiệm vụ này cho phép các nhà phát triển tập trung vào cải thiện từng phần mà không ảnh hưởng đến phần còn lại, ví dụ như thay đổi giao diện người dùng trên client không ảnh hưởng đến logic xử lý trên server. Kiến trúc client-server cũng cho phép tối ưu hóa cho từng mục đích cụ thể, sử dụng các công nghệ hiện đại cho giao diện trên client và công nghệ mạnh mẽ cho xử lý dữ liệu trên server.

Hình 4.1 mô tả kiến trúc tổng quan của ứng dụng, tuân theo mô hình client - server đã nêu ở trên. Cụ thể gồm 2 phần frontend (client) với các thành phần giao diện, hiển thị và backend (server) với vai trò xử lý các yêu cầu.

Phía frontend được phát triển theo kiến trúc Redux, cụ thể kiến trúc Redux cho phép quản lý trạng thái (state) của ứng dụng một cách dễ dàng hơn. Cụ thể, kiến trúc Redux bao gồm những thành phần chính sau: (i) state biểu thị trạng thái của ứng dụng, (ii) Store là nơi lưu trữ state, với một ứng dụng Redux thì store là duy nhất, (iii) Actions - định nghĩa các hành động, sự kiện và dữ liệu cần thiết gửi đến kho lưu trữ trạng thái (store) và (iv) Reducers - thành phần thực hiện thay đổi trạng thái (state) của ứng dụng dựa theo trạng thái hiện tại và hành động chuyển đến store.

Phía backend bao gồm các mô-đun (thư mục modules), thành phần của mỗi mô-đun gồm: (i) Routers, (ii) Services, (iii) Controllers và (iv) Models. Khi một yêu cầu HTTP/HTTPS được gửi từ client đến server, Router nhận yêu cầu và ánh xạ URL đến phương thức tương ứng trong Controller. Controller nhận yêu cầu từ Router, xử lý logic cơ bản và tương tác với service để thực hiện các tác vụ nghiệp vụ. Service nhận yêu cầu từ Controller, thực hiện các logic nghiệp vụ phức tạp

và tương tác với Model - thành phần để truy xuất hoặc cập nhật dữ liệu trong cơ sở dữ liệu. Sau khi Service xử lý xong, Controller sẽ định dạng lại kết quả và trả về phản hồi cho phía client. Việc tách biệt vai trò của Controller (xử lý yêu cầu HTTP/HTTPs) và Service (xử lý logic nghiệp vụ) làm đơn giản hóa Controller, đồng thời dễ kiểm thử, phát hiện lỗi và đảm bảo tính chính xác của nghiệp vụ trong Service.



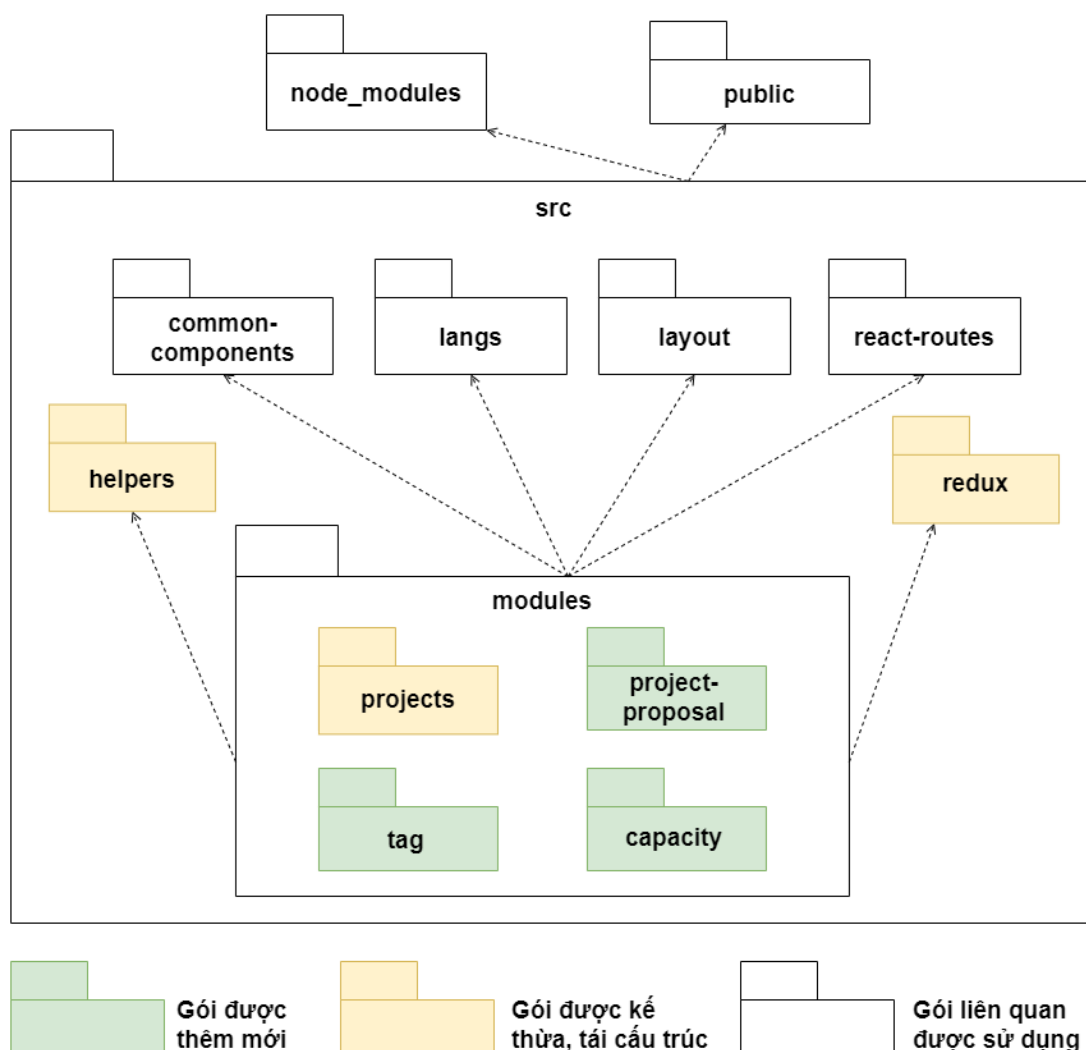
Hình 4.1: Kiến trúc tổng quan của hệ thống

4.1.2 Thiết kế tổng quan

Trên cơ sở mô hình kiến trúc tổng quan của ứng dụng, các chức năng mà ĐATN phát triển và tích hợp vào hệ thống DXClan được đóng gói và thiết kế tổng quan như dưới đây.

Hình 4.2 mô tả kiến trúc tổng quan của ứng dụng phía client. Các gói gồm có: (i) gói projects gồm các thành phần để xây dựng chức năng liên quan đến quản lý danh sách dự án; (ii) gói project-proposal để xây dựng các chức năng liên quan đến phân bổ nguồn lực dự án, lập lịch thực hiện dự án; (iii) gói tag cung cấp chức năng quản lý thẻ công việc; (iv) gói capacity cung cấp chức năng quản lý bộ năng lực nhân viên; (v) gói helpers gồm các hàm tiện ích và được sử dụng tại nhiều chỗ phía client; (vi) gói common-components chứa các thành phần giao diện dùng chung; (vii) gói langs hỗ trợ định nghĩa và chuyển đổi ngôn ngữ tiếng Anh - tiếng Việt; (viii) gói layout chứa các thành phần giao diện làm khung cho toàn bộ giao

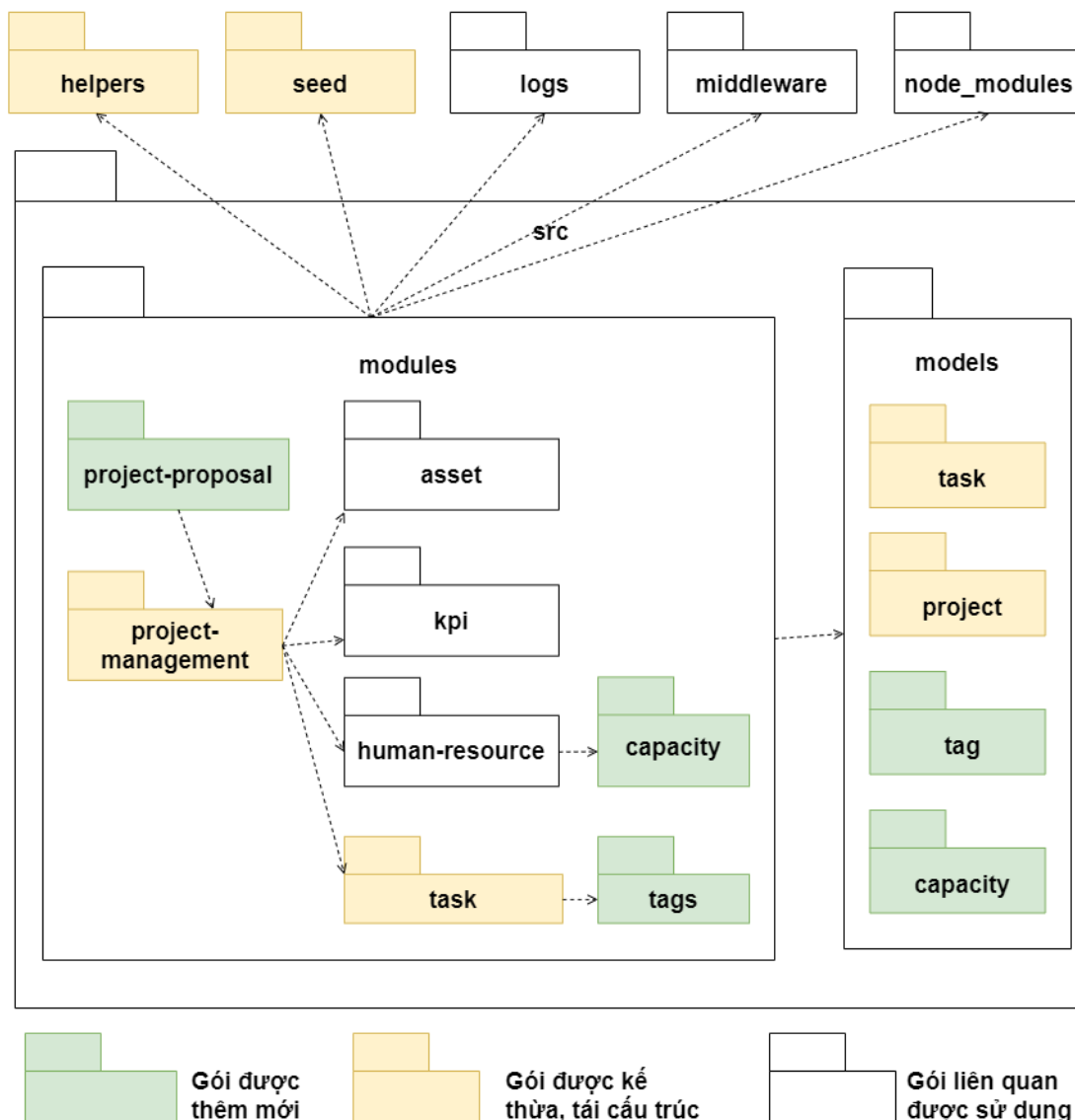
diện trang web; (ix) gói react-routes chứa các thành phần định tuyến để liên kết giữa URL trang và các thành phần trang hiển thị tương ứng; (x) gói redux định nghĩa kho lưu trữ trạng thái cục bộ (store) cho cả ứng dụng phía client; (xi) gói node_modules lưu trữ các thư viện đã cài đặt, các thành phần trong ứng dụng phía client sẽ sử dụng các thư viện được lưu trữ tại đây; (xii) gói public để lưu trữ một số file, hình ảnh được sử dụng chung cho cả ứng dụng.



Hình 4.2: Kiến trúc tổng quan của ứng dụng phía client

Hình 4.3 mô tả kiến trúc tổng quan của ứng dụng phía server. Các gói con thuộc gói modules cung cấp chức năng tiếp nhận và xử lý nghiệp vụ cho các yêu cầu đến từ client. Các gói con thuộc gói models cung cấp các thực thể để truy xuất trực tiếp tới các thành phần trong cơ sở dữ liệu. Cụ thể, các gói phía server bao gồm: (i) gói project-management chứa các thành phần để thực hiện chức năng liên quan đến quản lý danh sách dự án; (ii) gói project-proposal thực hiện các chức năng liên quan đến phân bổ nguồn lực dự án; (iii) gói tags thực hiện các chức năng liên quan đến quản lý thẻ công việc; (iv) gói capacity thực hiện các chức năng liên quan đến

quản lý bộ năng lực nhân viên; các gói (v) project, (vi) task, (vii) tag, (viii) capacity thuộc phần models định nghĩa các thực thể cho phép truy xuất và tương tác trực tiếp với cơ sở dữ liệu; (ix) gói helpers chứa các hàm tiện ích dùng chung cho các gói của modules; (x) gói seed phục vụ cho việc khởi tạo dữ liệu ban đầu cho ứng dụng; (xi) gói logs lưu các yêu cầu mà client gửi đến; (xii) gói middleware phục vụ việc xác thực các yêu cầu HTTP/HTTPS từ client; (xiii) gói node_modules, tương tự như trên để lưu trữ các thư viện đã cài; (xiv) gói task của phần modules liên quan đến việc thực hiện thêm, sửa xóa các thực thể công việc trong một dự án.

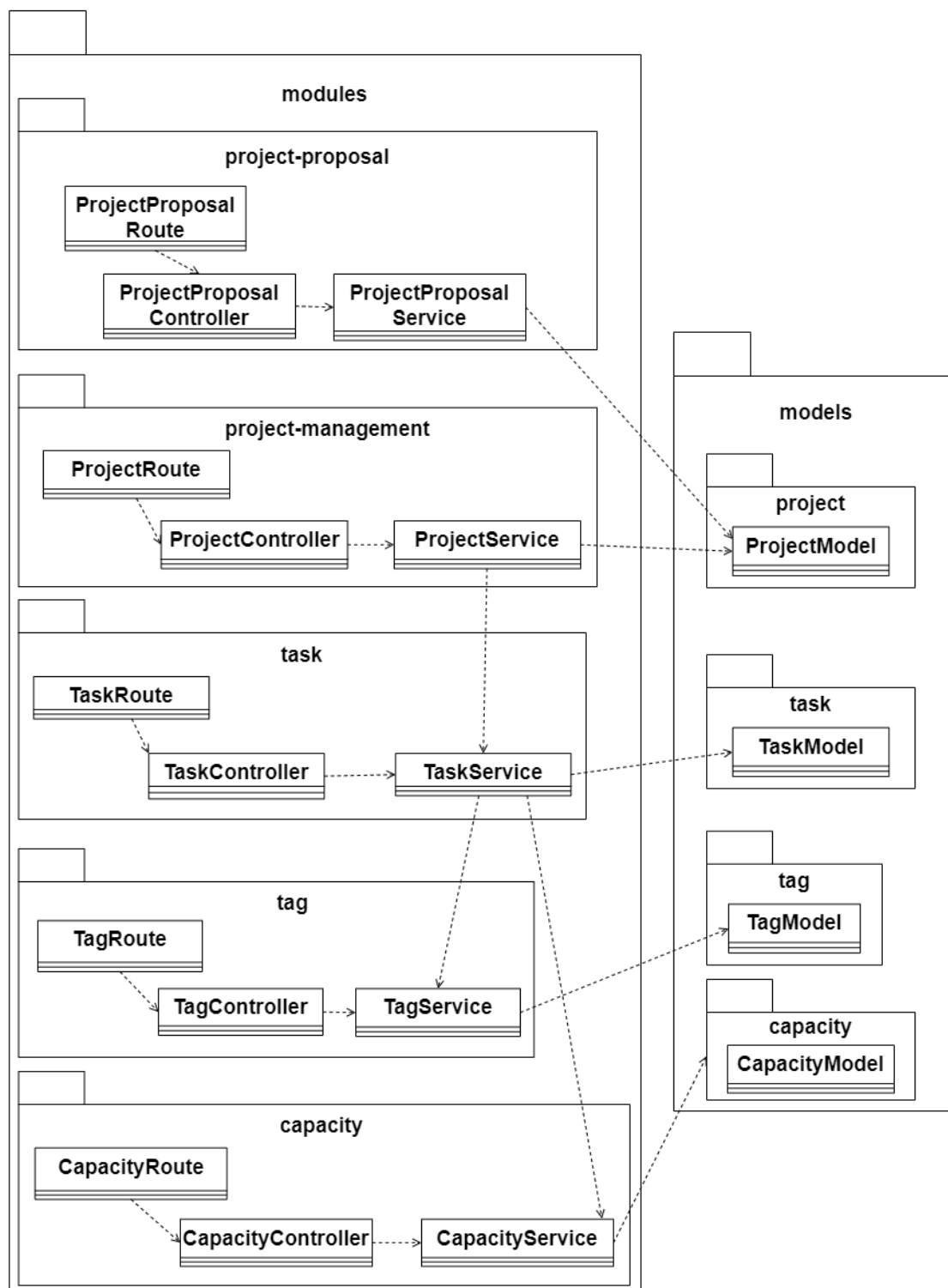


Hình 4.3: Kiến trúc tổng quan của ứng dụng phía server

4.1.3 Thiết kế chi tiết gói

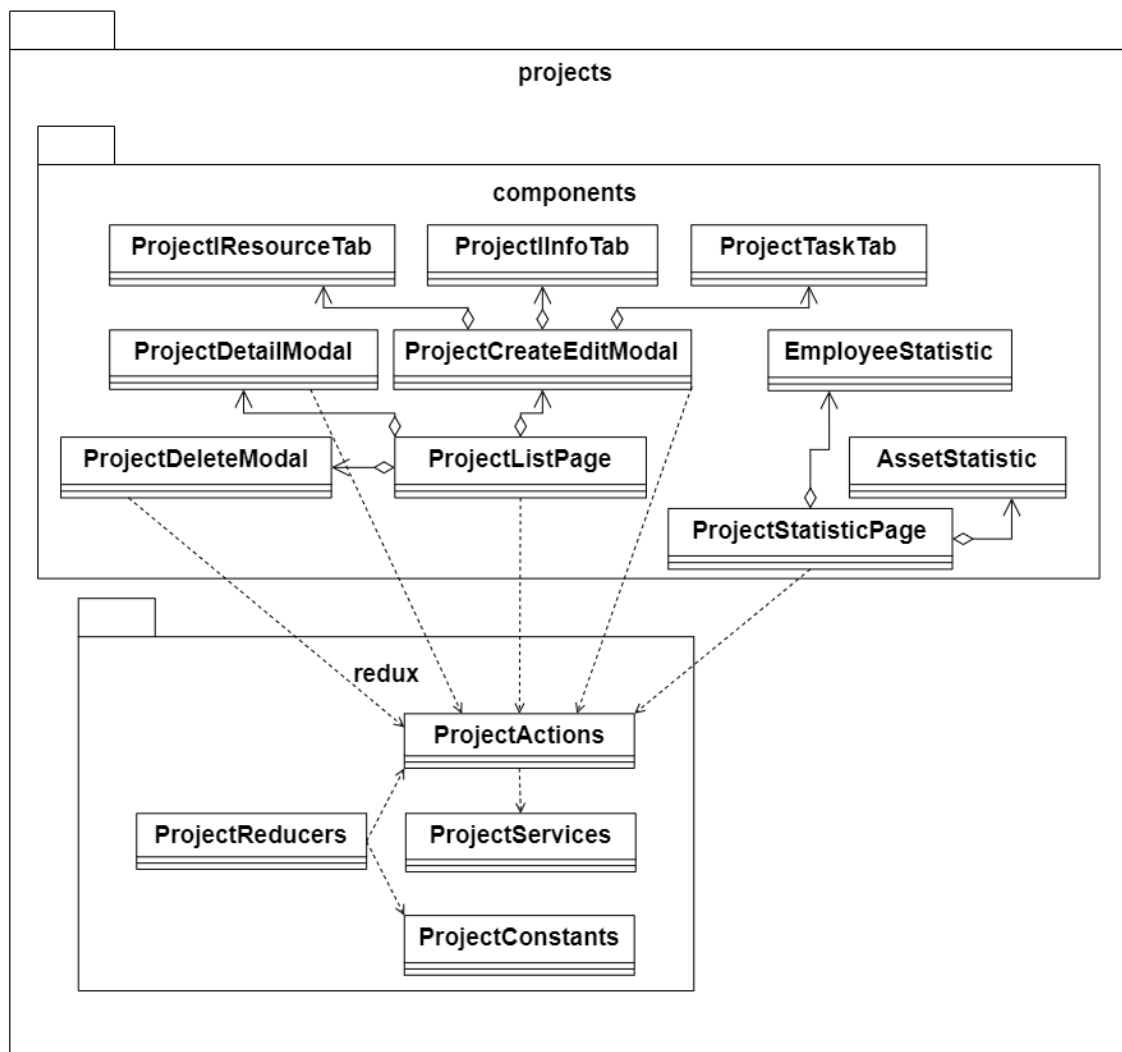
Hình 4.4 mô tả thiết kế chi tiết gói phía server. Mỗi gói thuộc phần modules sẽ bao gồm các lớp chính là Route, Controller, Service. Cụ thể, gói project-proposal sẽ có 3 lớp là: (i) ProjectProposalRoute định nghĩa các đường dẫn liên quan đến API

xử lý phân bổ nguồn lực dự án, (ii) ProjectProposalController tiếp nhận yêu cầu từ phía client thông qua ProjectProposalRoute và xử lý yêu cầu đó, (iii) ProjectProposalService sẽ xử lý logic nghiệp vụ của yêu cầu mà ProjectProposalController ủy thác. Các gói còn lại ở modules được thiết kế và giải thích tương tự. Các lớp Model thuộc gói models chịu trách nhiệm trích xuất, cập nhật thông tin từ cơ sở dữ liệu.



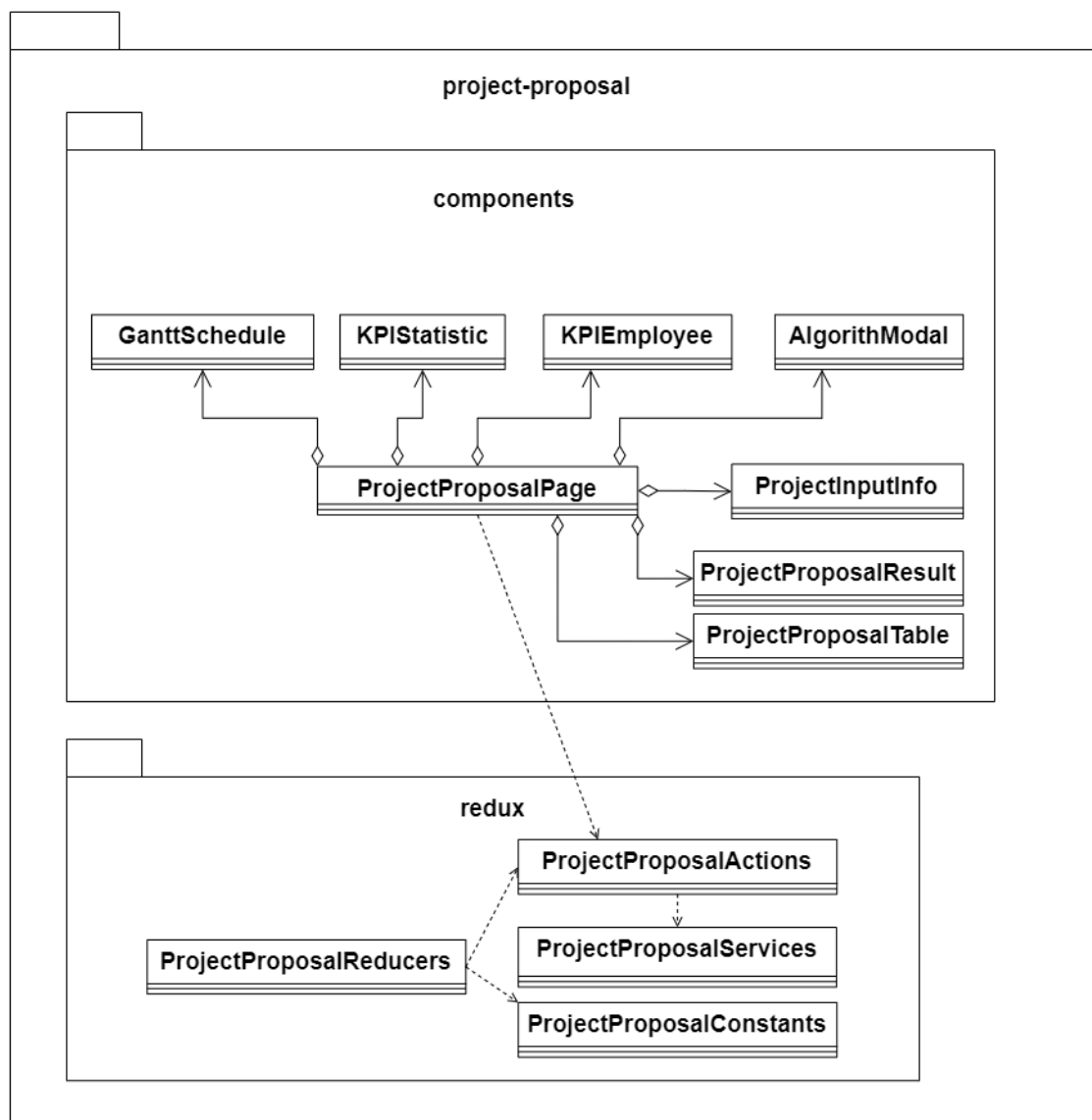
Hình 4.4: Thiết kế chi tiết gói phía server

Hình 4.5 mô tả về thiết kế chi tiết gói cho các chức năng liên quan đến quản lý danh sách dự án. Gói này gồm 2 gói con là (i) redux - chứa các thành phần phục vụ lưu trữ, cập nhật trạng thái của dữ liệu liên quan đến các dự án, (ii) components - chứa các thành phần giao diện hỗ trợ xử lý, hiển thị dữ liệu liên quan đến nhóm chức năng quản lý danh sách dự án và thu thập dữ liệu từ thao tác của người dùng để gửi yêu cầu đến server.



Hình 4.5: Thiết kế chi tiết gói phía client cho nhóm chức năng quản lý danh sách dự án

Hình 4.6 mô tả về thiết kế chi tiết gói cho các chức năng liên quan đến phân bổ nguồn lực dự án. Gói này gồm 2 gói con là (i) redux - chứa các thành phần phục vụ lưu trữ, cập nhật trạng thái của dữ liệu liên quan đến nhóm chức phân bổ nguồn lực dự án, (ii) components - chứa các thành phần giao diện hỗ trợ xử lý, hiển thị dữ liệu liên quan đến nhóm chức năng phân bổ nguồn dự án và thu thập dữ liệu từ thao tác của người dùng để gửi yêu cầu đến server.



Hình 4.6: Thiết kế chi tiết gói phía client cho nhóm chức năng phân bổ nguồn lực

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Ứng dụng cho phép hiển thị tương thích trên các loại màn hình khác nhau như máy tính cá nhân, laptop, máy tính bảng.

Bố cục của giao diện cần nhất quán trên toàn bộ ứng dụng. Các nút điều hướng, các dòng thông báo nhanh thông thường sẽ ở phía trên và bên phải của bố cục trang. Với các biểu mẫu, các nút sẽ ở phía dưới, bên phải. Hơn nữa, các trang cũng cần phải thống nhất theo bố cục sẵn có của ứng dụng DXClan, gồm các phần chính là: (i) header, (ii) sidebar, (iii) nội dung và (iv) footer. Cụ thể được biểu diễn và minh họa như **Hình 4.7**.

