**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Quản lý công việc trong dự án**

**TRƯƠNG ANH QUỐC**

quoc.ta163403@sis.hust.edu.vn

**Ngành Kỹ thuật phần mềm**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | TS. Trịnh Tuấn Đạt |
| **Bộ môn:** | Công nghệ phần mềm |
| **Viện:** | Công nghệ thông tin – Truyền thông |
| **HÀ NỘI, 06/2021** | |

# Lời cam kết

Họ và tên sinh viên: Trương Anh Quốc

Điện thoại liên lạc: 0989652445 Email: quoc.ta163403@sis.hust.edu.vn

Lớp: CNTT 2.02 – K61 Hệ đào tạo: KSCQ

Tôi – *Trương Anh Quốc* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *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 tháng năm*  Tác giả ĐATN  *Trương Anh Quốc* |

# Lời cảm ơn

Lời đầu tiên em xin phép cảm ơn và biết ơn sâu sắc đến TS. Trịnh Tuấn Đạt. Thầy là người  
hướng dẫn trực tiếp cũng như giúp đỡ em rất nhiều, đưa ra các định hướng, các giải pháp công nghệ quan trọng trong quá trình em thực hiện ĐATN. Bên cạnh đó, em xin phép gửi lời cảm ơn tới TS. Vũ Thị Hương Giang, người đã giúp em trong giai đoạn tìm kiếm, thẩm định và phát triển nghiệp vụ cho ĐATN của bản thân.

Em xin gửi lời cảm ơn Trường đại học Bách Khoa Hà Nội và các thầy cô giáo đã tạo môi trường học tập, xây dựng kiến thức nền tảng là yếu tố cốt lõi để em có thể thực hiện ĐATN của mình. Các thầy cô đã cho em những bài học bổ ích không chỉ về kiến thức trong nhà trường mà có cả những buổi thực tế tại doanh nghiệp và các kỹ năng mềm giúp chúng em có thêm hiểu biết về cuộc sống.

Em cũng xin được gửi lời cảm ơn đến gia đình, người thân – là hậu phương vững chắc cho những năm tháng ngồi trên ghế nhà trường, tạo điều kiện để em có thể theo học và tìm kiếm con đường phát triển sự nghiệp bản thân trong những chặng đường sắp tới của cuộc đời. Đồng thời, em xin gửi lời cảm ơn tới những người bạn, những người anh chị em đã luôn ở bên động viên, hỗ trợ và đồng hành cùng em trong suốt 5 năm học tại trường Đại học Bách Khoa Hà Nội.

Cuối cùng, em xin cảm ơn Công ty Cổ phần công nghệ an toàn thông tin và truyền thông Việt Nam – VNIST đã hỗ trợ cũng như tạo điều kiện để em thực hiện ĐATN của mình trên nền tảng ứng dụng của công ty.

Do thời gian để thực hiện ĐATN có hạn cũng như trình độ chuyên môn còn hạn chế nên ĐATN của em sẽ còn nhiều thiếu sót. Vì thế em rất mong nhận được đánh giá từ thầy cô để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

# Tóm tắt

Dự án là một trong những thành tố quan trọng, thiết yếu cho sự hình thành và phát triển của bất kỳ một doanh nghiệp nào. Trước kia, việc quản lý dự án bị phụ thuộc nhiều vào giấy tờ sổ sách, danh sách công việc và chi phí tính toán chỉ được kiểm soát bởi những phần mềm thô sơ như Microsoft Excel, Microsoft Project, v.v. Ngày nay, với sự phát triển công nghệ thông tin, việc sử dụng các trang web thay thế cho các “phần mềm cần cài đặt – installed software” dần trở nên phổ biến. Do đó, công cụ quản lý dự án online dần trở nên nhiều hơn, thân thuộc hơn với người dùng.

Trên thị trường hiện có rất nhiều phần mềm, công cụ online giúp các doanh nghiệp có thể lập kế hoạch dự án, quản lý tiến độ dự án và báo cáo kết quả dự án như Jira, Zoho, Backlog, Trello, Wrike, v.v. Tất cả những công cụ trên đã giải quyết được vấn đề quan trọng nhất đó chính là quản lý tiến độ dự án và báo cáo tiến độ dự án. Tuy nhiên, vấn đề lập kế hoạch dự án tốt cần phụ thuộc vào kinh nghiệm ước lượng, khả năng tính toán của nhà quản lý để có thể cân bằng chi phí – thời gian sao cho hợp lý. Đồng thời, vấn đề báo cáo kết quả dự án và báo cáo điểm số công việc, điểm số thành viên trong dự án mới chỉ có một số ít phần mềm chú trọng và xử lý triệt để. Cuối cùng, quản lý yêu cầu thay đổi cũng là một tính năng thiết yếu với những dự án có sự nghiêm ngặt về tham số và chất lượng.

Nhận biết được những hạn chế của những phần mềm quản lý dư án, em đã xây dựng ứng dụng **“Quản lý công việc theo dự án”**. Ứng dụng giúp người dùng tạo dự án, lập kế hoạch dự án, tạo công việc dự án, quản lý tiến độ, báo cáo tiến độ và kết quả dự án, đánh giá điểm số thành viên trong dự án, quản lý yêu cầu thay đổi trong dự án.

ĐATN được xây dựng và tích hợp vào hệ thống không gian làm việc số dxclan.com của công ty Cổ phần Công nghệ An toàn thông tin và Truyền thông Việt Nam. Các đóng góp chính trong ĐATN là (i) sử dụng thuật toán để tính toán thời gian – chi phí thoả hiệp khi lập kế hoạch dự án, (ii) xây dựng công thức đánh giá tính điểm số công việc, điểm số thành viên trong dự án, (iii) xây dựng báo cáo, biểu đồ trực quan thể hiện 2 yếu tố quan trọng trong dự án là tiến độ - chi phí dự án, (iv) xây dựng chức năng quản lý yêu cầu thay đổi trong dự án.

# Mục lục

[Lời cam kết 2](#_Toc73001475)

[Lời cảm ơn 3](#_Toc73001476)

[Tóm tắt 4](#_Toc73001477)

[Mục lục 5](#_Toc73001478)

[Danh mục hình vẽ 10](#_Toc73001479)

[Danh mục bảng 11](#_Toc73001480)

[Danh mục công thức 12](#_Toc73001481)

[Danh mục các từ viết tắt 13](#_Toc73001482)

[Danh mục thuật ngữ 14](#_Toc73001483)

[Chương 1 Giới thiệu đề tài 1](#_Toc73001484)

[1.1 Đặt vấn đề 1](#_Toc73001485)

[1.2 Mục tiêu và phạm vi đề tài 2](#_Toc73001486)

[1.3 Định hướng giải pháp 2](#_Toc73001487)

[1.4 Bố cục đồ án 4](#_Toc73001488)

[Chương 2 Khảo sát và phân tích yêu cầu 5](#_Toc73001489)

[2.1 Khảo sát hiện trạng 5](#_Toc73001490)

[2.2 Tổng quan chức năng 7](#_Toc73001491)

[2.2.1 Biểu đồ use case tổng quan 7](#_Toc73001492)

[2.2.2 Biểu đồ use case phân rã Quản lý dự án 8](#_Toc73001493)

[2.2.3 Biểu đồ use case phân rã Quản lý công việc 9](#_Toc73001494)

[2.2.4 Biểu đồ use case phân rã Đánh giá công việc 10](#_Toc73001495)

[2.2.5 Biểu đồ use case phân rã Xem báo cáo tiến độ dự án 11](#_Toc73001496)

[2.2.6 Biểu đồ use case phân rã Thống kê đánh giá dự án 12](#_Toc73001497)

[2.2.7 Biểu đồ use case phân rã Quản lý yêu cầu thay đổi 13](#_Toc73001498)

[2.2.8 Quy trình nghiệp vụ 14](#_Toc73001499)

[2.3 Đặc tả chức năng 17](#_Toc73001500)

[2.3.1 Đặc tả use case Quản lý dự án 17](#_Toc73001501)

[2.3.2 Đặc tả use case Quản lý công việc 17](#_Toc73001502)

[2.3.3 Đặc tả use case Đánh giá công việc 18](#_Toc73001503)

[2.3.4 Đặc tả use case Xem báo cáo tiến độ dự án 18](#_Toc73001504)

[2.3.5 Đặc tả use case Thống kê đánh giá dự án 18](#_Toc73001505)

[2.3.6 Đặc tả use case Quản lý yêu cầu thay đổi 18](#_Toc73001506)

[2.4 Yêu cầu phi chức năng 18](#_Toc73001507)

[2.4.1 Yêu cầu chung 18](#_Toc73001508)

[2.4.2 Yêu cầu về giao diện người dùng 18](#_Toc73001509)

[2.4.3 Yêu cầu về bảo mật 19](#_Toc73001510)

[2.4.4 Yêu cầu khác 19](#_Toc73001511)

[Chương 3 Công nghệ sử dụng 20](#_Toc73001512)

[3.1 Công nghệ xây dựng cơ sở dữ liệu 20](#_Toc73001513)

[3.2 Công nghệ xây dựng backend 22](#_Toc73001514)

[3.2.1 NodeJS 22](#_Toc73001515)

[3.2.2 Express 22](#_Toc73001516)

[3.3 Công nghệ xây dựng frontend 22](#_Toc73001517)

[3.3.1 ReactJS 22](#_Toc73001518)

[3.3.2 Redux 23](#_Toc73001519)

[3.3.3 CSS 24](#_Toc73001520)

[3.3.4 Bootstrap 24](#_Toc73001521)

[3.4 Tích hợp các công nghệ 24](#_Toc73001522)

[Chương 4 Phát triển và triển khai ứng dụng 26](#_Toc73001523)

[4.1 Thiết kế kiến trúc 26](#_Toc73001524)

[4.1.1 Lựa chọn kiến trúc phần mềm 26](#_Toc73001525)

[4.1.1.1 Mô hình kiến trúc của Redux 26](#_Toc73001526)

[4.1.1.2 Mô hình MVC 27](#_Toc73001527)

[4.1.2 Thiết kế tổng quan 27](#_Toc73001528)

[4.1.3 Thiết kế chi tiết gói 29](#_Toc73001529)

[4.2 Thiết kế chi tiết 31](#_Toc73001530)

[4.2.1 Thiết kế giao diện 31](#_Toc73001531)

[4.2.2 Thiết kế lớp 33](#_Toc73001532)

[4.2.3 Thiết kế cơ sở dữ liệu 39](#_Toc73001533)

[4.2.3.1 Thực thể liên kết 39](#_Toc73001534)

[4.2.3.2 Thiết kế chi tiết cơ sở dữ liệu 44](#_Toc73001535)

[4.3 Xây dựng ứng dụng 50](#_Toc73001536)

[4.3.1 Thư viện và công cụ sử dụng 50](#_Toc73001537)

[4.3.2 Kết quả đạt được 51](#_Toc73001538)

[4.3.3 Minh hoạ các chức năng chính 51](#_Toc73001539)

[4.4 Kiểm thử 52](#_Toc73001540)

[4.4.1 Compatibility Testing 52](#_Toc73001541)

[4.4.2 Black Box Testing 52](#_Toc73001542)

[4.5 Triển khai 55](#_Toc73001543)

[Chương 5 Các giải pháp và đóng góp nổi bật 56](#_Toc73001544)

[5.1 Thuật toán đề xuất giải pháp thoả hiệp chi phí - thời gian 56](#_Toc73001545)

[5.1.1 Vấn đề 56](#_Toc73001546)

[5.1.2 Giải pháp 57](#_Toc73001547)

[5.2 Công thức đánh giá công việc và thành viên trong dự án 60](#_Toc73001548)

[5.2.1 Vấn đề 60](#_Toc73001549)

[5.2.2 Giải pháp 60](#_Toc73001550)

[5.2.2.1 Công thức đánh giá công việc 60](#_Toc73001551)

[5.2.2.2 Công thức đánh giá thành viên trong công việc 62](#_Toc73001552)

[5.3 Báo cáo tự động và thống kê đánh giá 64](#_Toc73001553)

[5.3.1 Vấn đề 64](#_Toc73001554)

[5.3.2 Giải pháp 64](#_Toc73001555)

[5.4 Thuật toán đệ quy xác định tầm ảnh hưởng một công việc 67](#_Toc73001556)

[5.4.1 Vấn đề 67](#_Toc73001557)

[5.4.2 Giải pháp 68](#_Toc73001558)

[5.5 Yêu cầu thay đổi có yếu tố định lượng về chi phí – thời gian 69](#_Toc73001559)

[5.5.1 Vấn đề 69](#_Toc73001560)

[5.5.2 Giải pháp 69](#_Toc73001561)

[5.6 Tích hợp với module lương nhân viên 71](#_Toc73001562)

[5.6.1 Vấn đề 71](#_Toc73001563)

[5.6.2 Giải pháp 71](#_Toc73001564)

[5.7 Công việc bình thường và công việc theo dự án 72](#_Toc73001565)

[5.7.1 Vấn đề 72](#_Toc73001566)

[5.7.2 Giải pháp 73](#_Toc73001567)

[Chương 6 Kết luận và hướng phát triển 75](#_Toc73001568)

[6.1 Kết luận 75](#_Toc73001569)

[6.2 Hướng phát triển 75](#_Toc73001570)

[Tài liệu tham khảo 77](#_Toc73001571)

# Danh mục hình vẽ

[**Hình 2.1** Biểu đồ use case tổng quan hệ thống 7](#_Toc73003159)

[**Hình 2.2** Phân rã use case Quản lý dự án 8](#_Toc73003160)

[**Hình 2.3** Phân rã use case Quản lý công việc 9](#_Toc73003161)

[**Hình 2.4** Phân rã use case Đánh giá công việc 10](#_Toc73003162)

[**Hình 2.5** Phân rã use case Xem báo cáo tiến độ dự án 11](#_Toc73003163)

[**Hình 2.6** Phân rã use case Thống kê đánh giá dự án 12](#_Toc73003164)

[**Hình 2.7** Phân rã use case Quản lý yêu cầu thay đổi 13](#_Toc73003165)

[**Hình 2.8** Sơ đồ hoạt động lập kế hoạch dự án 14](#_Toc73003166)

[**Hình 2.9** Sơ đồ hoạt động đánh giá công việc dự án 15](#_Toc73003167)

[**Hình 2.10** Sơ đồ hoạt động tạo mới công việc (sau khi lập kế hoạch) 16](#_Toc73003168)

[**Hình 2.11** Sơ đồ hoạt động chỉnh sửa công việc (liên quan tới nguồn lực) 17](#_Toc73003169)

[**Hình 3.1** Sự khác biệt giữa các lưu trữ dữ liệu NoSQL DB và RDBMS 20](#_Toc73003170)

[**Hình 3.2** So sánh tốc độ ghi dữ liệu của MongoDB và MySQL 21](#_Toc73003171)

[**Hình 3.3** Luồng dữ liệu của Redux 23](#_Toc73003172)

[**Hình 3.4** Sự khác biệt giữa việc quản lý states dùng Redux và không dùng Redux 24](#_Toc73003173)

[**Hình 3.5** Tích hợp các công nghệ - MERN 25](#_Toc73003174)

[**Hình 4.1** Mô hình kiến trúc Redux 26](#_Toc73003175)

[**Hình 4.2** Sơ đồ kiến trúc MVC 27](#_Toc73003176)

[**Hình 4.3** Biểu đồ gói tổng quan phần server của ứng dụng QLDA 27](#_Toc73003177)

[**Hình 4.4** Biểu đồ gói tổng quan phần client của ứng dụng QLDA 28](#_Toc73003178)

[**Hình 4.5** Chi tiết các gói xây dựng chức năng Lập kế hoạch dự án phần server 29](#_Toc73003179)

[**Hình 4.6** Chi tiết các gói xây dựng chức năng Lập kế hoạch dự án phần client 30](#_Toc73003180)

[**Hình 4.7** Chi tiết các gói xây dựng chức năng Quản lý dự án phần client 31](#_Toc73003181)

[**Hình 4.8** Layout giao diện mock-up chung của ứng dụng 32](#_Toc73003182)

[**Hình 4.9** Giao diện mock-up lập kế hoạch dự án 33](#_Toc73003183)

[**Hình 4.10** Thiết kế chi tiết lớp EvaluateByResponsibleEmployeeProject 34](#_Toc73003184)

[**Hình 4.11** Thiết kế chi tiết lớp AddProjectTaskForm 36](#_Toc73003185)

[**Hình 4.12** Sơ đồ thực thể liên kết của ứng dụng QLDA 39](#_Toc73003186)

[**Hình 4.13** Chi tiết thuộc tính của các thực thể liên kết Project, Salary, ProjectChangeRequest, Employee, User 41](#_Toc73003187)

[**Hình 4.14** Chi tiết thuộc tính của thực thể Task 42](#_Toc73003188)

[**Hình 5.1** Minh hoạ tính toán đề xuất thoả hiệp dự án 58](#_Toc73003189)

[**Hình 5.2** Báo cáo tiến độ hiện tại của dự án 63](#_Toc73003190)

[**Hình 5.3** Báo cáo chi phí hiện tại của dự án 64](#_Toc73003191)

[**Hình 5.4** Báo cáo tình hình thành viên hiện tại của dự án 64](#_Toc73003192)

[**Hình 5.5** Thống kê thông số dự án với biểu đồ EVM theo tuần 65](#_Toc73003193)

[**Hình 5.6** Thống kê thông số các thành viên trong tháng 65](#_Toc73003194)

[**Hình 5.7** Hình ảnh minh hoạ thuật toán bằng code 67](#_Toc73003195)

[**Hình 5.8** Biểu mẫu yêu cầu tạo công việc mới không theo lập kế hoạch 69](#_Toc73003196)

[**Hình 5.9** Biểu mẫu yêu cầu chỉnh sửa công việc sau khi đã lập kế hoạch 69](#_Toc73003197)

[**Hình 5.10** Chi tiết yêu cầu thay đổi công việc 70](#_Toc73003198)

[**Hình 5.11** Bảng lương gợi ý cho người quản lý khi tạo mới dự án 71](#_Toc73003199)

[**Hình 5.12** Giao diện các nút trong công việc bình thường 72](#_Toc73003200)

[**Hình 5.13** Giao diện các nút trong công việc dự án 72](#_Toc73003201)

[**Hình 5.14** Giao diện công thức mặc định công việc dự án 73](#_Toc73003202)

# Danh mục bảng

[**Bảng 2.1** Bảng so sánh chức năng của Zoho và Base Wework với ĐATN 5](#_Toc73003203)

[**Bảng 4.1** Bảng đặc tả chi tiết thuộc tính và phương thức của lớp EvaluateByResponsibleEmployeeProject 34](#_Toc73003204)

[**Bảng 4.2** Bảng đặc tả chi tiết thuộc tính và phương thức của lớp AddProjectTaskForm 36](#_Toc73003205)

[**Bảng 4.3** Giải thích ý nghĩa của các thực thể liên kết 40](#_Toc73003206)

[**Bảng 4.4** Danh sách thư viện và công cụ sử dụng 49](#_Toc73003207)

[**Bảng 4.5** Thống kê thông tin ứng dụng 50](#_Toc73003208)

[**Bảng 4.6** Thống kê kiểm thử tưởng thích trên các thiết bị 50](#_Toc73003209)

[**Bảng 4.7** Kiểm thử nhóm chức năng Lập kế hoạch dự án 51](#_Toc73003210)

[**Bảng 4.8** Kiểm thử nhóm chức năng Báo cáo tự động dự án 53](#_Toc73003211)

[**Bảng 5.1** Danh mục các hoạt động của dự án bảo dưỡng hai cỗ máy 55](#_Toc73003212)

[**Bảng 5.2** Lịch trình hoạt động của dự án bảo dưỡng hai cỗ máy 56](#_Toc73003213)

[**Bảng 5.3** Dữ liệu về hoạt động bình thường và thoả hiệp của dự án 57](#_Toc73003214)

[**Bảng 5.4** Bảng danh sách công việc dùng cho thuật toán đệ quy 66](#_Toc73003215)

# Danh mục công thức

# Danh mục các từ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface  Giao diện lập trình ứng dụng |
| **HTML** | HyperText Markup Language  Ngôn ngữ đánh dấu siêu văn bản |
| **ĐATN** | Đồ án tốt nghiệp |
| **SV** | Sinh viên |
| **QLDA** | Quản lý dự án |
| **CSDL** | Cơ sở dữ liệu |

# Danh mục thuật ngữ

|  |  |
| --- | --- |
| Browser | Trình duyệt |
| **Website** | Trang web |
| **Module** | Một thành phần có nhiệm vụ nhất định nào đó trong hệ thống |
| **Backup** | Sao chép lại dữ liệu ra một vùng nhớ khác |
| **Server** | Máy chủ |
| **Client** | Máy khách |
| **Backend** | Lập trình bên phía máy chủ |
| **Frontend** | Lập trình bên phía máy khách |
| **NoSQL** | Cơ sở dữ liệu phi quan hệ |
| **SQL** | Cơ sở dữ liệu quan hệ |
| **RDBMS** | Relational Database Management System – Hệ quản trị cơ sở dữ liệu quan hệ |
| **Form** | Biểu mẫu để điền thông tin |

# Giới thiệu đề tài

## Đặt vấn đề

Trên góc độ tổng quát nhất, dự án là một quá trình hoạt động, thực hiện một danh sách các công việc để đạt được những mục tiêu đã hoạch định từ trước. Các mục tiêu có thể được định nghĩa thông qua đầu vào, đầu ra, kết quả và lợi nhuận kiếm được[[1]](#footnote-1). Để một dự án thành công, người quản lý cần có những phương pháp dựa trên năng lực và hiểu biết của mình, để dự án đạt được các mục tiêu theo những tiêu chí chấp nhận được của họ, trong khoảng thời gian, phạm vi và ngân sách đã thoả thuận.