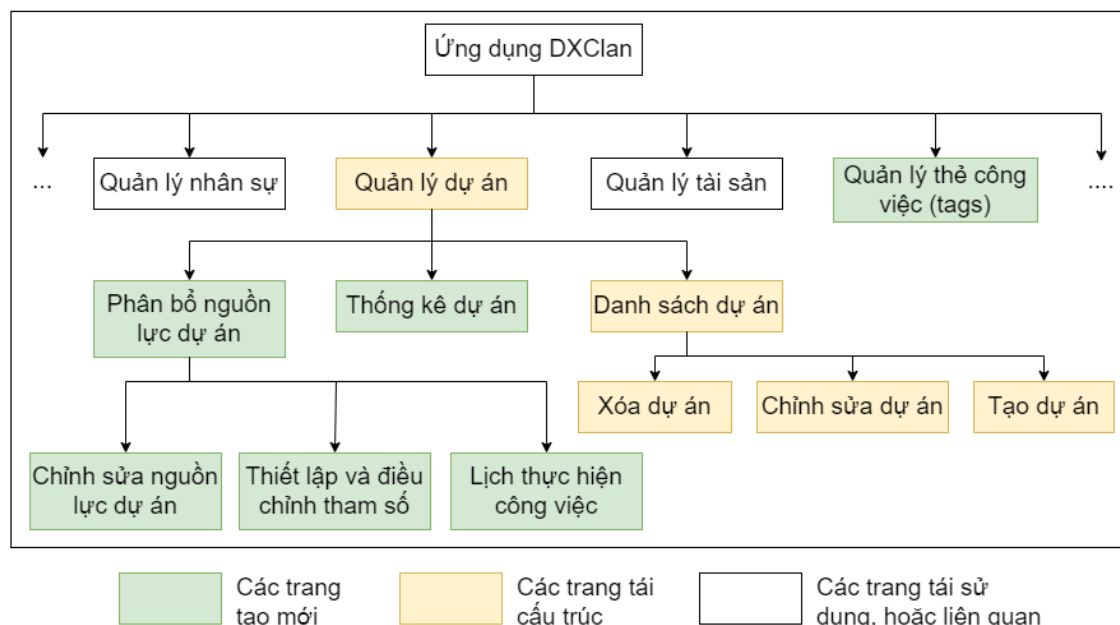
Quy định màu sắc giữa các nút phải thống nhất với nhau, cụ thể với nút xóa, hủy sẽ có màu đỏ, các nút liên quan đến thêm, lưu sẽ có màu xanh lá. Các nội dung

hiển thị thông tin thì có màu đen, xanh da trời, các nội dung hiển thị lỗi sẽ có màu đỏ, các nội dung hiển thị cảnh báo sẽ có màu vàng, các nội dung hiển thị thông báo thành công sẽ có màu xanh lá cây.



Hình 4.7: Cấu trúc giao diện phía DXClan

Hình 4.8 mô tả sitemap tổng quan của ứng dụng sau khi tích hợp các chức năng mà DATN xây dựng. Theo đó, các thành phần giao diện DATN xây dựng sẽ có màu xanh lá, các thành phần giao diện DATN tái cấu trúc, kế thừa sẽ có màu da cam, các thành phần liên quan còn lại có màu trắng.



Hình 4.8: Sitemap giao diện của ứng dụng

Tiếp theo, em xin đưa ra một số thiết kế về giao diện của ứng dụng. **Hình 4.9** là mock-up thiết kế giao diện cho màn danh sách dự án.

Tên ứng dụng

Header

UserRole

User

Danh sách dự án

Breadcrumbs

Lọc

Thêm dự án mới

Tên dự án

Thành viên

Người quản trị

Thời điểm bắt đầu

Thời điểm kết thúc

Tìm kiếm

Tên dự án	Ngày bắt đầu	Ngày kết thúc	Người quản trị	Thành viên	Tài sản	Trạng thái	Link MH phân bổ	Hành động
xxxx	yyyy/mm/dd	yyyy/mm/dd	xx, yy, zz	xx, yy, zz	xx, yy, zz	Đã có KQ phân bổ	Link	○ □ ▽
xxxx	yyyy/mm/dd	yyyy/mm/dd	xx, yy, zz	xx, yy, zz	xx, yy, zz	Chưa có KQ phân bổ	Link	○ □ ▽
xxxx	yyyy/mm/dd	yyyy/mm/dd	xx, yy, zz	xx, yy, zz	xx, yy, zz	Đã có KQ phân bổ	Link	○ □ ▽
xxxx	yyyy/mm/dd	yyyy/mm/dd	xx, yy, zz	xx, yy, zz	xx, yy, zz	Đã có KQ phân bổ	Link	○ □ ▽

Footer

Hình 4.9: Thiết kế giao diện màn danh sách dự án

Hình 4.10 là mock-up thiết kế giao diện cho màn tạo dự án mới.

Thêm mới dự án

Thông tin chung

Tài sản

Công việc

Thông tin chung

Tên dự án (*)

Ngày bắt đầu (*)

Ngày kết thúc dự kiến (*)

Đơn vị thời gian

Đơn vị chi phí

Mô tả dự án

Chỉ tiêu KPI

Chọn KPI Đơn vị

KPI PTPM

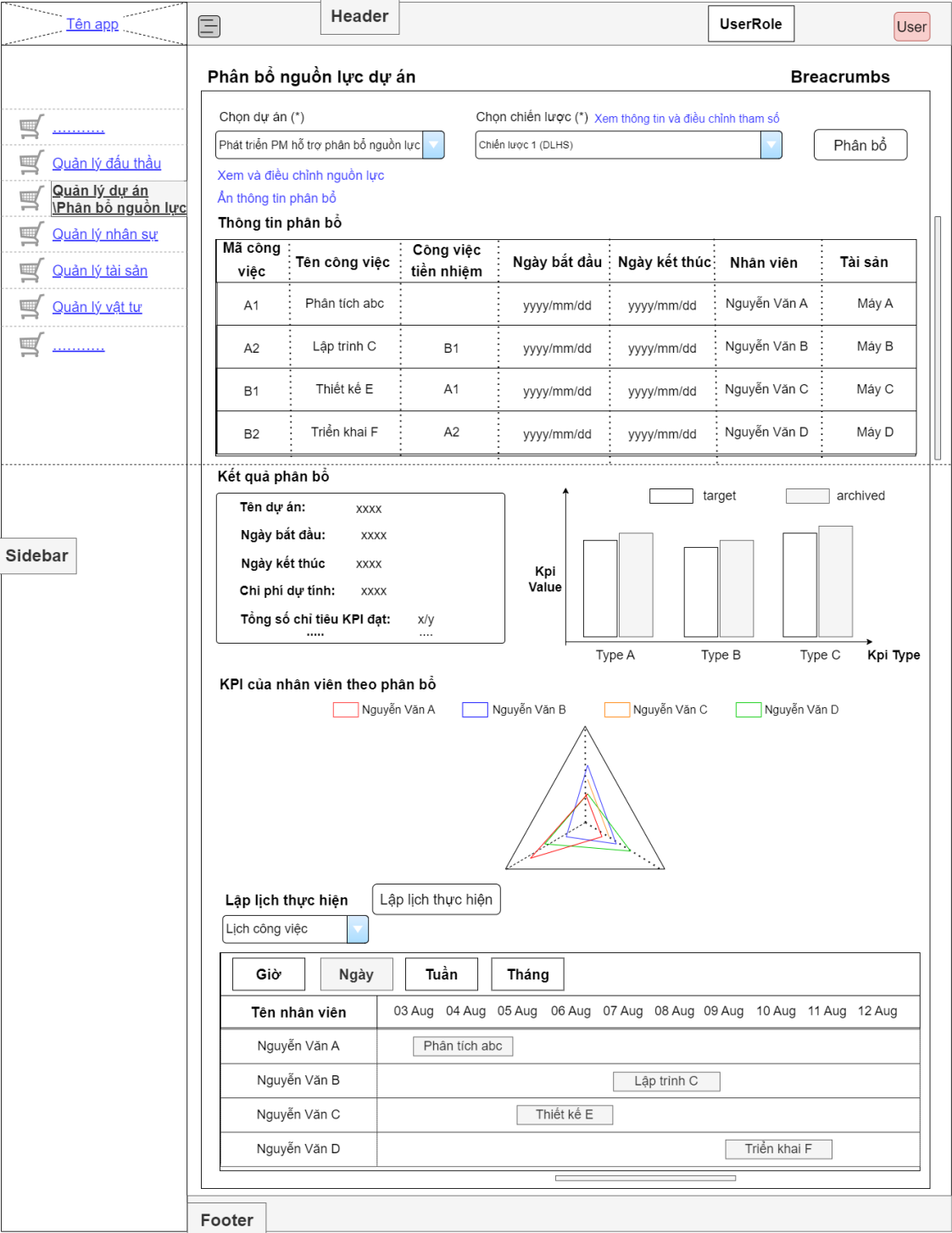
Loại KPI Mục tiêu	Tổng giao	Cần đạt
A	100	88
B	100	88
C	100	88

(*) chỉ các trường bắt buộc

Lưu Hủy

Hình 4.10: Thiết kế giao diện màn tạo mới dự án

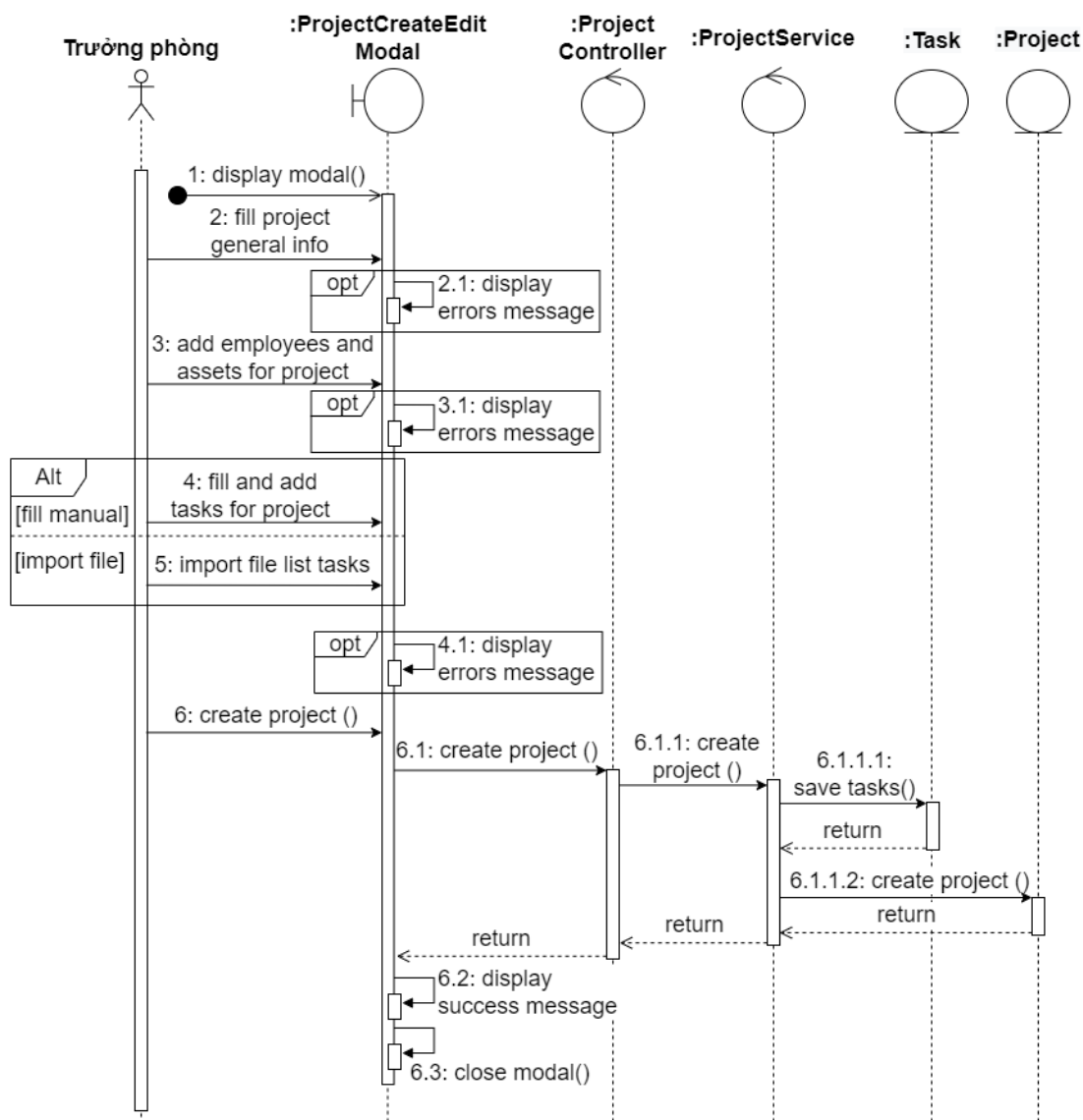
Hình 4.11 là mock-up thiết kế giao diện cho màn phân bổ nguồn lực dự án.



4.2.2 Thiết kế lớp

Trong phần này, DATN sẽ trình bày thiết kế chi tiết một số lớp chủ đạo. Trước tiên là biểu đồ trình tự của một số usecase quan trọng.

Hình 4.12 mô tả biểu đồ trình tự của usecase tạo dự án. Trên cơ sở luồng truyền thông điệp giữa các lớp của biểu đồ trình tự này, dưới đây là thiết kế chi tiết của các thành phần con của lớp ProjectCreateEditModal.



Hình 4.12: Biểu đồ trình tự cho usecase tạo dự án

Hình 4.13 mô tả thiết kế chi tiết các lớp ProjectInfoTab, ProjectResourceTab, ProjectTaskTab. Cụ thể, chi tiết về ý nghĩa các thuộc tính và phương thức sẽ được giải thích và trình bày ngay phía sau. Lớp ProjectInfoTab mô tả thành phần chứa thông tin chung về dự án cần tạo mới. Các thuộc tính, phương thức của lớp được mô tả dưới **Bảng 4.1**.

ProjectInfoTab	ProjectResourceTab
+ projectName: String + description: String + willStartTime: Date + willEndTime: Date + unitOfCost: String + unitOfTime: String + kpiSet: String + kpiTarget: Object<String, Number> [] + handleChangeGeneralForm(Event, String): void + handleChangeProjectStartTime(String): void + handleChangeProjectEndTime(String): void + handleChangeKPIUnit(String): void + setGeneralProjectInfo(Object): void + handleChangeKPITargetForm(Event, String, Number): void	+ projectMembers: Object + projectAssets: Object + handleChangeProjectManagers(Event, String): void + setProjectMembers(Object): void + setProjectAssets(Object): void + handleDeleteAssetRow(Number): void + handleAddAssetRow(Object []): void + handleAddEmployeesRow(Object []): void + handleDeleteEmployeesRow(String): void
	ProjectTaskTab
	+ projectTasks: Object + currentTask: Object + handleChangeCurrentTaskForm(Event, String): void + setProjectTasks(Object): void + handleAddTask(Object): void + handleDeleteTask(Number): void + setCurrentTask(Object): void

Hình 4.13: Thiết kế chi tiết lớp ProjectInfoTab, ProjectResourceTab, ProjectTaskTab

Thuộc tính	Ý nghĩa
projectName	tên dự án
description	mô tả dự án
willStartTime	thời gian bắt đầu dự kiến
willEndTime	thời gian kết thúc dự kiến
unitOfCost	đơn vị tiền tệ
unitOfTime	đơn vị thời gian
kpiSet	tập KPI mục tiêu
kpiTarget	danh sách KPI mục tiêu
Phương thức	Ý nghĩa
handleChangeProjectStartTime()	xử lý thay đổi thời gian bắt đầu dự kiến
handleChangeProjectEndTime()	xử lý thay đổi thời gian kết thúc dự kiến
handleChangeKPIUnit()	xử lý thay đổi tập KPI mục tiêu
handleChangeGeneralForm()	xử lý thay đổi các thông tin chung
handleChangeKPITargetForm()	xử lý thay đổi giá trị KPI mục tiêu
setGeneralProjectInfo()	thiết lập thông tin chung của dự án

Bảng 4.1: Thuộc tính và phương thức của lớp ProjectInfoTab

Lớp ProjectResourceTab mô tả thành phần chứa thông tin về nguồn lực gồm nhân lực và thiết bị sẽ tham gia dự án cần tạo mới. Các thuộc tính, phương thức của lớp được mô tả trong **Bảng 4.2**.

Thuộc tính	Ý nghĩa
projectMembers	người quản lý và thành viên tham gia dự án
projectAssets	danh sách tài sản tham gia dự án
Phương thức	Ý nghĩa
handleChangeProjectManagers()	xử lý thay đổi về người quản lý dự án
handleAddAssetRow()	thêm các tài sản cho dự án theo nhóm
handleDeleteAssetRow()	xóa các tài sản đã chọn cho dự án theo nhóm
handleDeleteEmployeesRow()	xóa danh sách nhân viên đã chọn cho dự án theo phòng ban
handleAddEmployeesRow()	thêm danh sách nhân viên đã chọn cho dự án theo phòng ban
setProjectMembers()	thiết lập danh sách nhân viên tham gia dự án
setProjectAssets()	thiết lập danh sách tài sản tham gia dự án

Bảng 4.2: Thuộc tính và phương thức của lớp ProjectResourceTab

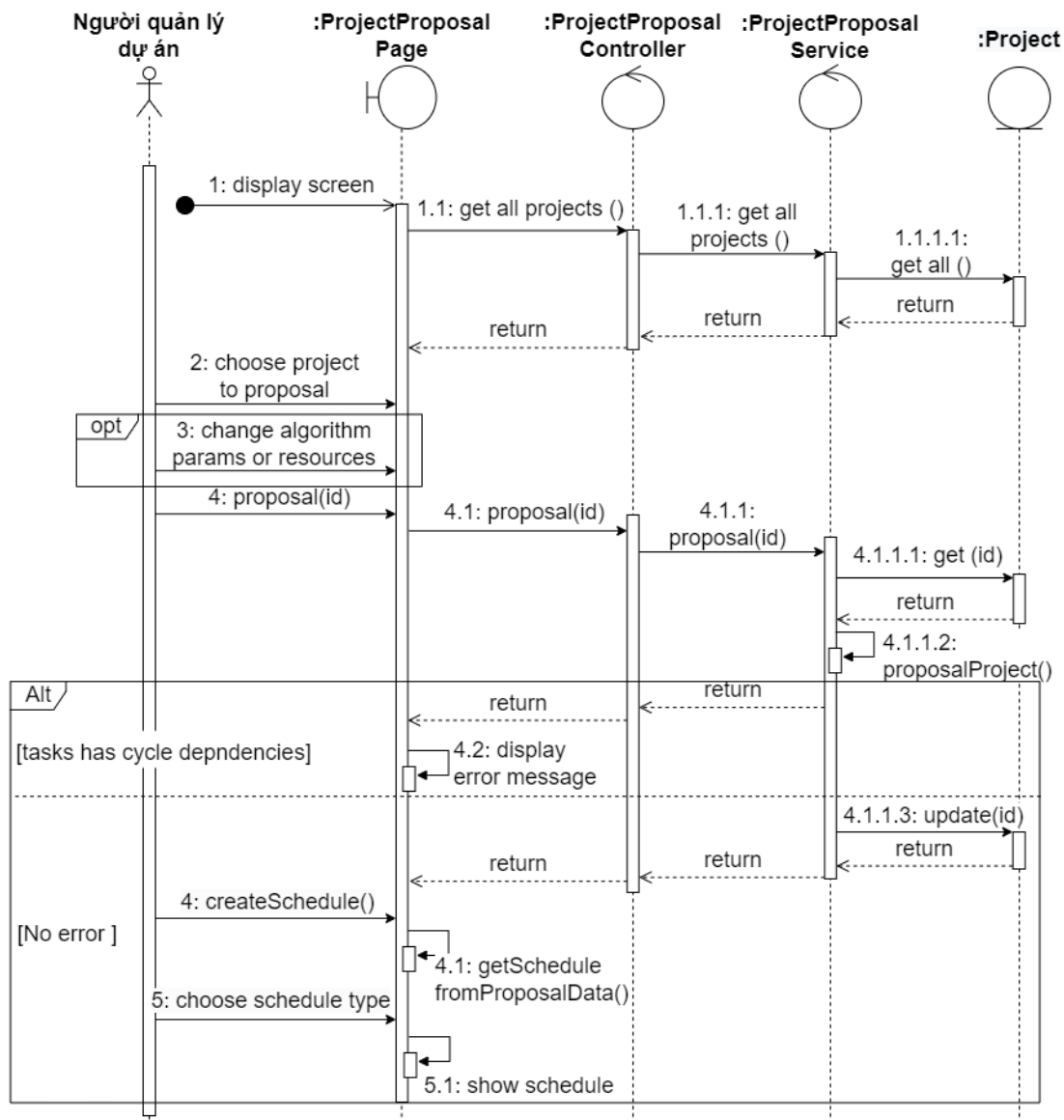
Lớp ProjectTaskTab mô tả thành phần chứa thông tin về danh sách các công việc trong dự án, cho phép thực hiện thêm các công việc vào dự án. Các thuộc tính, phương thức của lớp được mô tả trong **Bảng 4.3**.

Thuộc tính	Ý nghĩa
projectTasks	danh sách các công việc của dự án
currentTask	thông tin công việc hiện tại đang xét để thêm hoặc sửa
Phương thức	Ý nghĩa
handleChangeCurrentTaskForm()	xử lý thay đổi thông tin công việc hiện đang xét
handleAddTask()	thêm công việc cho dự án
handleDeleteTask()	xóa công việc của dự án
setCurrentTask()	thiết đặt công việc hiện đang xét dựa trên thay đổi của người dùng
setProjectTasks()	thiết đặt danh sách công việc cho dự án

Bảng 4.3: Thuộc tính và phương thức của lớp ProjectTaskTab

Hình 4.14 mô tả biểu đồ trình tự cho các usecase phân bổ nguồn lực và lập lịch thực hiện các công việc cho dự án. Trên cơ sở luồng truyền thông điệp giữa các lớp của biểu đồ trình tự này, **Hình 4.15** trình bày thiết kế chi tiết của lớp

ProjectProposalPage. Các thuộc tính và phương thức của lớp được mô tả trong **Bảng 4.4**.



Hình 4.14: Biểu đồ trình tự cho usecase phân bổ nguồn lực, lập lịch thực hiện dự án

Thuộc tính	Ý nghĩa
currentProject	dữ liệu về dự án hiện xét
scheduleType	loại lịch thực hiện công việc muốn hiển thị
isShowResourceInput	tùy chọn hiển thị dữ liệu nguồn lực dự án
isShowProposal	tùy chọn hiển thị dữ liệu phân bổ
algorithm	chiến lược phân bổ
algorithmParams	các tham số cho chiến lược phân bổ
Phương thức	Ý nghĩa
setCurrentProject()	thiết lập thông tin dự án hiện xét

setScheduleType()	thiết lập loại lịch trình thực hiện dự án theo kết quả phân bổ
setIsShowResourceInput()	thiết lập ẩn hay hiện thông tin nguồn lực
setIsShowProposal()	thiết lập ẩn hay hiện thông tin phân bổ
handleProposalProject()	phân bổ nguồn lực cho dự án đã chọn
setAlgorithm()	thiết lập chiến lược phân bổ
setAlgorithmParams()	thiết lập tham số cho thuật toán tương ứng chiến lược phân bổ
openAlgorithmModal()	mở biểu mẫu hiện thông tin chiến lược phân bổ và cài đặt các tham số
updateProject()	cập nhật lại thông tin nguồn lực dự án điều chỉnh
getListProject()	lấy danh sách dự án
useEffect()	phương thức thực hiện các hàm, hành động mỗi khi trang được gắn vào DOM ảo

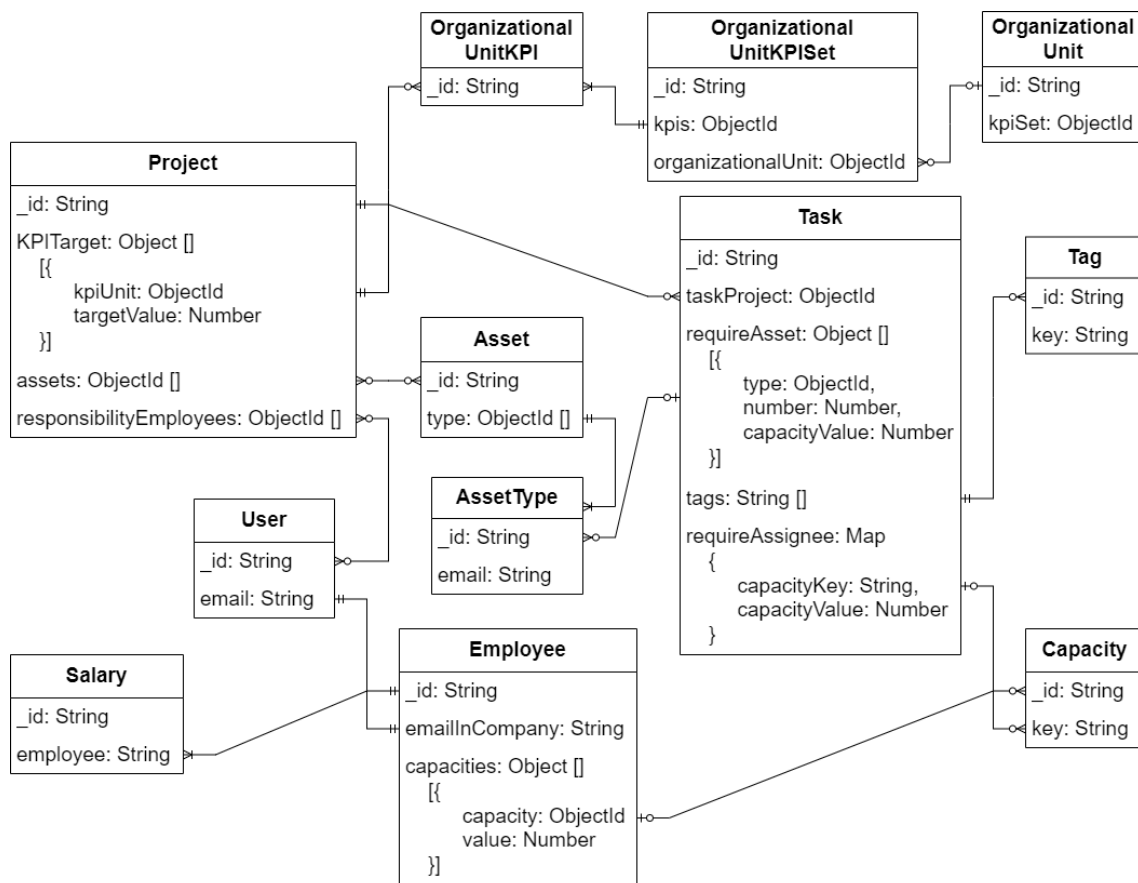
Bảng 4.4: Thuộc tính và phương thức của lớp ProjectProposalPage

ProjectProposalPage
+ currentProject: Object + scheduleType: String + isShowResourceInput: String + isShowProposal: String + algorithm: String + algorithmParams: Object
+ setCurrentProject(Event, Object): void + setScheduleType(String): void + setIsShowProposal(): void + setIsShowResourceInput(): void + handleProposalForProject(Object): void + setAlgorithm(String): void + setAlgorithmParams(Object): void + openAlgorithmModal(String): void + updateProject(Object): void + getListProjects(): Object + useEffect()

Hình 4.15: Thiết kế chi tiết cho lớp ProjectProposalService

4.2.3 Thiết kế cơ sở dữ liệu

Ứng dụng sử dụng hệ quản trị cơ sở dữ liệu MongoDB theo như hệ thống tích hợp DXClan.



Hình 4.16: Biểu đồ thực thể liên kết

Hình 4.16 mô tả sơ đồ thực thể liên kết giữa các thực thể trong ứng dụng. Ở đây, chỉ liệt kê các thuộc tính có quan hệ với thực thể khác. Cụ thể, các thực thể đã có sẵn trong cơ sở dữ liệu của hệ thống DXClan mà em tái sử dụng bao gồm: (i) OrganizationUnitKPI, (ii) OrganizationalUnitKPISet, (iii) OrganizationalUnit, (iv) AssetType, (v) User và (vi) Salary. Các thực thể mà em thực hiện xây dựng, sửa đổi, tái cấu trúc cho ĐATN bao gồm: (i) Project, (ii) Asset, (iii) Task, (iv) Employee, (v) Capacity và (vi) Tag. **Bảng 4.5** mô tả ý nghĩa của các thực thể trên.

Tên thực thể	Ý nghĩa	Tên thực thể	Ý nghĩa
Project	Dự án	Asset	Tài sản
User	Người dùng	AssetType	Loại tài sản
Employee	Nhân viên	OrganiaztionalUnitKPI	KPI đơn vị
Task	Công việc	OrganiaztionalUnitKPISet	Tập KPI đơn vị
Tag	Thẻ công việc	OrganiaztionalUnit	Đơn vị phòng ban
Capacity	Năng lực	Salary	Lương nhân viên

Bảng 4.5: Bảng liệt kê các thực thể và ý nghĩa

Thiết kế chi tiết cho thực thể Project trong ĐATN được trình bày ở **Bảng 4.6**.

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectId	Id của dự án
name	String	tên dự án
description	String	mô tả dự án
startDate	Date	ngày bắt đầu dự án
endDate	Date	ngày kết thúc dự án
unitTime	enum	đơn vị tiền tệ
unitCost	enum	đơn vị thời gian
status	String	trạng thái
projectManager	ObjectId[]	danh sách người quản lý
responsibleEmployees	ObjectId[]	danh sách thành viên tham gia
assets	ObjectId[]	danh sách tài sản
kpiTarget	kpiTargetType[]	danh sách KPI mục tiêu
kpiTargetType.type	ObjectId	KPI mục tiêu
kpiTargetType.value	Number	giá trị mục tiêu
tasks	ObjectId[]	danh sách công việc
proposals	Object	dữ liệu phân bổ nguồn lực
proposalLogs	Object	lịch sử phân bổ

Bảng 4.6: Thiết kế chi tiết cho thực thể Project

Thiết kế chi tiết cho thực thể Capacity trong DATN được trình bày ở **Bảng 4.7**.

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectId	Id của năng lực
name	String	tên năng lực
key	String	mã của năng lực
description	String	mô tả năng lực
values	valueType[]	các giá trị của năng lực
valueType.key	String	tên của giá trị năng lực
valueType.value	Number	giá trị quy đổi của giá trị năng lực

Bảng 4.7: Thiết kế chi tiết cho thực thể Capacity

Bảng 4.8, 4.9 lần lượt trình bày thiết kế chi tiết cho thực thể Task, Tag trong DATN. **Bảng 4.10, 4.11** lần lượt trình bày về thuộc tính, dữ liệu của thực thể Employee, Asset mà DATN kế thừa, tái cấu trúc. **Bảng 4.12** trình bày về thuộc

tính, dữ liệu của thực thể OrganizationalUnitKPI mà ĐATN tái sử dụng.

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectId	Id của công việc
name	String	tên công việc
description	String	mô tả công việc
startDate	Date	ngày bắt đầu công việc
endDate	Date	ngày kết thúc công việc
code	String	mã công việc
tags	String []	thẻ công việc
status	enum	trạng thái công việc
point	Number	điểm đánh giá công việc
requireAssignee	Map(key, value)	yêu cầu năng lực nhân sự
key	String	loại năng lực yêu cầu
value	Number	giá trị năng lực yêu cầu
requireAsset	requireAssetType[]	yêu cầu về tài sản
requireAssetType.type	ObjectId	loại tài sản yêu cầu
requireAssetType.number	Number	số lượng tài sản yêu cầu
requireAssetType.capacityValue	Number	giá trị của tài sản yêu cầu
requireAssetType.requireType	Number	loại yêu cầu
kpiInTask	ObjectId	KPI liên quan đến công việc
taskKPIWeight	Number	trọng số của công việc
assignee	ObjectId	người thực hiện
assets	ObjectId[]	tài sản thực hiện

Bảng 4.8: Thiết kế chi tiết cho thực thể Task

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectId	Id của thẻ công việc
name	String	tên thẻ công việc
description	String	mô tả về thẻ công việc
key	String	mã của thẻ công việc

Bảng 4.9: Thiết kế chi tiết cho thực thể Tag

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectId	Id của nhân viên
fullName	String	tên của nhân viên
employeeNumber	String	mã nhân viên
emailInCompany	String	email công ty của nhân viên
capacities	capacityType[]	mảng các giá trị năng lực
capacityType.capacity	ObjectId	Id của năng lực
capacityType.value	Number	giá trị năng lực tương ứng

Bảng 4.10: Thông tin mô tả thực thể Employee

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectId	Id của tài sản
group	Enum	nhóm tài sản
code	String	mã tài sản
assetName	String	tên tài sản
serial	String	số tài sản
assetType	ObjectId[]	loại tài sản
status	Enum	trạng thái tài sản
usageLogs	usageLogType[]	ghi chú thời gian sử dụng
usageLogType.startDate	Date	thời gian bắt đầu sử dụng
usageLogType.endDate	Date	thời gian kết thúc sử dụng
costPerHour	Number	chi phí vận hành theo giờ
capacityValue	Number	khả năng sử dụng

Bảng 4.11: Thông tin mô tả thực thể Asset

Tên trường	Kiểu dữ liệu	Mô tả
_id	ObjectId	Id của KPI
name	String	tên KPI
target	Number	giá trị mục tiêu
weight	Number	trọng số của KPI
criteria	String	tiêu chí đánh giá
unit	String	đơn vị đo chỉ tiêu KPI

Bảng 4.12: Thông tin mô tả thực thể OrganizationalUnitKPI

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Bảng 4.13 trình bày về danh sách thư viện, công cụ được sử dụng trong quá trình em thực hiện ĐATN.

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual Studio Code	https://code.visualstudio.com
Lập trình backend	NodeJS	https://nodejs.org/en
Lập trình backend	ExpressJS	https://expressjs.com
Lập trình frontend	ReactJS	https://react.dev
Quản lý trạng thái	Redux	https://redux.js.org
Hệ quản trị CSDL	MongoDB	https://www.mongodb.com
Thao tác với CSDL	mongoose	https://mongoosejs.com
Quản lý CSDL	MongoDB Compass	https://www.mongodb.com/docs
Kiểm tra API	Postman API Platform	https://www.postman.com

Bảng 4.13: Danh sách thư viện và công cụ sử dụng

4.3.2 Kết quả đạt được

Sau khi trải qua các bước nghiên cứu, phân tích, thiết kế, xây dựng, tích hợp, ĐATN đã thực hiện xây dựng phương pháp giải quyết cho bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI.

Trên cơ sở phương pháp giải quyết, ĐATN thực hiện xây dựng các chức năng chính sau: (i) thêm, sửa, xóa dự án với danh sách công việc, nguồn lực và chỉ tiêu KPI; (ii) phân bổ nguồn lực thực hiện công việc trong dự án; (iii) lập lịch thực hiện công việc trong dự án; (iv) xem lịch nhân công, lịch máy móc tương ứng dữ liệu phân bổ; (v) quản lý thẻ công việc; (vii) quản lý bộ năng lực của nhân viên. Các chức năng trên đã được tích hợp vào hệ thống DXClan, từ đó cung cấp ứng dụng hỗ trợ phân bổ nguồn lực thực hiện dự án theo chỉ tiêu KPI một cách tự động.

Bảng 4.14 trình bày thống kê về số file xây dựng, số file tái sử dụng, dung lượng mã nguồn và số thực thể sử dụng cho ĐATN.

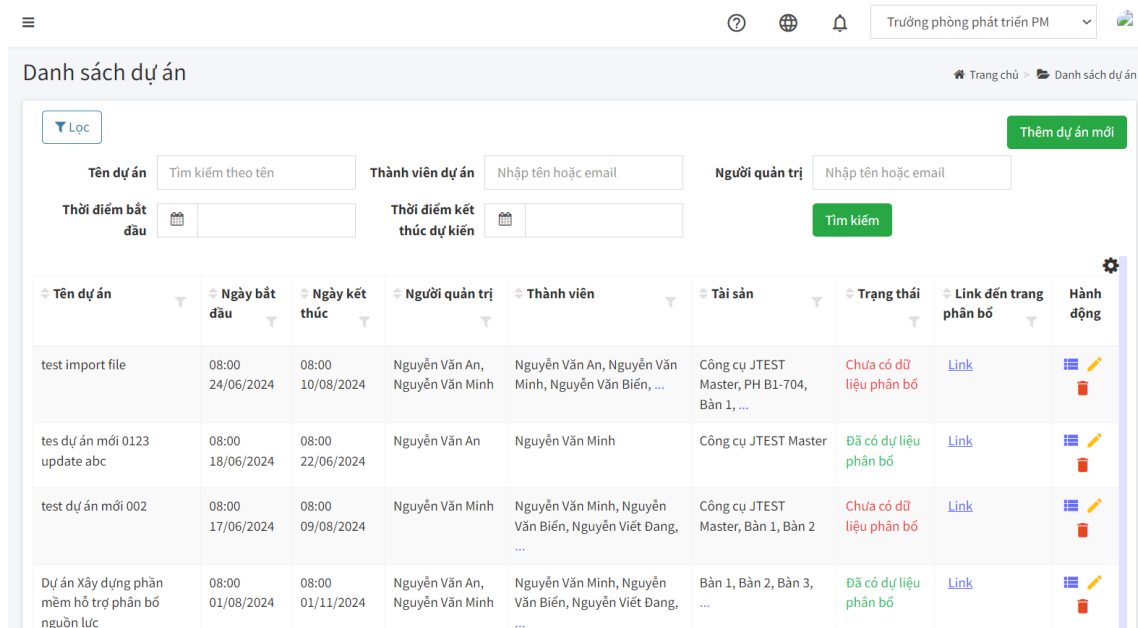
Thông tin thống kê	Kết quả thống kê
Số file(.js, .jsx) xây dựng phía frontend	28

Số file(.js, .jsx) tái sử dụng, tái cấu trúc phía frontend	6
Dung lượng phía frontend (cả thư viện)	1063.92 MB
Số file(.js) xây dựng phía backend	10
Số file(.js) tái sử dụng, tái cấu trúc phía backend	8
Dung lượng phía backend (cả thư viện)	312.02 MB
Dung lượng toàn bộ mã nguồn	1375.95 MB
Số collection sử dụng trong CSDL	12
Số collection tái sử dụng, sửa đổi, xây dựng	6

Bảng 4.14: Thống kê về ứng dụng

4.3.3 Minh họa các chức năng chính

a, Các chức năng liên quan đến quản lý danh sách dự án



Hình 4.17: Màn hình danh sách dự án

Hình 4.17 là màn hình danh sách dự án. Màn hình này đưa ra các dự án kèm theo thông tin về: (i) tên dự án, (ii) ngày bắt đầu, (iii) ngày kết thúc, (iv) người quản trị, (v) thành viên, (vi) trạng thái đã có dữ liệu phân bổ hay chưa, (vii) đường dẫn đến màn hình phân bổ. Ở đây, người dùng có thể tìm kiếm và lọc thông tin các dự án theo: (i) tên dự án, (ii) tên hoặc email của thành viên dự án, (iii) tên hoặc email của người quản trị dự án, (iv) thời điểm bắt đầu, (v) thời điểm kết thúc.

Để thêm một dự án mới, ta thực hiện ấn vào nút "Thêm dự án mới" trên màn hình danh sách. Sau đó, biểu mẫu thêm dự án sẽ hiện ra, người dùng cần điền các thông tin về dự án sẽ thêm. **Hình 4.18** hiển thị thông tin Tab Thông tin chung của

màn hình thêm dự án, ở đây, người dùng sẽ điền thông tin chung về dự án bao gồm: (i) tên dự án, (ii) thời điểm bắt đầu, kết thúc, (iii) mô tả dự án, (iv) các KPI mục tiêu của dự án. Các trường thông tin bắt buộc sẽ có dấu (*) màu đỏ, là những trường thông tin mà người dùng phải thực hiện điền đầy đủ thì mới có thể hoàn thành chức năng. Trong trường hợp có lỗi xảy ra trong quá trình nhập, điền biểu mẫu, hệ thống cũng sẽ thông báo ra màn hình cho người dùng.

Thêm mới dự án

Thông tin chung

Tài nguyên nhân lực và tài sản

Thêm công việc

Thông số dự án

Tên dự án*

Dự án

Giá trị phải có số ký tự từ 6 đến 255

Thời điểm bắt đầu (dự kiến)*

📅

30-06-2024

Đơn vị của thời gian

Ngày

Thời điểm kết thúc (dự kiến)*

📅

16-06-2024

Ngày kết thúc phải sau ngày bắt đầu

Đơn vị của thời gian

VND

Mô tả dự án

KPI mục tiêu

KPI đơn vị*

Phòng phát triển PM (66659c3f169e3547d0e0a881)

Giá trị KPI Mục tiêu*

KPI Mục tiêu	Tổng số giao	Chỉ tiêu cần đạt
Phản tích, xây dựng chức năng, module	100	991
Giá trị chỉ tiêu cần đạt phải nhỏ hơn bằng tổng số giao		
Chất lượng sản phẩm	0	0
Tài liệu và các yêu cầu khác	0	0

(*) : Các trường thông tin bắt buộc

Lưu

Đóng

Hình 4.18: Màn hình thêm dự án - Tab thông tin chung

Sau khi điền xong thông tin chung về dự án, người dùng tiếp tục thêm nhân lực, tài sản thực hiện dự án ở Tab Tài nguyên nhân lực và tài sản. **Hình 4.19** mô tả màn hình thêm tài nguyên nhân lực và tài sản cho dự án. Cụ thể, ở màn hình này, người dùng sẽ thêm (i) người quản trị dự án, (ii) thành viên tham gia dự án và (iii) tài sản tham gia dự án. Ở màn hình này, người dùng thực hiện các thao tác tìm kiếm và lựa chọn nguồn lực để thêm vào dự án.

Thêm mới dự án

Thông tin chung

Tài nguyên nhân lực và tài sản

Thêm công việc

Nhân lực

Người quản trị*

Nguyễn Văn Minh (Trưởng phòng phát triển PM, Nhân viên phòng phát triển PM)

Thành viên dự án*

Thuộc đơn vị

Phòng phát triển PM

Ban giám đốc

Thành viên tham gia

Nguyễn Viết Dang , Nguyễn Văn Minh

--- Chọn thành viên ---

Hành động

Thiết bị, tài sản

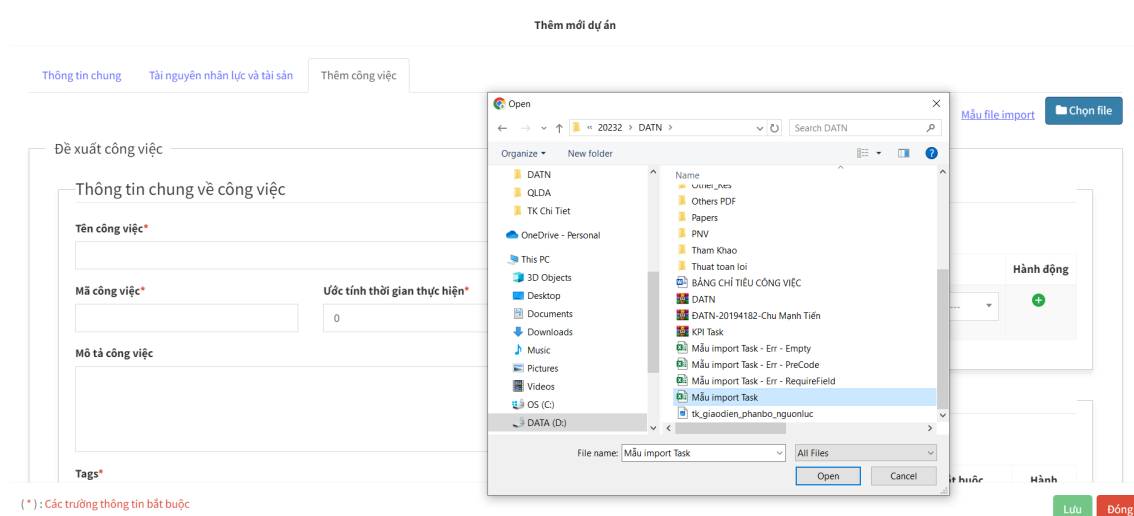
Lựa chọn tài sản*

Nhóm tài sản	Tài sản cho dự án	Hành động
Khác	Công cụ JTEST Master (SW.011)	
Mặt bằng	PH B1-704 (SW.007)	
Khác	--- Chọn tài sản --- <div><div>Khác</div><div>Công cụ JTEST Master (SW.011)</div><div>Bản 1 (SW.008)</div><div>Bản 2 (SW.009)</div><div>Bản 3 (SW.010)</div><div>Card GTX 2050Ti (VVMV18.0026)</div></div>	

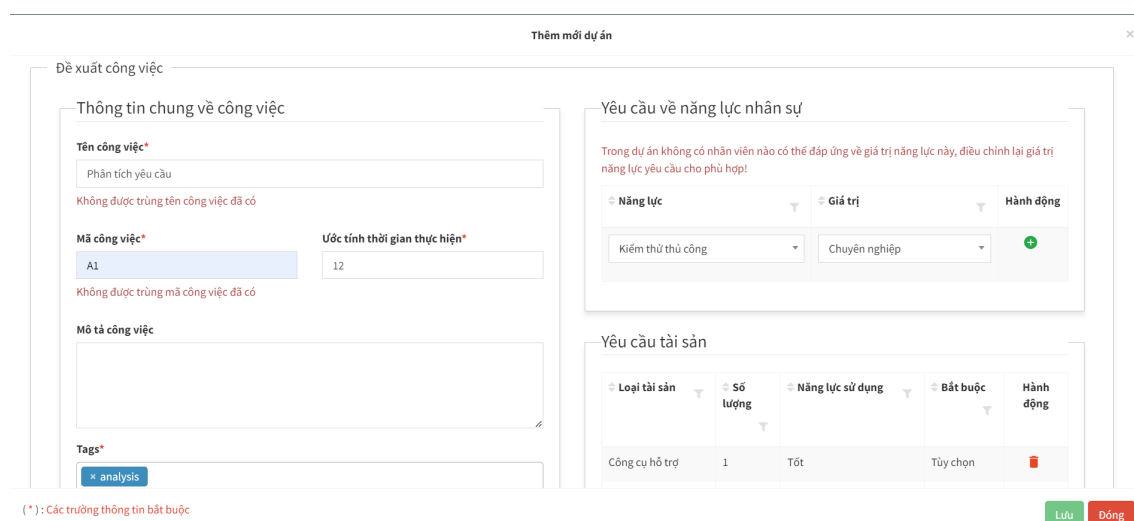
(*) : Các trường thông tin bắt buộc

Hình 4.19: Màn hình thêm dự án - Tab Tài nguyên nhân lực và tài sản

Bước tiếp theo, người dùng thêm công việc cho dự án ở Tab Thêm công việc. Ở đây, người dùng có thể chọn thêm các công việc bằng cách tải file lên, hoặc thêm lần lượt công việc bằng cách điền vào biểu mẫu. Trong trường hợp thêm công việc bằng cách tải file, người dùng cần tải file mẫu về, rồi nhập liệu và tải file lên hệ thống, minh họa ở **Hình 4.20**. Trong trường hợp thêm công việc thủ công, người dùng cần điền các thông tin liên quan đến công việc như: (i) tên công việc, (ii) mã công việc, (iii) mô tả, (iv) thể công việc, (v) yêu cầu về năng lực nhân sự, (vi) yêu cầu về năng lực tài sản để đáp ứng công việc, sau đó ấn nút "Thêm". Nếu có lỗi xảy ra, hệ thống sẽ hiển thị lỗi cho người dùng cho đến khi sửa hết lỗi mới có thể thực hiện thêm được công việc. Cụ thể cho bước thêm từng công việc được mô tả trong **Hình 4.21**.



Hình 4.20: Màn hình thêm dự án - Tab Thêm công việc



Hình 4.21: Màn hình thêm dự án - Tab Thêm công việc

Trong khi thêm các công việc, người dùng có thể xem danh sách công việc trong

dự án theo 2 dạng bảng và biểu đồ gantt.

Thêm mới dự án

Công việc tiên nhiệm

A1

KPI Liên quan*

Tài liệu và các yêu cầu khác

Trọng số*

0.123

Công cụ hỗ trợ

1

Bình thường

Tùy chọn

Chọn loại tài sản

0

Chọn năng lực sử dụng

Chọ...

Hủy

Lưu

Xóa trắng

Bảng

Biểu đồ Gantt

Mã công việc	Tên công việc	Công việc tiên nhiệm	Thời gian ước lượng	Tags công việc	KPI liên quan	Yêu cầu nhân lực	Yêu cầu tài sản	Hành động
A1	Phân tích yêu cầu khách hàng		3	planning	Phân tích, xây dựng chức năng, module	Tiếng Anh: B1, Xếp hạng tốt nghiệp: Khá	Không	
A2	Đề xuất ý tưởng mua hàng	A1	3	idea	Tài liệu và các yêu cầu khác	Xếp hạng tốt nghiệp: Khá	Loại: Công cụ hỗ trợ, Số lượng: 1, Khả năng sử dụng: Tốt, Yêu cầu: Bắt buộc	

(*) : Các trường thông tin bắt buộc

Lưu

Đóng

Hình 4.22: Màn hình thêm dự án - Tab Thêm công việc, danh sách công việc dạng bảng

Hình 4.22 mô tả danh sách công việc của dự án sẽ thêm ở dạng bảng, người dùng có thể chọn xóa, hoặc chỉnh sửa công việc thông qua các icon ở cột hành động trong bảng. Trong trường hợp sửa, thông tin công việc hiện xét sẽ hiển thị lại ở biểu mẫu, người dùng chỉnh sửa công việc rồi ấn nút "Lưu" để lưu thông tin chỉnh sửa, ấn "Hủy" để bỏ việc chỉnh sửa, ấn "Xóa trắng" để khởi tạo lại biểu mẫu tạo công việc. Người dùng cũng có thể xem danh sách công việc dưới dạng biểu đồ gantt để có cái nhìn trực quan hơn, minh họa ở **Hình 4.23**.

Thêm mới dự án

KPI Liên quan*

Chọn KPI liên quan

Trọng số*

0

Thêm

Xóa trắng

Bảng

Biểu đồ Gantt

Giờ	Ngày	Tuần	Tháng
# 26			
# 27			
Tên công việc	Thời lượng (Thứ)	29 Jun	30 Jun
Phân tích yêu cầu khách hàng	3		Now
Đề xuất ý tưởng mua hàng	3		

(*) : Các trường thông tin bắt buộc

Lưu

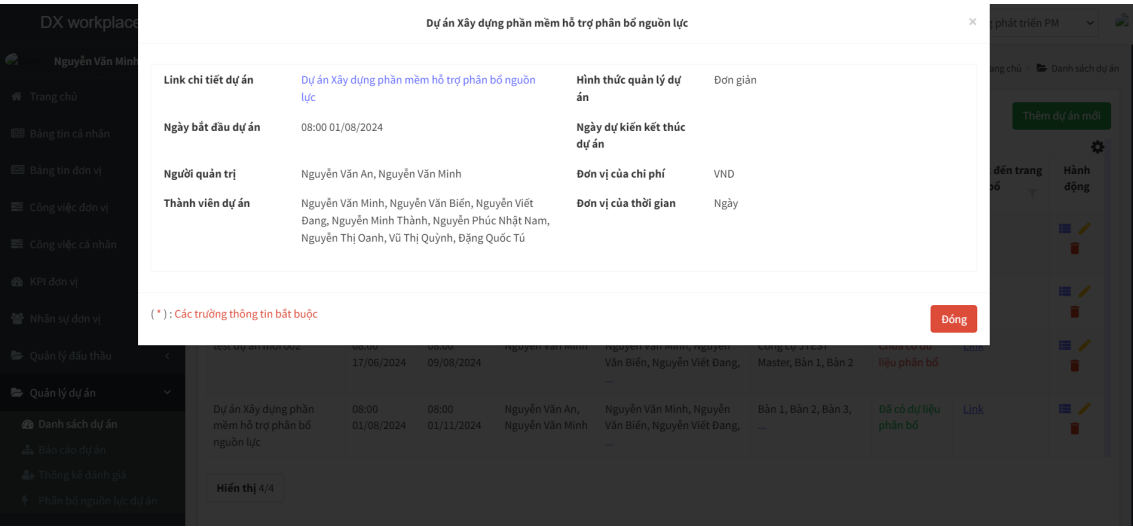
Đóng

Hình 4.23: Màn hình thêm dự án - Tab Thêm công việc, danh sách công việc dạng biểu đồ

Sau khi đã điền đầy đủ thông tin về dự án, người dùng ấn "Lưu", hệ thống sẽ xử lý yêu cầu thêm dự án, đóng biểu mẫu và hiển thị thông báo thành công ra màn hình, quay trở lại màn hình danh sách dự án cho người dùng.

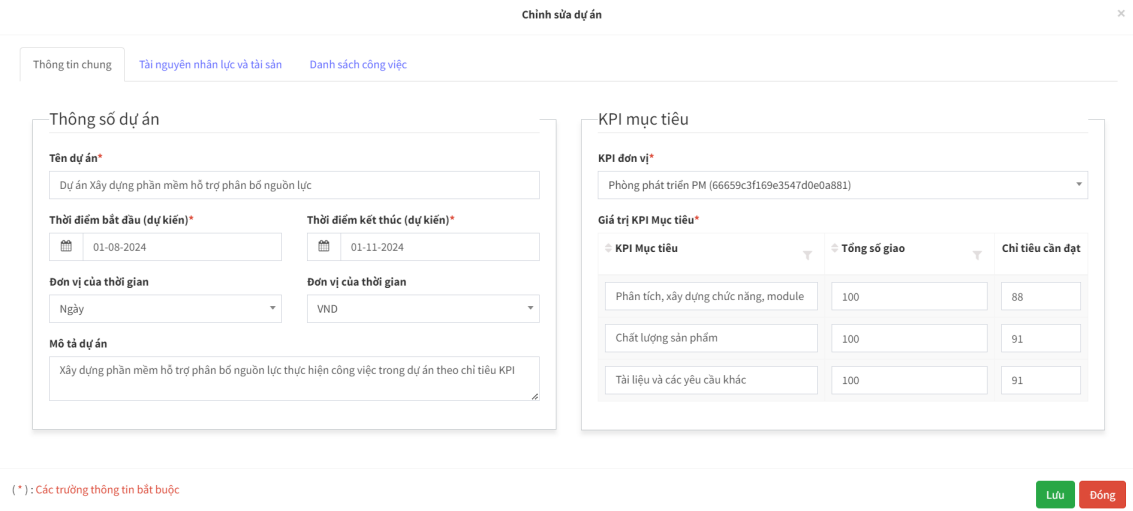
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Ở màn hình danh sách dự án, ta cũng có thể thực hiện xem thông tin và chỉnh sửa thông tin dự án cũng như xóa dự án nếu không cần thiết. Bằng cách bấm vào các icon hành động tương ứng.



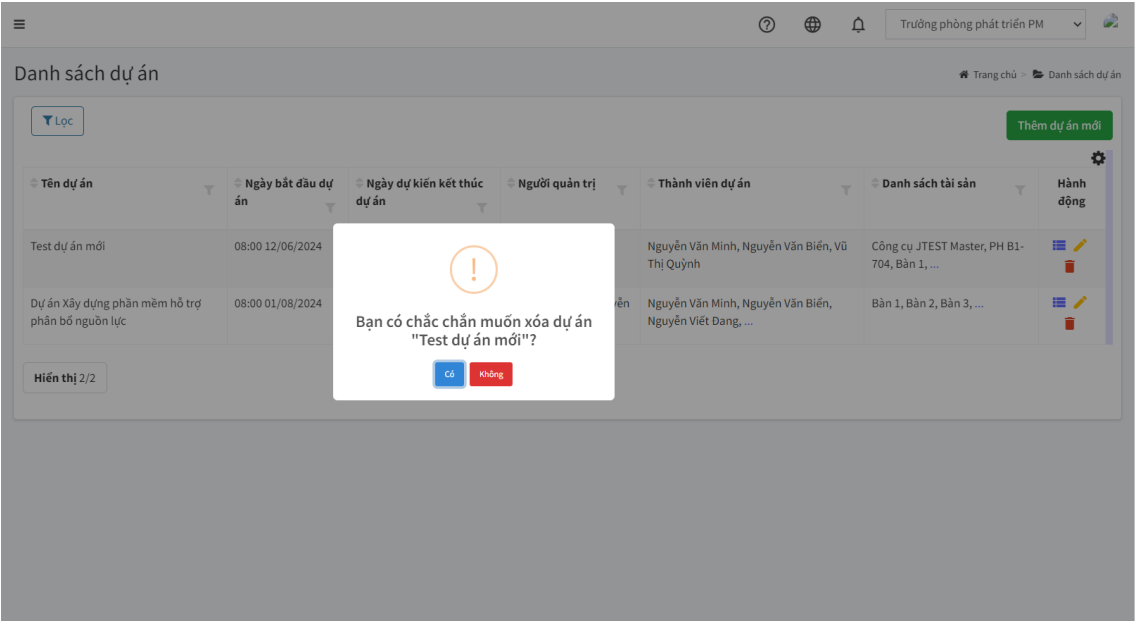
Hình 4.24: Màn hình thông tin dự án

Hình 4.24 là màn hình mô tả thông tin cơ bản của dự án khi ta ấn vào xem thông tin chi tiết của dự án ở phía trang danh sách.



Hình 4.25: Màn hình chỉnh sửa dự án

Hình 4.25 là màn hình sửa dự án sau khi chọn từ danh sách. Tương tự màn hình thêm dự án, màn hình này cũng có 3 Tab bao gồm: (i) thông tin chung, (ii) tài nguyên nhân lực và tài sản, (iii) danh sách công việc. Tuy nhiên thông tin dự án đã có và sẽ hiển thị cho người dùng. Người dùng muốn sửa thông tin của dự án tiến hành thực hiện các bước tương tự như mô tả phần tạo dự án mới ở bên trên.