Quản lý dự án bao gồm các bước chính sau: (i) lập kế hoạch, (ii) thực thi, (iii) giám sát tiến độ, (iv) đánh giá kết quả, (v) xử lý yêu cầu thay đổi trong dự án. Mỗi một bước trong công việc quản lý yêu cầu một khối lượng quy trình cụ thể, khác nhau cho từng loại phạm vi, tiến độ và chi phí. Dự án càng mở rộng về ba yếu tố trên, thì khối lượng quy trình sẽ tăng lên, số lượng rủi ro thất bại cũng sẽ nhiều hơn. Chỉ tính riêng với những dự án công nghệ thông tin trên thế giới, theo nghiên cứu của The Standish Group, trong năm 2019 có 31.1% dự án bị thất bại [[2]](#footnote-2), so với con số cùng kỳ năm 2020 là 19% [[3]](#footnote-3). Con số trên cho thấy, kĩ năng và kiến thức của những người quản lý vô cùng quan trọng trong việc quyết định dự án của họ thành hay bại.

Để giúp đỡ và hỗ trợ cho những người quản lý, nhiều công cụ, phần mềm với chức năng tương tự đã ra đời. Các phần mềm này đã giúp phần xoá bỏ việc quản trị dự án trên giấy tờ hay phần mềm thô sơ, thay vào đó giúp số hoá dự án và biểu diễn dưới dạng biểu đồ một cách trực quan, dễ hiểu hơn. Với sự phát triển của các công cụ, những người quản lý dự án mới sẽ có thể nhận được sự trợ giúp từ kiến thức và kinh nghiệm của những người quản lý dự án đi trước, có thâm niên - thông qua các chức năng tích hợp trong sản phẩm phần mềm. Tuy nhiên, các phần mềm này vẫn còn một số hạn chế như (i) chưa đưa ra được biện pháp đề xuất cân bằng chi phí – thời gian lúc lập kế hoạch, (ii) công thức đánh giá dự án chưa bao quát được hết các yếu tố quan trọng là tiến độ, chi phí và (iii) yêu cầu thay đổi đa phần dưới dạng mẫu báo cáo bình thường, không có yếu tố định lượng.

Giải quyết được các vấn đề nêu trên sẽ giúp cho phần mềm quản lý công việc theo dự án đạt được kết quả trực quan, dễ hiểu hơn. Đồng thời, người quản lý sẽ có thêm nhiều góc nhìn về dự án, từ đó họ có thể đưa ra những quyết định chính xác, tối ưu hơn dựa trên kết quả định lượng đã có.

## Mục tiêu và phạm vi đề tài

Hiện nay, có nhiều phần mềm, công cụ hỗ trợ quản lý công việc theo dự án theo nhiều loại khác nhau. Tuy nhiên, có hai loại phần mềm chính là (i) loại Desktop và (ii) loại Web-based. Loại phần mềm Destop như Microsoft Project yêu cầu cài đặt lên trên máy tính mới sử dụng được. Nhược điểm các phần mềm này là cần có hệ điều hành tương ứng mới có thể cài đặt và thực thi được. Loại Web-based chỉ cần yêu cầu trình duyệt, đa phần tương thích với mọi loại máy nên được người dùng ưa chuộng và sử dụng hơn như Zoho, Trello, Base Wework, Backlog, v.v.

Dựa trên phân tích, đánh giá ưu điểm và hạn chế của hai phần mềm QLDA tiêu biểu cho doanh nghiệp nêu trên, ĐATN của em hướng đến việc xây dựng ứng dụng QLDA tích hợp vào hệ thống “Không gian làm việc số[[4]](#footnote-4)” của Công ty Cổ phần công nghệ an toàn thông tin và truyền thông Việt Nam (VNIST). Ứng dụng cung cấp các chức năng như (i) lập kế hoạch dự án, (ii) giám sát tiến độ dự án, (iii) đánh giá công việc và thành viên dự án, (iv) thống kê đánh giá công việc và điểm số nhân viên theo tháng và (v) quản lý yêu cầu thay đổi dự án.

## Định hướng giải pháp

ĐATN đã khảo sát nhu cầu khách hàng doanh nghiệp, khảo sát tính năng của các ứng dụng QLDA hiện có trên thị trường. Dựa trên kết quả khảo sát đó, ĐATN thực hiện (i) phát triển thuật toán đề xuất giải pháp giảm thời gian tiến độ dự án ở giai đoạn lập kế hoạch, (ii) ứng dụng thuật toán đệ quy để tìm kiếm các công việc bị ảnh hưởng trong một yêu cầu thay đổi, (iii) xây dựng công thức đánh giá công việc và đánh giá thanh viên trong dự án, (iv) lựa chọn công nghệ phù hợp để xây dựng và (v) tích hợp ứng dụng vào hệ thống không gian làm việc số - hệ thống cung cấp các chức năng quản lý doanh nghiệp như: quản lý công việc, quản lý người dùng, quản lý lương nhân viên, v.v.

Để đề xuất được giải pháp giảm thời gian tiến độ dự án ở giai đoạn lập kế hoạch, em sử dụng thuật toán “trao đổi thời gian – chi phí” (cost/time tradeoff). Đây là thuật toán chuyên dụng trong phương pháp lập kế hoạch theo phương thức “Đường găng CPM”. Thuật toán sẽ sử dụng các tham số “thời gian ước lượng”, “thời gian thoả hiệp”, “chi phí ước lượng”, “chi phí thoả hiệp”. Chi tiết giải pháp và công thức thuật toán này sẽ được trình bày ở mục 5.1.

Sau khi lập kế hoạch, dự án sẽ có một danh sách công việc mà trong đó, một số công việc liên kết với nhau thành một chuỗi. Một khi một công việc trong chuỗi này thay đổi thì những công việc sau đó sẽ bị ảnh hưởng theo. Chi tiết giải pháp để xác định tầm ảnh hưởng sẽ được trình bày ở mục 5.4.

Công việc trong dự án có hai tham số quan trọng là “thời gian” và “chi phí”, điểm số của thành viên trong dự án sẽ dựa trên điểm số công việc mà thành viên đó tham gia. Vì vậy, giải pháp xây dựng công thức đánh giá công việc và đánh giá thành viên trong dự án sẽ được trình bày ở mục 5.2.

Đồ án sử dụng (i) MongoDB làm hệ quản trị cơ sở dữ liệu cho hệ thống, (ii) NodeJS, ExpressJS để lập trình backend, (iii) JWT để xác thực tài khoản và phiên đăng nhập và (iv) ReactJS, Redux để lập trình frontend. Các công nghệ lựa chọn đều dễ dàng triển khai trên hệ thống máy chủ thông thường hoặc trên nền tảng cloud hiện nay. MongoDB là cơ sở dữ liệu NoSQL dễ dàng sửa đổi bổ sung và thay đổi cấu trúc dữ liệu nên rất thích hợp cho xây dựng ứng dụng này. Cuối cùng, ReactJS là công nghệ được sử dụng để xây dựng giao diện ứng dụng trình duyệt có tính tương tác cao.

Ứng dụng được xây dựng và tích hợp với các module (i) quản lý công việc và (ii) quản lý lương nhân viên. Tích hợp với module quản lý công việc nhằm tái sử dụng và chỉnh sửa trường công việc cho phù hợp với công việc trong dự án. Tích hợp module quản lý lương nhân viên để có thể lây được lương nhân viên mặc định cho các thành viên trong dự án, từ đó người quản lý có thể thay đổi lương dựa trên mức mặc định đó.

## Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày khảo sát hiện trạng các phần mềm QLDA hiện nay, nhu cầu về việc sử dụng ứng dụng QLDA trong doanh nghiệp. Phân tích yêu cầu và đưa ra chức năng chính của ứng dụng.

Chương 3 giới thiệu về các công nghệ được sử dụng để xây dựng ứng dụng, điểm nổi bật của công nghệ đó, lý do lựa chọn công nghệ đó để xây dựng ứng dụng QLDA.

Chương 4 trình bày về cách thức thiết kế, triển khai và phát triển ứng dụng. Quy trình triển khai và phát triển ứng dụng bao gồm các bước như sau: (i) thiết kế kiến trúc, (ii) thiết kế chi tiết, (iii) xây dựng và phát triển ứng dụng, (iv) kiểm thử ứng dụng và cuối cùng là (v) triển khai ứng dụng.

Chương 5 trình bày về những điểm đặc biệt trong ĐATN và tính khả thi khi đem vào sử dụng trong thực tế. Những tính năng đã xây dựng mang lại những lợi ích gì đối với doanh nghiệp sản xuất và sự khác biệt với các ứng dụng tương tự khác trên thị trường.

Chương 6 trình bày về các các kết quả đạt được và chưa đạt được cùng với đó là định hướng phát triển trong tương lai với ứng dụng trong ĐATN.

# Khảo sát và phân tích yêu cầu

## Khảo sát hiện trạng

Trong phần này, em xin giới thiệu hai phần mềm QLDA chính và tiêu biểu là Zoho và Base Wework để so sánh và đánh giá ưu nhược điểm hiện có.

Zoho và Base Wework đã đáp ứng được đa phần chức năng chính trong QLDA như (i) lập kế hoạch dự án, (ii) theo dõi và giám sát dự án và (iii) đánh giá kết quả dự án. Tuy nhiên, với chức năng lập kế hoạch dự án, các phần mềm quản lý công việc như Zoho, Base Wework mới chỉ dừng lại ở việc tạo danh sách công việc nhanh chóng, xem xét tính khả thi của dự án nhưng chưa tự động đưa ra gợi ý giải pháp để tăng độ khả thi lên. Tiếp theo, với chức năng đánh giá kết quả dự án, Zoho và Base Wework chỉ sử dụng công thức mặc định liên quan tới tiến độ và không liên quan gì đến chi phí, ngân sách. Cuối cùng, với chức năng yêu cầu thay đổi trong dự án, Base Wework không có tính năng này, còn bên phía Zoho họ thiết kế tính năng này như một mẫu báo cáo bình thường, không có yếu tố định lượng.

Dưới đây là bảng so sánh các chức năng của hai phần mềm Zoho, Base Wework với ứng dụng mà ĐATN định hướng xây dựng.

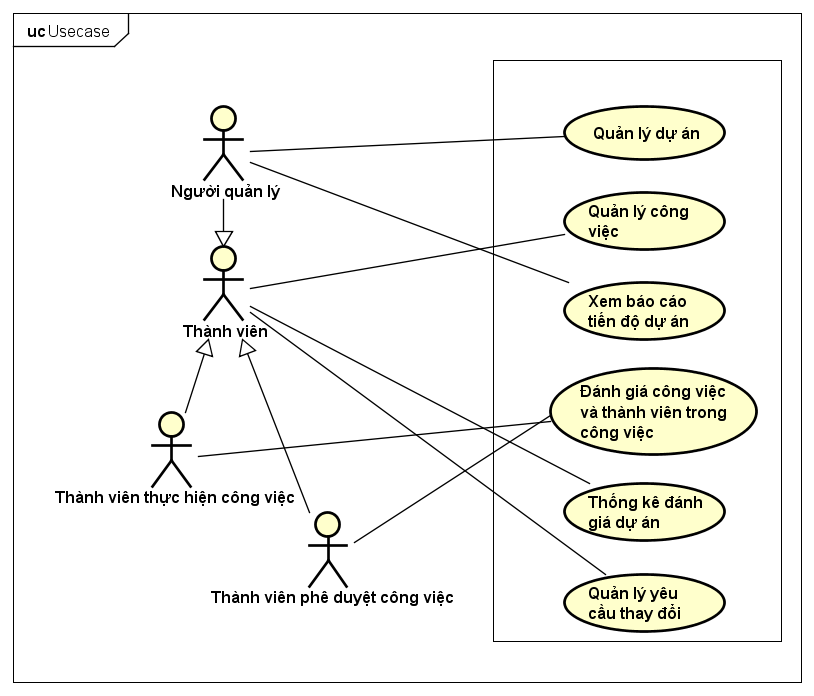
Chứ thích: X – có chức năng, O – không có chức năng

**Bảng 2.1** Bảng so sánh chức năng của Zoho và Base Wework với ĐATN

| STT | Tiêu chí | Zoho | Base Wework | ĐATN |
| --- | --- | --- | --- | --- |
| 1 | Tạo danh sách công việc từ file excel | X | X | X |
| 2 | Tự động đề xuất phương án thoả hiệp | O | O | X |
| 3 | Giám sát tiến độ dự án | X | X | X |
| 4 | Báo cáo tự động | X | X | X |
| 5 | Thống kê thông thường | X | X | X |
| 6 | Thống kê theo yếu tố thời gian, chi phí | O | O | X |
| 7 | Đánh giá công việc trong dự án theo mô hình RACI | O | O | X |
| 8 | Quản lý yêu cầu thay đổi thông thường | X | O | X |
| 9 | Tạo yêu cầu thay đổi có yếu tố định lượng | X | X | X |

## Tổng quan chức năng

### Biểu đồ use case tổng quan

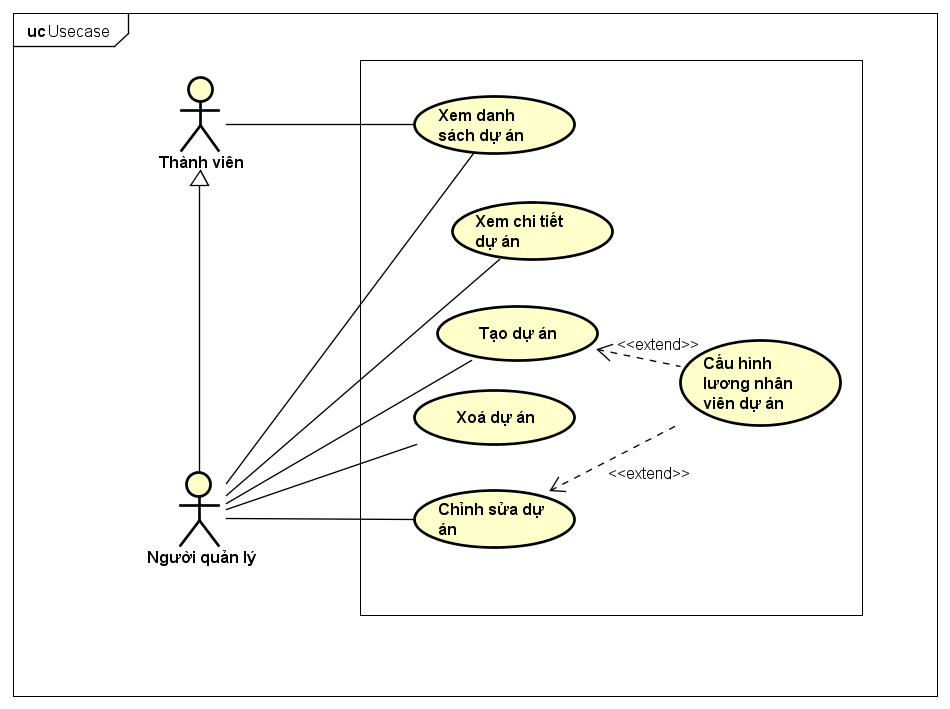


**Hình 2.1** Biểu đồ use case tổng quan hệ thống

**Hình 2.1** mô tả tổng quan về ứng dụng QLDA với các tác nhân và vai trò của tác nhân. Dựa vào hình vẽ, ta xác định được các tác nhân của ứng dụng bao gồm:

1. Người quản lý: Có vai trò quản trị dự án, thực hiện được use case quản lý dự án, quản lý công việc, xem báo cáo tiến độ dự án, thống kê đánh giá dự án, quản lý yêu cầu thay đổi..
2. Thành viên: Là thành viên của dự án nói chung, thực hiện được use case quản lý công việc, thống kê đánh giá dự án, quản lý yêu cầu thay đổi.
3. Thành viên thực hiện công việc: Là thành viên đóng vai trò thực hiện trong một công việc, thực hiện được các use case như của Thành viên, và thêm use case đánh giá công việc và thành viên trong công việc.
4. Thành viên phê duyệt công việc: Là thành viên đóng vai trò phê duyệt trong một công việc, thực hiện được các use case như của Thành viên, và thêm use case đánh giá công việc và thành viên trong công việc.

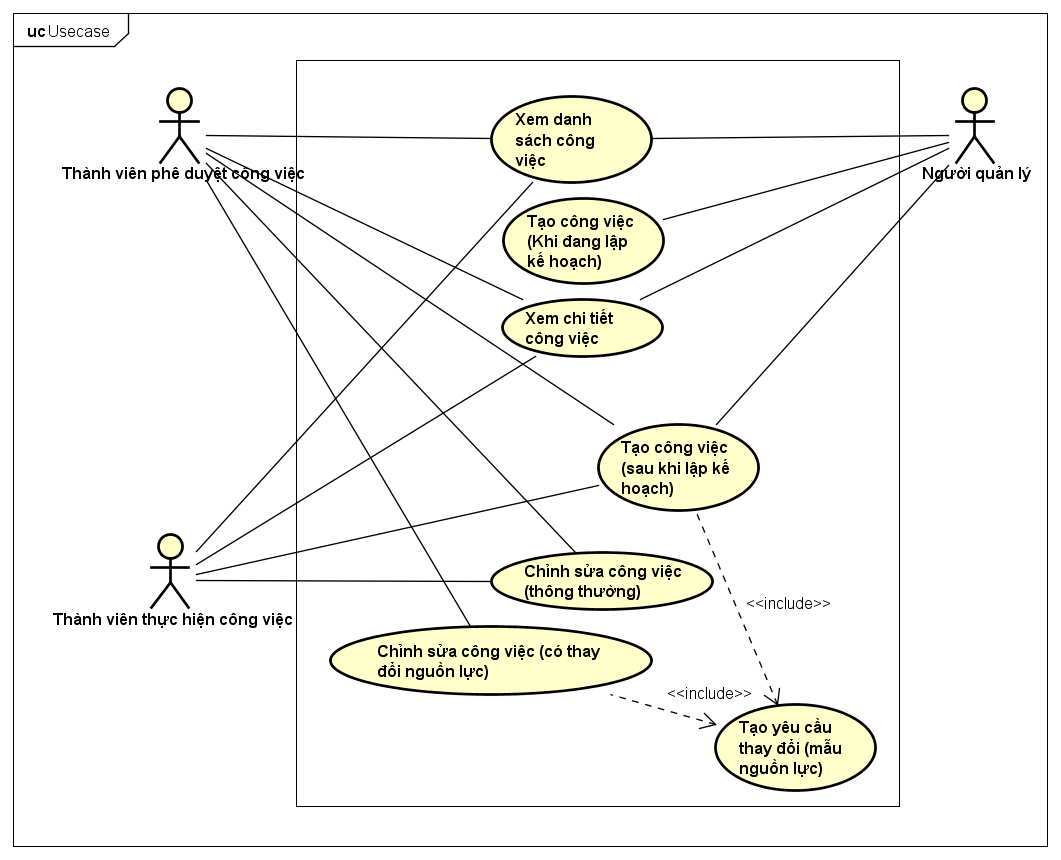
### Biểu đồ use case phân rã Quản lý dự án



**Hình 2.2** Phân rã use case Quản lý dự án

**Hình 2.2** là biểu đồ use case phân rã Quản lý dự án. Người quản lý dự án có quyền thực hiện use case (i) xem danh sách dự án, (ii) xem chi tiết dự án (iii), xoá dự án (iv), chỉnh sửa dự án (v), tạo dự án, ngoài ra còn có thể (vi) cấu hình lương nhân viên trong khi tạo hoặc chỉnh sửa dự án. Thành viên thông thường của dự án chỉ có thể (i) xem danh sách dự án và (ii) xem chi tiết dự án, tuy nhiên cả hai use case chỉ được sử dụng khi thành viên đó có trong dự án mà họ muốn xem.