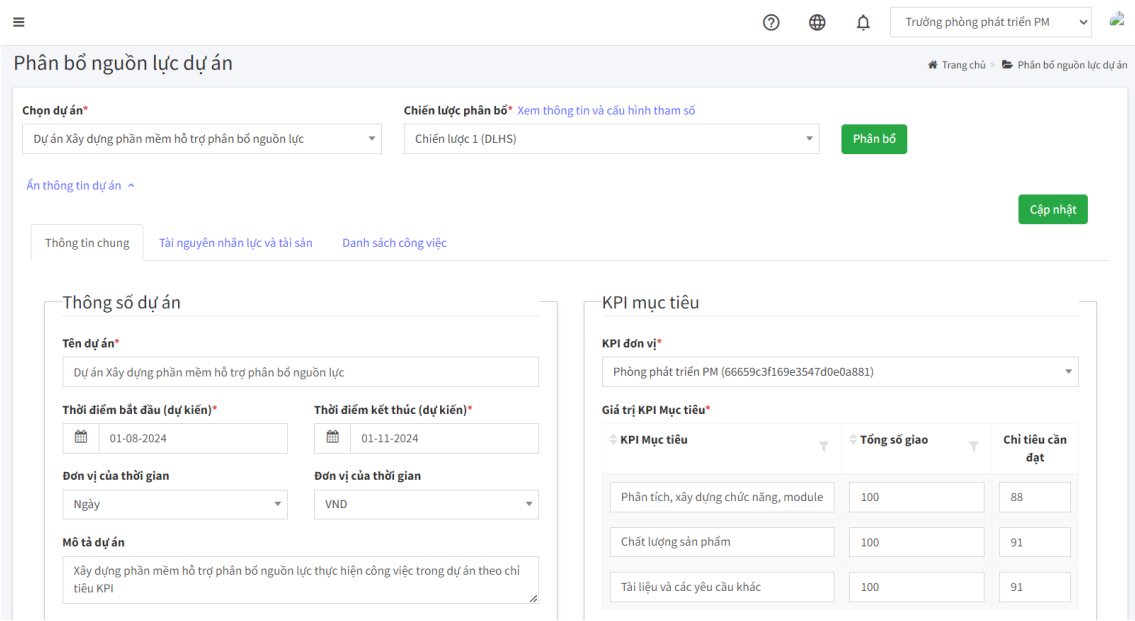


Hình 4.26: Màn hình sau khi thêm dự án thành công

Hình 4.26 là màn hình xác nhận khi người dùng thực hiện lựa chọn và muốn xóa một dự án khi cảm thấy không cần thiết. Người dùng xác nhận xóa bằng cách chọn "Có" hoặc hủy xóa bằng cách chọn "Không".

4.3.4 Các chức năng liên quan đến phân bổ nguồn lực

Sau khi đã có thông tin các dự án, người dùng có thể thực hiện phân bổ nguồn lực cho dự án.



Hình 4.27: Màn hình phân bổ nguồn lực dự án

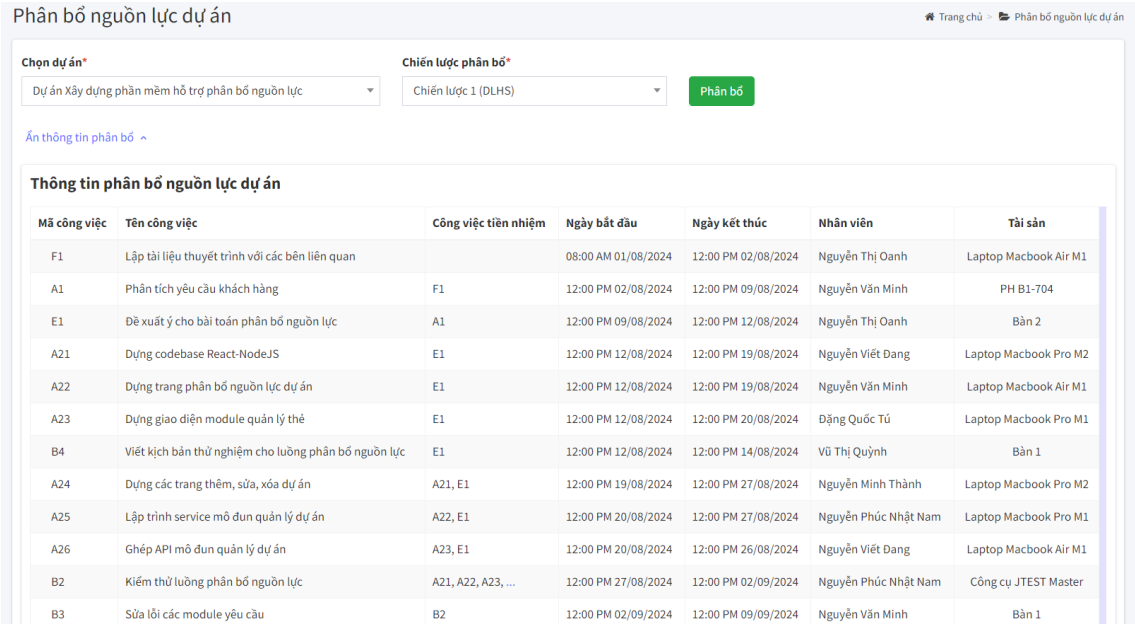
Hình 4.27 mô tả màn hình cho chức năng phân bổ nguồn lực dự án. Ban đầu,

người dùng cần tìm kiếm và lựa chọn dự án cần phân bổ. Sau đó, chọn chiến lược phân bổ nguồn lực: (i) Chiến lược 1 (DLHS), (ii) Chiến lược 2 (HS) tương ứng là thuật toán lựa chọn cho bước gán nhân lực thực hiện công việc trong quá trình phân bổ nguồn lực. Người dùng cũng có thể thực hiện xem xét và điều chỉnh lại nguồn lực, thông tin dự án ở màn này.

Về chi tiết chiến lược thực hiện cũng như tình chỉnh tham số, người dùng ấn vào phần "Xem thông tin và cấu hình tham số" để có thể thực hiện. Cụ thể minh họa như **Hình 4.28**. Chi tiết về chiến lược này sẽ được trình bày chi tiết ở **Chương 5** phía sau. Sau khi đã chọn xong, người dùng nhấn vào nút "Phân bổ", hệ thống sẽ tiếp nhận yêu cầu cùng biểu mẫu và tiến hành phân bổ nguồn lực cho dự án.

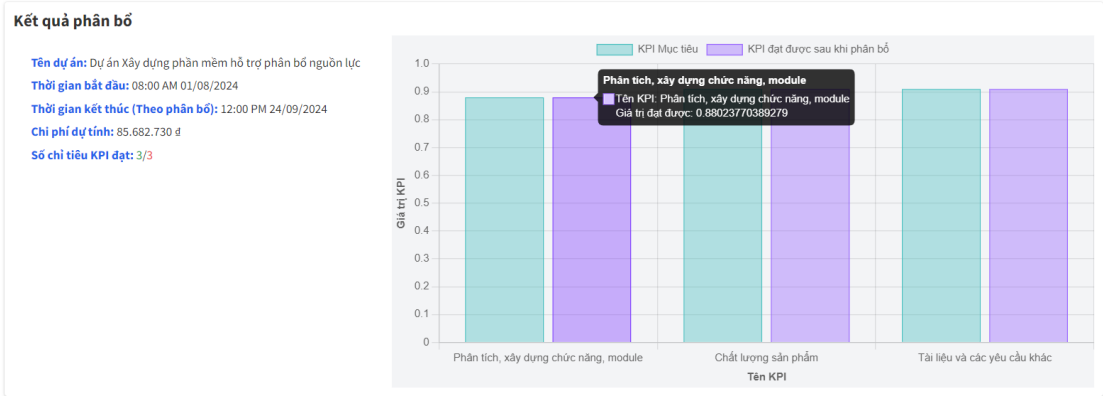
Hình 4.28: Màn hình phân bổ nguồn lực dự án, biểu mẫu thiết lập tham số

Sau khi xử lý yêu cầu, hệ thống sẽ trả về kết quả phân bổ nguồn lực cho người dùng, người dùng có thể xem kết quả phân bổ nguồn lực trả về.



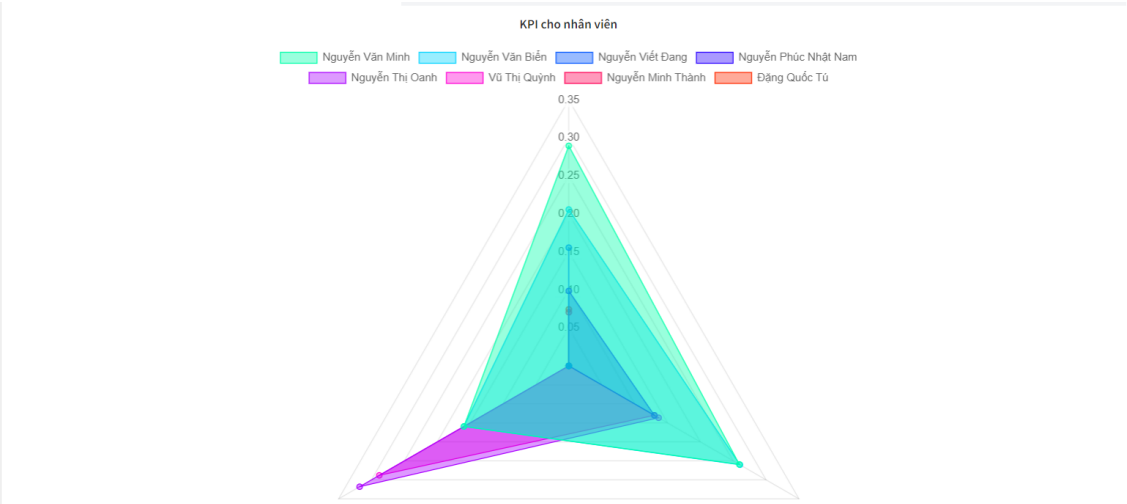
Hình 4.29: Màn hình phân bổ nguồn lực dự án, bảng thông tin phân bổ

Hình 4.29 minh họa về thông tin phân bổ nguồn lực dự án. Bao gồm các công việc kèm theo thời gian thực hiện, nhân sự và tài sản được phân công để thực hiện công việc đó.



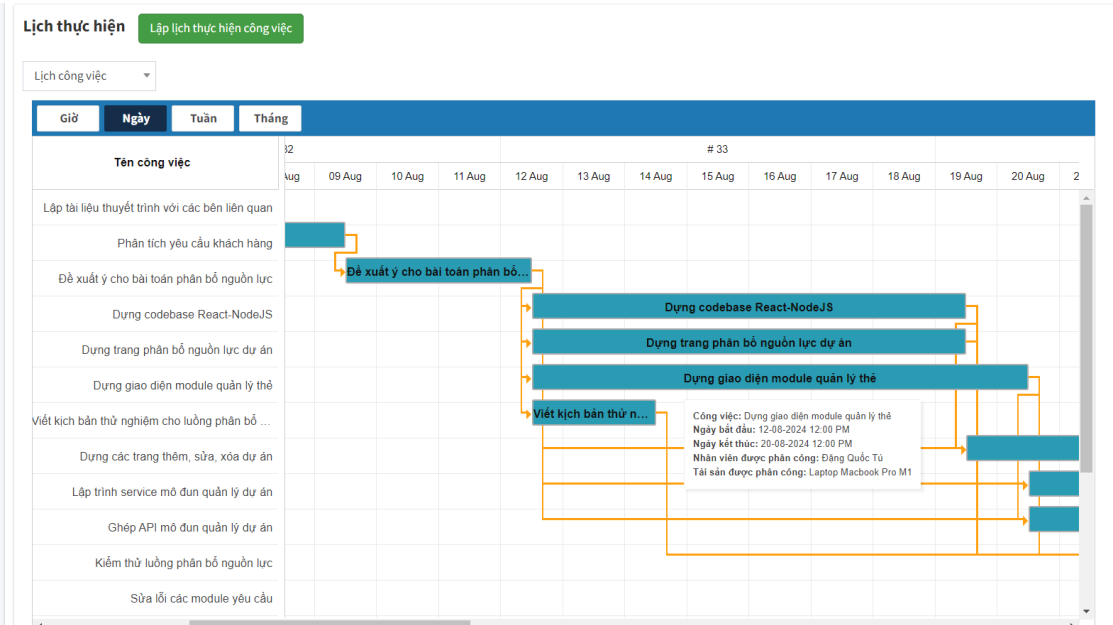
Hình 4.30: Màn hình phân bổ nguồn lực dự án, kết quả phân bổ

Người dùng cũng xem được kết quả phân bổ dự án, bao gồm (i) thời gian bắt đầu, (ii) thời gian kết thúc theo phân bổ, (iii) chi phí nhân công và số chỉ tiêu KPI dự kiến sẽ vượt theo phương án phân bổ trên. Kèm theo đó là biểu đồ thể hiện chỉ tiêu KPI mục tiêu so với giá trị KPI dự kiến sẽ đạt được theo kết quả phân bổ, để dễ dàng so sánh, đối chiếu. Hình 4.30 minh họa về kết quả phân bổ dự án. Ngoài ra, KPI của từng nhân viên tương ứng với kết quả phân bổ được minh họa như Hình 4.31.



Hình 4.31: Mảnh hình phân bổ nguồn lực dự án, kết quả phân bổ (2)

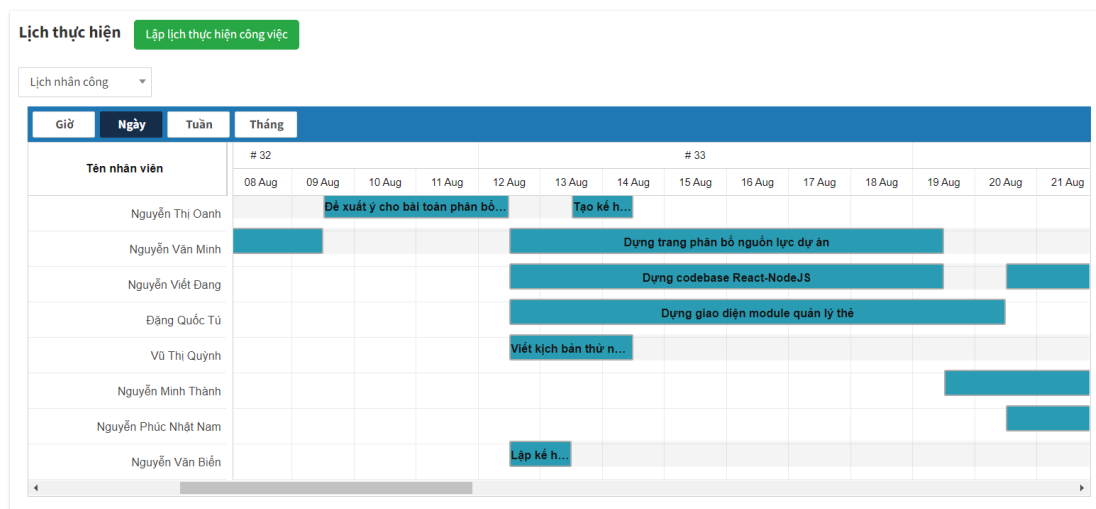
Sau khi có kết quả phân bổ nguồn lực dự án, người dùng có thể thực hiện chức năng lập lịch thực hiện dự án bằng cách ấn vào nút "Lập lịch thực hiện công việc". Hệ thống sẽ xử lý từ kết quả phân bổ, hiển thị lịch thực hiện các công việc dự án cho người dùng. Hình 4.32 biểu thị lịch công việc - bao gồm lịch thực hiện các công việc trong dự án theo thời gian được hiển thị trực quan dưới dạng biểu đồ gantt. Khi trở vào từng công việc, ta còn có thể xem được nhân viên, tài sản thực hiện và chi tiết thời gian bắt đầu, kết thúc của công việc đó.



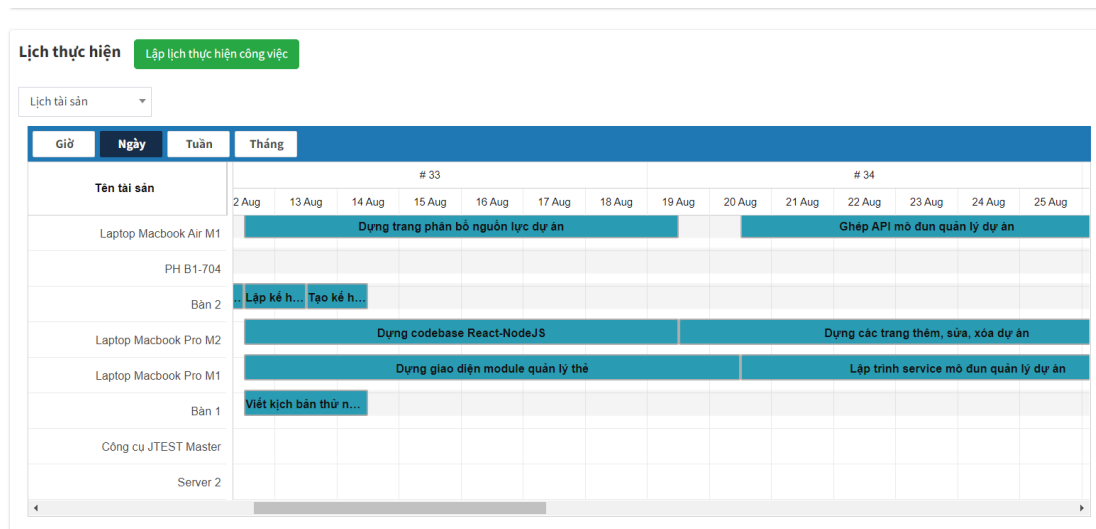
Hình 4.32: Màn hình phân bổ nguồn lực dự án, lịch công việc

Ngoài ra, người dùng có thể thực hiện thay đổi cách nhìn thông qua việc lựa chọn xem lịch nhân công hoặc lịch tài sản. Trong đó, lịch nhân công hiển thị danh sách công việc mà từng nhân viên thực hiện trong dự án (minh họa như Hình 4.33).

Lịch tài sản là lịch thực hiện các công việc của dự án, nhóm theo từng tài sản tham gia dự án, được minh họa trong **Hình 4.34**.



Hình 4.33: Màn hình phân bổ nguồn lực dự án, lịch nhân công



Hình 4.34: Màn hình phân bổ nguồn lực dự án, lịch tài sản

Việc cung cấp nhiều loại lịch giúp cho người dùng có cái nhìn đa chiều, khai thác được tối đa thông tin từ kết quả phân bổ nguồn lực thực hiện dự án.

4.4 Kiểm thử

4.4.1 Kiểm thử tính tương thích (Compatibility Testing)

Trong phần này, em sẽ trình bày về việc kiểm thử tính tương thích của ứng dụng sau khi thực hiện kiểm tra trên các thiết bị khác nhau. **Bảng 4.15** ở bên dưới trình bày kết quả về việc kiểm thử tính tương thích này.

Thiết bị	Thông tin	Chức năng	Giao diện
Laptop Dell Vostro 14	SSD: 256GB, Ram: 16GB, Màn hình: 14"	Đạt	Đạt
PC Dell	SSD: 256GB, Ram: 8GB, Màn hình: 24"	Đạt	Đạt
Laptop Macbook Pro M1 2020	SSD: 512GB, Ram: 8GB, Màn hình: 13"	Đạt	Đạt

Bảng 4.15: Kiểm thử tính tương thích trên các thiết bị

4.4.2 Kiểm thử hộp đen (Black Box Testing)

Danh sách các trường hợp kiểm thử kèm kết quả kiểm thử hộp đen cho nhóm chức năng quản lý danh sách dự án được trình bày trong **Bảng 4.16**.

Chức năng	Đầu vào	Đầu ra mong đợi	Kết quả
Tạo dự án mới	Thông tin chung của dự án thiếu trường thông tin bắt buộc là tên dự án	Hiển thị chữ màu đỏ báo lỗi "Không được bỏ trống trường này"	Đạt
Tạo dự án mới	Thông tin chung của dự án có ngày bắt đầu sau thời gian ngày kết thúc dự kiến	Hiển thị chữ màu đỏ báo lỗi "Ngày kết thúc phải sau ngày bắt đầu"	Đạt
Tạo dự án mới	Để trống trường thành viên tham dự án	Hiển thị chữ màu đỏ báo lỗi "Không được để trống"	Đạt
Tạo dự án mới	Để trống trường tài sản tham gia	Hiển thị chữ màu đỏ báo lỗi "Không được để trống"	Đạt
Tạo dự án mới	Điền 2 công việc có chung mã	Hiển thị lỗi "Mã công việc không được trùng nhau"	Đạt
Tạo dự án mới	Điền 2 công việc có chung tên	Hiển thị lỗi "Tên công việc không được trùng nhau"	Đạt
Tạo dự án mới	Khi thêm 1 công việc, chọn giá trị yêu cầu năng lực của nhân viên vượt quá so với tất cả nhân viên đã được chọn tham gia dự án	Hiển thị lỗi "Không có nhân viên nào tham gia dự án đáp ứng được bộ năng lực yêu cầu, điều chỉnh lại cho hợp lý"	Đạt

Tạo dự án mới	Khi thêm 1 công việc, chọn giá trị yêu cầu tài sản vượt quá so với tất cả tài sản đã được chọn tham gia dự án	Hiển thị lỗi "Tài sản tham gia dự án không đáp ứng đủ yêu cầu của công việc, hãy điều chỉnh lại cho hợp lý"	Đạt
Tạo dự án mới	Chọn thêm công việc bằng cách tải file, tải file danh sách công việc của dự án nhưng không giống như mẫu	Hiển thị lỗi cho người dùng "File tải lên bị lỗi, yêu cầu chỉnh sửa và tải file đúng mẫu"	Đạt
Tạo dự án mới	Chọn thêm công việc bằng cách tải file, tải file danh sách công việc của dự án theo mẫu	Danh sách công việc sẽ được thêm và hiển thị thông tin vào bảng bên dưới	Đạt
Tạo dự án mới	Điền đầy đủ các trường thông tin, không bị lỗi hiển thị, ấn nút Lưu	Đóng biểu mẫu tạo dự án, thông báo tạo dự án thành công, hiển thị lại danh sách dự án	Đạt
Xem danh sách dự án	Tìm kiếm danh sách dự án theo tên dự án, ngày bắt đầu, ngày kết thúc, tên thành viên	Hiển thị danh sách dự án theo các kết quả tìm kiếm	Đạt
Xem thông tin dự án	Tại màn hình danh sách dự án, chọn lấy một dự án, rồi ấn vào hành động xem thông tin dự án	Mở biểu mẫu hiển thị thông tin chi tiết của dự án đã chọn	Đạt
Chỉnh sửa thông tin dự án	Tại màn hình danh sách dự án, chọn lấy một dự án, rồi ấn vào hành động sửa dự án. Sửa thông tin dự án, tuy nhiên để một số trường thông tin không hợp lệ	Hiển thị lỗi cho người dùng về các trường thông tin không hợp lệ	Đạt
Chỉnh sửa thông tin dự án	Tại màn hình danh sách dự án, chọn lấy một dự án, rồi ấn vào hành động sửa dự án. Sửa thông tin dự án, đảm bảo không có lỗi này xảy ra, sau đó ấn nút "Lưu"	Dự án được thay đổi, đóng biểu mẫu, hiển thị thông báo sửa thành công, quay về màn hình danh sách dự án	Đạt

Bảng 4.16: Kiểm thử hộp đen cho nhóm chức năng quản lý danh sách dự án

Bảng 4.17 trình bày về các trường hợp kiểm thử và kết quả kiểm thử hộp đen cho nhóm chức năng phân bổ nguồn lực dự án.

Chức năng	Đầu vào	Đầu ra mong đợi	Kết quả
Phân bổ nguồn lực dự án	Chọn dự án cần phân bổ và chiến lược phân bổ, sau đó ấn "Phân bổ", dự án được chọn tồn tại một số công việc có quan hệ trước - sau theo một chu kỳ	Hiển thị thông báo lỗi "Không thể phân bổ vì công việc phụ thuộc theo chu kỳ"	Đạt
Phân bổ nguồn lực dự án	Chọn dự án cần phân bổ và chiến lược phân bổ, sau đó ấn "Phân bổ", dự án được chọn có thời gian thực hiện sao cho số nhân viên tham gia không đáp ứng được một trong các luồng công việc song song	Hiển thị thông báo "Không thể phân bổ trong khoảng thời gian yêu cầu của dự án, điều chỉnh lại"	Đạt
Phân bổ nguồn lực dự án	Chọn dự án cần phân bổ và chiến lược phân bổ, sau đó ấn "Phân bổ". Dự án chọn lần này không bị lỗi như 2 trường hợp thử nghiệm trên	Hiển thị thông báo phân bổ thành công, hiện kết quả phân bổ cho người dùng	Đạt
Lập lịch thực hiện dự án	Sau khi đã có kết quả phân bổ, chọn "Lập lịch thực hiện dự án"	Hiển thị lịch công việc gồm danh sách các công việc của dự án kèm nhân lực, tài sản được gán thực hiện công việc	Đạt
Xem lịch nhân công	Sau khi đã có kết quả phân bổ, đã chọn "Lập lịch thực hiện dự án", lựa chọn xem "Lịch nhân công"	Hiển thị lịch nhân công gồm danh sách các công việc theo từng nhân viên tham gia dự án	Đạt
Xem lịch tài sản	Sau khi đã có kết quả phân bổ, đã chọn "Lập lịch thực hiện dự án", lựa chọn xem "Lịch tài sản"	Hiển thị lịch tài sản gồm danh sách các công việc theo từng tài sản tham gia dự án	Đạt

Bảng 4.17: Kiểm thử hộp đen cho nhóm chức năng phân bổ nguồn lực dự án

4.5 Triển khai

Ứng dụng được triển khai trên trang <https://dxclan.com>.

Ngoài ra, ta có thể triển khai trực tiếp ứng dụng trên máy tính cá nhân, với các bước thực hiện như sau:

1. Cài đặt các thư viện, công cụ hỗ trợ đã nêu ở chương 3
2. Yêu cầu cài thêm gói npm hoặc yarn
3. Clone hoặc tải trực tiếp mã nguồn từ link github của ứng dụng
4. Tạo 2 file .env ở thư mục phía client-vite và server, rồi thực hiện tạo và gán giá trị cho các biến môi trường
5. Bật Terminal của 2 thư mục client-vite và server, chạy lệnh ***npm install*** hoặc ***yarn install*** để cài đặt các thư viện
6. Sau khi đã cài thư viện xong, bật Terminal phía server, chạy lệnh ***npm run init-db*** hoặc ***yarn init-db*** để thực hiện khởi tạo các dữ liệu mẫu vào cơ sở dữ liệu.
7. Chạy ứng dụng phía server: bật terminal phía server, rồi chạy lệnh ***npm run dev*** hoặc ***yarn start dev***
8. Chạy ứng dụng phía client: bật terminal phía client-vite, rồi chạy lệnh ***npm run dev*** hoặc ***yarn dev***

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Để xây dựng và tích hợp ứng dụng vào hệ thống DXClan, việc phát triển mô hình bài toán và thuật toán cốt lõi nhằm giải quyết bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI đóng vai trò vô cùng quan trọng. Chương này của ĐATN sẽ trình bày chi tiết về mô hình bài toán cũng như thuật toán cốt lõi để giải quyết vấn đề này và đánh giá hiệu quả của thuật toán đề xuất.

5.1 Mô hình bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI

5.1.1 Inputs (Đầu vào)

1. Thông tin chung về dự án

Thông tin chung về dự án được mô tả kèm theo ví dụ nằm trong **Bảng 5.1**.

Dự án P		
Thuộc tính	Ký hiệu	Ví dụ
ID	P^{id}	1
Tên dự án	P^n	Dự án phát triển PM prj-prp
Mô tả	P^d	Phần mềm hỗ trợ phân bổ nguồn lực dự án
Thời gian bắt đầu dự kiến	P^s	8:00 01/08/2024
Thời gian kết thúc dự kiến	P^e	8:00 01/11/2024
Danh sách chỉ tiêu KPI	$KPI_{target} = \{K_i\}$	

Bảng 5.1: Đầu vào thông tin chung về dự án

Trong đó một chỉ tiêu KPI đề ra $K_i \in KPI_{target}$ được mô tả như **Bảng 5.2**.

Chỉ tiêu KPI $K_i \in KPI_{target}$		
Thuộc tính	Ký hiệu	Ví dụ
ID	K_i^{id}	1
Tên KPI	K_i^n	Phân tích, phát triển chức năng, mô-đun
Tiêu chí đánh giá	K_i^{cr}	Số mô-đun hoàn thành đúng hạn
Đơn vị giá trị	K_i^u	mô-đun
Tổng giá trị giao	K_i^{total}	100
Giá trị cần đạt	$K_i^{require}$	88
Giá trị KPI mục tiêu	K_i^{tv}	0.88

Bảng 5.2: Đầu vào thông tin về KPI chỉ tiêu

2. Danh sách công việc $T = \{T_i\}$

Bảng 5.3 mô tả về một công việc $T_i \in T$.