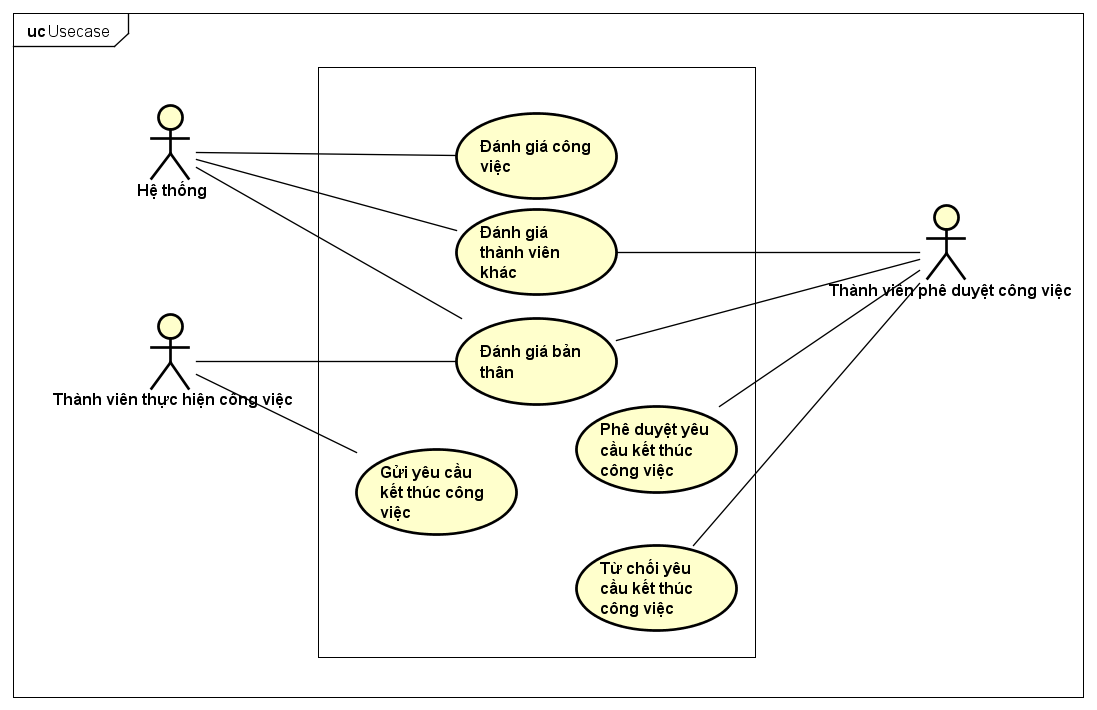
### Biểu đồ use case phân rã Quản lý công việc



**Hình 2.3** Phân rã use case Quản lý công việc

**Hình 2.3** là biểu đồ use case phân rã use case Quản lý công việc. Người quản lý có quyền thực hiện use case (i) xem danh sách công việc, (ii) tạo công việc (khi đang lập kế hoạch), (iii) xem chi tiết công việc và (iv) tạo công việc mới (sau khi lập kế hoạch). Thành viên thực hiện công việc có quyền thực hiện use case (i) xem danh sách công việc, (ii) tạo công việc (sau khi lập kế hoạch), (iii) chỉnh sửa công việc (thông thường) và (iv) xem chi tiết công việc. Thành viên phê duyệt công việc có quyền thực hiện use case (i) xem danh sách công việc, (ii) xem chi tiết công việc, (iii) tạo công việc (sau khi lập kế hoạch), (iii) chỉnh sửa công việc (thông thường) và (iv) chỉnh sửa công việc (có thay đổi nguồn lực). Điều chú ý là khi thực hiện use case (i) tạo công việc (sau khi lập kế hoạch) và (ii) chỉnh sửa công việc (có thay đổi nguồn lực) sẽ tự động kích hoạt use case tạo yêu cầu thay đổi (mẫu nguồn lực).

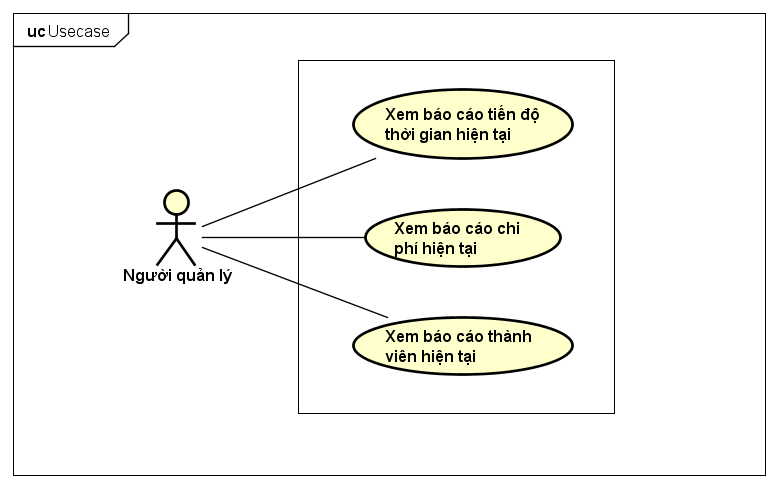
### Biểu đồ use case phân rã Đánh giá công việc



**Hình 2.4** Phân rã use case Đánh giá công việc

**Hình 2.4** là biểu đồ phân rã use case Đánh giá công việc. Thành viên thực hiện công việc có thể thực hiện use case (i) tự đánh giá bản thân trong công việc và (ii) gửi yêu cầu kết thúc công việc. Thành viên phê duyệt công việc có thể thực hiện use case (i) đánh giá thành viên khác, (ii) đánh giá bản thân, (iii) phê duyệt yêu cầu kết thúc công việc và (iv) từ chối yêu cầu kết thúc công việc. Những lần đánh giá bên trên đều là đánh giá bằng tay, còn hệ thống sẽ thực hiện các use case sau một cách tự động gồm (i) đánh giá công việc, (ii) đánh giá thành viên khác và (iii) đánh giá bản thân người đó.

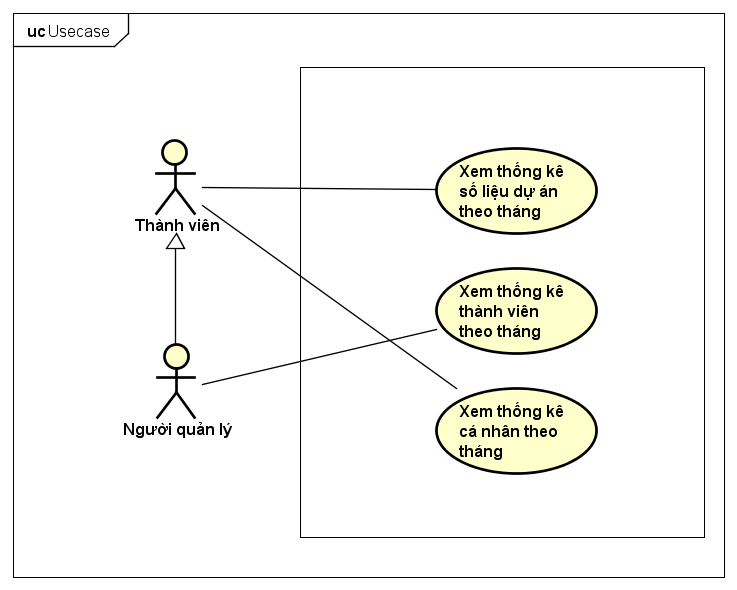
### Biểu đồ use case phân rã Xem báo cáo tiến độ dự án



**Hình 2.5** Phân rã use case Xem báo cáo tiến độ dự án

**Hình 2.5** là biểu đồ phân rã use case Xem báo cáo tiến độ dự án. Người quản lý mới có quyền thực hiện các use case này gồm (i) xem báo cáo tiến độ thời gian hiện tại, (ii) xem báo cáo chi phí hiện tại và (iii) xem báo cáo tình hình thành viên hiện tại.

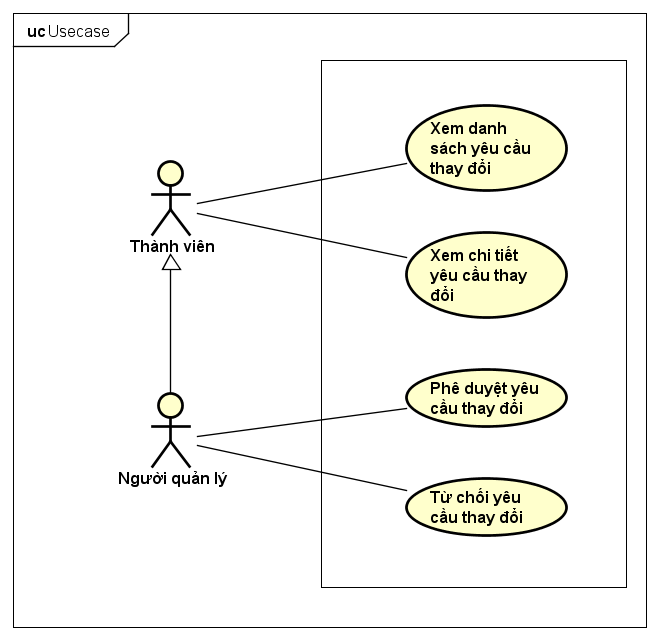
### Biểu đồ use case phân rã Thống kê đánh giá dự án



**Hình 2.6** Phân rã use case Thống kê đánh giá dự án

**Hình 2.6** là biểu đồ phân rã use case Thống kê đánh giá dự án. Thành viên trong dự án chỉ có thể thực hiện được hai use case gồm (i) xem thống kê số liệu dự án theo tháng và (ii) xem thống kê số liệu cá nhân theo tháng. Người quản lý thực hiện được hai use case đã nói trên đồng thời thêm use case thứ ba là xem thống kê thành viên theo tháng.

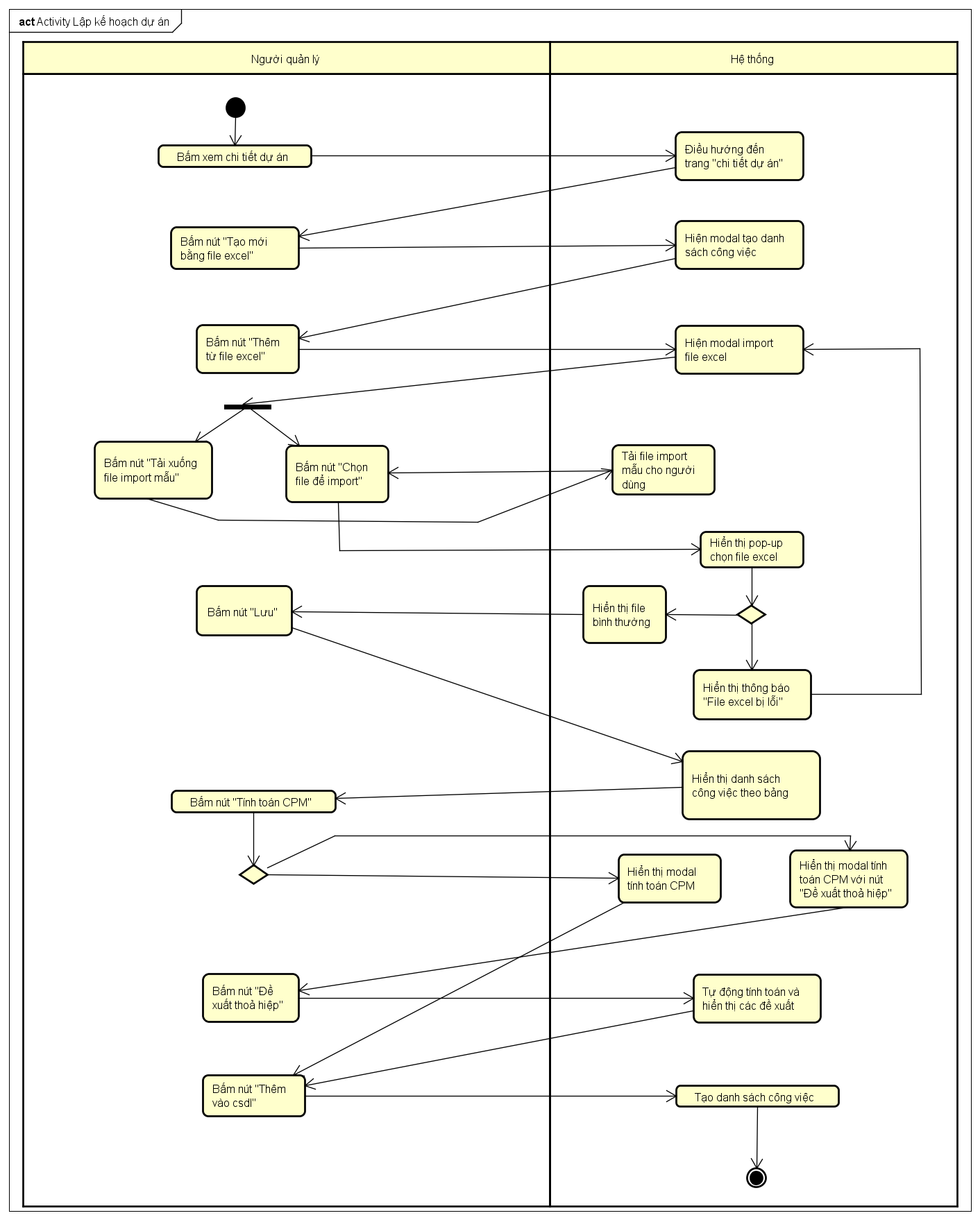
### Biểu đồ use case phân rã Quản lý yêu cầu thay đổi

****

**Hình 2.7** Phân rã use case Quản lý yêu cầu thay đổi

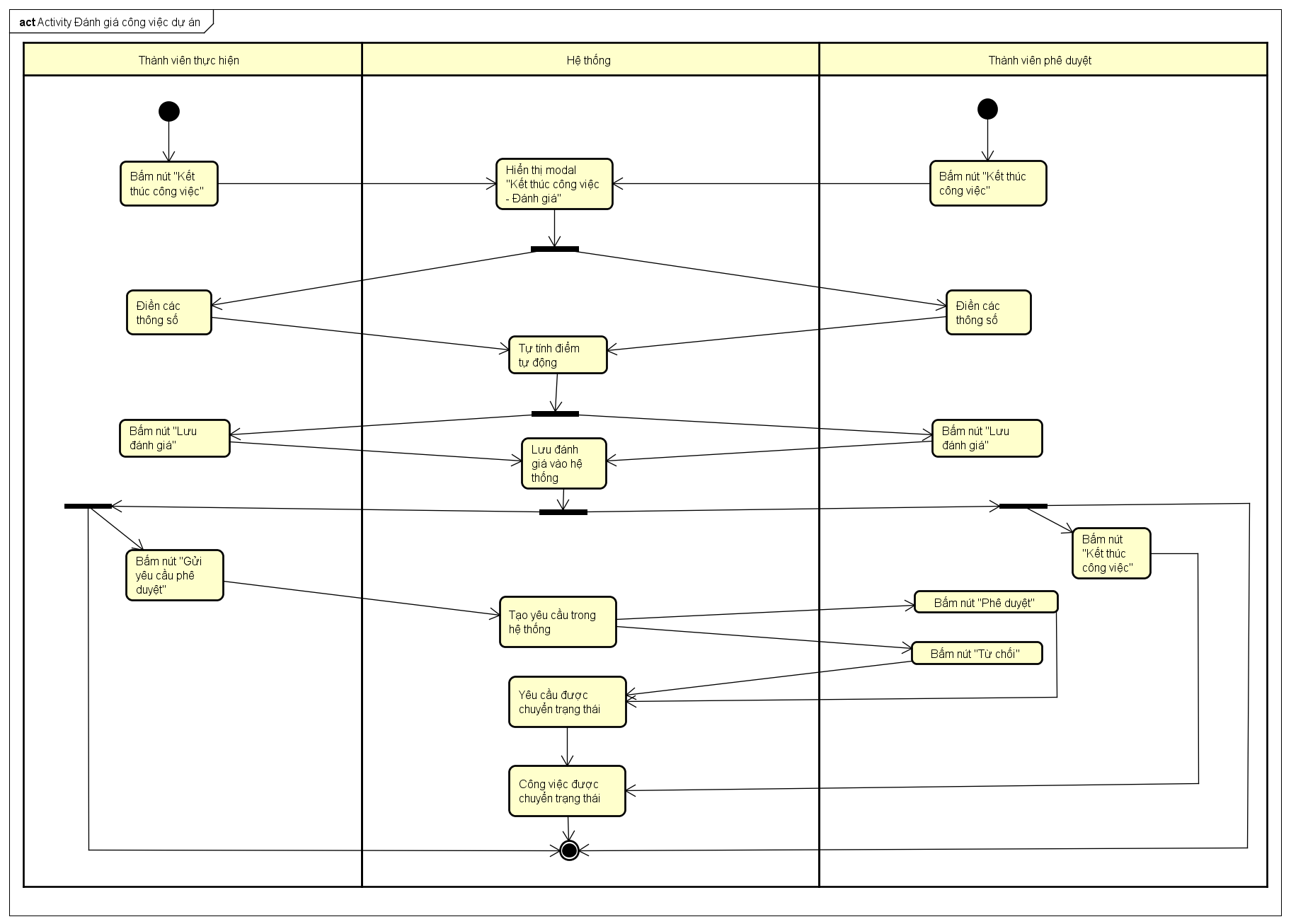
**Hình 2.7** là biểu đồ phân rã use case Quản lý yêu cầu thay đổi. Mọi thành viên trong dự án có thể (i) xem danh sách yêu cầu thay đổi và (ii) xem chi tiết yêu cầu thay đổi. Người quản lý dự án bên cạnh việc thực hiện hai use case đã nói trên, họ còn thêm được hai use case nữa là (i) phê duyệt yêu cầu thay đổi và (ii) từ chối yêu cầu thay đổi.

### Quy trình nghiệp vụ



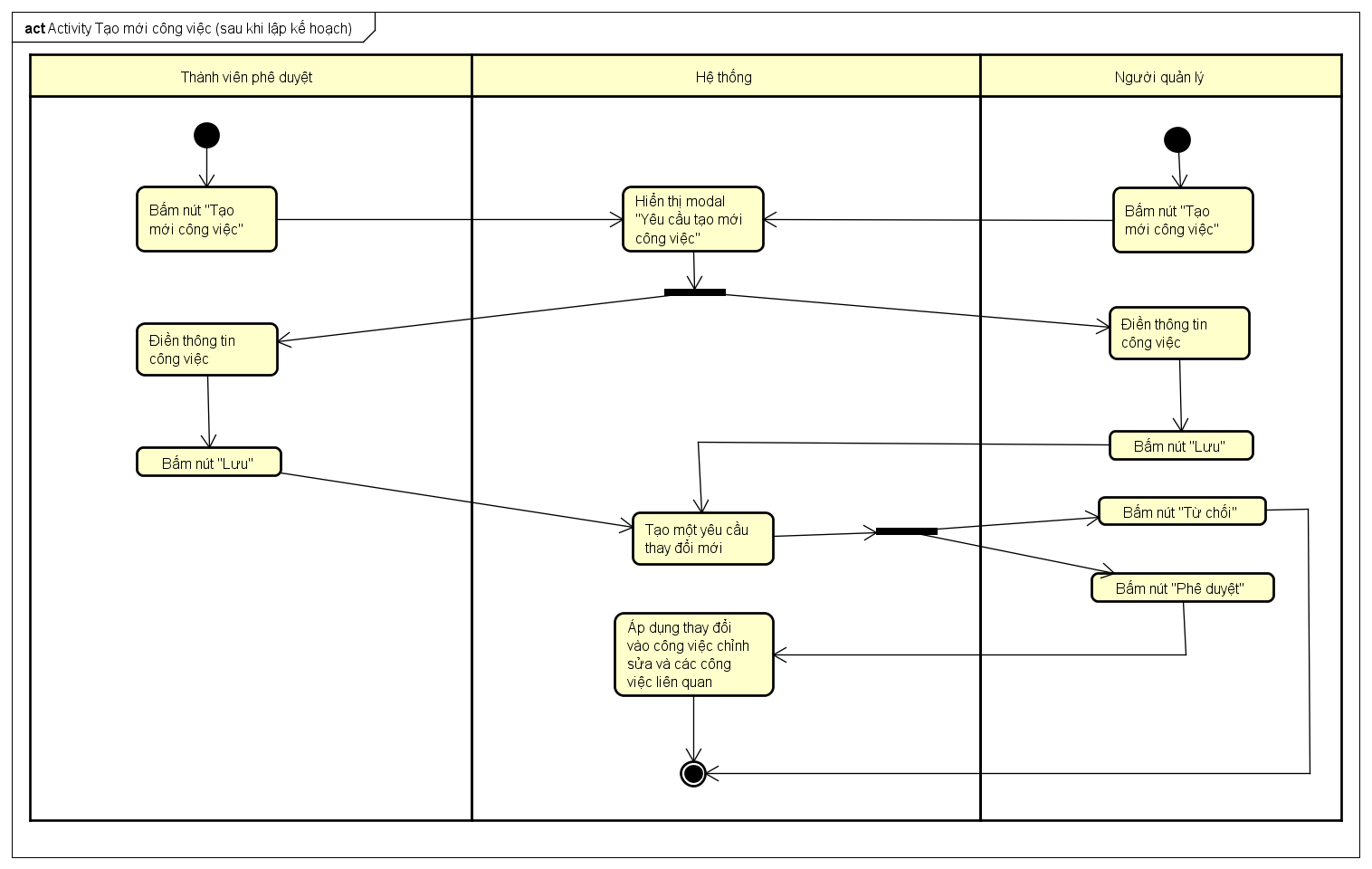
**Hình 2.8** Sơ đồ hoạt động lập kế hoạch dự án

**Hình 2.8** mô tả nghiệp vụ lập kế hoạch cho dự án. Sau khi tạo một dự án mới, người quản lý sẽ bấm vào chi tiết để hệ thống điều hướng sang trang chi tiết dự án. Khi người quản lý bấm nút “Thêm mới công việc từ file excel”, một modal sẽ hiện lên yêu cầu nhập tập tin excel từ máy tính. Sau khi nhập tập tin excel đúng định dạng, một bảng danh sách các công việc theo thông tin có sẵn trong tập tin excel sẽ hiện ra, người quản lý có thể chọn chỉnh sửa từng công việc hoặc bấm nút “Tính toán CPM” để đi đến bước tiếp theo. Ở modal “Tính toán CPM”, dữ liệu về danh sách công việc dự án sẽ cho người quản lý thấy được ngày bắt đầu, ngày kết thúc dự kiến của từng công việc. Nếu thời gian kết thúc tính toán được từ CPM lớn hơn thời gian dự kiến kết thúc ban đầu của dự án, người quản lý có thể sử dụng nút “Tính toán đề xuất” để hiển thị danh sách những phương án đề xuất giảm thời gian cho dự án. Sau khi người quản lý hài lòng với danh sách công việc, bấm nút “Thêm vào cơ sở dữ liệu” để tạo mới các công việc đó.