Công việc $T_i \in T$		
Thuộc tính	Ký hiệu	Ví dụ
Tên công việc	T_i^n	Dựng trang thống kê dự án
Mã công việc	T_i^c	A4
Thẻ công việc (tags)	T_i^{tags}	frontend
Công việc tiền nhiệm	$T_i^p = \{t_i\} \subseteq \{T_i^c\}$	A2, A3
Ước tính thời gian thực hiện	T_i^d	3 ngày
Yêu cầu năng lực nhân viên	$re_i = \{(re_{ij}^{qa}, re_{ij}^{qv})\}$	
Yêu cầu năng lực tài sản	ra_i	
KPI tương ứng công việc	T_i^{kpi}	
Yêu cầu về năng lực nhân viên $re_{ij} = (re_{ij}^{qa}, re_{ij}^{qv}) \in re_i$		
Thuộc tính	Ký hiệu	Ví dụ
Năng lực yêu cầu	re_{ij}^{qa}	Backend
Giá trị năng lực yêu cầu	re_{ij}^{qv}	Thành thạo
Yêu cầu về năng lực tài sản $ra_i = (ra_i^t, ra_i^{qa}, ra_i^{qv})$		
Thuộc tính	Ký hiệu	Ví dụ
Loại tài sản	ra_i^t	Laptop
Loại yêu cầu	ra_i^{rt}	Bắt buộc
Số lượng	ra_i^n	1
Giá trị năng lực yêu cầu	ra_i^{qv}	Tốt
KPI tương ứng với task $T_i^{kpi} = (k_i^{id}, k_i^w)$		
Thuộc tính	Ký hiệu	Ví dụ
ID của KPI	$k_i^{id} \in \{k_i\}$	1
Trọng số	k_i^w	0.125

Bảng 5.3: Đầu vào danh sách công việc trong dự án

3. Danh sách nhân viên tham gia dự án $E = \{E_i\}$

Nhân viên $E_i \in E$		
Thuộc tính	Ký hiệu	Ví dụ
ID	E_i^{id}	1
Tên nhân viên	E_i^n	Nguyễn Văn M

Mã nhân viên	E_i^c	SW20210901
Email công ty	E_i^m	nguyenvanm.vnist@gmail.com
Lương	E_i^s	10,000,000 VNĐ
Tập năng lực	$E_i^q = \{E_{ij}^q\}$	
Năng lực $E_{ij}^q = (E_{ij}^{qa}, E_{ij}^{qv}) \in E_i^q$		
Thuộc tính	Ký hiệu	Ví dụ
Tên năng lực	E_{ij}^{qa}	Backend
Giá trị năng lực	E_{ij}^{qv}	Chuyên nghiệp

Bảng 5.4: Đầu vào danh sách nhân viên tham gia dự án

Chi tiết thông tin về nhân viên E_i thuộc dự án được mô tả trong **Bảng 5.4**.

4. Danh sách tài sản tham gia dự án $A = \{A_i\}$

Bảng 5.5 trình bày về thông tin một tài sản $A_i \in A$.

Tài sản $A_i \in A$		
Thuộc tính	Ký hiệu	Ví dụ
ID	A_i^{id}	1
Tên tài sản	A_i^n	Macbook Air M1
Mã tài sản	A_i^{code}	SW.001
Loại tài sản	A_i^t	Laptop
Chi phí vận hành	A_i^c	240,000 VNĐ
Năng lực sử dụng	A_i^{cv}	Tốt

Bảng 5.5: Đầu vào danh sách tài sản tham gia dự án

Trong các **Bảng 5.3, 5.4, 5.5**, giá trị năng lực của nhân viên, tài sản cũng như các giá trị năng lực yêu cầu trong công việc tương ứng với tên giá trị năng lực trong cơ sở dữ liệu, trong quá trình tính toán sẽ ánh xạ sang điểm quy đổi của các năng lực đó. Ví dụ tên năng lực "Backend", giá trị năng lực "Chuyên nghiệp", điểm năng lực quy đổi sẽ là 3.

5. Giá trị hiệu suất thực hiện công việc của nhân viên trong quá khứ

Phần dữ liệu này được tính toán dựa trên danh sách công việc mà nhân viên thực hiện trong quá khứ.

$$lastKPIs = p = \{p_{ij}\} \quad (5.1)$$

Trong đó p_{ij} là giá trị hiệu quả của nhân viên E_i thực hiện công việc T_j được tính toán dựa trên các kết quả đạt được khi E_i thực hiện các công việc có tính tương tự với T_j ở trong quá khứ. Cụ thể sẽ được trình bày ở phần sau.

$$p_{ij} = \begin{cases} 0 & \text{nếu nhân viên chưa thực hiện công việc} \\ -1 & \text{nếu trong quá khứ nhân viên không đạt, bị điều chỉnh} \\ (0, 1] & \text{hiệu quả thực hiện công việc trong quá khứ} \end{cases} \quad (5.2)$$

5.1.2 Outputs (Đầu ra)

Từ dữ liệu đầu vào được cung cấp, sau khi thực hiện tính toán qua các bước, đầu ra mong muốn của bài toán sẽ bao gồm:

1. Kế hoạch thực hiện dự án

Kế hoạch thực hiện dự án (PS) bao gồm: (i) thời gian bắt đầu, (ii) thời gian kết thúc dự án, (iii) chi phí dự toán cho nhân viên và tài sản tham gia dự án sau khi phân công, (iv) danh sách chỉ tiêu KPI của dự án kèm giá trị sẽ đạt được theo phân bổ. Cụ thể được mô tả kèm theo ví dụ trong **Bảng 5.6**.

Kế hoạch thực hiện dự án PS		
Thuộc tính	Ký hiệu	Ví dụ
Thời gian bắt đầu	PS^s	8:00 01/08/2024
Thời gian kết thúc	PS^e	10:00 26/09/2024
Chi phí dự tính	PS^c	88,099,000 VNĐ
Danh sách chỉ tiêu KPI	$PS^{kpi} = KPI_{assign}$	
Lịch thực hiện tập công việc	$PS^t = \{T_i\}$	
Danh sách chỉ tiêu KPI đạt được	$KPI_{assign} = \{Ka_i\}$	

Bảng 5.6: Đầu ra kế hoạch thực hiện dự án

Kết quả đạt được của mỗi chỉ tiêu KPI sau phân bổ được mô tả trong **Bảng 5.7**.

Chỉ tiêu KPI theo kết quả phân bổ $Ka_i \in KPI_{assign} : Ka_i = (Ka_i^{id}, Ka_i^{av})$		
Thuộc tính	Ký hiệu	Ví dụ
ID của chỉ tiêu KPI	Ka_i^{id}	1
Giá trị KPI theo kết quả phân bổ	Ka_i^{av}	0.91

Bảng 5.7: Đầu ra kế hoạch thực hiện dự án

2. Lịch thực hiện các công việc của dự án

Danh sách công việc của dự án sau khi phân bổ là $PS^t = \{T_i\}$. Mỗi công việc đã được bổ sung thông tin về thời gian bắt đầu, kết thúc, nhân lực và tài sản thực hiện công việc đó. Cụ thể, lịch thực hiện các công việc trong dự án có thể ánh xạ qua một trong các loại sau đây:

- Lịch công việc: danh sách công việc trong dự án, mỗi công việc đã được gán thông tin người và tài sản thực hiện tương ứng.
- Lịch nhân công: danh sách nhân viên trong dự án, mỗi nhân viên đã được gán công việc, mỗi công việc đã được gán tài sản tương ứng.
- Lịch tài sản: danh sách tài sản tham gia dự án, tài sản đã được gán công việc, mỗi công việc đã được gán người thực hiện tương ứng.

Thông tin lịch thực hiện của một công việc $T_i \in PS^t$ được minh họa trong **Bảng 5.8**.

Lịch thực hiện công việc $T_i \in PS^t$		
Thuộc tính	Ký hiệu	Ví dụ
Thời gian bắt đầu	T_i^s	8:00 01/08/2024
Thời gian kết thúc	T_i^e	10:00 02/08/2024
Khoảng thời gian	T_i^d	1.25 ngày
Nhân sự thực hiện	$T_i^{assignee} \in E$	Nguyễn Văn A
Tài sản thực hiện	$T_i^{assets} \in A$	Macbook Air M1

Bảng 5.8: Đầu ra lịch trình thực hiện công việc của dự án

3. KPI dự kiến sẽ giao cho nhân viên

$KE = \{KE_i, E_i \in E\}$ là tập KPI dự kiến sẽ giao cho từng nhân viên trong dự án để đảm bảo rằng khi giao như vậy, tồn tại phương án phân bổ để nhân viên có thể thực hiện công việc và đạt được mục tiêu KPI của bản thân, cũng như KPI của dự án đề ra.

Trong đó KE_i là tập KPI sẽ giao cho nhân viên E_i :

$$KE_i = \{KE_{ij} = (KE_{ij}^{id}, KE_{ij}^{av})\}, j = 1, 2, 3, \dots, |KPI_{target}| \quad (5.3)$$

- KE_{ij} : KPI giao cho nhân viên tương ứng một trong các chỉ tiêu KPI của dự án, được mô tả ở **Bảng 5.9**

KPI dự kiến sẽ giao cho nhân viên $KE_{ij} \in KE_i$		
Thuộc tính	Ký hiệu	Ví dụ
ID của KPI	KE_{ij}^{id}	1
Giá trị KPI sẽ giao	KE_{ij}^{av}	0.143

Bảng 5.9: Đầu ra KPI sẽ giao cho nhân viên

5.1.3 Constraints (Ràng buộc)

1. Tất cả tài nguyên (nhân lực, tài sản) khả dụng tại thời điểm mốc. Thời điểm mốc là thời gian hiện tại của hệ thống hoặc thời gian bắt đầu dự án.
2. Tất cả các công việc đều sẵn sàng có thể thực hiện, phân bổ tại thời điểm mốc. Ngược lại, một công việc chỉ được bắt đầu khi tất cả các công việc tiền nhiệm của nó kết thúc.

$$\forall T_i \in T : \begin{cases} T_i^s \geq P^s & \text{nếu } T_i^p = \emptyset \\ T_i^s \geq \max(T_j^s), T_j^s \in T_i^p & \text{nếu } T_i^p \neq \emptyset \end{cases} \quad (5.4)$$

3. Tại một thời điểm, một nguồn lực (nhân lực, tài sản) chỉ có thể thực hiện một công việc.

$$\forall T_i, T_j \in T, i \neq j, [T_i^s, T_i^e] \cap [T_j^s, T_j^e] \neq \emptyset : T_i^{assignee} \neq T_j^{assignee}, T_i^{assets} \neq T_j^{assets} \quad (5.5)$$

4. Thời gian thực hiện công việc phải nằm trong thời gian quy định làm việc, không được tính các khoảng thời gian nghỉ ngơi cho phép của nhân viên và không tính các ngày nghỉ. Cụ thể, thời gian làm việc gồm: (i) buổi sáng từ 08:00 đến 12:00, buổi chiều từ 13:00 đến 17:00. Không làm việc vào thứ Bảy, Chủ nhật.

$$\forall T_i \in T : T_i^s, T_i^e \in [8 : 00, 12 : 00] \cup [13 : 00, 17 : 00] \quad (5.6)$$

5. Thời gian bắt đầu, kết thúc theo kế hoạch đề ra phải đảm bảo nằm trong khoảng thời gian dự kiến thực hiện đã định trước của dự án.

$$[PS^s, PS^e] \subseteq [P^s, P^e] \quad (5.7)$$

Một số giả thiết và phạm vi:

- Trong quá trình phân bổ và thời gian dự kiến thực hiện các công việc, giả sử

máy không bị hỏng hóc và trục trặc gì.

- Trong quá trình phân bổ và thời gian dự kiến thực hiện công việc, giả sử rằng không có sự thay đổi, nghỉ việc của nhân viên trong dự án.
- Mỗi công việc chỉ yêu cầu tối đa một người và một loại tài sản.

5.1.4 Objectives (Hàm mục tiêu)

1. Phân bổ nguồn lực thực hiện công việc với KPI tối ưu

- Tối ưu theo đầu vào: KPI 1 người được coi là tối ưu khi được giao công việc đúng năng lực. Do đó một phương án phân bổ nguồn lực tối ưu KPI theo đầu vào phải đảm bảo KPI sau khi phân bổ dự kiến sẽ đạt được là xấp xỉ nhau đối với những nhân viên có năng lực tương đương.
- Tối ưu theo đầu ra: KPI 1 người được coi là tối ưu khi họ có khả năng hoàn thành các công việc tương ứng với KPI đó theo mức chất lượng được yêu cầu. Một phương án phân bổ được gọi là tối ưu KPI theo đầu ra nếu đạt được tất cả các mục tiêu KPI ban đầu của dự án.

$$\begin{aligned} \forall K_i \in KPI_{target}, Ka_i \in KPI_{assign}, K_i^{id} = Ka_i^{id} : \\ Ka_i^{av} \geq K_i^{tv}, \quad i = 1, 2, \dots, |KPI_{target}| \end{aligned} \quad (5.8)$$

- Tối ưu theo quá trình xử lý: KPI 1 người được coi là tối ưu khi giao KPI cho người đó thì KPI của tổ chức cũng tối ưu mà không cần điều chỉnh lại. Nói cách khác, một phương án phân bổ tối ưu KPI theo quá trình đảm bảo không có trường hợp giao việc cho nhân viên đã từng thực hiện không đạt công việc tương ứng trong quá khứ và phải điều chỉnh.

$$\forall T_k \in T, T_k^{assignee} = E_i : p_{ij} \neq -1 \quad (5.9)$$

2. Chi phí thực hiện

Chi phí thực hiện dự án sẽ được tính dựa trên tổng chi phí của từng công việc trong dự án. Với mỗi công việc, chi phí sẽ được tính toán dựa trên thời gian thực hiện, lương của nhân viên và chi phí vận hành của tài sản được phân bổ tương ứng.

$$F_{total_cost} = \sum_{i=1}^{|T|} T_i^d \cdot (\alpha \cdot E_j^s + \beta \cdot \sum_k A_k^c) \rightarrow minimize \quad (5.10)$$

Trong đó:

- $\alpha, \beta > 0$ là các hệ số để quy lương nhân viên, chi phí vận hành máy móc về chung 1 hệ quy chiếu
- $E_j^s = T_i^{assignee}$, $A_k = T_i^{assets}$ lần lượt là nhân viên, tài sản thực hiện công việc T_i

3. Thời gian thực hiện

Thời gian thực hiện dự án sẽ được tính bằng khoảng thời gian tính từ thời điểm bắt đầu sớm nhất và kết thúc muộn nhất của các công việc trong dự án. Do nhân viên và tài sản chỉ có thể thực hiện các công việc trong các khung thời gian làm việc hợp lệ nên việc tính thời gian này sẽ bao gồm cả thời gian nghỉ ngơi hợp lệ của nguồn lực.

$$F_{total_time} = \max_{T_i \in T}(T_i^e) - \min_{T_i \in T}(T_i^s) \rightarrow minimize \quad (5.11)$$

5.2 Thuật toán đề xuất sử dụng để giải quyết bài toán

Giải thuật đề xuất cho bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI với đầu vào, đầu ra, hàm mục tiêu đã đề cập ở trên bao gồm các bước được đề cập dưới đây.

Một số quy ước sử dụng trong việc tính toán độ phức tạp thuật toán:

- V, E_G lần lượt là tập đỉnh, tập cạnh của đồ thị $G(V, E_G)$ tạo bởi tập công việc, và mối quan hệ tiền nhiệm của các công việc.
- $E_G = \{(T_i, T_j) : T_i, T_j \in T, T_i \in T_j^p \mid T_j \in T_i^p\}$, giả sử $T_i \in T_j^p$ thì (T_i, T_j) là 1 cạnh của E_G
- $n = |V| = |T|$: số công việc trong dự án
- $m = |E_G|$: số cạnh của đồ thị G
- $a = |A|$: số nhân viên tham gia dự án
- $e = |E|$: số tài sản tham gia dự án

5.2.1 Tiền xử lý: tính toán giá trị hiệu suất thực hiện công việc

Bước này tính toán giá trị hiệu suất thực hiện từng công việc của từng nhân viên trong dự án dựa trên dữ liệu thực hiện công việc tương tự trong quá khứ phục vụ cho các bước thực hiện của thuật toán được trình bày phía sau.

Đầu vào: danh sách nhân viên $E = \{E_i\}$, danh sách công việc $T = \{T_i\}$, danh sách công việc đã thực hiện trong quá khứ $TP = \{TP_i\}$

Đầu ra: hiệu suất thực hiện công việc $lastKPIs = \{p_{ij}\}$. Trong đó p_{ij} là hiệu

suất thực hiện công việc T_j của nhân viên E_i .

Các bước thực hiện để tính toán được tập các giá trị hiệu suất trên bao gồm:

1. **Xác định công việc tương đồng:** duyệt qua từng công việc và tìm các nhân viên có đủ năng lực để thực hiện công việc đó. Đối với mỗi nhân viên, tìm công việc trong quá khứ có yêu cầu năng lực gần nhất với công việc hiện tại. Hai công việc có thể coi là tương đồng nếu thỏa mãn đủ 2 điều kiện sau: (i) chúng cùng thuộc một lĩnh vực nào đó (cùng thẻ công việc - tags), (ii) yêu cầu năng lực giống nhau và chỉ sai khác nhiều nhất một loại năng lực. Độ tương đồng được đo bằng khoảng cách giữa các véc-tơ yêu cầu năng lực của công việc, khoảng cách càng nhỏ thì độ tương đồng càng cao. Ví dụ:

- công việc T_a của dự án, $T_a^{tags} = [backend, frontend]$ yêu cầu về năng lực nhân viên là $T_A^{ra} = (\text{Backend: Đã làm, Frontend: Thành thạo, Kinh nghiệm: Dưới 1 năm})$ ánh xạ qua khóa năng lực và điểm quy đổi giá trị ta có (backend: 1, frontend: 2, year_of_exp: 1) tương ứng với véc-tơ $d_a = (1, 2, 1)$
- công việc T_b đã thực hiện trong quá khứ, $T_b^{tags} = [backend, frontend]$ yêu cầu về năng lực nhân viên là $T_B^{ra} = (\text{Backend: Thành thạo, Frontend: Thành thạo, Kinh nghiệm: Dưới 1 năm})$ ánh xạ qua khóa năng lực và điểm quy đổi giá trị ta có (backend: 2, frontend: 2, year_of_exp: 1) tương ứng với véc-tơ $d_b = (2, 2, 1)$
- Chú ý rằng hai véc-tơ phải đưa về chung hệ quy chiếu và giá trị tương ứng với các năng lực yêu cầu, các chiều phải có cùng năng lực.

Khi đó độ tương đồng là $dT(T_a, T_b) = d(d_a, d_b) = \|d_b - d_a\| = \|(1, 0, 0)\| = 1$

2. **Tính toán hiệu suất:** sau khi tìm được công việc tương đồng trong quá khứ, tính toán hiệu suất thực hiện công việc dựa trên công việc đó. Giả sử xét nhân viên E_i thực hiện công việc T_a trong dự án, và công việc tương đồng là T_b với điểm hiệu suất trong quá khứ là T_b^{point} . Công thức 5.12 tính hiệu suất thực hiện công việc T_a của E_i được trình bày bên dưới.

$$lastKPIs(E_i, T_a) = p_{ia} = \begin{cases} -1, & \text{nếu } T_b^{point} = -1 \text{ \& } \|d_a\| \geq \|d_b\| \\ T_b^{point}, & \text{nếu } dT(T_a, T_b) = 0 \text{ \& } \|d_a\| \leq \|d_b\| \\ T_b^{point} \cdot \sqrt{\frac{d_b}{d_a}}, & \text{nếu } dT(T_a, T_b) > 0 \text{ \& } \|d_a\| > \|d_b\| \\ 0, & \text{các TH còn lại} \end{cases} \quad (5.12)$$

Sau khi đã tính toán xong, duyệt lại một lần nữa để tính hiệu suất, nếu nhân

viên E_i chưa từng thực hiện công việc tương tự, sử dụng công thức 5.13.

$$lastKPIs(E_i, T_a) = p_{ia} = \begin{cases} \min_{E_j \text{ available } T_a, p_{ja} \neq -1} (p_{ja}) \\ random(0.8, 1), \text{ các TH còn lại} \end{cases} \quad (5.13)$$

Giá trị hiệu suất sẽ là giá trị nhỏ nhất trong các hiệu suất của những nhân viên khác có thể thực hiện công việc đó, để đảm bảo ước lượng chặt chẽ và giảm rủi ro. Nếu không có nhân viên nào có giá trị hiệu suất, gán ngẫu nhiên một giá trị từ 0.8 đến 1, giả định rằng nhân viên có thể thực hiện tốt công việc.

Độ phức tạp tính toán cho 2 lần duyệt: $O(2 \cdot n \cdot a \cdot n_{cv}) = O(2na)$ với $n = |T|$, $a = |E|$, $n_{cv} = |TP|$ là hằng số.

5.2.2 Bước 1: Sắp thứ tự, ước tính thời gian thực hiện công việc

a, Bước 1.1: Sắp thứ tự topo các công việc

Mã giả 1 trình bày về mã giả của thuật toán sắp topo.

Mã giả 1: Thuật toán sắp topo

Data: Danh sách công việc $T = \{T_i\}$

Result: Thứ tự thực hiện công việc đã được sắp xếp hoặc lỗi nếu xảy ra sự phụ thuộc công việc theo chu trình

```

1 Function TopologicalSort ( $\{T_i\}$ ) :
2    $result, visited, inProgress \leftarrow \emptyset$ ;
3   Function visit ( $T_i$ ) :
4     if  $T_i \in inProgress$  then
5       return Error 'Tasks have cyclic dependencies';
6     if  $T_i \notin visited$  then
7        $inProgress \leftarrow inProgress \cup \{T_i\}$ ;
8       foreach  $dep \in T_i^p$  do
9         visit ( $dep$ );
10       $inProgress \leftarrow inProgress \setminus \{T_i\}$ ;  $visited \leftarrow visited \cup \{T_i\}$ ;
11       $result \leftarrow \{T_i\} \cup result$ ;
12  foreach  $T_i \in \{T_i\}$  do
13    visit ( $T_i$ );
14  Reverse  $result$ ;
15  return  $result$ ;
```

Trong dự án, một số công việc có thể có danh sách công việc tiên nhiệm, cần đảm bảo các công việc tiên nhiệm được hoàn thành trước. Vấn đề có thể xảy ra là vòng lặp tiên nhiệm khép kín, dẫn đến vòng lặp vô hạn và không thể phân bổ nguồn lực. Để giải quyết các vấn đề trên, thuật toán sắp xếp topo (Topological Sort) được

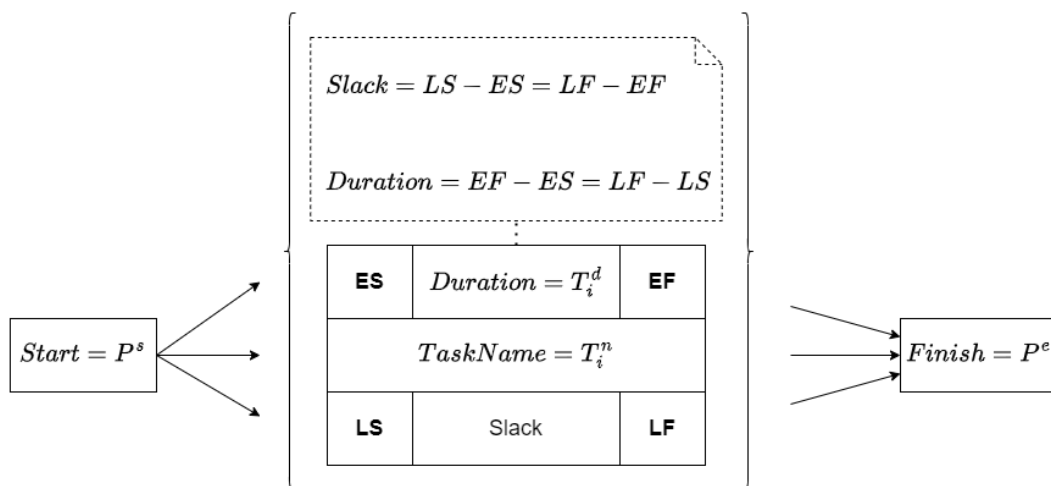
sử dụng.

Kết quả là, thứ tự công việc được sắp xếp đảm bảo mỗi công việc chỉ bắt đầu khi các công việc tiền nhiệm đã hoàn thành, đồng thời, xử lý được các tập công việc đầu vào ngoại lệ của bài toán.

Đánh giá độ phức tạp tính toán: $O(|V| + |E|) = O(n + m)$

b, Bước 1.2: Sử dụng phương pháp CPM (Critical Path Method) để định hình thời gian cho từng công việc trong dự án

Sau khi xác định thứ tự công việc và thời gian bắt đầu, kết thúc dự kiến của dự án, phương pháp đường găng - CPM sẽ được sử dụng để xác định, ước lượng thời gian thực hiện các công việc trong dự án.



Hình 5.1: Minh họa phương pháp CPM cho các công việc

Hình 5.1 minh họa phương pháp CPM cho các công việc. Theo phương pháp CPM, ta sẽ xác định được các thời điểm của từng công việc như: (i) thời gian bắt đầu sớm nhất - Earliest Start Time (ES), (ii) thời gian kết thúc sớm nhất - Earliest Finish Time (EF), (iii) thời gian bắt đầu muộn nhất - Latest Start Time (LS), (iv) thời gian kết thúc muộn nhất - Latest Finish Time (LF).

Sau đó, ta sẽ tính được những khoảng thời gian có thể các công việc cần thực hiện song song để dự trù được nhân sự cần thiết tương ứng. Cụ thể, thứ tự thực hiện ở bước này như sau:

1. Duyệt tập công việc đã được sắp xếp thứ tự theo chiều xuôi, từ đó tính được ES và EF của từng công việc. Giá trị ES, EF của từng công việc được xác định

lần lượt theo công thức 5.14, 5.15

$$ES(T_i) = \begin{cases} \max_{T_j \in T_i^p} (EF(T_j)), & \text{nếu } T_i^p \neq \emptyset \\ P^s, & \text{nếu } T_i^p = \emptyset \end{cases} \quad (5.14)$$

$$EF(T_i) = ES(T_i) + T_i^d \quad (5.15)$$

2. Duyệt tập công việc đã được sắp xếp thứ tự theo chiều ngược, từ đó tính được LF và LS của từng công việc. Giá trị LF, LS của từng công việc được xác định lần lượt theo công thức 5.16, 5.17

$$LF(T_i) = \begin{cases} \min_{T_j \in T_i^{successor}} (LS(T_j)), & \text{nếu } T_i^{successor} \neq \emptyset \\ P^e, & \text{nếu } T_i^{successor} = \emptyset \end{cases} \quad (5.16)$$

với $T_i^{successor} = \{T_j \in T : T_i \in T_j^p\}$ là tất cả các công việc T_j thực hiện ngay sau công việc hiện xét.

$$EF(T_i) = ES(T_i) + T_i^d \quad (5.17)$$

3. Sau khi có các giá trị ES, EF, LS, LF, duyệt qua các khoảng thời gian $[ES, LF]$ của từng công việc để tính số công việc có thể cần thực hiện song song. Số nhân viên cần tại khoảng thời gian đó tối thiểu bằng số công việc song song, vì mỗi người chỉ làm một việc tại một thời điểm. Nếu số nhân viên không đáp ứng đủ, thông báo lỗi và yêu cầu điều chỉnh thời gian.

Đánh giá độ phức tạp: $2 \cdot O(n + m) + O(n^2 + m) = O(n^2 + m)$, Với $O(n + m)$ cho từng bước tính ES, EF, LS, LF và $O(n^2 + m)$ cho bước tính số công việc có thể cần thực hiện song song.

Ở bước này, người quản lý dự án có thể nắm bắt thời gian thực hiện từng công việc và xác định tính khả thi của thời gian dự án. Về mặt thuật toán, ta có thể phát hiện và thông báo các ngoại lệ về thời gian, nguồn lực đáp ứng ngay lập tức nếu dự án không khả thi, tránh tính toán không cần thiết ở bước sau.

Tóm lại, sau bước 1, ta đã có tập công việc được sắp thứ tự, kèm theo các mốc thời gian thực hiện theo phương pháp CPM, và có thể loại bỏ được những ngoại lệ nếu có về mặt thời gian, thứ tự công việc.

5.2.3 Bước 2: Gán tài sản, khung thời gian cho từng công việc đã được sắp thứ tự

Sau khi sắp xếp các công việc trong dự án theo thứ tự, ta sẽ gán tài sản và khung thời gian thực hiện sao cho các công việc bắt đầu sớm nhất có thể, đồng thời thỏa mãn các ràng buộc về công việc tiền nhiệm và thời gian thực hiện của tài sản mà công việc đó yêu cầu. Điều này giúp tối thiểu hóa thời gian hoàn thành dự án.

Mã giả 2: Gán tài sản, khung thời gian cho công việc

Data: công việc đã sắp thứ tự $T = \{T_i\}$, dự án P , $\{A_i\}$, danh sách công việc đang được giao thực hiện của các nhân viên $T' = \{T'_i\}$

Result: Danh sách công việc $\{T_i\}$ đã được gán thời gian, tài sản

```

1 function scheduleTasksWithAsset ( $\{T_i\}, P, \{A_i\}$ ) :
2    $currentAssets \leftarrow \{A_i\}, T_0^s \leftarrow P^s$  ;
3   foreach  $T_i \in T$  do
4      $taskAssets, availableTime \leftarrow$ 
        $getAvailableTimeForTask (currentAssets, T_i, T')$  ;
5      $T_i^{assets} \leftarrow taskAssets$ ;
6     Calculate  $T_i^s$  base on  $availableTime, T_i^p$ ;
7     Calculate  $T_i^e$  base on  $T_i^s, T_i^d$  ;
8      $updateCurrentAssets (currentAssets, taskAssets, T_i^s, T_i^e)$  ;
9   return  $\{T_i\}$  assigned with assets and time

```

Mã giả 2 mô tả cách gán tài sản, thời gian thực hiện cho từng công việc. Hàm $getAvailableTimeForTask()$ sẽ tìm thời gian sớm nhất mà tất cả tài sản cần thiết rảnh rỗi và tồn tại nhân viên có thể được gán cho công việc. Quá trình này ưu tiên tài sản bắt buộc theo yêu cầu công việc, nhằm đảm bảo đạt được mục tiêu thời gian thực hiện của dự án. Các công việc có yêu cầu tài sản tùy chọn sẽ được xem xét sau.

Thời gian bắt đầu công việc sẽ là thời gian gần nhất đã tìm được hoặc thời gian hoàn thành của tất cả các công việc tiền nhiệm, tùy theo thời gian nào muộn hơn. Khi đó tất cả ràng buộc tiền nhiệm, yêu cầu về tài sản cũng như đảm bảo có nhân lực để thực hiện công việc đều thỏa mãn. Thời gian kết thúc được tính toán dựa trên thời gian bắt đầu và ước tính khoảng thời gian thực hiện của từng công việc.

Hàm $updateCurrentAssets()$ thực hiện cập nhật lại trạng thái của tài sản sau khi chúng đã được gán cho công việc hiện tại.

Đánh giá độ phức tạp tính toán, xét **Mã giả 2** ta có:

- Dòng 3 thực hiện vòng lặp qua từng công việc, độ phức tạp là $O(n)$

- Dòng 4 thực hiện tìm thời gian thỏa mãn về yêu cầu tài sản, tồn tại nhân viên có thể thực hiện, do số công việc mà các nhân viên thực hiện ngoài dự án T' là hằng số nên mất chi phí là $O(e + a)$ cho một vòng lặp. Tuy nhiên, thực thể mỗi công việc thì số nhân viên có thể thực hiện được rất nhỏ so với tập nhân viên, nên chi phí trên là trong trường hợp tồi nhất.
- Dòng 6 tính toán thời gian dựa trên các công việc tiền nhiệm, sau khi xét cả n vòng lặp thì sẽ xét tất cả công việc tiền nhiệm, chính bằng số cạnh của đồ thị nên độ phức tạp là $O(m)$ cho toàn bộ vòng lặp.
- Dòng 8 cập nhật lại các tài sản thực hiện công việc, độ phức tạp $O(a)$ cho một vòng lặp
- Các dòng còn lại độ phức tạp hằng số

Độ phức tạp của bước trên là: $O(n \cdot (e + a + a) + m) = O(n \cdot e + n \cdot a + m)$

Như vậy, sau bước 2, ta đã có danh sách công việc của dự án được sắp xếp theo một thứ tự, kèm theo việc gán tài sản và gian thực hiện từng công việc.

5.2.4 Bước 3: Gán nhân lực, tài sản tùy chọn thực hiện công việc

a, Bước 3.1: Phân chia KPI mục tiêu của dự án cho các nhân viên

Mã giả 3 mô tả cho việc thực hiện chia KPI mục tiêu của dự án cho nhân viên.

Mã giả 3: Hàm thực hiện chia KPI cho nhân viên

Data: $KPI_{target} = \{K_i\}$, $E = \{E_i\}$, $assetKPIWeight$, $T = \{T_i\}$ đã được gán tài sản và thời gian thực hiện

Result: KS biểu thị KPI được phân bổ cho nhân viên theo năng lực

```

1 function splitKPIToEmployees ( $T, E, KPI_{target},$ 
    $assetKPIWeight$ ) :
2    $kpiOfEmployeesSplitTmp \leftarrow \emptyset$ , get  $k$  base on  $E$  ;
3    $clusters \leftarrow$  kMeanWithEmployees ( $E, k$ ) ;
4   foreach  $T_i \in T$  do
5      $kpiOfEmpsInTask \leftarrow$ 
       splitKPIOfTask ( $T_i, KPI_{target}, clusters, assetKPIWeight$ ) ;
6     Update  $kpiOfEmployeesSplitTmp$  base on  $kpiOfEmpsInTask$  ;
7    $minKpis \leftarrow$  findMinKPIUnit ( $T, KPI_{target}, assetKPIWeight$ ) ;
8    $KS \leftarrow$  reSplitKPI ( $minKpis, kpiOfEmployeesSplitTmp$ ) ;
9   return  $KS$  ;
```

Bước này thực hiện phân chia KPI mục tiêu của dự án cho nhân viên dựa trên năng lực, kết quả thu được làm ngưỡng KPI đầu vào tạm xét của từng nhân viên. Điều này đảm bảo điều kiện cần về mục tiêu tối ưu KPI đầu vào: chia KPI mục tiêu

theo năng lực như đã nêu ở mô hình bài toán.

Ban đầu, nhân viên được phân thành 4 nhóm năng lực: Trung bình, Khá, Giỏi, Xuất sắc, bằng thuật toán K-Mean thông qua hàm *kMeanWithEmployee()*. Điểm trung bình của mỗi nhóm được tính dựa trên tổng giá trị năng lực của nhân viên trong nhóm đó. Tuy nhiên, việc phân nhóm có thể thay đổi tùy thuộc vào số lượng nhân viên và bộ năng lực cụ thể của từng nhân viên.

KPI của dự án sẽ được tính bằng tổng KPI của các công việc (với trọng số tương ứng) cộng lại. Với mỗi công việc, hàm *splitKPIOfTask()* sẽ phân bổ KPI mục tiêu cho từng người có thể thực hiện công việc đó.

Đầu tiên, KPI được phân bổ theo tỷ lệ điểm của các nhóm năng lực. Điều này đảm bảo việc phân bổ KPI theo các nhóm năng lực. Sau đó, trong cùng một nhóm, KPI được phân bổ tiếp dựa trên tỷ lệ năng lực của từng nhân viên so với yêu cầu công việc. Điều này để đảm bảo rằng trong cùng một nhóm năng lực, nhân viên nào có năng lực phù hợp hơn với yêu cầu công việc sẽ nhận được KPI nhiều hơn.

Mỗi công việc chỉ được giao cho một nhân viên, dẫn đến việc chia KPI có thể bị phân mảnh, tức là tạo ra các chỉ tiêu KPI nhỏ hơn nhiều so với KPI đạt được khi hoàn thành một công việc, ngay cả với công việc có KPI nhỏ nhất. Để phân bổ KPI hợp lý hơn, ta xác định các giá trị KPI nhỏ nhất - *minKpis* đạt được khi thực hiện công việc với từng loại chỉ tiêu KPI, thông qua hàm *findMinKPIUnit()*. Sau đó, hàm *reSplitKPI()* sẽ tái phân bổ các KPI nhỏ bé bị phân mảnh trên cho các nhân viên khác, dựa trên các giá trị *minKpis* tìm được, đảm bảo việc phân chia KPI trở nên hợp lý và khả thi hơn.

Đánh giá độ phức tạp tính toán, xét **Mã giả 3** ta có:

- Dòng 3 thực hiện thuật toán K-Mean, với k đã được xác định là 4 nhóm, độ phức tạp là $O(4t \cdot e) = O(t \cdot e)$, t là số lần lặp để hội tụ, thường rất nhỏ nên chi phí thực tế có thể thấp hơn độ phức tạp tồi tệ nhất $O(t \cdot e)$.
- Dòng 4 - 7 thực hiện n vòng lặp, mỗi vòng lặp chia KPI của công việc cho từng nhân viên thực hiện được theo cụm, chi phí tồi tệ nhất là $O(n \cdot (e+e)) = O(n \cdot e)$, tuy nhiên, thực tế với một công việc thì số nhân viên có thể thực hiện công việc sẽ nhỏ hơn toàn bộ tập nhân viên, nên chi phí thực tế có thể nhỏ hơn nhiều so với chi phí tồi tệ nhất $O(n \cdot e)$.
- Dòng 8 chỉ cần duyệt qua danh sách công việc, và số lượng KPI mục tiêu trong dự án (hằng số) nên độ phức tạp $O(n)$.
- Dòng 9 thực hiện tái phân bổ lại KPI thực hiện cho nhân viên, cần duyệt danh sách nhân viên 2 lần (1 lần cho việc xét KPI và 1 lần chia lại KPI phân mảnh),

độ phức tạp tồi tệ nhất có thể là $O(e^2)$.

Tóm lại, độ phức tạp tính toán của bước trên là $O(t \cdot e + n \cdot e + n + e^2) = O(n \cdot e + e^2)$.

b, Bước 3.2: Tìm phương án gán nhân lực, tài sản tùy chọn cho công việc bằng cách sử dụng thuật toán tìm kiếm hài hòa

Ở bước này, sau khi đã gán tài sản và khung thời gian cho công việc, cũng như có dữ liệu ngưỡng KPI tạm thời của nhân viên dựa trên năng lực, ta sẽ sử dụng thuật toán HS hoặc DLHS để gán nhân lực và tài sản tùy chọn cho các công việc dự án. Mong muốn là hoàn thành việc phân bổ nguồn lực nhằm đạt được các mục tiêu tối ưu đã đề ra.

Như đã nêu ở **Chương 3** khi trình bày về thuật toán tìm kiếm hài hòa, thuật toán đề xuất sẽ thực hiện cải tiến cho bước tìm kiếm lời giải ngẫu nhiên của thuật toán tìm kiếm hài hòa HS, cũng như DLHS. Cụ thể, thay vì chọn ngẫu nhiên nhân viên rồi gán thực hiện cho từng công việc như cách mà thuật toán HS, DLHS đã đề cập, sẽ thực hiện gán ngẫu nhiên nhân viên có khả năng thực hiện công việc đó. Điều này giúp giảm đáng kể không gian tìm kiếm đem lại kết quả không tốt. Theo đó, trong cả thuật toán HS hay DLHS, việc khởi tạo một lời giải ngẫu nhiên (phục vụ khởi tạo bộ nhớ hài hòa) đều thực hiện như sau:

1. Duyệt từng công việc, với mỗi công việc, thực hiện gán ngẫu nhiên nhân viên có thể thực hiện công việc đó (đảm bảo về yêu cầu năng lực, rảnh rỗi trong khoảng thời gian thực hiện công việc).
2. Sau khi đã gán nhân viên thực hiện công việc, nếu công việc có yêu cầu tài sản tùy chọn, ta sẽ xem xét xem có tài sản phù hợp nào không. Nếu có, kiểm tra và so sánh chi phí việc sử dụng tài sản với không sử dụng tài sản. Nếu tiết kiệm hơn, thêm tài sản vào để thực hiện công việc.
3. Sau khi đã duyệt hết công việc, tính toán các giá trị mục tiêu cho phương án ngẫu nhiên, đóng gói thành một phương án lời giải.

Trong mỗi vòng lặp của cả hai thuật toán, việc tạo ra một phương án cải tiến (improvised solution) được thực hiện theo ba bước tương tự. Tuy nhiên, ở bước thứ hai, thay vì tạo ra các phương án ngẫu nhiên, thuật toán sẽ tìm kiếm các phương án cải tiến dựa trên các tham số của thuật toán và bộ nhớ hài hòa. Quá trình này sẽ khác nhau tùy thuộc vào việc sử dụng thuật toán HS hay DLHS.

Cụ thể, mỗi một phương án lời giải trong thuật toán HS, hoặc DLHS cho bài toán sẽ gồm:

1. Tập công việc đã sắp tự, gán thời gian, tài sản, nhân viên thực hiện. Dữ liệu này ánh xạ tới kế hoạch dự án PS , lịch trình thực hiện công việc $PS^t = \{T_i\}$

trong đầu ra của bài toán.

2. Các giá trị mục tiêu của phương án phân bổ, bao gồm:

- Tổng chi phí thực hiện: gồm chi phí nhân công và tài sản các công việc.
- Số trường hợp gán nhân lực bị đọng độ: là số trường hợp mà tồn tại 1 nhân viên thực hiện 2 công việc trong cùng 1 thời điểm.
- Số trường hợp gán nhân lực bị thiếu: là số nhân viên không được gán thực hiện công việc trong dự án.
- KPI mục tiêu đạt được theo phương án phân bổ: ánh xạ tới đầu ra danh sách chỉ tiêu KPI KPI_{assign} của dự bài toán.
- KPI của từng nhân viên theo phương án phân bổ: ánh xạ tới đầu ra KPI dự kiến sẽ giao cho nhân viên KE của bài toán.
- Khoảng cách Euclid giữa 2 vector KPI chia cho nhân viên theo năng lực KS , và KPI của từng nhân viên theo phân bổ KE , $d = (KE, KS)$.

Một phương án lời giải tốt sẽ là phương án thỏa mãn được các mục tiêu (một lời giải hài hòa), cụ thể, các mục tiêu cho một lời giải hài hòa trong bài toán là:

- Tất cả nhân viên đều được phân công thực hiện công việc, không có trường hợp giao việc mà nhân viên đã không đạt và bị điều chỉnh trong quá khứ. Điều này đảm bảo mục tiêu tối ưu KPI theo quy trình của bài toán.
- KPI đạt được vượt các KPI chỉ tiêu đề ra với dự án. Điều này đảm bảo mục tiêu tối ưu KPI đầu ra của bài toán.
- Khoảng cách Euclid giữa 2 véc-tơ KPI chia cho nhân viên theo năng lực KS , và KPI của từng nhân viên theo phân bổ KE , $d = (KE, KS)$ là nhỏ nhất. $d < \epsilon$, $\epsilon > 0$ rất nhỏ. Từ đó đảm bảo mục tiêu tối ưu KPI đầu vào của bài toán.

Phương pháp so sánh giữa 2 phương án lời giải được thực hiện theo thứ tự ưu tiên như sau:

1. Tất cả nhân viên đều được phân công công việc: phương án tốt hơn là phương án có càng nhiều nhân viên được giao công việc.
2. Vượt các chỉ tiêu KPI của dự án: phương án tốt hơn là phương án có nhiều chỉ tiêu KPI đạt hơn, nếu tất cả chỉ tiêu KPI đều đạt, so sánh về tổng giá trị KPI đạt được.
3. Khoảng cách Euclid giữa 2 vector KPI chia cho nhân viên theo năng lực KS , và KPI của từng nhân viên theo phân bổ KE , $d = (KE, KS)$. Phương án tốt hơn sẽ có giá trị d nhỏ hơn.

4. Số trường hợp gán nhân lực bị đựng độ. Khi so sánh và lựa chọn giải pháp tốt hơn, sự chổng chéo không còn, đảm bảo được về thời gian thực hiện dự án, do tỷ lệ phải điều chỉnh thời gian ở bước sau sẽ giảm đi.
5. Tổng chi phí: phương án tốt hơn sẽ có chi phí thực hiện nhỏ hơn. Khi đạt được phương án tốt nhất theo thuật toán tìm kiếm, ta sẽ đạt được mục tiêu về chi phí thực hiện của bài toán.

Mã giả 4: Tìm kiếm phương án cải tiến cho thuật toán HS

Data: KPI_{target} , $E = \{E_i\}$, $assetKPIWeight$, $T = \{T_i\}$ đã được gán tài sản và thời gian, KS , $lastKPIs$, HS_Args

$inputs \leftarrow \{T, E, lastKPIs, KPI_{target}, KS, assetKPIWeight\}$

$HS_Args \leftarrow \{HMS, bw, HMCR, PAR\}$

Result: *improviseSolution* một phương án lời giải cải tiến

```

1 function getImproviseHSSolution(HM, inputs, HS_Args) :
2   HMCR, PAR, bw  $\leftarrow$  HS_Args;
3   improviseAssignment  $\leftarrow$   $\emptyset$ ;
4   foreach  $T_i \in T$  do
5     get availaleEmps for  $T_i$ ;
6     assignee  $\leftarrow$  get random from availaleEmps;
7     if  $random(0, 1) < HMCR$  then
8       randomSolution  $\leftarrow$  get random solution from HM;
9       randomAssignment  $\leftarrow$  assignment of from randomSolution;
10      assignee  $\leftarrow$  employee assigned for  $T_i$  of randomAssignment;
11      if  $random(0, 1) < PAR$  then
12        newAssignee  $\leftarrow$  member from availaleEmps base on
13          bw, employee;
14        assignee  $\leftarrow$  newAssignee;
15      assign assignee for  $T_i$ ;
16      check and assign optionAssets for  $T_i$ ;
17    Calculate objectives values for improviseAssignment;
18    improviseSolution  $\leftarrow$   $\{improviseAssignment, objectives\}$ ;
19  return improviseSolution

```

Mã giả 4 trình bày về cách tính phương án cải tiến dựa trên bộ nhớ HM cho thuật toán HS.

Mã giả 5: Thuật toán HS

Data: $KPI_{target} = \{K_i\}$, $E = \{E_i\}$, $assetKPIWeight$, $T = \{T_i\}$ đã được
gán tài sản và thời gian, KS , $lastKPIs$, HS_Args

$inputs \leftarrow \{T, E, lastKPIs, KPI_{target}, KS, assetKPIWeight\}$

$HS_Args \leftarrow \{HMS, bw, HMCR, MaxIter, PAR\}$

Result: $solution$, một phương án phân bổ nguồn lực tối ưu cục bộ

```

1 function harmonySearch ( $HS\_Args$ ,  $inputs$ ) :
2    $HM \leftarrow \emptyset$  ;
3    $HMS, maxIter \leftarrow HS\_Args$ ;
4    $initHM(HM, HMS, T, E, lastKPIs, KS, assetKPIWeight)$  ;
5   for  $i = 1$  to  $maxIter$  do
6      $get\ worstSolution, bestSolution$  from  $HM$ ;
7     if  $bestSolution$  is fitness then
8       return  $bestSolutions$ 
9      $improviseSolution \leftarrow getImproviseHSSolution(HM,$ 
        $inputs, HM\_Args)$  ;
10    if  $improviseSolution$  better than  $worstSolution$  then
11      Replace  $worstSolution$  with  $improviseSolution$  in  $HM$ ;
12     $i \leftarrow i + 1$ ;
13  return  $HM[0]$ ;

```

Mã giả 5 mô tả về thuật toán HS áp dụng cho bước hiện tại của bài toán. Hàm $initHM()$ thực hiện khởi tạo HMS lời giải ngẫu nhiên và lưu vào bộ nhớ hài hòa HM . Sau mỗi vòng lặp, một phương án cải tiến được tạo ra qua hàm $getImproveHSSolution()$.

Mã giả 6: Tìm kiếm phương án cải tiến cho thuật toán DLHS

Data: KPI_{target} , $E = \{E_i\}$, $assetKPIWeight$, $T = \{T_i\}$ đã được gán tài sản và thời gian, KS , $lastKPIs$, $DLHS_Args$

$inputs \leftarrow \{T, E, lastKPIs, KPI_{target}, KS, assetKPIWeight\}$

$DLHS_Args \leftarrow \{HMS, bwMax, bwMin, PSLSize, numSub, R, maxFEs\}$

Result: $improviseSolution$ một phương án lời giải cải tiến

```

1 function getImproviseSolution( $subHM, PSL, inputs, FEs$ ):
2    $HMCR, PAR \leftarrow$  select random and remove from  $PSL$ ;
3   Calculate  $bw$  base on  $bwMax, bwMin, FEs, maxFEs$ ;
4    $bestSolution, worstSolution \leftarrow$  best and worst solution from  $subHM$ ;
5    $bestAssignment \leftarrow$  assignment of  $bestSolution$ ;
6    $improviseAssignment \leftarrow \emptyset$ ;
7   foreach  $T_i \in T$  do
8     get  $availaleEmps$  for  $T_i$ ;
9      $assignee \leftarrow$  get random from  $availaleEmps$ ;
10    if  $random(0, 1) < HMCR$  then
11       $assignee \leftarrow$  employee assigned for  $T_i$  of  $bestAssignment$ ;
12      if  $random(0, 1) < PAR$  then
13         $newAssignee \leftarrow$  new member from  $availaleEmps$  base on
14           $bw, employee$ ;
15         $assignee \leftarrow newAssignee$ ;
16      assign  $assignee$  for  $T_i$ ;
17      check and assign  $optionAssets$  for  $T_i$ ;
18    Calculate  $objectives$  values for  $improviseAssignment$ ;
19     $improviseSolution \leftarrow \{improviseAssignment, objectives\}$ ;
20  return  $improviseSolution, HMCR, PAR$ 

```

Mã giả 6 thực hiện tạo một phương án lời giải cải tiến cho thuật toán DLHS.

Mã giả 7 trình bày thuật toán DLHS áp dụng cho bài toán. Tương tự so với thuật toán HS, hàm $initHM()$ để khởi tạo bộ nhớ hài hòa HM với kích thước HMS . Hàm $initPSL()$ dùng để khởi tạo bộ nhớ PSL lưu trữ các cặp tham số ($HMCR, PAR$). Hàm $divideHM()$ chia bộ nhớ hài hòa thành các bộ nhớ con, hay các quần thể động $subHMs$. Giai đoạn tính toán trên các bộ nhớ con, mỗi vòng lặp tạo một phương án lời giải cải tiến thông qua hàm $getImproviseSolution()$. Hàm $reFillPSL()$ thực hiện điền các bộ tham số ($HMCR, PAR$) vào bộ nhớ PSL khi nó rỗng. Hàm $regroupSubHMs()$ thực hiện chia lại các bộ nhớ con (tiểu quần thể)

$subHMs$ theo chu kỳ, để khắc phục nhược điểm tối ưu cục bộ và mất tính đa dạng, giúp trao đổi thông tin giữa các quần thể và tăng độ đa dạng của từng tiểu quần thể. Khi đã đạt đến giới hạn vòng lặp nhất định, hàm $newHMFromSubs()$ sẽ gom nhóm các tiểu quần thể lại để tạo thành bộ nhớ HM mới và thực hiện tìm kiếm hài hòa trên bộ nhớ này, nhằm tối đa hóa việc khai thác cục bộ.

Mã giả 7: Thuật toán DLHS

Data: $KPI_{target} = \{K_i\}$, $E = \{E_i\}$, $assetKPIWeight$, $T = \{T_i\}$ đã được gán tài sản và thời gian, KS , $lastKPIs$, $DLHS_Args$

$inputs \leftarrow \{T, E, lastKPIs, KPI_{target}, KS, assetKPIWeight\}$

$DLHS_Args \leftarrow \{HMS, bwMax, bwMin, PSLSize, numSub, R, maxFEs\}$

Result: $solution$, một phương án phân bổ nguồn lực tối ưu cục bộ

```

1 function DLHS ( $DLHS\_Args$ ,  $inputs$ ) :
2    $FEs \leftarrow 0$ ,  $PSL \leftarrow \emptyset$ ,  $HM \leftarrow \emptyset$ ,  $WPSL \leftarrow \emptyset$ ;
3    $initHM(HM, HMS, T, E, lastKPIs, KS, assetKPIWeight)$ ;
4    $initPSL(PSL, PSLSize)$ ;  $lastPSL \leftarrow PSL$ ;
5    $subHMs \leftarrow divideHM(HM, numOfSub)$ ;
6   while  $FEs < 0.9 * maxFEs$  do
7     foreach  $subHM \in subHMs$  do
8        $improviseSolution \leftarrow getImproviseSolution(subHM,$ 
9          $PLS, inputs, FEs)$ ;  $FEs \leftarrow FEs + 1$ ;
10       $get\ worstSolution, bestSolution$  from  $subHM$ ;
11      if  $bestSolution$  is fitness then
12        return  $bestSolutions$ 
13      if  $improviseSolution$  better than  $worstSolution$  then
14        Replace  $worstSolution$  with  $improviseSolution$  in  $subHM$ ;
15        add  $\{HMCR, PAR\}$  to  $WPSL$ ;
16      if  $PSL == \emptyset$  then
17         $reFillPSL(PSL, WPSL, lastPSL, PSLSize)$ ;
18      if  $FEs \% R == 0$  then
19         $regroupSubHMs(subHMs, numOfSub)$ ;
20     $newHM \leftarrow newHMFromSubs(subHMs, inputs)$ ;
21    while  $FEs < maxFEs$  do
22      Repeat lines 8-16 and do them with  $newHM$ 
23  return  $newHM[0]$ ;

```

Đánh giá độ phức tạp thuật toán: trong cả 2 trường hợp sử dụng thuật toán HS

và DLHS, độ phức tạp thuật toán sẽ được tính như sau: $O(NI \cdot HMS \cdot t)$, với t là thời gian tính toán của việc thực hiện tạo ra một lời giải ngẫu nhiên hoặc lời giải cải tiến. Cụ thể $t = O(n \cdot (e + a) + n \cdot e + n \cdot a) = O(n \cdot (e + a))$ bao gồm việc duyệt qua từng công việc, chọn và gán nhân viên, tài sản tùy chọn (nếu có) cho công việc, và thực hiện tính toán các mục tiêu về chi phí, KPI. Tuy nhiên, mỗi công việc thì số nhân viên thực hiện được và số tài sản tùy chọn là rất ít, nên chi phí thực tế sẽ nhỏ hơn chi phí tồi tệ nhất rất nhiều. Tóm lại, độ phức tạp của bước này là $O(NI \cdot HMS \cdot n \cdot (e + a))$, với NI là số vòng lặp thực hiện.

5.2.5 Bước 4: Xử lý các công việc thực hiện chồng chéo nếu có

Mã giả 8 trình bày cách xử lý các công việc thực hiện chồng chéo.

Mã giả 8: Hàm thực hiện xử lý các công việc chồng chéo nếu có

Data: $assignment = \{T_i \text{ đã được gán thời gian, tài sản, nhân lực} \}$, $T' = T'_i$

là danh sách công việc ngoài dự án

Result: Cập nhật thời gian thực hiện các công việc trong $assignment$

```

1 function reScheduleTasks (assignment) :
2     Sort the assignment by endDate in ascending order;
3     endTimeSaves  $\leftarrow \emptyset$ ; assetAssignments  $\leftarrow \emptyset$ ;
4     foreach  $T_i \in assignment$  do
5         get assignee, assets assigned from  $T_i$ ;
6          $T_i^s \leftarrow \max(T_i^s, \max_{T_j \in T_i^p}(T_j^s))$ ; isConflictOfAsset  $\leftarrow false$ ;
7         Calculate isConflictOfEmp base on endDate of assignee in
            endTimeSaves,  $T_i^s$ ;
8         if isConflictOfEmp then
9              $T_i^s \leftarrow \text{endDate of assignee in } endTimeSaves$ ;
10        foreach  $asset \in assets$  do
11            Calculate isConflictOfAsset base on endTime of asset in
                assetAssignments;
12            if isConflictOfEmp then
13                 $T_i^s \leftarrow \text{endDate of asset in } assetAssignments$ ;
14            Calculate  $T_i^e$  base on  $T_i^s, T_i^d$ ;
15            Update endDate of assignee in endTimeSaves;
16            if  $assets \neq \emptyset$  then
17                foreach  $asset \in assets$  do
18                    Update endDate of asset in assetAssignments;
    
```

Sau khi bước 3 hoàn thành, ta đã có kế hoạch phân bổ nguồn lực và lịch thực hiện công việc thỏa mãn các mục tiêu đề ra. Tuy nhiên, vẫn có thể xảy ra trường hợp một người phải thực hiện nhiều công việc cùng lúc do các công việc ban đầu có thể thực hiện song song. Để giải quyết vấn đề này, ta sẽ điều chỉnh các công việc bị chồng chéo, đồng thời giữ cho tổng thời gian thực hiện dự án là ngắn nhất có thể.

Độ phức tạp thuật toán:

- Việc thực hiện sắp xếp ở dòng 2: $O(n \log n)$
- Lặp lại n vòng lặp, mỗi vòng lặp thực hiện cần $O(p + a)$, với p là số công việc tiền nhiệm của công việc hiện xét, nên sau n vòng lặp thì cần e bước thực hiện, vậy sau n vòng lặp cần $O(e + n \cdot a)$. Vì số tài sản thực hiện ở mỗi công việc rất nhỏ so với tổng số tài sản a , nên độ phức tạp tính toán sẽ nhỏ hơn so với độ phức tạp trên.

Tóm lại độ phức tạp tính toán trong trường hợp tồi tệ nhất của bước trên là: $O(n \log n + n \cdot a + e)$.

5.3 Đánh giá thuật toán

5.3.1 So sánh thuật toán đề xuất với thuật toán đã có

Trong phần này, DATN thực hiện so sánh giữa thuật toán đề xuất so với thuật toán phân bổ nguồn lực thực hiện gói thầu [1].

Về nghiệp vụ thực hiện, kết quả so sánh được để hiện ở **Bảng 5.10**.