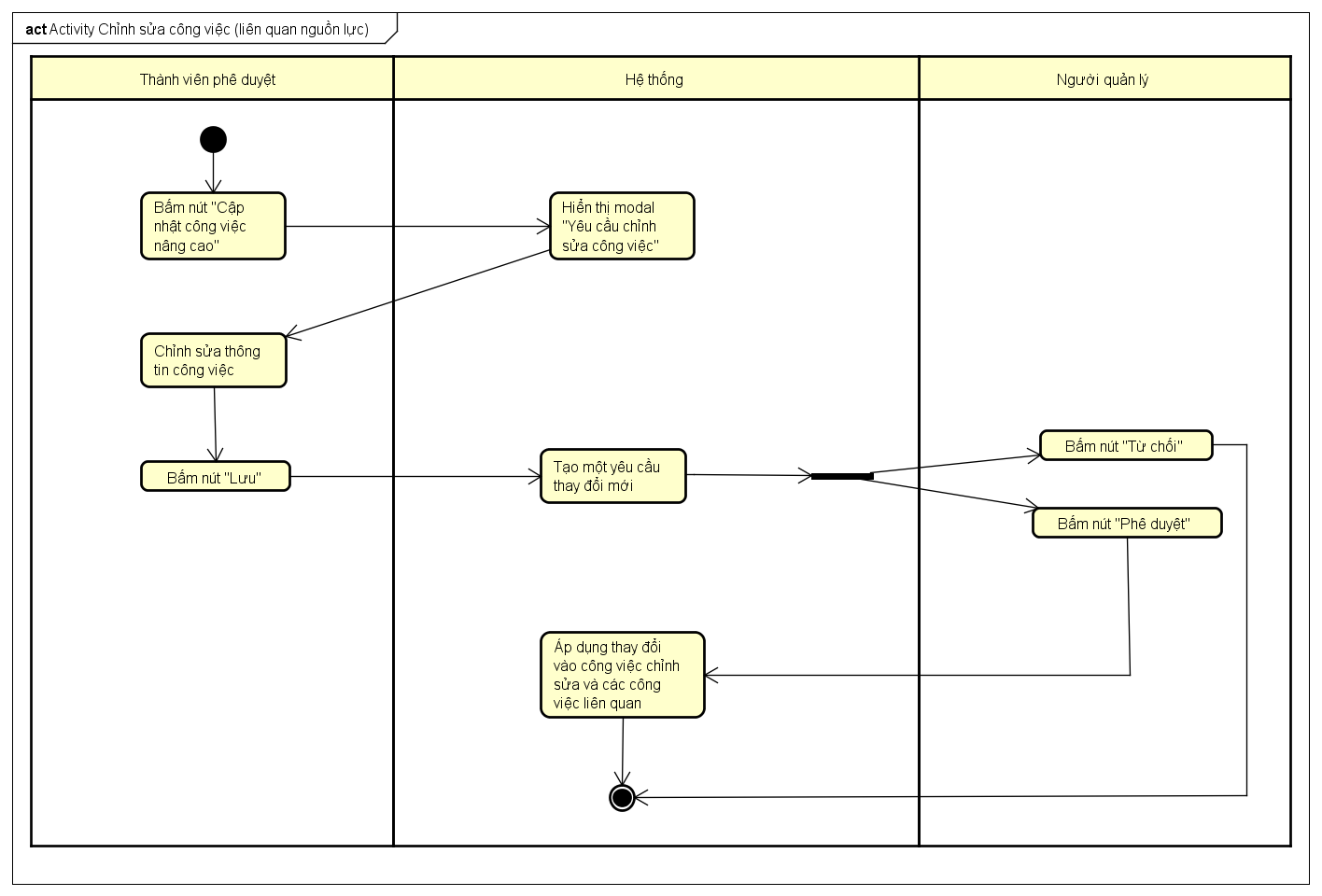
**Hình 2.9** Sơ đồ hoạt động đánh giá công việc dự án

**Hình 2.9** mô tả hoạt động đánh giá công việc trong dự án. Một công việc trong dự án được đánh giá khi công việc này chuẩn bị giai đoạn hồi kết thúc. Thành viên thực hiện và thành viên phê duyệt đều có thể mở modal đánh giá công việc. Khi một bên điền thông số đánh giá xong thì bên kia sẽ nhận được các thông số đã lưu. Việc tính điểm tự động sẽ do hệ thống đảm nhiệm. Hai loại thành viên trong công việc chỉ cần nhập tay thông số công việc, điểm tự đánh giá và những điểm đánh giá bằng tay vào modal đó. Sau đó, thành viên thực hiện có thể gửi “Yêu cầu kết thúc công việc” tới thành viên phê duyệt để chấp nhận hoặc từ chối yêu cầu đó. Bên cạnh đó, thành viên phê duyệt có thể tự bản thân kết thúc công việc mà không cần chờ thành viên thực hiện gửi yêu cầu qua.



**Hình 2.10** Sơ đồ hoạt động tạo mới công việc (sau khi lập kế hoạch)

**Hình 2.10** mô tả hoạt động tạo mới công việc sau khi đã lập kế hoạch. Việc tạo mới công việc này khác với tạo công việc khi lập kế hoạch ở khía cạnh công việc mới này có thể thay đổi phạm vi nguồn lực của dự án. Nguồn lực ở đây bao gồm thời gian, chi phí và nhân lực. Vì thế mỗi khi tạo mới công việc, một yêu cầu thay đổi được hệ thống tự động sinh ra và lưu trong cơ sở dữ liệu. Mọi thành viên trong dự án có quyền tạo công việc mới, xem danh sách các yêu cầu thay đổi và xem chi tiết yêu cầu thay đổi có liên quan tới hành động tạo mới công việc đó. Tuy nhiên, chỉ có người quản lý dự án có quyền phê duyệt hoặc từ chối để yêu cầu tạo mới công việc đó có hiệu lực hoặc bị bãi bỏ.



**Hình 2.11** Sơ đồ hoạt động chỉnh sửa công việc (liên quan tới nguồn lực)

**Hình 2.11** mô tả hoạt động chinh sửa các thuộc tính của công việc có liên quan tới nguồn lực. Như đã trình bày ở **Hình 2.10**, một số hoạt động có thể làm thay đổi và ảnh hưởng tới phạm vi nguồn lực của dự án, việc chỉnh sửa công việc này chính là một trong số đó. Mọi quy trình đều giống với hoạt động tạo công việc ở **Hình 2.10**, tuy nhiên chỉ có thành viên phê duyệt công việc mới có quyền chỉnh sửa các thuộc tính nâng cao này của công việc.

## Đặc tả chức năng

### Đặc tả use case Quản lý dự án

**Bảng 2.2** Bảng đặc tả use case Quản lý dự án

### Đặc tả use case Quản lý công việc

**Bảng 2.3** Bảng đặc tả use case Quản lý công việc

### Đặc tả use case Đánh giá công việc

**Bảng 2.4** Bảng đặc tả use case Đánh giá công việc

### Đặc tả use case Xem báo cáo tiến độ dự án

**Bảng 2.5** Bảng đặc tả use case Xem báo cáo tiến độ dự án

### Đặc tả use case Thống kê đánh giá dự án

**Bảng 2.6** Bảng đặc tả use case Thống kê đánh giá dự án

### Đặc tả use case Quản lý yêu cầu thay đổi

**Bảng 2.7** Bảng đặc tả use case Quản lý yêu cầu thay đổi

## Yêu cầu phi chức năng

### Yêu cầu chung

Ứng dụng cần phải cho phép phân quyền theo các chức năng, vai trò của người dùng. Dữ liệu phải được hiển thị chính xác tuỳ theo từng quyền khi đăng nhập vào hệ thống. Việc hiển thị dữ liệu trên giao diện cũng được thay đổi một cách dễ dàng, linh hoạt để không làm mất đi dữ liệu nhưng giảm được đáng kể lượt chi phí truy vấn. Đồng thời, ứng dụng có tính khả chuyển khi cho phép người dùng sử dụng trên nhiều trình duyệt khác nhau như Google Chrome, Microsoft Edge v.v.

### Yêu cầu về giao diện người dùng

Giao diện người dùng phải luôn đạt được yếu tố dễ dùng, đơn giản hoá các thao tác để người dùng không cần suy nghĩ và mất quá nhiều thời gian khi sử dụng. Màu sắc trong website cần phải đồng nhất ở mọi trang, mọi components, chỉ sử dụng tối đa ba màu trong cùng một trang. Giao diện hỗ trợ song ngữ Anh – Việt và có thể chuyển đổi giữa hai ngôn ngữ một cách nhanh chóng và ngược lại để phù hợp với nhiều đối tượng người dùng khác nhau. Với những nút bấm có ý nghĩa đơn giản, hệ thống sử dụng những icon mang tính gợi nhớ cao và tái sử dụng được ở nhiều nơi. Bên cạnh đó, việc thay đổi kích thước màn hình theo từng loại thiết bị cũng là một yếu tố quan trọng trong thiết kế giao diện người dùng.

### Yêu cầu về bảo mật

Các tiêu chuẩn về bảo mật thông tin được sử dụng (i) các thông tin quan trọng như mật khẩu không được lưu lại ở trình duyệt để tránh bị xâm nhập, (ii) sử dụng phương thức Json web token để mã hoá thông tin đăng nhập người dùng và thực hiện đăng nhập theo phiên, (iii) ghi lại lịch sử hoạt động người dùng trên hệ thống và (iv) luôn có biện pháp để lưu bản sao dữ liệu hệ thống đề phòng mất mát.

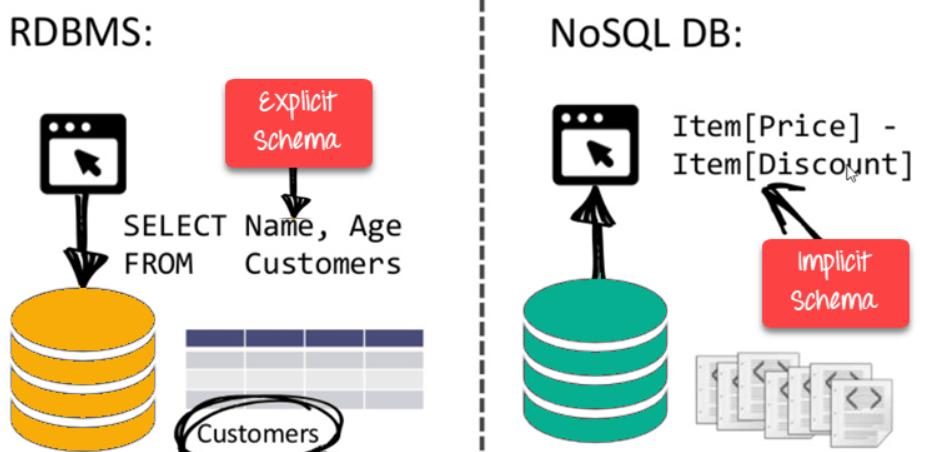
### Yêu cầu khác

Bên cạnh yêu cầu được nêu ở mục 2.4.1, 2.4.2 và 2.4.3, ứng dụng cần được xây dựng để đảm bảo thêm những yêu cầu sau (i) tính khả thi - ứng dụng được xây dựng đáp ứng nhu cầu thực tế của người dùng, (ii) tính linh động - có thể dễ dàng thay đổi theo yêu cầu người dùng và (iii) tính an toàn – dữ liệu luôn được backup thường xuyên để đảm bảo tính toàn vẹn dữ liệu, tránh mất mát do sự cố hay bị tấn công.

# Công nghệ sử dụng

## Công nghệ xây dựng cơ sở dữ liệu

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là cơ sở dữ liệu thuộc NoSQL (Not only SQL – là một dạng cơ sở dữ liệu phi quan hệ). MongoDB là cơ sở dữ liệu hướng tài liệu (document) và sử dụng ngôn ngữ truy vấn không có cấu trúc. Với cơ sở dữ liệu quan hệ (RDBMS – Relational Database Management System) ta sử dụng các khái niệm bảng để lưu trữ các bản ghi thì với cơ sở dữ liệu phi quan hệ ta sử dụng collection để lưu trữ các document kiểu JSON.



**Hình 3.1** Sự khác biệt giữa các lưu trữ dữ liệu NoSQL DB và RDBMS

MongoDB có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế với mục đích hướng đến đối tượng, hoạt động dựa vào các khái niệm Collection và Document. Điều này cho phép dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau nên ta không cần phải kiểm tra sự thoả mãn ràng buộc như trong RDBMS.

MongoDB có tốc độ truy vấn (find, update, insert, delete) nhanh hơn rất nhiều so với RDBMS. Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy tốc độ ghi của MongoDB có thể nhanh gấp 100 lần so với MySQL[[5]](#footnote-5).

Chart, bar chart

Description automatically generated

**Hình 3.2** So sánh tốc độ ghi dữ liệu của MongoDB và MySQL

ĐATN sử dụng Mongoose – trình ánh xạ tài liệu đối tượng (ODM – Object Document Mapper) giúp sử dụng Mongodb dễ dàng hơn bằng cách dịch tài liệu trong Mongodb sang các đối tượng. Mongoose cho phép định nghĩa các object với một schema được định nghĩa rõ ràng, được ánh xạ tới một MongoDB document. Mongoose cung cấp các phương thức truy vấn đa dạng, sử dụng chức năng xâu chuỗi thay vì phải ghi nhớ các câu lệnh, điều này dẫn đến mã nguồn linh hoạt, dễ đọc và dễ bảo trì hơn.

## Công nghệ xây dựng backend

### NodeJS

NodeJS là nền tảng (Platform) được xây dựng trên V8 Javascript Engine – trình thông dịch thực thi mã Javascript, giúp xây dựng phần backend của ứng dựng web QLDA một cách đơn giản và dễ dàng mở rộng. Phần core của NodeJS được xây dựng hầu hết bằng C++ nên có tốc độ xử lý và hiệu năng cao. Các restful API được xây dựng để có thể được sử dụng ở nhiều module khác nhau trong client.

### Express

Express là một trong những framework phổ biến của NodeJS dùng để xây dựng API cho website cũng như là mobile. Express được phát hành lần đầu vào tháng 11 năm 2010 và hiện tại đang ở phiên bản 5.0.0-alpha.8. Express hỗ trợ các giao thức HTTP và middleware tạo ra các API dễ sử dụng và hiệu quả. Express có nhiều package giúp người dùng dễ dàng hơn trong việc xây dựng các ứng dụng. Đồng thời, Express còn hỗ trợ thêm nhiều plugin, bảo mật và an toàn tốt hơn so với việc xây dựng thuần.

## Công nghệ xây dựng frontend

### ReactJS

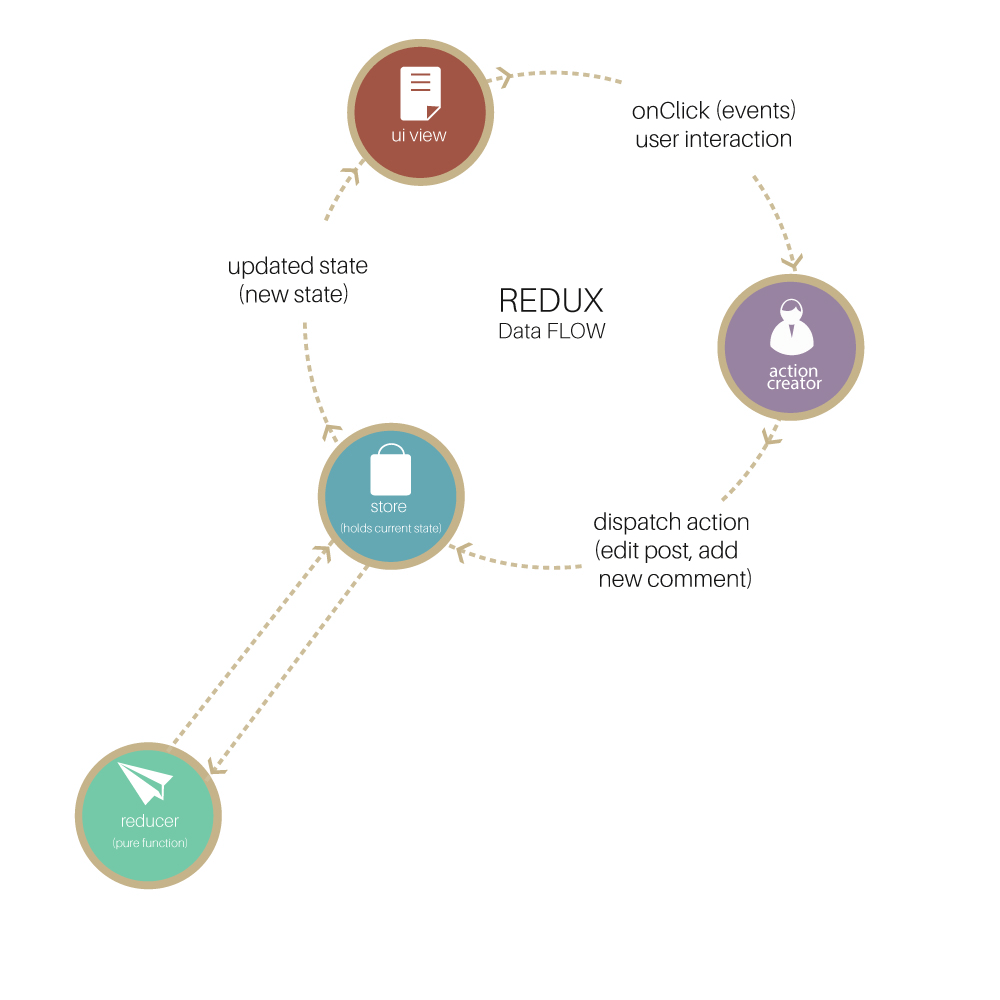
ReactJS, hay còn được gọi là React, là một thư viện mã nguồn mở của JavaScript, được sử dụng để xây dựng giao diện website và các ứng dụng mobile. React được triển khải lần đầu tiên trên newsfeed của Facebook vào năm 2011 và trên Instagram vào năm 2012.

React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện JavaScript khác, trong đó phải kể đến việc React cho phép nhúng code HTML vào trong code JavaScript nhờ JSX, điều đó làm cho tư duy lập trình và những dòng code dễ hiểu hơn. Một điểm nổi bật nữa của React là việc quản lý states và props của các components trong toàn bộ cấu trúc dự án. Các components có thể được xây dựng phân cấp dạng cha con, và components con có thể kế thừa các thuộc tính cũng như các hàm của components cha dựa vào việc truyền props từ components cha. Một component có thể được triển khai theo hai cách: (i) sử dụng class component với life-cycle và (ii) sử dụng functional component với hooks.

### Redux

Ở mục 3.3.1 bên trên có đề cập đến việc React sử dụng props và state để quản lý ứng dụng được viết ra. Tuy nhiên trong một dự án có thể có nhiều states phức tạp mà không thể nào chỉ kiểm soát trong một vài components được. Đó là lý do Redux ra đời.

Redux là một thư viện JavaScript được sử dụng như một kho lưu trữ các trạng thái của các ứng dụng được viết bằng JavaScript. Redux được tạo ra vào tháng 6 năm 2015, và dần trở nên phổ biến nhanh chóng vì tính đơn giản, kích thước nhỏ, tài liệu chi tiết và hướng tiếp cận dễ hiểu. Redux được dùng để duy trì trạng thái của toàn bộ ứng dụng trong một cây trạng thái duy nhất, không thể thay đổi trực tiếp. Khi một dữ liệu nào đó bị thay đổi thì một đối tượng mới được tạo ra (sử dụng actions và reducers).