	Thuật toán [1]	Thuật toán đề xuất
Phân bổ nguồn lực tối ưu gồm: nhân viên, tài sản	Có	Có
Có yêu cầu về năng lực của nhân viên thực hiện trong công việc	Không có	Có
Có yêu cầu về năng lực của tài sản trong công việc	Không có	Có
Có xử lý khung thời gian trùng lặp thiết bị khi gán công việc	Không có	Có
Có quan tâm đến yếu tố chỉ tiêu KPI của dự án, KPI của nhân viên	Không có	Có

Bảng 5.10: Bảng so sánh về nghiệp vụ thực hiện phân bổ nguồn lực

Về độ phức tạp thuật toán, **Bảng 5.11** ở phía dưới trình bày kết quả so sánh độ phức tạp giữa 2 thuật toán.

	Thuật toán [1]	Thuật toán đề xuất
Bước 1	Sắp topo: $O(n + m)$	Sắp thứ tự, ước tính thời gian thực hiện công việc: $O(n^2 + m) + O(n + m)$
Bước 2	Branch and bound để gán thời gian, tài sản: $O(e^n)$	Gán tài sản, thời gian cho từng công việc: $O(ne + na + m)$
Bước 3	Branch and bound để gán nhân viên: $O(2^n)$	HS, DLHS: $O(IN \cdot HMS \cdot n(e + a))$
Bước 4		Điều chỉnh lại lịch nếu độ: $O(n \log n + na + m)$

Bảng 5.11: Bảng so sánh độ phức tạp thuật toán

Ta có thể thấy độ phức tạp trong trường hợp tồi tệ nhất của thuật toán đề xuất tốt hơn so với thuật toán đã có [1]. Cụ thể, với IN (số vòng lặp), HMS (kích thước bộ nhớ hài hòa) là hằng số, thuật toán đề xuất có độ phức tạp xấp xỉ hàm bậc hai, trong khi đó thuật toán [1] có độ phức tạp lên tới hàm số mũ.

Thứ ba, em cũng thực hiện so sánh với các bộ dữ liệu thử nghiệm. Phạm vi về nghiệp vụ thực hiện để so sánh sẽ gồm mục tiêu tối ưu về thời gian và chi phí, theo thuật toán [1]. **Bảng 5.12** mô tả về bộ dữ liệu để so sánh hai thuật toán.

	Bộ dữ liệu thử nghiệm 1	Bộ dữ liệu thử nghiệm 2	Bộ dữ liệu thử nghiệm 3
Số công việc	8	11	12
Số nhân viên	3	3	4
Số tài sản	5	5	6
Số ràng buộc công việc tiền nhiệm	9	14	15
Mục tiêu tối ưu	Thời gian, chi phí thực hiện	Thời gian, chi phí thực hiện	Thời gian, chi phí thực hiện

Bảng 5.12: Mô tả bộ dữ liệu so sánh 2 thuật toán

Chạy thử các thuật toán độc lập 100 lần, trên cùng môi trường Visual Studio Code, trình biên dịch Node.js v21.2.0, máy tính Ram 16GB, SSD 256GB, ta thu được các kết quả được thể hiện trong **Bảng 5.13** so sánh phía dưới. Các tiêu chí so sánh bao gồm: (i) thời gian thực hiện, (ii) chi phí thực hiện, (iii) thời gian tính toán.

	Bộ dữ liệu	Thuật toán [1]	Thuật toán đề xuất (sử dụng HS)	Thuật toán đề xuất (sử dụng DLHS)
Thời gian thực hiện dự án nhỏ nhất	1	22	22	22
Thời gian thực hiện dự án trung bình	1	22	22	22
Chi phí thực hiện nhỏ nhất	1	3312	3240	3240
Chi phí thực hiện trung bình	1	3312	3240	3240
Thời gian tính toán nhỏ nhất (ms)	1	45,06	243,40	190,12
Thời gian tính toán trung bình (ms)	1	542,69	301,82	204,83
Thời gian thực hiện dự án nhỏ nhất	2	32	28	28
Thời gian thực hiện dự án trung bình	2	32	28	28
Chi phí thực hiện nhỏ nhất	2	4968	4896	4896
Chi phí thực hiện trung bình	2	4968	4902,24	4896.08
Thời gian tính toán nhỏ nhất (ms)	2	706,70	385,23	318,24
Thời gian tính toán trung bình (ms)	2	1293,44	524.21	481,92
Thời gian thực hiện dự án nhỏ nhất	3	35	28	28
Thời gian thực hiện dự án trung bình	3	35	30,26	30.97
Chi phí thực hiện nhỏ nhất	3	5328	5272	5268
Chi phí thực hiện trung bình	3	5328	5338,28	5338,68
Thời gian tính toán nhỏ nhất (ms)	3	917,92	433.59	354,89
Thời gian tính toán trung bình (ms)	3	1381,88	629.67	410,96

Bảng 5.13: Bảng so sánh kết quả thuật toán trên 3 bộ dữ liệu

Từ kết quả so sánh đề cập trong **Bảng 5.13**, có thể thấy, với bộ dữ liệu nhỏ, và các công việc ít phụ thuộc vào nhau, các thuật toán thực hiện cho kết quả tương

đương nhau, thậm chí thuật toán [1] trong trường hợp tốt nhất còn có thời gian tính toán rất nhỏ. Tuy nhiên, khi số lượng công việc và quan hệ ràng buộc tăng lên, thuật toán đề xuất sẽ cho kết quả tối ưu hơn. Việc sử dụng thuật toán HS ở bước 3 thay vì DLHS sẽ tốt đối với những bài toán có dữ liệu nhỏ, thời gian tính toán thấp. Đối với dữ liệu đầu vào lớn, phức tạp, thì thuật toán DLHS đem lại kết quả tốt hơn, thời gian tính toán nhanh hơn, mặc dù kết quả tối ưu có thể kém hơn so với dùng HS một chút nhưng là chấp nhận được.

Đặc biệt, khi tăng số nhân viên, tức số luồng công việc, thuật toán [1] sẽ tồi hơn, chạy lâu hơn do độ phức tạp tăng theo hàm mũ của số luồng công việc.

Tóm lại, thông qua những so sánh đã đề cập thuật toán đề xuất có những ưu điểm hơn so với thuật toán đã có [1].

5.3.2 Thực nghiệm với bộ dữ liệu chuẩn

Bộ dữ liệu chuẩn được lấy trong bài báo trình bày cách đánh giá hiệu quả các giải thuật phân bổ nguồn lực - *Benchmarking and Comparison of Software Project Human Resource Allocation Optimization Approaches* [19]. Bài báo đề cập tới phương pháp để đánh giá hiệu quả của các giải thuật phân bổ nguồn lực, thông qua việc so sánh hiệu quả của một số thuật toán khi thực hiện bài toán phân bổ nguồn lực (nhân viên) thực hiện danh sách các công việc trong dự án với mục tiêu tối ưu theo thời gian thực hiện. Bộ dữ liệu chuẩn bao gồm các thông tin được mô tả trong **Bảng 5.14**.

Vì bài toán đề cập trong bài báo [19] cho phép nhiều nhân viên cùng thực hiện một mô-đun công việc tại một thời điểm, để có thể đưa về ngữ cảnh của bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI của ĐATN, em đã thực hiện chia các mô-đun công việc thành những công việc nhỏ hơn, với thời gian thực hiện như nhau, đồng thời thêm ràng buộc về công việc tiên nhiệm để đảm bảo một nhân viên chỉ thực hiện một công việc tại một thời điểm. Bộ dữ liệu chuẩn sau khi thực hiện chia lại các mô-đun công việc thành các công việc đơn lẻ bao gồm:

- Task: tập công việc, gồm 96 công việc đơn lẻ ứng với 8 mô-đun công việc (bộ dữ liệu Module)
- Resource: 12 nhân viên của bộ dữ liệu chuẩn
- Số ràng buộc công việc tiên nhiệm: 1008

Thực hiện chạy thuật toán đề xuất 100 lần với bộ dữ liệu trên, và thu thập kết quả. Sau đó, so sánh các kết quả này với những kết quả của các thuật toán đã được thực hiện với bộ dữ liệu chuẩn trong bài báo [19]. Với các tiêu chí (i) thời gian thực

Bộ dữ liệu chuẩn		
Tên	Số lượng	Mô tả
Module	8	Bộ dữ liệu về các mô-đun công việc cần thực hiện, mỗi bản ghi gồm các trường thông tin: (i) ID - định danh mô-đun công việc, (ii) Preceedings - công việc tiền nhiệm, (iii) Estimate Effort: ước tính thời gian thực hiện
Resource	12	Bộ dữ liệu về danh sách nguồn lực nhân viên, gồm các trường thông tin: (i) ID - định danh nhân viên, (ii) Name - tên nhân viên, (iii) Productivity - hiệu suất (trong bộ dữ liệu này hiệu suất của tất cả nhân viên đều bằng 1)
Các thông tin khác		
Tên	Giá trị	Mô tả
Min	80.33	Thời gian thực hiện ngắn nhất có thể đạt được theo phân bố
Max	964	Thời gian thực hiện dài nhất theo phân bố

Bảng 5.14: Mô tả bộ dữ liệu chuẩn

hiện dự án nhỏ nhất sau các lần chạy (EPT), (ii) thời gian thực hiện dự án trung bình sau các lần chạy (MEAN), (iii) độ lệch chuẩn giữa các thời gian thực hiện.

	Thuật toán Constraint- based [11]	Thuật toán robust search-based [12]	Thuật toán Search-based [10]	Thuật toán đề xuất
Thời gian dự án nhỏ nhất (EPT)	111.5	80.33	80.6	80.33
Thời gian dự án trung bình (MEAN)	111.5	80.83	85.13	80.33
Độ lệch chuẩn (STDEV)	0	1.139	2.61	0

Bảng 5.15: Bảng so sánh kết quả thực nghiệm với bộ dữ liệu chuẩn

Bảng 5.15 trình bày kết quả so sánh của thuật toán đề xuất với các thuật toán khác trong bài báo [19]. Từ kết quả trên ta có thể thấy thuật toán đề xuất đạt được kết quả tốt nhất có thể từ bộ dữ liệu chuẩn, cũng như mang lại kết quả tối ưu, ổn định hơn so với các thuật toán [11], [12], [10] đã đề cập trong bài báo.

5.3.3 Thực nghiệm với bộ dữ liệu thực tế

Các tiêu chí để đánh giá thuật toán thực hiện bài toán phân bổ nguồn lực trong dự án theo chỉ tiêu KPI được trình bày trong **Bảng 5.16**.

Tiêu chí đánh giá	Ý nghĩa
Thời gian thực hiện dự án	Khoảng thời gian kể từ lúc bắt đầu dự án đến thời điểm tất cả các công việc trong dự án hoàn thành, mong muốn tối thiểu hóa thời gian thực hiện
Chi phí thực hiện dự án	Chi phí tính cho nhân viên, tài sản thực hiện các công việc trong dự án, mong muốn tối thiểu hóa chi phí
Số nhân viên được giao việc	Số nhân viên được giao thực hiện công việc trong dự án, mong muốn tất cả nhân viên đều được giao việc
Số công việc bị xung đột	Số trường hợp mà một nhân viên được giao cho nhiều công việc có thời gian chồng chéo nhau, mong muốn không có trường hợp nào như vậy, nếu có thì xảy ra ít nhất có thể
Số chỉ tiêu KPI đạt theo phân bổ	Số chỉ tiêu KPI trong dự án mà theo kết quả phân bổ vượt giá trị mục tiêu đề ra, mong muốn tất cả các chỉ tiêu KPI đều đạt và có thể vượt ngưỡng đề ra
Khoảng cách giữa KPI của nhân viên dựa trên chia theo năng lực và kết quả phân bổ	Khoảng cách giữa vec-tơ KPI của nhân viên chia theo năng lực và vec-tơ KPI của nhân viên theo phân bổ, khoảng cách càng nhỏ thì việc chia KPI theo năng lực càng đúng đắn
Số nhân viên được giao việc mà họ đã không hoàn thành trước đó	Số nhân viên được giao việc mà họ đã không hoàn thành hoặc bị điều chỉnh công việc tương tự trong quá khứ, mong muốn không có trường hợp nào như vậy

Bảng 5.16: Tiêu chí đánh giá thuật toán

Bộ dữ liệu thử nghiệm bao gồm: (i) thông tin chung về dự án, (ii) danh sách chỉ tiêu KPI, (iii) danh sách nhân viên tham gia dự án, (iv) danh sách tài sản tham gia dự án, (v) danh sách công việc nhân viên đang thực hiện ngoài dự án, (vi) danh sách công việc mà nhân viên đã thực hiện trong quá khứ, và (vii) hiệu suất thực hiện từng công việc của nhân viên được tính thông qua dữ liệu trung gian là hiệu suất thực hiện các công việc tương tự trong quá khứ tương ứng. Chi tiết về bộ dữ liệu thử nghiệm được mô tả trong các bảng dữ liệu ở phần **PHỤ LỤC A**.

Thực hiện thuật toán đề xuất với bộ dữ liệu trên, thu được kết quả phân bổ nguồn

lực được mô tả trong **Bảng 5.17**.

Tên công việc	Ngày bắt đầu	Ngày kết thúc	Nhân viên	Tài sản
Lập tài liệu thuyết trình với các bên liên quan	08:00 01/08/2024	12:00 02/08/2024	Nguyễn Thị Oanh	Macbook Air M1
Phân tích yêu cầu khách hàng	12:00 02/08/2024	12:00 09/08/2024	Nguyễn Văn Minh	PH B1-704
Đề xuất ý cho bài toán phân bổ nguồn lực	12:00 09/08/2024	12:00 12/08/2024	Nguyễn Thị Oanh	Bàn 2
Dựng codebase React-NodeJS	12:00 12/08/2024	12:00 19/08/2024	Nguyễn Minh Thành	Macbook Pro M2
Dựng trang phân bổ nguồn lực dự án	12:00 12/08/2024	12:00 19/08/2024	Nguyễn Văn Biễn	Macbook Air M1
Dựng giao diện module quản lý thẻ	12:00 12/08/2024	12:00 20/08/2024	Nguyễn Phúc Nhật Nam	Macbook Pro M1
Viết kịch bản thử nghiệm cho luồng phân bổ nguồn lực	12:00 12/08/2024	12:00 14/08/2024	Vũ Thị Quỳnh	Bàn 1
Dựng các trang thêm, sửa, xóa dự án	12:00 19/08/2024	12:00 27/08/2024	Nguyễn Văn Biễn	Macbook Pro M2
Lập trình service mô đun quản lý dự án	12:00 20/08/2024	12:00 27/08/2024	Nguyễn Văn Minh	Macbook Pro M1
Ghép API mô đun quản lý dự án	12:00 20/08/2024	12:00 26/08/2024	Đặng Quốc Tú	Macbook Air M1
Kiểm thử luồng phân bổ nguồn lực	12:00 27/08/2024	12:00 02/09/2024	Nguyễn Phúc Nhật Nam	JTest Master
Sửa lỗi các module yêu cầu	12:00 02/09/2024	12:00 09/09/2024	Nguyễn Văn Biễn	Bàn 1
Thu thập phản hồi của khách hàng	12:00 09/09/2024	12:00 11/09/2024	Vũ Thị Quỳnh	PH B1-704
Điều chỉnh chức năng phân bổ nguồn lực	12:00 11/09/2024	12:00 16/09/2024	Nguyễn Văn Biễn	Server 16GB SSD
Triển khai hệ thống DXClan lên server	12:00 16/09/2024	12:00 23/09/2024	Nguyễn Viết Đang	Server 16GB SSD
Lập kế hoạch cho dự án	12:00 12/08/2024	12:00 13/08/2024	Nguyễn Văn Minh	Bàn 2
Tạo kế hoạch kiểm thử các chức năng	12:00 13/08/2024	12:00 14/08/2024	Nguyễn Thị Oanh	Bàn 2
Viết tài liệu công nghệ sử dụng	12:00 23/09/2024	12:00 24/09/2024	Nguyễn Văn Minh	Bàn 2

Bảng 5.17: Kết quả thực hiện phân bổ

Bảng 5.18 tổng hợp lại các tiêu chí đánh giá ứng với kết quả phân bổ trên.

Tiêu chí đánh giá	Kết quả
Thời gian thực hiện dự án	08:00 01/08/2024 đến 12:00 24/09/2024 (54.5 ngày công)
Chi phí thực hiện dự án	86,375,067 VNĐ
Số nhân viên được giao việc	8/8
Số công việc bị xung đột	0/18
Số chỉ tiêu KPI đạt theo phân bổ	3/3
Khoảng cách giữa KPI của nhân viên dựa trên chia theo năng lực và kết quả phân bổ	0.00984
Số nhân viên được giao việc mà họ đã không hoàn thành trước đó	0/8

Bảng 5.18: Kết quả thực hiện phân bổ trên các tiêu chí đánh giá

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Sau quá trình nghiên cứu và phát triển, ĐATN đã phát triển ứng dụng hỗ trợ phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI một cách tự động. Ứng dụng này khắc phục một số hạn chế của các phần mềm hiện tại, bao gồm: (i) xem xét yếu tố KPI trong phân bổ nguồn lực, (ii) mô hình hóa công việc trong dự án với yêu cầu về năng lực nhân viên, yêu cầu về tài sản thực hiện, (iii) phân bổ nguồn lực tự động và lập lịch thực hiện công việc trong dự án.

Về mặt cơ sở lý luận, ĐATN đã nghiên cứu, đề xuất, xây dựng giải pháp cho bài toán phân bổ nguồn lực thực hiện công việc trong dự án theo chỉ tiêu KPI. Giải pháp này được thử nghiệm trên các bộ dữ liệu chuẩn và thực tế, so sánh với thuật toán phân bổ nguồn lực thực hiện gói thầu [1]. Kết quả cho thấy giải pháp của ĐATN tốt hơn về độ phức tạp tính toán, các giá trị mục tiêu tối ưu về chi phí, thời gian và KPI so với các thuật toán hiện có.

Về mặt ứng dụng, ĐATN đã phát triển các chức năng chính bao gồm: (i) quản lý danh sách dự án, (ii) phân bổ nguồn lực dự án, (iii) lập lịch công việc, (iv) quản lý thẻ công việc, (v) quản lý bộ năng lực nhân viên. Các chức năng này đã được tích hợp thành công vào hệ thống DXClan [2].

Tuy nhiên, do hạn chế về mặt thời gian cũng như năng lực, nên sản phẩm ĐATN của em còn nhiều tồn tại và thiếu sót, kính mong quý thầy, cô đưa ra những nhận xét đánh giá để sản phẩm ngày càng hoàn thiện hơn. Trong quá trình làm ĐATN, em đã học hỏi được rất nhiều kiến thức về quy trình, nghiệp vụ phân bổ nguồn lực, quản lý dự án cũng như các kiến thức về công nghệ, phát triển ứng dụng. Điều này giúp đem lại thêm nhiều kiến thức, trải nghiệm và sẽ là hành trang cho em trong tương lai.

6.2 Hướng phát triển

Một số hướng phát triển cho ĐATN được đề xuất sẽ thực hiện trong tương lai bao gồm: (i) mở rộng thêm các điều kiện, ràng buộc về yêu cầu nhân lực, tài sản trong công việc (ii) xây dựng được thuật toán có thể thực hiện phân bổ nguồn lực thực hiện công việc trong nhiều dự án đồng thời theo chỉ tiêu KPI, (iii) tự động hóa thêm các bước trong quy trình nghiệp vụ thực hiện phân bổ nguồn lực của ứng dụng hiện có, (iv) cải thiện trải nghiệm người dùng: nâng cao, cải thiện giao diện người dùng để sử dụng, thân thiện hơn.

TÀI LIỆU THAM KHẢO

- [1] T.-H.-G. Vu and V.-D. Nguyen, “Efficient resource allocation for multiple bid packages and projects in enterprise,” *ICISN 24, Paper 88*, 2023.
- [2] VNIST, C. ty CP Công nghệ An toàn thông tin và truyền thông Việt Nam, *DX - Giải pháp không gian làm việc số*, <https://dxclan.com/>, 2024.
- [3] Meta Platforms, Inc., *React Documentation - Learn React*. [Online]. Available: <https://react.dev/learn> (visited on 06/09/2024).
- [4] Redux, *Redux Documentation*. [Online]. Available: <https://redux.js.org/introduction/getting-started> (visited on 06/09/2024).
- [5] OpenJS Foundation, *Node.js Documentation*. [Online]. Available: <https://nodejs.org/en/docs> (visited on 06/09/2024).
- [6] NodeJS Foundation, *Express.js Documentation*. [Online]. Available: <https://expressjs.com/en/starter/installing.html> (visited on 06/09/2024).
- [7] MongoDB, Inc., *MongoDB Documentation*. [Online]. Available: <https://docs.mongodb.com> (visited on 06/09/2024).
- [8] J. wen Huang, X. xia Wang, and R. Chen, “Genetic algorithms for optimization of resource allocation in large scale construction project management,” *J. Comput.*, vol. 5, pp. 1916–1924, 2010. [Online]. Available: <https://api.semanticscholar.org/CorpusID:7618034>.
- [9] S. Chaharsooghi and A. H. Meimand Kermani, “An effective ant colony optimization algorithm (aco) for multi-objective resource allocation problem (morap),” *Applied Mathematics and Computation*, vol. 200, no. 1, pp. 167–177, 2008, ISSN: 0096-3003. DOI: <https://doi.org/10.1016/j.amc.2007.09.070>.
- [10] G. Antoniol, M. Di Penta, and M. Harman, “Search-based techniques applied to optimization of project planning for a massive maintenance project,” vol. 2005, Oct. 2005, pp. 240–249, ISBN: 0-7695-2368-4. DOI: 10.1109/ICSM.2005.79.
- [11] D. Kang, J. Jung, and D.-H. Bae, “Constraint-based human resource allocation in software projects,” *Softw., Pract. Exper.*, vol. 41, pp. 551–577, Apr. 2011. DOI: 10.1002/spe.1030.
- [12] G. Antoniol, M. Di Penta, and M. Harman, “A robust search-based approach to project management in the presence of abandonment, rework, error and uncertainty,” in *10th International Symposium on Software Metrics, 2004. Proceedings.*, 2004, pp. 172–183. DOI: 10.1109/METRIC.2004.1357901.

- [13] TailwindCSS, *TailwindCSS Documentation*. [Online]. Available: <https://tailwindcss.com/docs/utility-first> (visited on 06/09/2024).
- [14] Mongoose, *Mongoose Documentation Guide*. [Online]. Available: <https://mongoosejs.com/docs> (visited on 06/09/2024).
- [15] S. W. Kim and H. S. Park, “Periodic task scheduling algorithm under precedence constraint based on topological sort,” in *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2011, pp. 727–729. DOI: 10.1109/URAI.2011.6145920.
- [16] ProjectManager, *Critical path method (cpm) - the ultimate guide to project planning*, Accessed: 2024-06-18, 2023. [Online]. Available: <https://www.projectmanager.com/guides/critical-path-method>.
- [17] X. Z. Gao, V. Govindasamy, H. Xu, X. Wang, and K. Zenger, “Harmony search method: Theory and applications,” *Computational Intelligence and Neuroscience*, vol. 2015, no. 1, p. 258491, 2015. DOI: <https://doi.org/10.1155/2015/258491>.
- [18] Q.-K. Pan, P. N. Suganthan, J. J. Liang, and M. F. Tasgetiren, “A local-best harmony search algorithm with dynamic subpopulations,” en, *Engineering Optimization*, vol. 42, no. 2, pp. 101–117, Feb. 2010, ISSN: 0305-215X, 1029-0273. DOI: 10.1080/03052150903104366.
- [19] S. Al Khatib and J. Noppen, “Benchmarking and comparison of software project human resource allocation optimization approaches,” *ACM SIGSOFT Software Engineering Notes*, vol. 41, pp. 1–6, Jan. 2017. DOI: 10.1145/3011286.3011305.

PHỤ LỤC

CHƯƠNG A. BỘ DỮ LIỆU THỬ NGHIỆM THỰC

Bảng A.1, A.2 chứa thông tin chung và các chỉ tiêu KPI của dự án.

ID dự án	Tên dự án	Mô tả	Thời gian bắt đầu dự kiến	Thời gian kết thúc dự kiến
1	Dự án prj-proposal	Dự án phát triển phần mềm hỗ trợ phân bổ nguồn lực	08:00 01/08/2024	08:00 01/10/2024

Bảng A.1: Thông tin chung về dự án

ID KPI	Tên KPI	Tiêu chí đánh giá	Đơn vị giá trị	Tổng giao	Giá trị cần đạt	Giá trị KPI mục tiêu
1	Phân tích, phát triển chức năng, mô-đun	Số chức năng hoàn thiện đúng hạn	Chức năng	50	44	0.88
2	Chất lượng sản phẩm	Số trường hợp kiểm thử được thực hiện mà không gặp lỗi	Testcase	200	182	0.91
3	Tài liệu và các yêu cầu khác	Số tài liệu và sản phẩm giao nộp đúng hạn	Tài liệu, sản phẩm	100	91	0.91

Bảng A.2: Danh sách chỉ tiêu KPI của dự án

Bảng A.3, A.4 mô tả bộ dữ liệu chứa thông tin về danh sách công việc trong dự án.

ID công việc	Mã công việc	Tên công việc	Công việc tiên nhiệm	Ước tính thời gian (ngày)	Tags	KPI liên quan	Trọng số
1	A1	Phân tích yêu cầu khách hàng	F1	7	analysis	Phân tích, phát triển chức năng, mô-đun	0.228
2	A21	Dựng codebase React-NodeJS	E1	7	backend, frontend	Phân tích, phát triển chức năng, mô-đun	0.086
3	A22	Dựng trang phân bổ nguồn lực dự án	E1	6	frontend	Phân tích, phát triển chức năng, mô-đun	0.086

Bảng A.3: Danh sách công việc của dự án (01)

ID công việc	Mã công việc	Tên công việc	Công việc tiền nhiệm	Ước tính thời gian (ngày)	Tags	KPI liên quan	Trọng số
4	A23	Dựng giao diện module quản lý thẻ	E1	8	frontend	Phân tích, phát triển chức năng, mô-đun	0.114
5	A3	Triển khai hệ thống DXClan lên server	D2	5	devops	Phân tích, phát triển chức năng, mô-đun	0.171
6	B1	Tạo kế hoạch kiểm thử các chức năng	C1	1	testing	Chất lượng sản phẩm	0.143
7	B2	Kiểm thử luồng phân bổ nguồn lực	A21, A22, A23, B4, A24, A25, A26	5	testing	Chất lượng sản phẩm	0.143
8	B3	Sửa lỗi các module yêu cầu	B2	7	frontend, backend	Chất lượng sản phẩm	0.286
9	B4	Viết kịch bản thử nghiệm cho luồng phân bổ nguồn lực	E1	2	testing	Chất lượng sản phẩm	0.143
10	C1	Lập kế hoạch cho dự án		1	planning	Chất lượng sản phẩm	0.285
11	D1	Thu thập phản hồi khách hàng	B3	2	collecting_data	Tài liệu và các yêu cầu khác	0.333
12	D2	Điều chỉnh chức năng phân bổ nguồn lực	D1	4	backend, frontend	Tài liệu và các yêu cầu khác	0.167
13	E1	Đề xuất ý cho bài toán phân bổ nguồn lực	A1	2	idea	Tài liệu và các yêu cầu khác	0.166
14	F1	Lập tài liệu hợp với các bên liên quan		1.5	document	Tài liệu và các yêu cầu khác	0.167
15	F2	Viết tài liệu công nghệ sử dụng	A3	1	document	Tài liệu và các yêu cầu khác	0.167
16	A24	Dựng các trang thêm, sửa, xóa dự án	A21, E1	8	frontend	Phân tích, phát triển chức năng, mô-đun	0.143
17	A24	Lập trình service mô-đun quản lý dự án	A22, E1	7	backend	Phân tích, phát triển chức năng, mô-đun	0.086
18	A26	Ghép API mô-đun quản lý dự án	A23, E1	6	frontend, backend	Phân tích, phát triển chức năng, mô-đun	0.086

Bảng A.4: Danh sách công việc của dự án (02)

Bảng A.5, A.6 mô tả bộ dữ liệu chứa thông tin yêu cầu nhân lực, yêu cầu tài sản tương ứng từng công việc trong dự án.

ID công việc	Mã công việc	Tên công việc	Yêu cầu năng lực nhân viên	Yêu cầu tài sản	Số lượng	Năng lực sử dụng	Bắt buộc
1	A1	Phân tích yêu cầu khách hàng	Tiếng Anh: A2 Kinh nghiệm: 1-2 Năm	Phòng họp	1	Tốt	Có
2	A21	Dựng codebase React-NodeJS	Backend: Thành thạo Frontend: Đã làm Docker: Đã làm Kinh nghiệm: Dưới 1 năm	Laptop	1	Tốt	Có
3	A22	Dựng trang phân bổ nguồn lực dự án	Xếp hạng tốt nghiệp: Khá Frontend: Thành thạo	Laptop	1	Bình thường	Có
4	A23	Dựng giao diện module quản lý thẻ	Backend: Đã làm Frontend: Thành thạo Kinh nghiệm: Dưới 1 năm	Laptop	1	Tốt	Có
5	A3	Triển khai hệ thống DXClan lên server	CI_CD: Thành thạo Docker: Thành thạo Kinh nghiệm: Dưới 1 năm	Server	1	Tốt	Có
6	B1	Tạo kế hoạch kiểm thử các chức năng	Kiểm thử thủ công: Thành thạo Kiểm thử đơn vị: Đã làm	Bàn	1	Bình thường	Không
7	B2	Kiểm thử luồng phân bổ nguồn lực	Kiểm thử thủ công: Đã làm Kiểm thử đơn vị: Đã làm Kiểm thử tự động: Đã làm	Công cụ hỗ trợ	1	Tốt	Có
8	B3	Sửa lỗi các mô-đun yêu cầu	Backend: Thành thạo Frontend: Đã làm Kinh nghiệm: Dưới 1 năm	Bàn	1	Tốt	Có
9	B4	Viết kịch bản thử nghiệm cho luồng phân bổ nguồn lực	Xếp hạng tốt nghiệp: Khá Kiểm thử đơn vị: Thành thạo	Bàn	1	Bình thường	Có
10	B4	Viết kịch bản thử nghiệm cho luồng phân bổ nguồn lực	Xếp hạng tốt nghiệp: Khá Kiểm thử đơn vị: Thành thạo	Bàn	1	Bình thường	Có
11	C1	Lập kế hoạch cho dự án	Xếp hạng tốt nghiệp: Khá Kinh nghiệm: 2-5 năm	Bàn	1	Bình thường	Có

Bảng A.5: Yêu cầu về nhân lực và tài sản tương ứng của các công việc (01)

ID công việc	Mã công việc	Tên công việc	Yêu cầu năng lực nhân viên	Yêu cầu tài sản	Số lượng	Năng lực sử dụng	Bắt buộc
12	D1	Thu thập phản hồi của khách hàng	Tiếng Anh: C1	Phòng họp	1	Bình thường	Có
13	D2	Điều chỉnh chức năng phân bổ nguồn lực	Kinh nghiệm: 1-2 năm CI_CD: Thành thạo Backend: Thành thạo	Server	1	Bình thường	Có
14	E1	Đề xuất ý tưởng cho bài toán phân bổ nguồn lực	Kinh nghiệm: 1-2 năm Tiếng Anh: B1	Bàn	1	Bình thường	Có
15	F2	Viết tài liệu công nghệ sử dụng	Kinh nghiệm: 1-2 năm Backend: Thành thạo Frontend: Thành thạo Docker: Thành thạo CD_CD: Thành thạo	Bàn	1	Bình thường	Có
16	A24	Dựng các trang thêm, sửa, xóa dự án	Kinh nghiệm: Dưới 1 năm Backend: Đã làm Frontend: Thành thạo	Laptop	1	Tốt	Có
17	A25	Lập trình service mô đun quản lý dự án	Backend: Thành thạo Frontend: Thành thạo	Laptop	1	Tốt	Có
18	A26	Ghép API mô-đun quản lý dự án	Backend: Đã làm Frontend: Thành thạo Kiểm thử thủ công: Đã làm	Laptop	1	Bình thường	Có

Bảng A.6: Yêu cầu về nhân lực và tài sản tương ứng của các công việc (02)

Bảng A.7 chứa bộ dữ liệu về nhân viên tham gia dự án.

ID	Tên nhân viên	Lương cơ bản	Năng lực nhân viên
1	Nguyễn Văn Minh	40,000,000	Xếp hạng tốt nghiệp: Xuất sắc, Kinh nghiệm: Trên 10 năm, Tiếng Anh: B2, Backend: Bậc thầy, Frontend: Chuyên nghiệp, Docker: Chuyên gia, CD_CD: Chuyên gia, Kiểm thử đơn vị: Đã làm, Kiểm thử thủ công: Đã làm, Kiểm thử tự động: Chuyên nghiệp
2	Nguyễn Văn Biển	30,000,000	Xếp hạng tốt nghiệp: Giỏi, Kinh nghiệm: 5-10 năm, Tiếng Anh: B1, Backend: Chuyên gia, Frontend: Chuyên nghiệp, Docker: Chuyên gia, CD_CD: Chuyên nghiệp, Kiểm thử đơn vị: Thành thạo, Kiểm thử thủ công: Đã làm, Kiểm thử tự động: Chuyên gia
3	Nguyễn Việt Đang	19,500,000	Xếp hạng tốt nghiệp: Khá, Kinh nghiệm: 1-2 năm, Tiếng Anh: B1, Backend: Chuyên nghiệp, Frontend: Thành thạo, Docker: Thành thạo, CD_CD: Thành thạo, Kiểm thử đơn vị: Đã làm, Kiểm thử thủ công: Đã làm, Kiểm thử tự động: Chuyên nghiệp
4	Nguyễn Phúc Nhật Nam	21,500,000	Xếp hạng tốt nghiệp: Giỏi, Kinh nghiệm: 1-2 năm, Tiếng Anh: B2, Backend: Thành thạo, Frontend: Chuyên nghiệp, Docker: Thành thạo, CD_CD: Thành thạo, Kiểm thử đơn vị: Thành thạo, Kiểm thử thủ công: Đã làm, Kiểm thử tự động: Thành thạo
5	Nguyễn Thị Oanh	13,000,000	Xếp hạng tốt nghiệp: Khá, Kinh nghiệm: 1-2 năm, Tiếng Anh: C1, Kiểm thử đơn vị: Chuyên gia, Kiểm thử thủ công: Bậc thầy, Kiểm thử tự động: Chuyên nghiệp
6	Vũ Thị Quỳnh	21,000,000	Xếp hạng tốt nghiệp: Khá, Kinh nghiệm: 2-5 năm, Tiếng Anh: C2, Kiểm thử đơn vị: Chuyên nghiệp, Kiểm thử thủ công: Chuyên gia, Kiểm thử tự động: Thành thạo
7	Nguyễn Minh Thành	10,500,000	Xếp hạng tốt nghiệp: Khá, Kinh nghiệm: Dưới 1 năm, Tiếng Anh: B1, Backend: Thành thạo, Frontend: Thành thạo, Docker: Thành thạo, CD_CD: Đã làm, Kiểm thử đơn vị: Thành thạo, Kiểm thử thủ công: Đã làm
8	Đặng Quốc Tú	11,000,000	Xếp hạng tốt nghiệp: Khá, Kinh nghiệm: Dưới 1 năm, Tiếng Anh: B2, Backend: Đã làm, Frontend: Chuyên nghiệp, Docker: Đã làm, Kiểm thử đơn vị: Thành thạo, Kiểm thử thủ công: Đã làm

Bảng A.7: Danh sách nhân viên tham gia dự án

Bảng A.8 chứa bộ dữ liệu tài sản tham gia dự án.

ID	Loại tài sản	Tên tài sản	Chi phí vận hành (theo giờ)	Khả năng sử dụng	Trạng thái	Thời gian sử dụng
1	Laptop	Macbook Air M1	10,000	Bình thường	Rảnh rỗi	
2	Laptop	Macbook Pro M1	15,000	Tốt	Đang dùng	15:00 01/07/2024 - 15:00 20/07/2024
						10:00 21/07/2024 - 10:00 01/08/2024
						10:00 01/07/2024 - 15:00 10/08/2024
3	Laptop	Macbook Pro M2	25,000	Rất tốt	Đang dùng	08:00 01/07/2024 - 09:00 13/07/2024
						09:00 13/07/2024 - 15:00 25/07/2024
						09:00 26/07/2024 - 15:00 08/08/2024
4	Server	Server 12GB SSD	20,000	Tốt	Đang dùng	15:00 01/07/2024 - 15:00 15/07/2024
						08:00 16/07/2024 - 15:00 18/08/2024
						08:00 19/08/2024 - 15:00 24/09/2024
5	Server	Server 16GB SSD	25,000	Tốt	Rảnh rỗi	
6	Phòng họp	PH B1-704	25,000	Rất tốt	Rảnh rỗi	
7	Bàn	Bàn 1	10,000	Tốt	Rảnh rỗi	
8	Bàn	Bàn 2	8,000	Bình thường	Rảnh rỗi	
9	Bàn	Bàn 3	15,000	Rất tốt	Đang dùng	13:00 01/07/2024 - 12:00 12/08/2024
						13:00 12/08/2024 - 12:00 08/09/2024
10	Công cụ hỗ trợ	JTest Master	1,000	Rất tốt	Rảnh rỗi	

Bảng A.8: Danh sách tài sản tham gia dự án

Bảng A.9 chứa thông tin về danh sách công việc ngoài dự án mà nhân viên đang được phân công thực hiện. Dữ liệu này dùng để đánh dấu những khoảng thời gian mà nhân viên không thể thực hiện công việc trong dự án bị xung đột với khoảng thời gian đó.

PHỤ LỤC A. BỘ DỮ LIỆU THỬ NGHIỆM THỰC

STT	Tên nhân viên	Công việc	Thời gian bắt đầu	Thời gian kết thúc
1	Nguyễn Văn Minh	Phân tích nhu cầu học ngoại ngữ của sinh viên	15:00 27/07/2024	11:00 01/08/2024
2		Phân tích yêu cầu sắp xếp lịch giảng dạy	15:00 30/09/2024	17:00 03/10/2024
3		Lập kế hoạch cho dự án SSLG	08:00 04/10/2024	17:00 06/10/2024
4	Nguyễn Văn Biển	Dựng code base Java core	09:00 27/07/2024	17:00 01/08/2024
5		Sửa lỗi chức năng Thêm vào giỏ	08:00 02/08/2024	15:00 03/08/2024
6		Triển khai mô-đun quản lý thuê lên server	15:00 03/08/2024	17:00 08/08/2024
7		Dựng trang giới thiệu sản phẩm	13:00 24/09/2024	17:00 24/09/2024
8	Nguyễn Viết Đang	Dựng giao diện mô-đun mua hàng	08:00 27/07/2024	15:00 01/08/2024
9		Lập trình Infra	13:00 04/09/2024	13:00 08/09/2024
10		Kiểm thử luồng mua hàng	13:00 08/09/2024	17:00 10/09/2024
11		Đề xuất ý tưởng cho giao diện mới	13:00 23/09/2024	17:00 24/09/2024
12	Nguyễn Phúc Nhật Nam	Sửa lỗi mô-đun quản lý thuế TNCN	12:00 09/08/2024	12:00 11/08/2024
13		Triển khai ứng dụng trên github	13:00 23/09/2024	17:00 24/09/2024
14	Nguyễn Thị Oanh	Viết kịch bản kiểm thử tích hợp các mô-đun	08:00 17/08/2024	17:00 21/08/2024
15	Vũ Thị Quỳnh	Xây dựng kịch bản thử nghiệm cho luồng quản lý hóa đơn	08:00 20/08/2024	17:00 24/08/2024
16	Nguyễn Minh Thành	Dựng giao diện mô-đun đặt lịch	13:00 28/07/2024	17:00 03/08/2024
17		Lập trình trang quản lý phân quyền	08:00 04/08/2024	17:00 08/08/2024
18		Đọc hiểu code base	08:00 09/08/2024	17:00 11/08/2024
19		Ghép API mô-đun bán hàng	08:00 24/09/2024	17:00 02/10/2024
20		Lập trình trang quảng bá, giới thiệu sản phẩm	08:00 04/10/2024	17:00 09/10/2024
21	Đặng Quốc Tú	Dựng các màn hình chatbot	11:00 29/07/2024	17:00 02/08/2024
22		Lập trình service chat với socket	08:00 09/08/2024	17:00 11/08/2024
23		Dựng các màn hình trò chuyện	08:00 24/08/2024	17:00 29/08/2024
24		Kiểm thử chức năng tạo cuộc họp	08:00 30/08/2024	17:00 01/09/2024
25		Dựng các màn hình mô-đun quản lý lớp học	09:00 17/09/2024	15:00 22/09/2024

Bảng A.9: Danh sách công việc mà các nhân viên đang thực hiện ngoài dự án

Bảng A.10, A.11, A.12, A.13, A.14 là bộ dữ liệu danh sách công việc từng nhân viên đã thực hiện trong quá khứ. Bộ dữ liệu này phục vụ tính giá trị hiệu suất thực hiện công việc trong dự án của từng nhân viên.

PHỤ LỤC A. BỘ DỮ LIỆU THỬ NGHIỆM THỰC

STT	Nhân viên	Công việc	Tags	Thời gian thực hiện	Yêu cầu năng lực nhân viên	Điểm
1	Nguyễn Văn Minh	Phân tích quy trình nghiệp vụ khám bệnh	analysis	10:00 10/06/2024 - 17:00 17/06/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: A2	0.9
2		Viết code base Java-Angular	backend, frontend	8:00 18/06/2024 - 14:00 24/06/2024	Kinh nghiệm: 1-2 năm Backend: Thành thạo Frontend: Đã làm Docker: Đã làm	0.9
3		Dựng trang homepage	frontend	14:00 24/06/2024 - 14:00 27/06/2024	Xếp hạng tốt nghiệp: Khá Frontend: Thành thạo Kinh nghiệm: Dưới 1 năm	0.95
4		Đề xuất chiến lược kinh doanh gạch lát	idea	14:00 27/06/2024 - 17:00 29/06/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: B1	0.95
5		Lập kế hoạch cho dự án DCMA	planning	08:00 30/06/2024 - 08:00 02/07/2024	Xếp hạng tốt nghiệp: Khá Kinh nghiệm: 2-5 năm	0.9
6		Dựng giao diện module quản lý thầu	frontend	08:00 02/07/2024 - 12:00 08/07/2024	Kinh nghiệm: Dưới 1 năm Backend: Đã làm Frontend: Thành thạo	0.9
7		Lập trình phía server module quản lý thầu	backend	13:00 08/07/2024 - 17:00 14/07/2024	Kinh nghiệm: Dưới 1 năm Backend: Thành thạo Frontend: Thành thạo	0.95
8		Tách service quản lý thầu	frontend, backend	08:00 15/07/2024 - 09:00 19/07/2024	Backend: Thành thạo Frontend: Thành thạo Kiểm thử thủ công: Đã làm	0.85
9		Kiểm thử luồng quản lý tài liệu	testing	10:00 19/07/2024 - 17:00 21/07/2024	Kiểm thử thủ công: Đã làm Kiểm thử đơn vị: Đã làm Kiểm thử tự động: Đã làm	0.8
10		Điều chỉnh module quản lý thầu	backend, frontend	08:00 21/07/2024 - 12:00 25/07/2024	Kinh nghiệm: 1-2 năm Backend: Thành thạo Frontend: Đã làm CI_CD: Thành thạo	0.95
11		Viết tài liệu công nghệ sử dụng	document	13:00 25/07/2024 - 17:00 26/07/2024	Kinh nghiệm: 1-2 năm Backend: Thành thạo Frontend: Thành thạo Docker: Thành thạo CI_CD: Thành thạo	0.95

Bảng A.10: Danh sách công việc mà các nhân viên đã thực hiện trong quá khứ (1)

STT	Nhân viên	Công việc	Tags	Thời gian thực hiện	Yêu cầu năng lực nhân viên	Điểm
12	Nguyễn Văn Biển	Lập trình micro-service module quản lý sách	frontend, backend	09:00 10/06/2024 - 12:00 13/06/2024	Backend: Đã làm Frontend: Thành thạo Kiểm thử thủ công: Đã làm	-1
13		Phân tích quá trình nhập kho	analysis	13:00 13/06/2024 - 17:00 16/06/2024	Kinh nghiệm: Dưới 1 năm Tiếng Anh: A2	0.96
14		Đề xuất chiến lược mua hàng	idea	08:00 17/06/2024 - 09:00 20/06/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: B1	0.95
15		Viết kịch bản kiểm thử cho luồng nhập kho	testing	12:00 18/06/2024 - 17:00 24/06/2024	Xếp hạng tốt nghiệp: Khá Kiểm thử đơn vị: Thành thạo	0.9
16		Dựng trang landingpage	frontend	09:00 20/06/2024 - 12:00 23/06/2024	Xếp hạng tốt nghiệp: Khá Frontend: Thành thạo	0.9
17		Dựng code base Laravel-React	backend, frontend	13:00 23/06/2024 - 17:00 29/06/2024	Kinh nghiệm: 1-2 năm Backend: Thành thạo Frontend: Đã làm Docker: Đã làm	0.95
18		Lập trình phía server module quản lý sách	backend	10:00 01/07/2024 - 12:00 05/07/2024	Backend: Thành thạo Frontend: Thành thạo	0.95
19		Dựng giao diện module quản lý nhân sự	frontend	13:00 05/07/2024 - 14:00 09/07/2024	Xếp hạng tốt nghiệp: TB Kinh nghiệm: Dưới 1 năm Backend: Đã làm Frontend: Thành thạo	0.9
20		Triển khai module đầu tàu	devops	14:00 09/07/2024 - 17:00 13/07/2024	Kinh nghiệm: Dưới 1 năm Docker: Thành thạo CI_CD: Thành thạo	0.9
21		Sửa lỗi module quản lý phòng khách sạn	frontend, backend	08:00 14/07/2024 - 12:00 19/07/2024	Kinh nghiệm: Dưới 1 năm Backend: Thành thạo Frontend: Đã làm	0.9
22		Điều chỉnh module phân quyền	backend, frontend	13:00 19/07/2024 - 13:00 24/07/2024	Kinh nghiệm: 1-2 năm Backend: Thành thạo CI_CD: Thành thạo	0.95
23		Viết tài liệu hướng dẫn cài đặt	document	13:00 24/07/2024 - 17:00 26/07/2024	Kinh nghiệm: 1-2 năm Backend: Thành thạo Frontend: Thành thạo Docker: Thành thạo CI_CD: Thành thạo	0.95

Bảng A.11: Danh sách công việc mà các nhân viên đã thực hiện trong quá khứ (2)

PHỤ LỤC A. BỘ DỮ LIỆU THỬ NGHIỆM THỰC

STT	Nhân viên	Công việc	Tags	Thời gian thực hiện	Yêu cầu năng lực nhân viên	Điểm
24	Nguyễn Viết Đang	Phân tích tính khả thi dự án	analysis	08:00 13/06/2024 - 10:00 17/06/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: A2	0.85
25		Đề xuất chiến lược nhập kho	idea	08:00 18/06/2024 - 10:00 22/06/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: B1	0.9
26		Dựng code base NodeJS-React	backend, frontend	10:00 22/06/2024 - 12:00 29/06/2024	Backend: Thành thạo Frontend: Đã làm Docker: Đã làm	0.95
27		Lập trình service phân quyền	backend	14:00 29/06/2024 - 14:00 02/07/2024	Backend: Thành thạo Frontend: Đã làm	0.9
28		Kiểm thử chức năng nhập kho	testing	14:00 02/07/2024 - 17:00 04/07/2024	Kiểm thử thủ công: Đã làm Kiểm thử đơn vị: Thành thạo Kiểm thử tự động: Đã làm	0.9
29		Thu thập phản hồi của đối tác	collecting data	08:00 05/07/2024 - 17:00 11/07/2024	Xếp hạng tốt nghiệp: TB Tiếng Anh: 5	0.9
30		Điều chỉnh module quản lý nhân viên	backend, frontend	08:00 12/07/2024 - 10:00 17/07/2024	Kinh nghiệm: 1-2 năm Backend: Thành thạo CD_CD: Thành thạo	0.85
31		Triển khai module thanh toán	devops	10:00 17/07/2024 - 12:00 22/07/2024	Kinh nghiệm: Dưới 1 năm Docker: Thành thạo CD_CD: Thành thạo Backend: Đã làm	0.9
32	Nguyễn Minh Thành	Viết tài liệu triển khai	document	13:00 22/07/2024 - 17:00 24/07/2024	Kinh nghiệm: 1-2 năm Backend: Thành thạo Frontend: Thành thạo Docker: Thành thạo CD_CD: Thành thạo	0.9
33		Dựng code base NestJS-React	backend, frontend	08:00 05/07/2024 - 17:00 11/07/2024	Backend: Thành thạo Frontend: Đã làm Docker: Đã làm	0.9
34		Dựng trang chủ, thống kê phía admin	frontend	08:00 12/07/2024 - 12:00 18/07/2024	Xếp hạng tốt nghiệp: Khá Frontend: Thành thạo	0.8
35		Lập trình service module quản lý thẻ	backend	13:00 18/07/2024 - 13:00 22/07/2024	Backend: Thành thạo Frontend: Thành thạo	-1

Bảng A.12: Danh sách công việc mà các nhân viên đã thực hiện trong quá khứ (3)

PHỤ LỤC A. BỘ DỮ LIỆU THỬ NGHIỆM THỰC

STT	Nhân viên	Công việc	Tags	Thời gian thực hiện	Yêu cầu năng lực nhân viên	Điểm
36	Nguyễn Thị Oanh	Tạo kế hoạch kiểm thử đơn vị	testing	10:00 02/07/2024 - 10:00 07/07/2024	Kiểm thử thủ công: Thành thạo Kiểm thử đơn vị: Đã làm Kiểm thử tự động: Đã làm	0.95
37		Phân tích và tư vấn mua hàng cho KH	analysis	08:00 28/06/2024 - 10:00 02/07/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: A2	0.75
38		Kiểm thử tính năng thêm vào giỏ	testing	10:00 07/07/2024 - 12:00 09/07/2024	Kiểm thử thủ công: Đã làm Kiểm thử đơn vị: Đã làm Kiểm thử tự động: Đã làm	0.95
39		Viết kịch bản kiểm thử cho luồng thanh toán	testing	13:00 09/07/2024 - 17:00 12/07/2024	Xếp hạng tốt nghiệp: Khá Kiểm thử đơn vị: Thành thạo Kinh nghiệm: Dưới 1 năm	0.95
40		Đề xuất chiến lược kinh doanh trực tuyến	idea	08:00 13/07/2024 - 12:00 17/07/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: B1	0.8
41		Lập tài liệu cuộc họp với khách hàng	document	13:00 17/07/2024 - 13:00 20/07/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: C1	0.8
42	Vũ Thị Quỳnh	Phân tích nhu cầu học tiếng Anh	analysis	08:00 09/06/2024 - 17:00 12/06/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: A2	0.95
43		Lập kế hoạch cho dự án	planning	08:00 20/06/2024 - 12:00 24/06/2024	Xếp hạng tốt nghiệp: Khá Kinh nghiệm: 2-5 năm	0.85
44		Lập tài liệu cuộc họp với đối tác	document	13:00 24/06/2024 - 12:00 27/06/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: C1	0.9
45		Tạo kế hoạch kiểm thử tích hợp	testing	13:00 27/06/2024 - 17:00 01/07/2024	Kiểm thử thủ công: Thành thạo Kiểm thử đơn vị: Đã làm	0.9
46		Viết kịch bản kiểm thử cho luồng đặt hàng	testing	08:00 02/07/2024 - 08:00 04/07/2024	Xếp hạng tốt nghiệp: Khá Kiểm thử đơn vị: Thành thạo	0.95
47		Kiểm thử tính năng Thêm chứng chỉ	testing	08:00 04/07/2024 - 12:00 05/07/2024	Kiểm thử thủ công: Đã làm Kiểm thử đơn vị: Đã làm Kiểm thử tự động: Đã làm	0.8
48		Thu thập phản hồi của người dùng	collecting data	13:00 05/07/2024 - 17:00 11/07/2024	Tiếng Anh: C1	0.95

Bảng A.13: Danh sách công việc mà các nhân viên đã thực hiện trong quá khứ (4)

PHỤ LỤC A. BỘ DỮ LIỆU THỬ NGHIỆM THỰC

STT	Nhân viên	Công việc	Tags	Thời gian thực hiện	Yêu cầu năng lực nhân viên	Điểm
49	Nguyễn Phúc Nhật Nam	Phân tích quy trình nhập kho	analysis	08:00 09/06/2024 - 17:00 12/06/2024	Kinh nghiệm: 1-2 năm Tiếng Anh: A2	-1
50		Đựng trang chủ phía khách	frontend	08:00 13/06/2024 - 12:00 15/06/2024	Xếp hạng tốt nghiệp: Khá Frontend: Thành thạo	-1
51		Đựng giao diện module quản lý phòng ban	frontend	13:00 15/06/2024 - 17:00 21/06/2024	Xếp hạng tốt nghiệp: TB Kinh nghiệm: 1-2 năm Backend: Đã làm Frontend: Thành thạo	0.85
52		Kiểm thử tính năng đồng bộ dữ liệu	testing	08:00 22/06/2024 - 17:00 25/06/2024	Kiểm thử thủ công: Đã làm Kiểm thử đơn vị: Thành thạo Kiểm thử tự động: Đã làm	0.9
53		Sửa lỗi mô-đun quản lý khách sạn	frontend, backend	08:00 27/06/2024 - 10:00 02/07/2024	Kinh nghiệm: Dưới 1 năm Backend: Thành thạo Frontend: Đã làm	0.9
54		Lập trình micro-service module phân quyền	frontend, backend	10:00 02/07/2024 - 12:00 09/07/2024	Backend: Đã làm Frontend: Thành thạo Kiểm thử thủ công: Đã làm	-1
55		Triển khai ứng dụng ở heroku	devops	13:00 09/07/2024 - 12:00 15/07/2024	Kinh nghiệm: Dưới 1 năm Docker: Thành thạo CI_CD: Thành thạo	0.8
56	Đặng Quốc Tú	Kiểm thử tính năng Sửa chứng chỉ	testing	08:00 25/06/2024 - 17:00 29/06/2024	Kiểm thử thủ công: Đã làm Kiểm thử đơn vị: Đã làm Kiểm thử tự động: Đã làm	0.75
57		Viết kịch bản kiểm thử luồng xuất kho	testing	08:00 30/06/2024 - 12:00 03/07/2024	Xếp hạng tốt nghiệp: Khá Kiểm thử đơn vị: Thành thạo	0.8
58		Lập trình service gán nguồn lực	backend	08:00 30/06/2024 - 12:00 03/07/2024	Backend: Thành thạo Frontend: Thành thạo	0.85
59		Đựng giao diện thống kê nhân sự	frontend	13:00 03/07/2024 - 15:00 06/07/2024	Kinh nghiệm: Dưới 1 năm Backend: Đã làm Frontend: Thành thạo	0.8
60		Lập trình service kết nối CSDL	backend, frontend	15:00 06/07/2024 - 15:00 12/07/2024	Backend: Đã làm Frontend: Thành thạo Kiểm thử thủ công: Đã làm	0.85

Bảng A.14: Danh sách công việc mà các nhân viên đã thực hiện trong quá khứ (5)

Từ dữ liệu công việc trong quá khứ, hiệu suất thực hiện các công việc trong dự

án của từng nhân viên được tính toán và kết quả được trình bày trong **Bảng A.15**.

Tên công việc	Nguyễn Văn Minh	Nguyễn Văn Biển	Nguyễn Viết Đang	Nguyễn Phúc Nhật Nam	Nguyễn Thị Oanh	Vũ Thị Quỳnh	Nguyễn Minh Thành	Đặng Quốc Tú
Phân tích yêu cầu khách hàng	0.9	0.854	0.85	-1	0.75	0.95	0	0
Dựng codebase React-NodeJS	0.9	0.866	0.914	0.866	0	0	0.866	0
Dựng trang phân bổ nguồn lực dự án	0.95	0.9	0.8	-1	0	0	0.8	0.8
Dựng giao diện module quản lý thẻ	0.9	0.9	0.8	0.85	0	0	0.8	0.8
Triển khai hệ thống DXClan lên server	0.8	0.9	0.9	0.8	0	0	0	0
Tạo kế hoạch kiểm thử các chức năng	0	0	0	0	0.95	0.9	0	0
Kiểm thử luồng phân bổ nguồn lực	0.8	0.8	0.9	0.9	0.95	0.8	0	0
Sửa lỗi các module yêu cầu	0.9	0.9	0.9	0.9	0	0	0.9	0
Viết kịch bản thử nghiệm cho luồng phân bổ nguồn lực	0	0.9	0	0.8	0.95	0.95	0.8	0.8
Lập kế hoạch cho dự án	0.9	0.85	0	0	0	0.85	0	0
Thu thập phản hồi của khách hàng	0	0	0	0	0.95	0.95	0	0
Điều chỉnh chức năng phân bổ nguồn lực	0.95	0.95	0.85	0.85	0	0	0	0
Đề xuất ý cho bài toán phân bổ nguồn lực	0.95	0.95	0.9	0.8	0.8	0.8	0	0
Lập tài liệu thuyết trình với các bên liên quan	0	0	0	0	0.8	0.9	0	0
Viết tài liệu công nghệ sử dụng	0.95	0.95	0.9	0.9	0	0	0	0
Dựng các trang CRUD dự án	0.9	0.9	0.8	0.85	0	0	0.8	0.8
Lập trình service mô-đun quản lý dự án	0.95	0.95	0.8	0.8	0	0	-1	0
Ghép API mô-đun quản lý dự án	0.85	-1	0.85	-1	0	0	0.85	0.85

Bảng A.15: Hiệu suất thực hiện công việc của nhân viên