**Hình 3.3** Luồng dữ liệu của Redux

A close up of a logo

Description automatically generated

**Hình 3.4** Sự khác biệt giữa việc quản lý states dùng Redux và không dùng Redux

### CSS

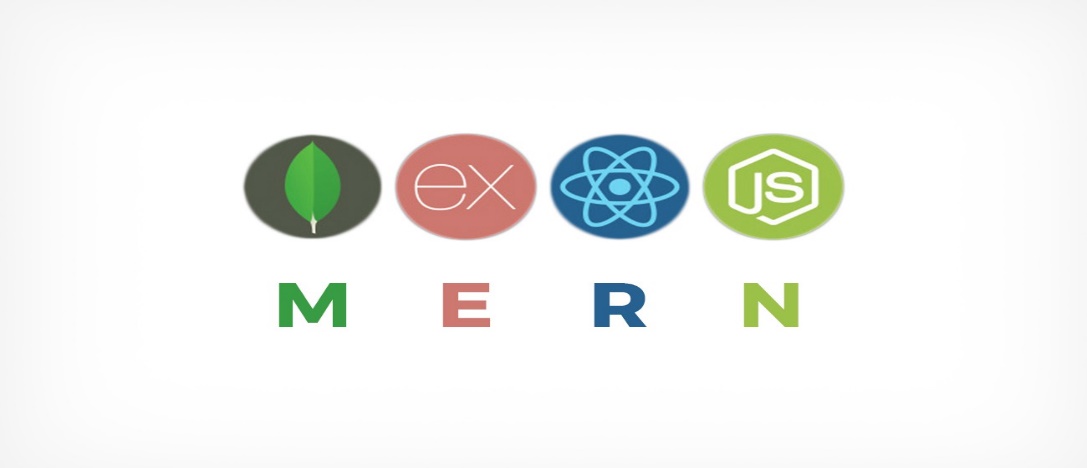
CSS là viết tắt của từ “Cascading Style Sheets”. Nếu ta coi HTML là khung của một giao diện ứng dụng thì CSS là phần da phần thịt đắp lên bộ xương đó để tạo nên một giao diện hoàn chỉnh. CSS được sử dụng để định kiểu cho ứng dụng, thiết kế bố cục và hiển thị cho các thiết bị có kích thước màn hình khác nhau. Việc sử dụng CSS có nhiều cách như (i) dạng inline – mã CSS được nhúng ngay trong thuộc tính style của phần từ HTML, (ii) dạng internal – mã CSS nằm trong thẻ <style/> của văn bản HTML và (iii) dạng external – mã CSS nằm trong một tệp tin riêng biệt sau đó được gọi đến và sử dụng trong tài liệu HTML.

### Bootstrap

Bootstrap là một framework của CSS cho phép xây dựng nhanh giao diện website responsive (tương thích với mọi nền tảng) dựa trên những thành phần đã được xây dựng sẵn trong đó. Bootstrap bao gồm các HTML templates, CSS templates và Javascript để tạo ra các thành phần như navbar, form, v.v. để có thể sử dụng để phát triển giao diện của ứng dụng mà không cần phải tự xây dựng từ đầu.

## Tích hợp các công nghệ

MERN là viết tắt của bốn công nghệ MongoDB, Express, Reactjs và Node.js đã được trình bày ở phần 3.1, 3.2 và 3.3. Các ứng dụng trên có thể kết hợp được với nhau để cùng tạo ra một ứng dụng mà mọi dòng mã được viết bằng Javascript. Đây là ngôn ngữ lập trình xây dựng được cả client lẫn server. Khi kết hợp các công nghệ khác nhau, nhà phát triển cần phải tìm cách tích hợp được sự khác biệt của các ngôn ngữ lập trình. Tuy nhiên, đối với MERN, lập trình viên chỉ cần thành thạo JavaScript và JSON và không cần thiết phải chuyển đồi môi trường. ĐATN đã lựa chọn MERN để xây dựng ứng dụng QLDA với hiệu năng cao.



**Hình 3.5** Tích hợp các công nghệ - MERN

# Phát triển và triển khai ứng dụng

## Thiết kế kiến trúc

### Lựa chọn kiến trúc phần mềm

Hệ thống được xây dựng thoe kiến trúc client – server. Trong mô hình này, server cung cấp API cho phép client thực hiện request thông qua các API đó. Khi nhân được yêu cầu từ client server thực hiện xác thực thông tin người dùng, xử lý dữ liệu, lưu trữ, cập nhật và trả về thông tin cho client. Còn client đóng vai trò cung cấp giao diện tương tác để thu thập thông tin và thực hiện gửi request đến server để yêu cầu xử lý thông tin. Client và server được xây dựng độc lập với nhau. Client được xây dựng dựa trên kiến trúc của Redux và xấy dựng ứng dụng theo từng module. Server được xây dựng theo mô hình MVC.

#### Mô hình kiến trúc của Redux

A picture containing clock

Description automatically generated

**Hình 4.1** Mô hình kiến trúc Redux

Redux bao gồm 3 thành phần chính giúp hỗ trợ việc xử lý và hiển thị dữ liệu nhận về từ phía server là: (i) Actions – là các sự kiện (event), giúp người dùng gửi dữ liệu từ ứng dụng đến kho lưu trữ trạng thái (state) cục bộ của Redux (Redux store), (ii) Reducer – là các function thực hiện lấy các state hiện tại của ứng dụng, thực hiện một action và trả về state mới, (iii) Store – kho lưu trữ state của ứng dụng, không thể trực tiếp sửa đổi mà phải sửa đổi khi thực hiện các Actions.

#### Mô hình MVC

A close up of a logo

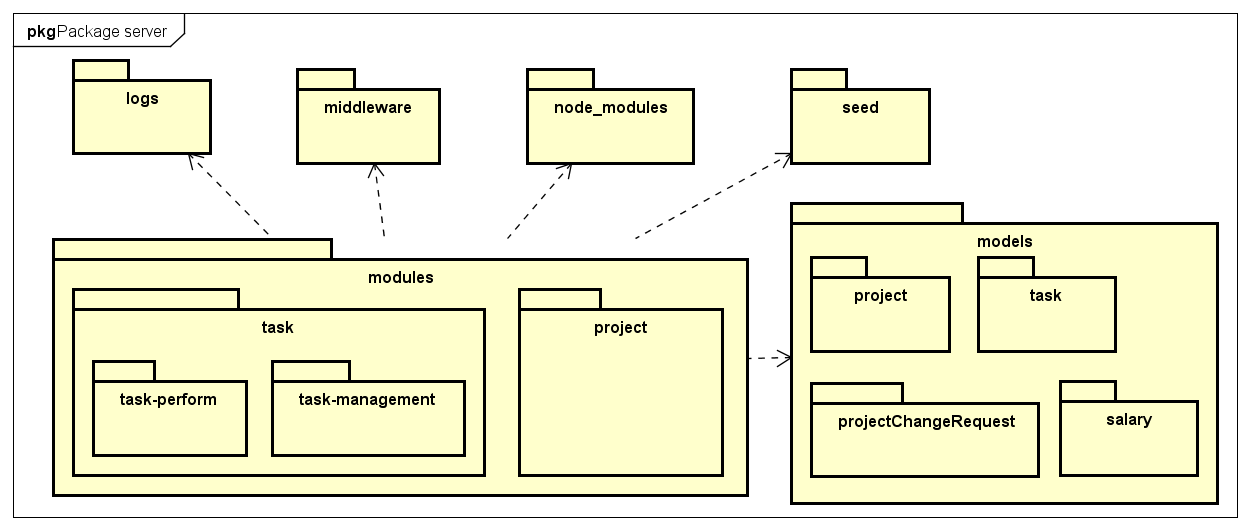
Description automatically generated

**Hình 4.2** Sơ đồ kiến trúc MVC

MVC là một kiến trúc 3 lớp chia ứng dụng thành 3 phần: (i) Model – lưu trữ cấu trúc dữ liệu của ứng dụng, (ii) View – giữ vai trò hiển thị dữ liệu cho người dùng và (iii) Controller – đóng vai trò như một công cụ liên kết giữa giao diện người dùng và cấu trúc dữ liệu, điều hướng các request của người dùng để xử lý dữ liệu.

### Thiết kế tổng quan

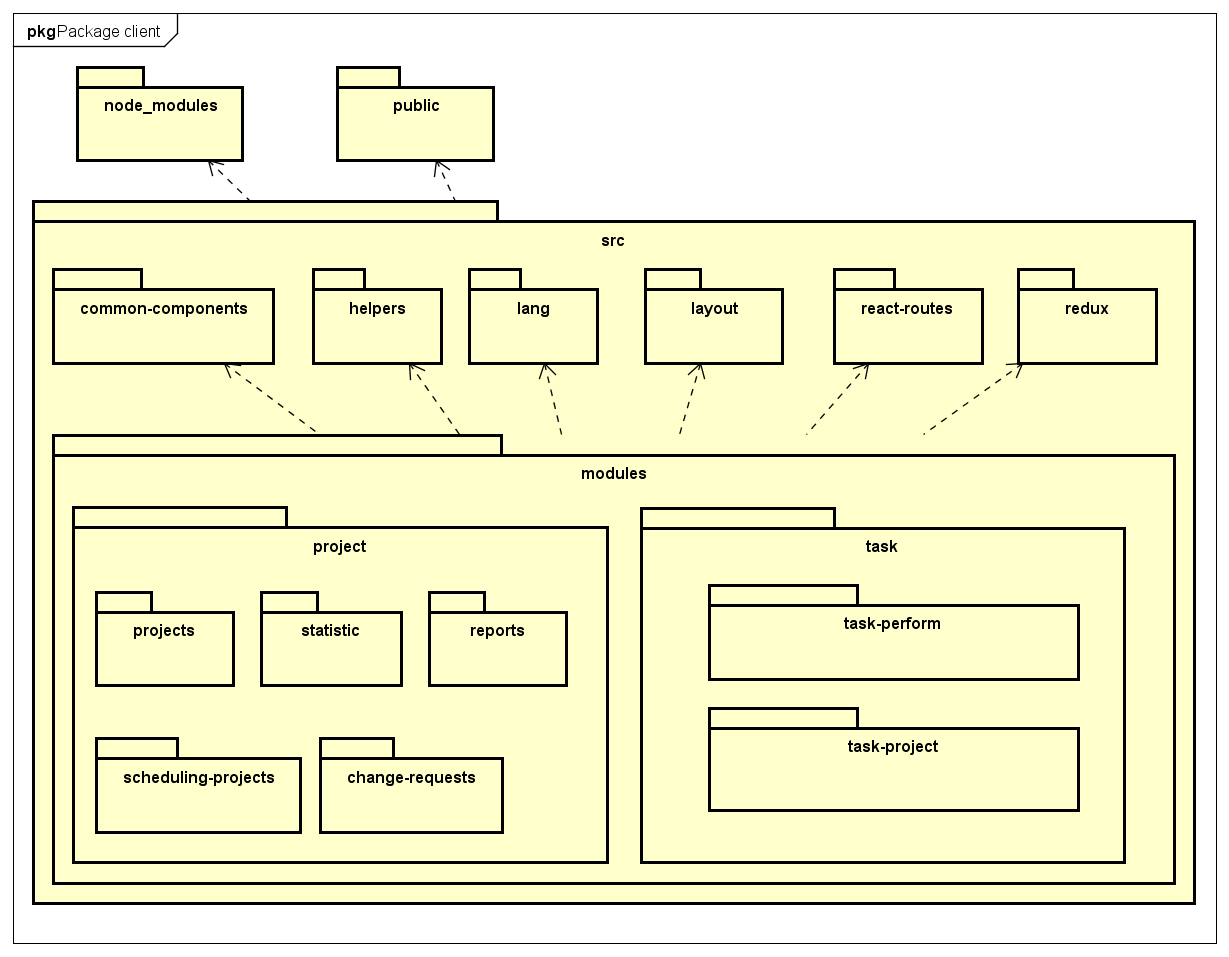
Hình 4.3 mô tả biểu đồ gói tổng quan phần server của ứng dụng QLDA.



**Hình 4.3** Biểu đồ gói tổng quan phần server của ứng dụng QLDA

Phần server bao gồm các gói: (i) logs – ghi lại lịch sử các request được gửi đến từ phía client, (ii) middleware – có vai trò xác thực các request gửi đến, (iii) models – lưu trữ cấu trúc dữ liệu, (iv) node\_modules – các thư viện JavaScript được sử dụng trong ứng dụng, (v) seed – khởi tạo dữ liệu gốc cho server, (vi) modules – các chức năng chính của hệ thống, (vii) task – một chức năng nằm trong modules, có chức năng quản lý công việc gồm task-perform để quản lý đánh giá và task-management để quản lý CRUD công việc, (viii) project – một chức năng nằm trong modules, chức năng quản lý dự án.

Hình 4.4 mô tả biểu đồ gói tổng quan phần client của ứng dụng QLDA

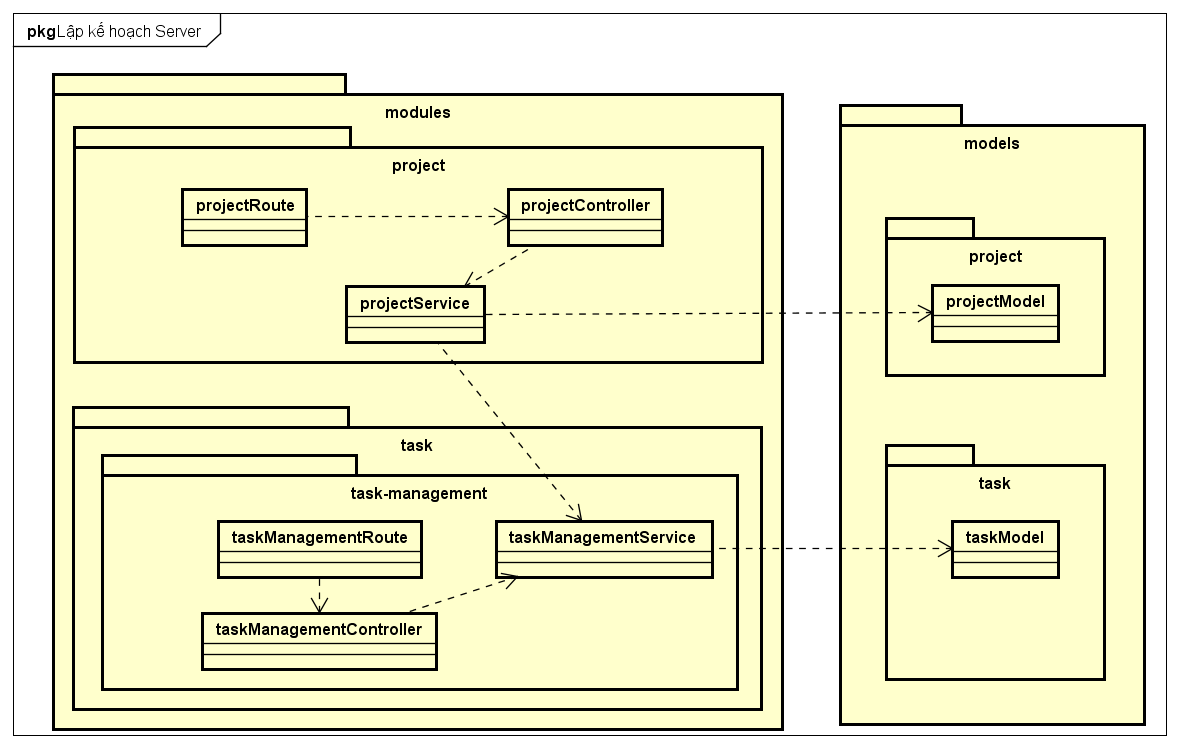


**Hình 4.4** Biểu đồ gói tổng quan phần client của ứng dụng QLDA

Các chức năng chính của phần client nằm trong thư mục src, trong đó: (i) common-components – chứa các component giao diện với mục đích tái sử dụng, (ii) helpers – chứa các hàm tiện ích dùng chung ở nhiều nơi trong client, (iii) lang – định dạng các ngôn ngữ, (iv) layout – cấu trúc cách hiển thị của website, (v) react-routes – kết hợp các trang tương ứng với các chức năng của client, (vi) redux – kho lưu trữ để quản lý các state cục bộ trong toàn ứng dụng, (vii) modules – chứa các chức năng của phần client, (viii) project – xây dựng giao diện ứng dụng QLDA, (ix) projects – xây dựng chức năng CRUD dự án, (x) statistic – xây dựng chức năng thống kê đánh giá dự án, (xi) reports – xây dựng chức năng báo cáo tiến độ dự án, (xii) scheduling-projects – xây dựng chức năng lập kế hoạch dự án (xiii) change-requests – xây dựng chức năng quản lý yêu cầu thay đổi. Ngoài ra ứng dụng cũng sử dụng chung gói task gồm (i) task-perform – chức năng quản lý đánh giá công việc và (ii) task-project – chức năng quản lý CRUD công việc cho dự án.

### Thiết kế chi tiết gói

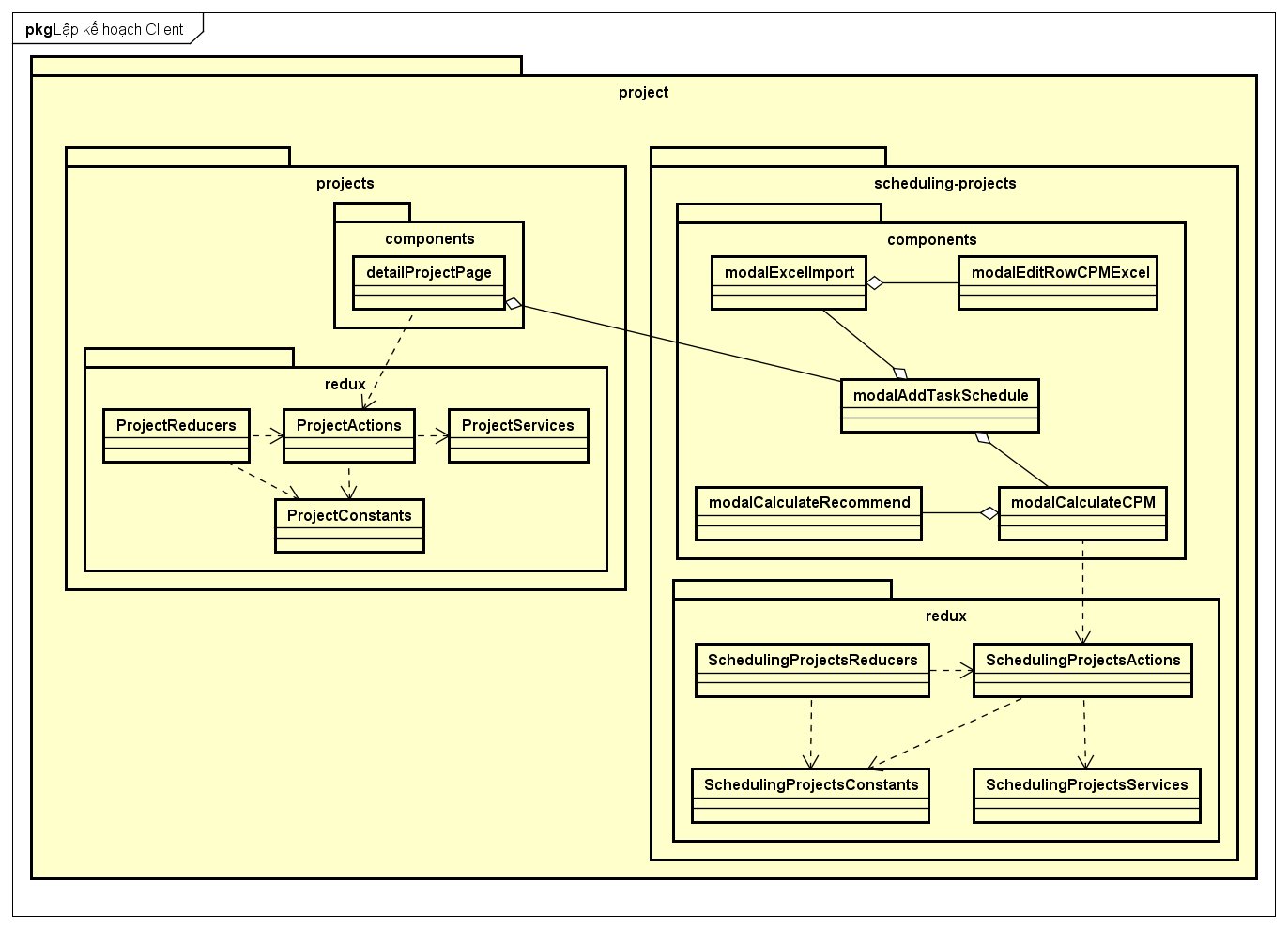
Hình 4.5 mô tả chi tiết gói ở phần server để xây dựng chức năng Lập kế hoạch dự án.



**Hình 4.5** Chi tiết các gói xây dựng chức năng Lập kế hoạch dự án phần server

Có ba gói chính là project, task-management và models. Gói project chứa các class sau: (i) projectRoute – định nghĩa các đường dẫn API như CRUD dự án, (ii) projectController – xử lý request liên quan đến CRUD dự án và trả về dữ liệu cho client, (iii) projectService – xử lý logic nghiệp vụ bên trong, tương tác với các models. Gói task-management chứa các class sau: (i) taskManagementRoute – định nghĩa các đường dẫn API như CRUD công việc, (ii) taskManagementController – xử lý request liên quan đến CRUD công việc và trả về dữ liệu cho client, (iii) taskManagementService – xử lý logic nghiệp vụ bên trong, tương tác với các models. Gói models gồm hai phần là (i) projectModel – định nghĩa cấu trúc dữ liệu của dự án và (ii) taskModel – định nghĩa cấu trúc dữ liệu của công việc.

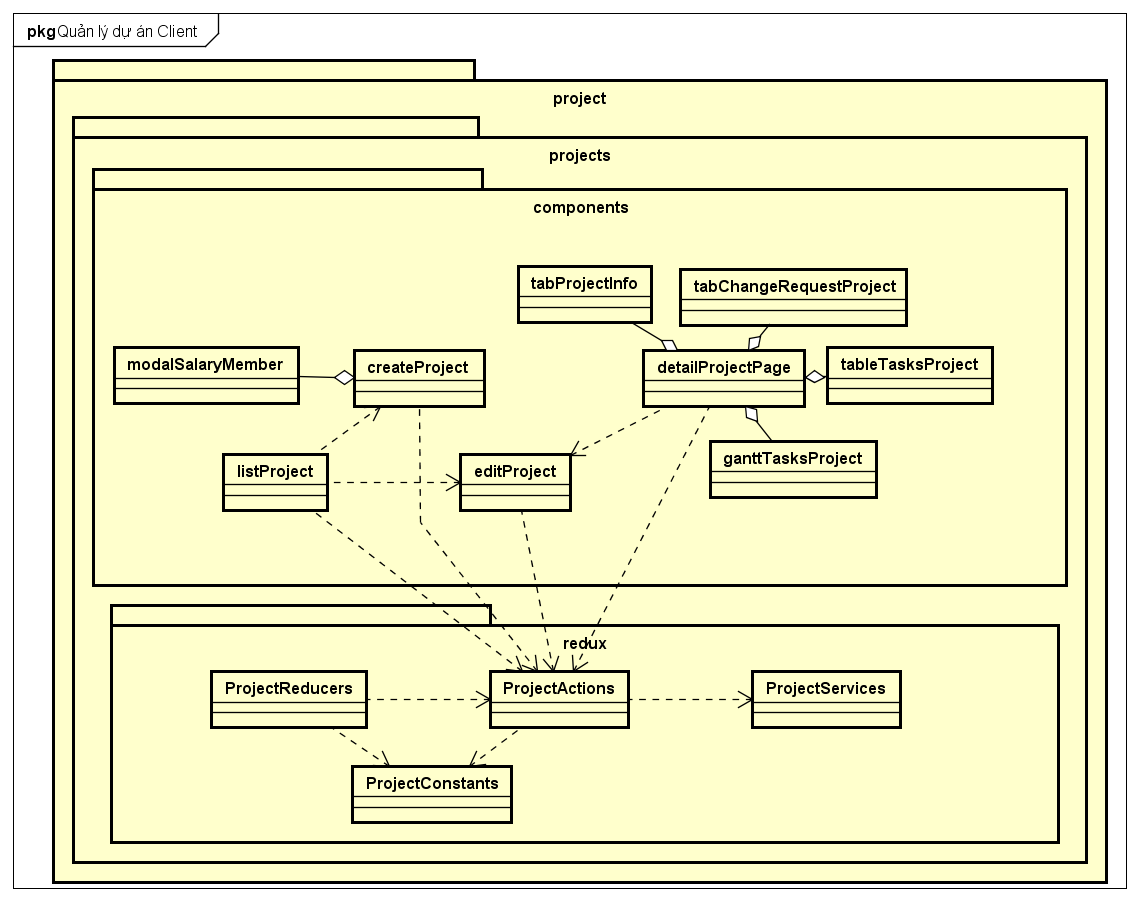
Hình 4.6 mô tả chi tiết gói ở phần client để xây dựng chức năng Lập kế hoạch dự án.



**Hình 4.6** Chi tiết các gói xây dựng chức năng Lập kế hoạch dự án phần client

Chức năng Lập kế hoạch dự án ở client tập trung chủ yếu vào gói scheduling-projects, gói projects là những phần liên kết bên ngoài. Gói scheduling-projects gồm hai gói chính là components và redux: (i) gói components chứa các để hiển thị dữ liệu và thực hiện xử lý dữ liệu, (ii) gói redux lưu trữ trạng thái (state) liên quan đến chức năng lập kế hoạch dự án. Gói components và gói redux được tách riêng ra nhằm mục đích tái sử dụng và bảo trì khi ứng dụng được mở rộng. các thành phần của redux như class SchedulingProjectsAction, SchedulingProjectsConstant, SchedulingProjectsService, SchedulingProjectsReducer được tái sử dụng ở các gói components khác trong phần client.

Hình 4.7 mô tả chi tiết gói ở phần client để xây dựng chức năng Quản lý dự án.



**Hình 4.7** Chi tiết các gói xây dựng chức năng Quản lý dự án phần client

Gói projects gồm hai gói chính là components và redux: (i) gói components chứa các để hiển thị dữ liệu và thực hiện xử lý dữ liệu, (ii) gói redux lưu trữ trạng thái (state) liên quan đến chức năng CRUD dự án. Gói components và gói redux được tách riêng ra nhằm mục đích tái sử dụng và bảo trì khi ứng dụng được mở rộng. các thành phần của redux như class ProjectAction, ProjectConstant, ProjectService, ProjectReducer được tái sử dụng ở các gói components khác trong phần client.

## Thiết kế chi tiết

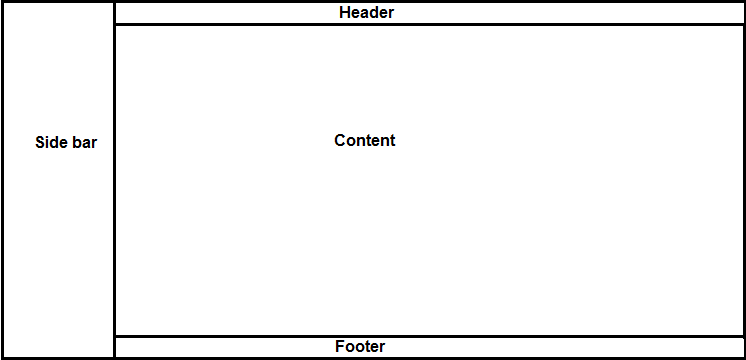
### Thiết kế giao diện

Giao diện được thiết kế tối ưu cho các màn hình laptop và máy tính để bàn. Tuy nhiên, giao diện cũng cần đáp ứng được cho thiết bị nhỏ hơn như điện thoại thông minh hay máy tính bảng. Thông tin trong giao diện đáp ứng phải đáp ứng đầy đủ những gì người dùng cần đồng thời che dấu các thông tin không cần thiết. Đối với các dạng dữ liệu có kích thước hiển thị trên màn hình quá lớn thì tjam ẩn hoặc chỉ hiển thị khi người dùng yêu cầu thông tinh bằng cách sử dụng modal cho phép người dùng xem chi tiết thông tin.

Giao diện ứng dụng có tính tương tác cao với người dùng, có thể phản hồi thông tin người dùng nhanh chóng khi tương tác với hệ thống. Ngoài ra giao diện ứng dụng mang tính gợi ý cho người dùng, giúp người dùng có thể tương tác với hệ thống mà không cần tìm hiểu nhiều, bằng cách sử dụng các icon thông dụng và các thanh điều hướng.

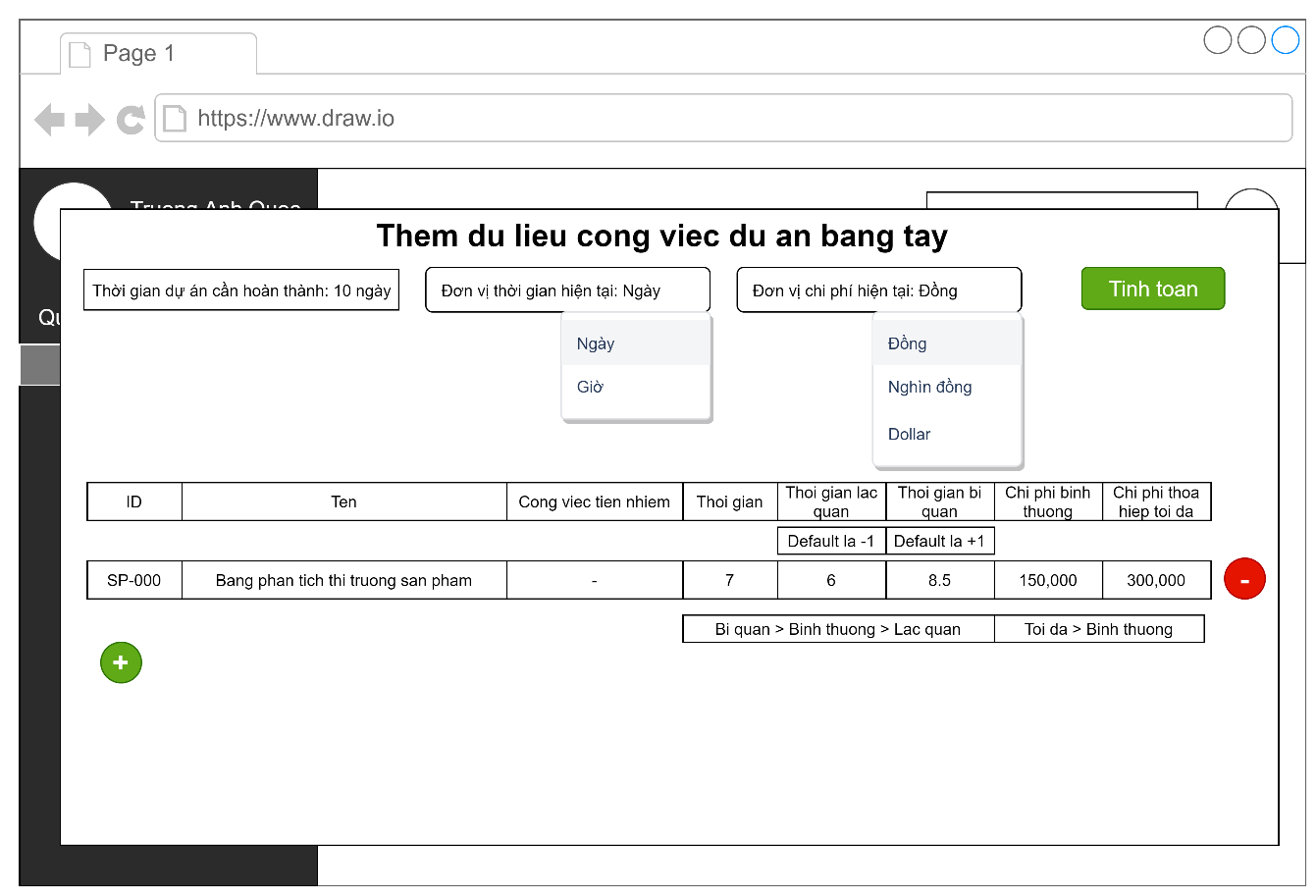
Màu sắc được sử dụng trong ứng dụng thể hiện vai trò của từng thành phần. Ví dụ màu xanh lục tượng trưng cho việc thêm mới, các thống báo thành công. Màu vàng thể hiện cho chức năng chỉnh sửa. Màu đỏ thể hiện cho việc xoá, thông báo lỗi. Bên cạnh việc xây dựng ứng dụng thân thiện với người dùng thì việc xây dựng ứng dụng cũng hạn chế tối đa việc sử dụng quá nhiều màu sắc, và ưu tiên xây dựng giao diện với màu sắc đơn giản nhưng bắt mắt.

Phần giao diện chung của ứng dụng sẽ gồm có bốn phần chính là (i) header – phần thanh tiêu đề trang, (ii) sidebar – phần menu điều hướng, (iii) content – phần nội dụng trang và (iv) footer – phần phía cuối của trang.



**Hình 4.8** Layout giao diện mock-up chung của ứng dụng

**Hình 4.9** là thiết kế giao diện chức năng phần lập kế hoạch dự án.



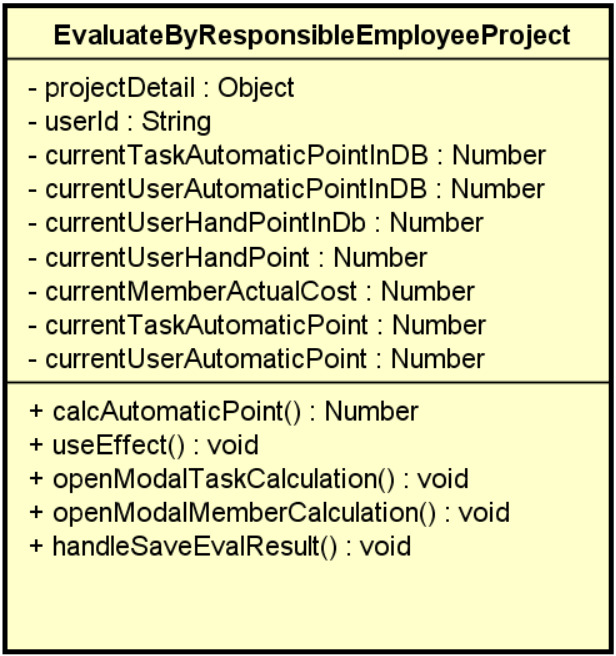
**Hình 4.9** Giao diện mock-up lập kế hoạch dự án

### Thiết kế lớp

Dưới đây là phần trình bày một số lớp chủ đạo của ứng dụng QLDA.

**Hình 4.10** mô tả thiết kế lớp EvaluateByResponsibleEmployeeProject – có vai trò trong việc tạo form đánh giá công việc với vai trò thành viên thực hiện. Các thuộc tính của lớp bao gồm điểm số tự động, điểm số đã được lưu trong cơ sở dữ liệu, và điểm số thành viên tự đánh giá bằng tay. Các hàm của lớp là những hàm thực hiện tính toán, thực hiện mở modal chi tiết điểm số tự động và thực hiện lưu kết quả đánh giá.

**Hình 4.11** mô tả thiết kế lớp AddProjectTaskForm – có vai trò trong việc tạo một công việc mới sau khi đã lập kế hoạch (có liên quan đến sự thay đổi nguồn lực). Các thuộc tính của lớp bao gồm thông số của công việc mới, một biến để lưu trữ thông tin dự án hiện tại và danh sách người dùng của toàn bộ hệ thống để hỗ trợ chuyển đổi id sang tên. Các hàm của lớp là những hàm thực hiện thay đổi trạng thái công việc hiện tại khi điền thông tin vào form, những hàm kiểm tra xem thông tin nhập vào có hợp lệ không.

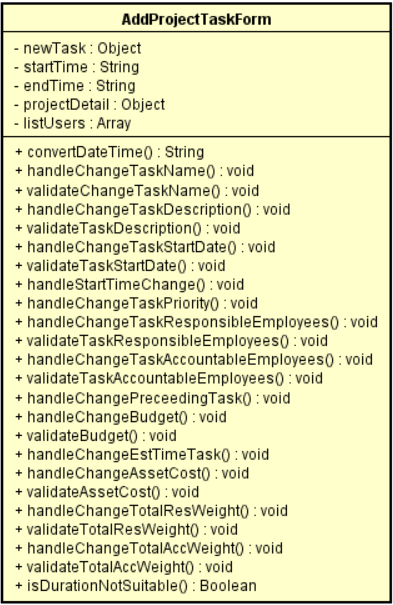


**Hình 4.10** Thiết kế chi tiết lớp EvaluateByResponsibleEmployeeProject

**Bảng 4.1** dưới đây là bảng đặc tả chi tiết thuộc tính và phương thức của lớp EvaluateByResponsibleEmployeeProject.

**Bảng 4.1** Bảng đặc tả chi tiết thuộc tính và phương thức của lớp EvaluateByResponsibleEmployeeProject

|  |  |  |  |
| --- | --- | --- | --- |
| Tên lớp: | EvaluateByResponsibleEmployeeProject | | |
| Danh sách thuộc tính | | | |
| Tên thuộc tính | **Kiểu dữ liệu** | | **Ý nghĩa** |
| projectDetail | Object | | Thông tin dự án hiện tại |
| userId | String | | Id của người dùng hiện tại |
| currentTaskAutomaticPointInDB | Number | | Điểm số công việc tự động đang được lưu trong CSDL |
| currentUserAutomaticPointInDB | Number | | Điểm số thành viên tự động đang được lưu trong CSDL |
| currentUserHandPointInDb | Number | | Điểm số thành viên nhập tay đang được lưu trong CSDL |
| currentUserHandPoint | Number | | Điểm số thành viên nhập tay hiện tại |
| currentMemberActualCost | Number | | Chi phí thực của thành viên hiện tại |
| currentTaskAutomaticPoint | Number | | Điểm số công việc tự động đang tính hiện tại |
| currentUserAutomaticPoint | Number | | Điểm số thành viên tự động đang tính hiện tại |
| Danh sách phương thức | | | |
| Tên phương thức | **Đầu vào** | **Đầu ra** | **Ý nghĩa** |
| useEffect | void | void | React hook để kiểm soát re-render component mỗi khi có state thay đổi |
| calcAutomaticPoint | String | Number | Tính toán điểm tự động |
| openModalTaskCalculation | void | void | Mở modal chi tiết điểm số công việc tự động |
| openModalMemberCalculation | void | void | Mở modal chi tiết điểm số thành viên tự động |
| handleSaveEvalResult | void | void | Lưu kết quả đánh giá |



**Hình 4.11** Thiết kế chi tiết lớp AddProjectTaskForm

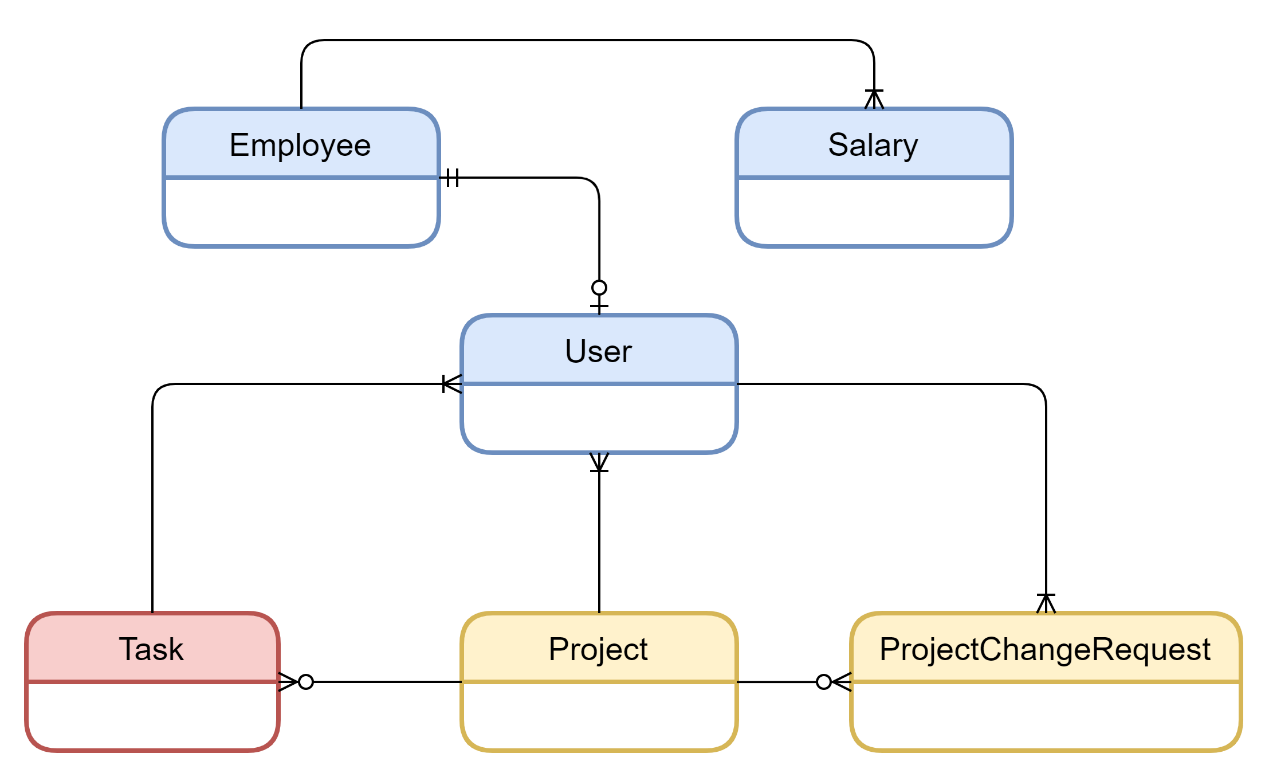
**Bảng 4.2** dưới đây là bảng đặc tả chi tiết thuộc tính và phương thức của lớp AddProjectTaskForm.

**Bảng 4.2** Bảng đặc tả chi tiết thuộc tính và phương thức của lớp AddProjectTaskForm

|  |  |  |  |
| --- | --- | --- | --- |
| Tên lớp: | AddProjectTaskForm | | |
| Danh sách thuộc tính | | | |
| Tên thuộc tính | **Kiểu dữ liệu** | | **Ý nghĩa** |
| newTask | Object | | Thông tin công việc được thêm mới |
| startTime | String | | Thời gian bắt đầu công việc |
| endTime | String | | Thời gian dự kiến kết thúc công việc |
| projectDetail | Object | | Thông tin dự án hiện tại |
| listUsers | Array | | Danh sách người dùng của hệ thống |
| Danh sách phương thức | | | |
| Tên phương thức | **Đầu vào** | **Đầu ra** | **Ý nghĩa** |
| convertDateTime | String, String | String | Ghép ngày và giờ thành một chuỗi hoàn chỉnh |
| handleChangeTaskName | String | void | Thay đổi tên công việc |
| validateChangeTaskName | String, Boolean | Boolean | Kiểm tra tên công việc |
| handleChangeTaskDescription | String | void | Thay đổi mô tả công việc |
| validateTaskDescription | String, Boolean | Boolean | Kiểm tra mô tả công việc |
| handleChangeTaskStartDate | String | void | Thay đổi ngày bắt đầu công việc |
| validateTaskStartDate | String, Boolean | Boolean | Kiểm tra ngày bắt đầu công việc |
| handleStartTimeChange | String | void | Thay đổi giờ bắt đầu công việc |
| handleChangeTaskPriority | String | void | Thay đổi độ ưu tiên công việc |
| handleChangeTaskResponsibleEmployees | String | void | Thay đổi người thực hiện |
| validateTaskResponsibleEmployees | String, Boolean | Boolean | Kiểm tra người thực hiện |
| handleChangeTaskAccountableEmployees | String | void | Thay đổi người phê duyệt |
| validateTaskAccountableEmployees | String, Boolean | Boolean | Kiểm tra người phê duyệt |
| handleChangePreceedingTask | Array | void | Thay đổi danh sách công việc tiền nhiệm |
| handleChangeBudget | String | void | Thay đổi ước lượng chi phí cho công việc |
| validateBudget | String, Boolean | Boolean | Kiểm tra ước lượng chi phí cho công việc |
| handleChangeEstTimeTask | String | void | Thay đổi khoảng thời gian dự kiến hoàn thành công việc |
| handleChangeAssetCost | String | void | Thay đổi chi phí tài sản |
| validateAssetCost | String, Boolean | Boolean | Kiểm tra chi phí tài sản |
| handleChangeTotalResWeight | String | void | Thay đổi tổng trọng số thành viên thực hiện |
| validateTotalResWeight | String, Boolean | Boolean | Kiểm tra tổng trọng số thành viên thực hiện |
| handleChangeTotalAccWeight | String | void | Thay đổi tổng trọng số thành viên phê duyệt |
| validateTotalAccWeight | String, Boolean | Boolean | Kiểm tra tổng trọng số thành viên phê duyệt |
| isDurationNotSuitable | String | Boolean | Kiểm tra khoảng thời gian công việc có hợp lệ không |

### Thiết kế cơ sở dữ liệu

#### 4.2.3.1 Thực thể liên kết



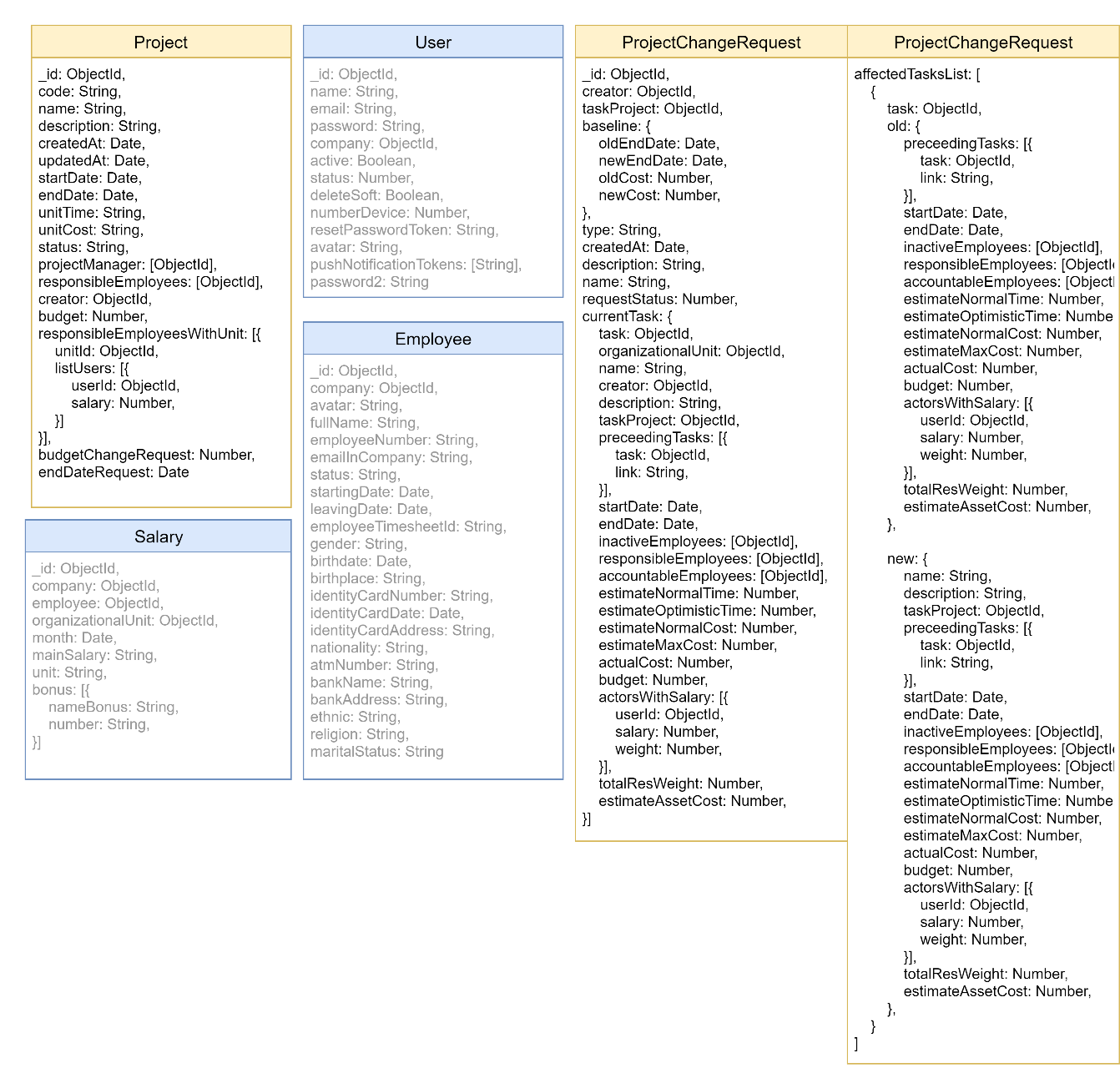
**Hình 4.12** Sơ đồ thực thể liên kết của ứng dụng QLDA

**Hình 4.12** Sơ đồ thực thể liên kết của ứng dụng QLDA là sơ đồ thực thể liên kết của ứng dụng QLDA. Màu sắc của các thực thể trên sơ đồ có ý nghĩa như sau (i) màu vàng cam thể hiện cách thực thể mà ứng dụng ĐATN xây dựng, (ii) màu hồng thể hiện những thực thể xây dựng chung với module khác, và (iii) màu xanh thể hiện các thực thể trong cơ sở dữ liệu của toàn hệ thống mà ứng dụng liên kết đến.

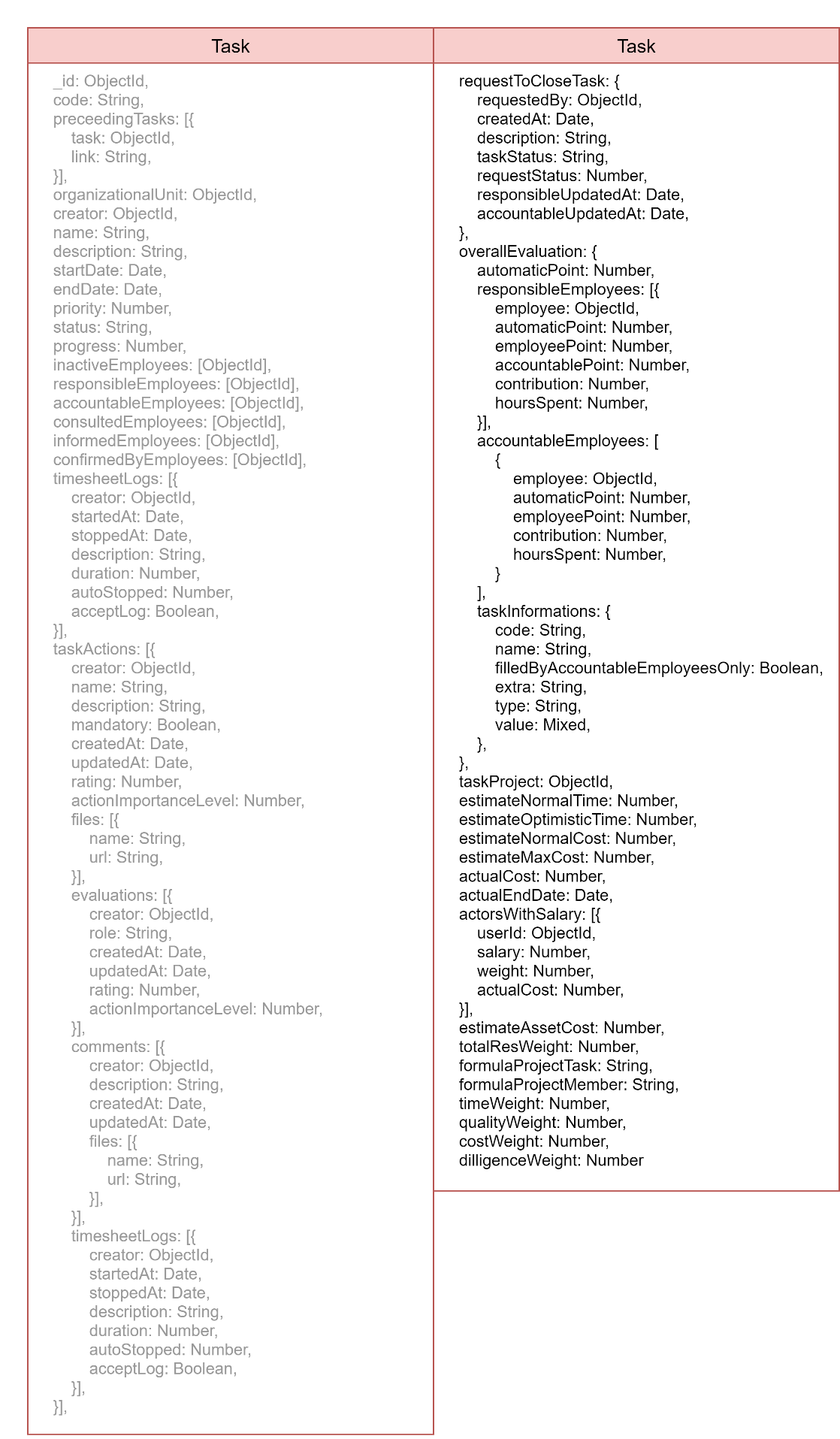
Ý nghĩa của từng thực thể được mô tả trong **Bảng 4.3**, chi tiết các thuộc tính của các thực thể mà ứng dụng ĐATN xây dựng được mô tả trong **Hình 4.13** và **Hình 4.14**.

**Bảng 4.3** Giải thích ý nghĩa của các thực thể liên kết

| Tên thực thể | Ý nghĩa của thực thể |
| --- | --- |
| Project | Dự án |
| ProjectChangeRequest | Yêu cầu thay đổi trong dự án |
| Task | Công việc |
| User | Người dùng hệ thống |
| Employee | Nhân viên |
| Salary | Bảng lương |



**Hình 4.13** Chi tiết thuộc tính của các thực thể liên kết Project, Salary, ProjectChangeRequest, Employee, User



**Hình 4.14** Chi tiết thuộc tính của thực thể Task

#### 4.2.3.2 Thiết kế chi tiết cơ sở dữ liệu

Thiết kế cơ sở dữ liệu collection Project

{

\_id: ObjectId,

code: String,

name: String,

description: String,

createdAt: Date,

updatedAt: Date,

startDate: Date,

endDate: Date,

unitTime: String,

unitCost: String,

status: String,

projectManager: [ObjectId],

responsibleEmployees: [ObjectId],

creator: ObjectId,

budget: Number,

responsibleEmployeesWithUnit: [{

unitId: ObjectId,

listUsers: [{

userId: ObjectId,

salary: Number,

}]

}],

budgetChangeRequest: Number,

endDateRequest: Date

}

Thiết kế cơ sở dữ liệu collection ProjectChangeRequest (phần 1)

{

\_id: ObjectId,

creator: ObjectId,

taskProject: ObjectId,

baseline: {

oldEndDate: Date,

newEndDate: Date,

oldCost: Number,

newCost: Number,

},

type: String,

createdAt: Date,

description: String,

name: String,

requestStatus: Number,

currentTask: {

task: ObjectId,

organizationalUnit: ObjectId,

name: String,

creator: ObjectId,

description: String,

taskProject: ObjectId,

preceedingTasks: [{

task: ObjectId,

link: String,

}],

startDate: Date,

endDate: Date,

inactiveEmployees: [ObjectId],

responsibleEmployees: [ObjectId],

accountableEmployees: [ObjectId],

estimateNormalTime: Number,

estimateOptimisticTime: Number,

estimateNormalCost: Number,

estimateMaxCost: Number,

actualCost: Number,

budget: Number,

actorsWithSalary: [{

userId: ObjectId,

salary: Number,

weight: Number,

}],

totalResWeight: Number,

estimateAssetCost: Number,

},

}

Thiết kế cơ sở dữ liệu collection ProjectChangeRequest (phần 2)

{

affectedTasksList: [{

task: ObjectId,

old: {

preceedingTasks: [{

task: ObjectId,

link: String,

}],

startDate: Date,

endDate: Date,

inactiveEmployees: [ObjectId],

responsibleEmployees: [ObjectId],

accountableEmployees: [ObjectId],

estimateNormalTime: Number,

estimateOptimisticTime: Number,

estimateNormalCost: Number,

estimateMaxCost: Number,

actualCost: Number,

budget: Number,

actorsWithSalary: [{

userId: ObjectId,

salary: Number,

weight: Number,

}],

totalResWeight: Number,

estimateAssetCost: Number,

},

new: {

name: String,

description: String,

taskProject: ObjectId,

preceedingTasks: [{

task: ObjectId,

link: String,

}],

startDate: Date,

endDate: Date,

inactiveEmployees: [ObjectId],

responsibleEmployees: [ObjectId],

accountableEmployees: [ObjectId],

estimateNormalTime: Number,

estimateOptimisticTime: Number,

estimateNormalCost: Number,

estimateMaxCost: Number,

actualCost: Number,

budget: Number,

actorsWithSalary: [{

userId: ObjectId,

salary: Number,

weight: Number,

}],

totalResWeight: Number,

estimateAssetCost: Number,

},

}]

}

Thiết kế cơ sở dữ liệu collection User

{

\_id: ObjectId,

name: String,

email: String,

password: String,

company: ObjectId,

active: Boolean,

status: Number,

deleteSoft: Boolean,

numberDevice: Number,

resetPasswordToken: String,

avatar: String,

pushNotificationTokens: [String],

password2: String

}

Thiết kế cơ sở dữ liệu collection Employee

{

\_id: ObjectId,

company: ObjectId,

avatar: String,

fullName: String,

employeeNumber: String,

emailInCompany: String,

status: String,

startingDate: Date,

leavingDate: Date,

employeeTimesheetId: String,

gender: String,

birthdate: Date,

birthplace: String,

identityCardNumber: String,

identityCardDate: Date,

identityCardAddress: String,

nationality: String,

atmNumber: String,

bankName: String,

bankAddress: String,

ethnic: String,

religion: String,

maritalStatus: String

}

Thiết kế cơ sở dữ liệu collection Salary

{

\_id: ObjectId,

company: ObjectId,

employee: ObjectId,

organizationalUnit: ObjectId,

month: Date,

mainSalary: String,

unit: String,

bonus: [{

nameBonus: String,

number: String,

}]

}

Thiết kế cơ sở dữ liệu collection Task (phần 1)

{

\_id: ObjectId,

code: String,

preceedingTasks: [{

task: ObjectId,

link: String,

}],

organizationalUnit: ObjectId,

creator: ObjectId,

name: String,

description: String,

startDate: Date,

endDate: Date,

priority: Number,

status: String,

progress: Number,

inactiveEmployees: [ObjectId],

responsibleEmployees: [ObjectId],

accountableEmployees: [ObjectId],

consultedEmployees: [ObjectId],

informedEmployees: [ObjectId],

confirmedByEmployees: [ObjectId],

timesheetLogs: [{

creator: ObjectId,

startedAt: Date,

stoppedAt: Date,

description: String,

duration: Number,

autoStopped: Number,

acceptLog: Boolean,

}],

taskActions: [{

creator: ObjectId,

name: String,

description: String,

mandatory: Boolean,

createdAt: Date,

updatedAt: Date,

rating: Number,

actionImportanceLevel: Number,

files: [{

name: String,

url: String,

}],

evaluations: [{

creator: ObjectId,

role: String,

createdAt: Date,

updatedAt: Date,

rating: Number,

actionImportanceLevel: Number,

}],

comments: [{

creator: ObjectId,

description: String,

createdAt: Date,

updatedAt: Date,

files: [{

name: String,

url: String,

}],

}],

timesheetLogs: [{

creator: ObjectId,

startedAt: Date,

stoppedAt: Date,

description: String,

duration: Number,

autoStopped: Number,

acceptLog: Boolean,

}],

}],

}

Thiết kế cơ sở dữ liệu collection Task (phần 2)

{

requestToCloseTask: {

requestedBy: ObjectId,

createdAt: Date,

description: String,

taskStatus: String,

requestStatus: Number,

responsibleUpdatedAt: Date,

accountableUpdatedAt: Date,

},

overallEvaluation: {

automaticPoint: Number,

responsibleEmployees: [{

employee: ObjectId,

automaticPoint: Number,

employeePoint: Number,

accountablePoint: Number,

contribution: Number,

hoursSpent: Number,

}],

accountableEmployees: [

{

employee: ObjectId,

automaticPoint: Number,

employeePoint: Number,

contribution: Number,

hoursSpent: Number,

}

],

taskInformations: {

code: String,

name: String,

filledByAccountableEmployeesOnly: Boolean,

extra: String,

type: String,

value: Mixed,

},

},

taskProject: ObjectId,

estimateNormalTime: Number,

estimateOptimisticTime: Number,

estimateNormalCost: Number,

estimateMaxCost: Number,

actualCost: Number,

actualEndDate: Date,

actorsWithSalary: [{

userId: ObjectId,

salary: Number,

weight: Number,

actualCost: Number,

}],

estimateAssetCost: Number,

totalResWeight: Number,

formulaProjectTask: String,

formulaProjectMember: String,

timeWeight: Number,

qualityWeight: Number,

costWeight: Number,

dilligenceWeight: Number

}

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

**Bảng 4.4** Danh sách thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| IDE lập trình | Visual Studio Code | https://code.visualstudio.org/ |
| Lập trình phía server | Node.js | https://nodejs.org |
| Lập trình phía server | Express | https://expressjs.com |
| Lập trình phía client | ReactJS | https://reactjs.org |
| Quản lý state cho ReactJS | Redux | https://redux.js.org |
| Hệ quản trị cơ sở dữ liệu | MongoDB | https://www.mongodb.com |
| Thao tác với cơ sở dữ liệu thông qua code | Mongoose | https://mongoosejs.com |

### Kết quả đạt được

Sau khi nghiên cứu, triển khai và phát triển, em đã xây dựng được ứng dụng QLDA và được tích hợp được vào hệ thống không gian làm việc số dxclan.com[[6]](#footnote-6) của công ty Cổ phần Công nghệ An toàn thông tin và truyền thông Việt Nam. Các chức năng chính của ĐATN đã triển khai và áp dụng được vào dữ liệu thực tế là: (i) lập kế hoạch dự án, (ii) quản lý tiến độ dự án, (iii) báo cáo tự động dự án hiện tại, (iv) thống kê đánh giá theo tháng và(v) quản lý yêu cầu thay đổi (vi).

**Bảng 4.5** Thống kê thông tin ứng dụng

|  |  |
| --- | --- |
| **Thông tin** | **Thống kê** |
| Số lớp trong mã nguồn (file jsx) | lớp |
| Số gói trong mã nguồn | gói |
| Số gói trực tiếp chứa mã nguồn | gói |
| Số lượng file js react-redux | file |
| Số lượng file js backend | file |
| Số bảng CSDL được sử dụng trong hệ thống | bảng |
| Số bảng CSDL thiết kế cho ứng dụng | bảng |

### Minh hoạ các chức năng chính

Sinh viên lựa chọn và đưa ra màn hình cho các chức năng chính, quan trọng, và thú vị nhất. Mỗi giao diện cần phải có lời giải thích ngắn gọn. Khi giải thích, sinh viên có thể kết hợp với các chú thích ở trong hình ảnh giao diện.

## Kiểm thử

### Compatibility Testing

**Bảng 4.6** Thống kê kiểm thử tưởng thích trên các thiết bị

| Thiết bị | Thông tin thiết bị | Giao diện | Chức năng |
| --- | --- | --- | --- |
| HP laptop 15-bs0xx | Màn hình: 15.6’’, Ram: 16GB | Đạt | Đạt |
| Samsung LED s19b150b | Màn hình: 17.7’’ HD, Ram: 16GB | Đạt | Đạt |
| Samsung Galaxy J7 Pro | Màn hình: 5.7” HD, Ram: 16GB | Đạt | Đạt |

### Black Box Testing

**Bảng 4.7** Kiểm thử nhóm chức năng Lập kế hoạch dự án

| Chức năng | Giá trị đầu vào | Đầu ra cần đạt | Kết quả |
| --- | --- | --- | --- |
| Import file excel danh sách công việc dự án | Nhấn nút “Thêm từ file excel” | Bấm được nút “Lưu” và hiển thị được danh sách các công việc dưới dạng bảng | Đạt |
| Import file excel danh sách công việc dự án không giống như mẫu | Nhấn nút “Thêm từ file excel” | Hiển thị dòng chữ đỏ báo lỗi “File không hợp lệ” và không bấm nút “Lưu” được | Đạt |
| Import file excel danh sách công việc dự án có một số không điền email | Nhấn nút “Thêm từ file excel” | Những công việc không điền email thì “người phê duyệt” và “người thực hiện” để trống. “Chi phí thông thường” và “Chi phí thoả hiệp tối đa” chưa tính được | Đạt |
| Làm mới danh sách công việc | Nhấn nút refresh | Danh sách trở về không còn công việc nào | Đạt |
| Tính toán CPM với danh sách công việc hiện tại | Nhấn nút “Tính toán CPM” | Hiển thị modal “Tính toán CPM” | Đạt |
| Hiển thị modal “Tính toán CPM” khi thời gian tính được lớn hơn thời gian dự kiến | Khi hiển thị modal “Tính toán CPM” | Không hiển thị nút “Tính toán đề xuất thoả hiệp dự án” | Đạt |
| Hiển thị modal “Tính toán CPM” khi thời gian tính được nhỏ hơn thời gian dự kiến | Khi hiển thị modal “Tính toán CPM” | Hiển thị nút “Tính toán đề xuất thoả hiệp dự án” | Đạt |
| Tính toán đề xuất thoả hiệp dự án (có thể thoả hiệp được) | Bấm nút “Tính toán” trong modal “Tính toán đề xuất thoả hiệp dự án” | Hiển thị danh sách các bước gồm (i) tên công việc (ii) thời gian cần rút (iii) chi phí cần tăng | Đạt |
| Tính toán đề xuất thoả hiệp dự án (không thể thoả hiệp được) | Bấm nút “Tính toán” trong modal “Tính toán đề xuất thoả hiệp dự án” | Hiển thị thông báo “Không có kế hoạch giảm thời gian” kèm theo những gợi ý | Đạt |
| Tạo danh sách công việc dự án | Bấm nút “Thêm vào cơ sở dữ liệu” | Đóng các modal và danh sách công việc dạng bảng hiện ra ở màn hình “Chi tiết dự án” | Đạt |

**Bảng 4.8** Kiểm thử nhóm chức năng Báo cáo tự động dự án

| Chức năng | Giá trị đầu vào | Đầu ra cần đạt | Kết quả |
| --- | --- | --- | --- |
| Xem danh sách báo cáo với tự cách là thành viên dự án | Nhấn nút “Báo cáo dự án” ở phần sidemenu | Không hiển thị dự án đó trong bảng danh sách | Đạt |
| Xem danh sách báo cáo với tự cách là quản lý dự án | Nhấn nút “Xem báo cáo” | Hiển thị dự án đó trong bảng danh sách | Đạt |
| Xem chi tiết “Tiến độ dự án” | Nhấn vào tab “Tiến độ dự án” | Hiển thị “tổng quan tiến độ dự án công việc” | Đạt |
| Xem chi tiết “Chi phí dự án” | Nhấn vào tab “Chi phí dự án” | Hiển thị “tổng quan chi phí dự án công việc” và “biểu đồ chi phí dự án công việc” | Đạt |
| Xem chi tiết “Thành viên dự án” | Nhấn vào tab “Thành viên dự án” | Hiển thị “ Các thành viên điểm số cao”, “Các thành viện hay bị chậm tiến độ”, “Các thành viên hay để lãng phí chi phí” và “Tổng quan thành viên dự án” | Đạt |

## Triển khai

Ứng dụng QLDA được tích hợp vào hệ thống không gian làm việc số với tên miền dxclan.com[[7]](#footnote-7). Ngoài ra ứng dụng còn có thể được cài đặt và thử nghiệm trên server localhost của máy tính. Ứng dụng QLDA hoạt động được trên cả ba hệ điều hành là Linux, Window và MacOS. Các bước để thử nghiệm sản phẩm trên server ubuntu 16.04 với bộ nhớ RAM 2GB như sau:

Bước 1: Cài đặt và cấu hình môi trường (Git, Node.js, MongoDB, NPM).

Bước 2: Clone mã nguồn.

Bước 3: Chạy lệnh “npm install” trên cả hai thư mục client và server của project để cài đặt thư viện.

Bước 4: Cấu hình các file biến môi trường (.env) cho cả client và server.

Bước 5: Di chuyển đến thư mục seed của phần server và chạy lệnh “npm run init-db” để tạo dữ liệu mẫu cho hệ thống.

Bước 6: Chạy “npm run dev” ở server và “npm start” ở client để khởi chạy project.

# Các giải pháp và đóng góp nổi bật

## Thuật toán đề xuất giải pháp thoả hiệp chi phí - thời gian

### Vấn đề

Trong giai đoạn lập kế hoạch dự án, người quản lý có thể dựa trên nhiều phương pháp và kinh nghiệm bản thân để đưa ra được giải pháp hợp lý cho từng hoàn cảnh. Trong đó, “Đường găng - CPM” là một trong những phương pháp phổ biến khi lập kế hoạch dự án. Không chỉ tổ chức danh sách công việc một cách hợp lý, “Đường găng - CPM” còn có thể đưa ra những thuật toán gợi ý khi việc lập kế hoạch không phù hợp với dự kiến ban đầu của người quản lý dự án.

Đường găng của dự án là đường xuyên mạng đi từ thời điểm khởi công dự án tới thời điểm kết thúc dự án, có chiều dài trên trục thời gian lớn nhất, qua các công việc có dự trữ thời gian toàn phần bằng không[[8]](#footnote-8). Điều này có nghĩa là, công việc găng là những công việc chỉ có đúng một khoảng thời gian thực hiện mà không có dự trữ thời gian. Đường găng sẽ cho chúng ta chúng ta biết công việc nào buộc phải theo kế hoạch, vì nếu thay đổi sẽ ảnh hưởng ngay đến tiến độ của dự án. ĐATN của em sẽ tạo danh sách công việc trong dự án dựa trên phương pháp đường găng – Critical Path Method (CPM)

Sau khi tạo được danh sách các công việc và tính toán được đường găng, thời gian dự kiến kết thúc của dự án, người quản lý sẽ gặp một vấn đề là thời gian yêu cầu kết thúc có thể nhỏ hơn so với thời gian tính toán dự kiến từ phương pháp đường găng. Ta thử xét một ví dụ sau.

**Bảng 5.1** Danh mục các hoạt động của dự án bảo dưỡng hai cỗ máy

|  |  |  |  |
| --- | --- | --- | --- |
| **Hoạt động** | **Mô tả** | **Hoạt động tiền nhiệm** | **Thời gian kỳ vọng (ngày)** |
| A | Kiểm tra máy I | - | 7 |
| B | Điều chỉnh máy I | A | 3 |
| C | Kiểm tra máy II | - | 6 |
| D | Điều chỉnh máy II | C | 3 |
| E | Hệ thống kiểm tra | B, D | 2 |

Từ danh sách các hoạt động trên, ta thực hiện tính toán đường găng để ra được bảng kế tiếp

**Bảng 5.2** Lịch trình hoạt động của dự án bảo dưỡng hai cỗ máy

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Hoạt động** | **ES** | **LS** | **EF** | **LF** | **Thời gian dự trữ** | **Đường găng** |
| A | 0 | 0 | 7 | 7 | 0 | Có |
| B | 7 | 7 | 10 | 10 | 0 | Có |
| C | 0 | 1 | 6 | 7 | 1 |  |
| D | 6 | 7 | 9 | 10 | 1 |  |
| E | 10 | 10 | 12 | 12 | 0 | Có |

Dựa vào thông tin thời gian dự trữ bằng không, đường găng của dự án này được xác định là đường đi (A-B-E). Do vậy, chiều dài của đường găng, hay chính là tổng thời gian cần thiết để hoàn thành dự án là 12 ngày.

Vậy giả sử trình độ sản xuất hiện tại yêu cầu bắt buộc hoàn thành dự án bảo dưỡng trong vòng 10 ngày, mà tổng thời gian đường găng theo tính toán CPM bên trên là 12 ngày, ta sẽ giải quyết bài toán này như thế nào? Và cách giải quyết đó sẽ cần sự tính toán bằng tay của người lập kế hoạch hay sẽ được thực thi tự động bằng thuật toán?

### Giải pháp

Như vấn đề được đặt ra trong bài toán nêu trên, việc đáp ứng thời gian hoàn thành dự án theo mong muốn là không thể được trừ phi ta có thể rút ngắn thời gian hoạt động một cách có chọn lọc. Việc rút ngắn thời gian hoạt động, mà thông thường có thể đạt được bằng cách bổ sung nguồn lực, được xem như là “thoả hiệp”. Tuy nhiên, các nguồn lực bổ sung đi liền với các thời gian hoạt động thoả hiệp thường dẫn đến chi phí dự án gia tăng. Do vậy, yêu cầu đặt ra cho giải pháp này là ta cần xác định các hoạt động đòi hỏi ít chi phí nhất để thoả hiệp và tiếp theo thoả hiệp những hoạt động chỉ bằng khoản thời gian cần thiết để đáp ứng thời gian hoàn thành dự án theo mong muốn. Đây chính là thuật toán “thoả hiệp chi phí – thời gian” (cost/time tradeoff).

Để thực thi thuật toán này, ta cần có thông tin các dữ liệu đầu vào gồm (i) ti là thời gian kỳ vọng của hoạt động i, (ii) t’i là thời gian của hoạt động i theo thoả hiệp thấp nhất (iii) Ci là chi phí của hoạt động i theo kỳ vọng, (iv) C’i là chi phí hoạt động i theo thoả hiệp cao nhất.

Ta gọi Mi là mức rút giảm thời gian tối đa có thể của hoạt động i theo thoả hiệp, được xác định bởi công thức:

Mi = ti – t’i

Ta gọi Ki là chi phí thoả hiệp của mỗi hoạt động i theo đơn vị thời gian, được xác định bởi công thức:

Ki =

Với những công thức và tham số đầu vào trên, ta có thể viết lại danh sách các hoạt động của dự án “bảo dưỡng hai cỗ máy” thành một bảng mới như sau.

**Bảng 5.3** Dữ liệu về hoạt động bình thường và thoả hiệp của dự án

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Hoạt động** | **Thời gian (ngày)** | | **Tổng chi phí** | | **Mi** | **Ki** |
| **Bình thường** | **Thoả hiệp** | **Bình thường** | **Thoả hiệp** |  |  |
| A | 7 | 4 | 500 | 800 | 3 | 100 |
| B | 3 | 2 | 200 | 350 | 1 | 150 |
| C | 6 | 4 | 500 | 900 | 2 | 200 |
| D | 3 | 1 | 200 | 500 | 2 | 150 |
| E | 2 | 1 | 300 | 550 | 1 | 250 |

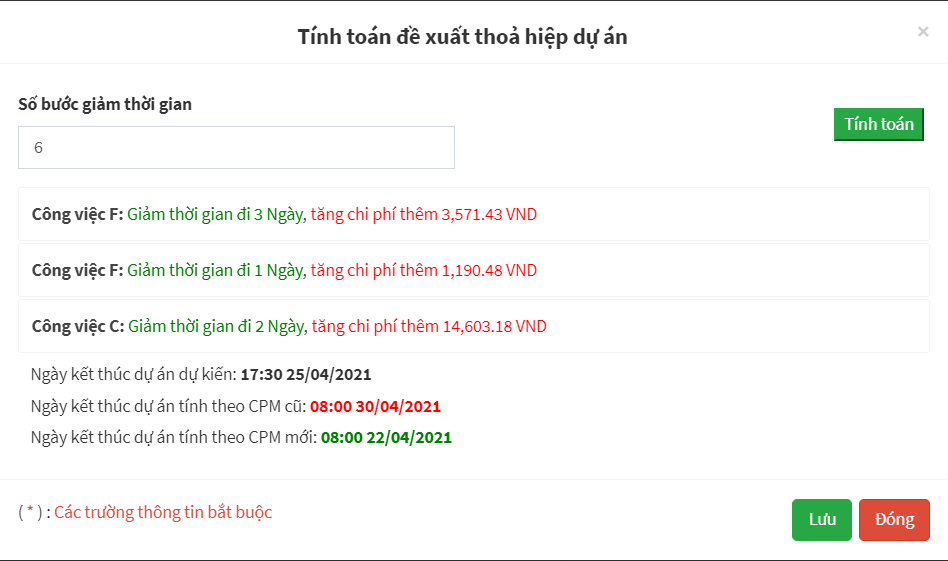
Để tính toán đề xuất thoả hiệp bằng tay, trước tiên ta cần quan tâm đến các hoạt động găng và ưu tiên rút ngắn hoạt động có chi phí thoả hiệp theo đơn vị thời gian thấp nhất. Đối với dự án này thì A, B, E là hoạt động găng. Xét về mặt chi phí Ki thì việc thoả hiệp nên ưu tiên rút ngắn hoạt động A trước, sau đó mới xem xét đến các hoạt động B, E. Lưu ý ở đây ta chỉ được rút ngắn mỗi hoạt động trong mức rút giảm thời gian tối đa (Mi). Do vậy, đề xuất phương án cho dự án này là rút ngắn hoạt động A: 2 ngày.

Tuy nhiên, khi thoả hiệp hoạt động găng hiện tại, các đường đi khác có thể trở thành đường găng. Do đó, cần thiết phải rà soát mạng dự án và tính toán lại đường găng, đồng thời nếu thời gian tính toán vẫn lớn hơn thời gian yêu cầu kết thúc thì ta cần đề xuất bổ sung các hoạt động cần rút ngắn. Trong trường hợp này, sau khi rút ngắn hoạt động A đi 2 ngày, đường găng đã thay đổi thành C-D-E. Và vì thời gian vẫn chưa đáp ứng yêu cầu nên ta cần bổ sung thêm phương án thoả hiệp là rút ngắn hoạt động D: 1 ngày.

Vậy quyết định phương án thoả hiệp là rút ngắn hoạt động A: 2 ngày và hoạt động D: 1 ngày. Chi phí thoả hiệp là 350$.

Từ phương pháp tính tay trên, em đã tổng kết và cho ra được một quy trình, thuật toán để tự động đề xuất thoả hiệp như sau:

* Xác định trước số lần đề xuất tối đa, với mỗi lần đề xuất thực hiện những việc sau.
* Xem xét các hoạt động găng có Ki thấp nhất thì ưu tiên thoả hiệp trước.
* Đề xuất phương án thoả thuận nhằm đáp ứng yêu cầu về thời gian.
* Rà soát mạng dự án được điều chỉnh, nếu thời gian hoàn thành dự án vẫn vượt quá thời gian theo yêu cầu thì tiếp tục đề xuất hoạt động thoả hiệp bổ sung.
* Nếu quá số lần đề xuất tối đa mà thời gian hoàn thành dự án vẫn chưa đạt được yêu cầu thì dừng thuật toán lại và báo lỗi.



**Hình 5.1** Minh hoạ tính toán đề xuất thoả hiệp dự án

## Công thức đánh giá công việc và thành viên trong dự án

### Vấn đề

Đánh giá kết quả dự án là một công việc quan trọng trong giai đoạn thực thi và giám sát tiến độ dự án. Từ số liệu cụ thể, trực quan, mọi thành viên trong dự án sẽ biết được chất lượng công việc mình thực hiện, và đồng thời cũng xem được điểm số của bản thân qua từng công việc trong dự án để làm động lực, kinh nghiệm cho những công việc sau này. Đồng thời, công thức đánh giá công việc sẽ thể hiện điểm số của dự án và khả năng quán trị của người quản lý dự án. Bên cạnh đó, công thức đánh giá thành viên theo công việc trong dự án sẽ làm cơ sở để đánh giá thành viên và trả lương dự án theo tháng.

Các phần mềm QLDA hiện tại trên thị trường đã có những công thức mặc định riêng để đánh giá công việc và đánh giá thành viên. Tuy nhiên, công thức của họ đa phần chỉ chú trọng vào tiến độ, số lượng công việc hoàn thành của thành viên mà chưa đề cập đến yếu tố quan trọng thứ hai trong QLDA – chi phí. Bên cạnh đó, yếu tố chất lượng công việc và yếu tố chuyên cần của thành viên trong công việc cũng là những vấn đề cần quan tâm khi xây dựng công thức. ĐATN của em xin được đề xuất công thức đánh giá mặc định có sự kết hợp của bốn yếu tố là (i) tiến độ công việc, (ii) chất lượng công việc, (iii) chi phí công việc và (iv) chuyên cần công việc.

### Giải pháp

#### Công thức đánh giá công việc

Yếu tố tiến độ công việc được xác định bởi tham số TaskTimePoint. Tuy nhiên, TaskTimePoint sẽ được quy đổi qua tham số SPI là “chỉ số hiệu suất tiến độ công việc” được xác định bởi công thức sau: SPI = [p / (dused / dest)]. Trong đó:

* p – tiến độ công việc hiện tại (đơn vị là %)
* dtused – tổng thời gian sử dụng cho công việc tính từ lúc bắt đầu công việc đến thời điểm hiện tại (đơn vị là milliseconds)
* dtest – tổng thời gian ước lượng cho công việc (đơn vị là milliseconds)

Công thức quy đổi từ SPI sang TaskTimePoint kèm với tham số wtime là trọng số yếu tố thời gian (giá trị mặc định là 0.25):

* SPI < 0.5 → TaskTimePoint = 0
* 0.5 <= SPI < 0.75 → TaskTimePoint = 40 \* wtime
* 0.75 <= SPI < 1 → TaskTimePoint = 60 \* wtime
* 1 <= SPI < 1.25 → TaskTimePoint = 80 \* wtime
* 1.25 <= SPI < 1.5 → TaskTimePoint = 90 \* wtime
* 1.5 <= SPI → TaskTimePoint = 100 \* wtime

Lý do tại sao SPI nằm trong khoảng từ 1 đến 1.25 mới chỉ được 80 điểm, bởi vì một công việc ở mức đó mới chỉ là ở mức hoàn thành đúng chỉ tiêu tiến độ thời gian. Công việc nào sử dụng thời gian thực càng ít hơn thời gian ước lượng bao nhiêu thì điểm sẽ càng tăng thêm trong khoảng bấy nhiêu

Yếu tố chất lượng công việc có liên quan đến hoạt động trong công việc. Một hoạt động trong một công việc sẽ có độ quan trọng riêng và số điểm đánh giá riêng từ người phê duyệt. Chất lương công việc sẽ được xác định bởi công thức sau: TaskQualityPoint = (∑ (ri \* ii) / ∑ ii) \* wquality. Trong đó:

* ri – điểm đánh giá hoạt động i của công việc hiện tại
* ii – độ quan trọng hoạt động i của công việc hiện tại
* wquality – trọng số yếu tố chất lượng (giá trị mặc định là 0.25)

Yếu tố chuyên cần trong công việc được xác định bởi công thức: TaskDilligencePoint = (dtlog / dtest) \* 100 \* wdilligence. Trong đó:

* dtlog – tổng thời gian bấm giờ cho công việc (đơn vị là milliseconds)
* dtest – tổng thời gian ước lượng cho công việc (đơn vị là milliseconds)
* wdilligence – trọng số yếu tố chuyên cần (giá trị mặc định là 0.25)

Lưu ý nếu dtlog / dtest > 1 thì ta sẽ lấy giá trị 1.

Yếu tố chi phí trong công việc được xác định bởi tham số TaskCostPoint. Tuy nhiên, TaskCostPoint sẽ được quy đổi qua tham số CPI là “chỉ số hiệu suất chi phí công việc” được xác định bởi công thức sau: CPI = (p \* ctest) / ctact. Trong đó:

* p – tiến độ công việc hiện tại (đơn vị là %)
* ctest – tổng chi phí ước lượng cho công việc
* ctact – tổng chi phí thực cho công việc (người dùng sẽ nhập vào)

Công thức quy đổi từ CPI sang TaskCostPoint kèm với tham số wcost là trọng số yếu tố chi phí (giá trị mặc định là 0.25):

* CPI < 0.5 → TaskCostPoint = 0
* 0.5 <= CPI < 0.75 → TaskCostPoint = 40 \* wcost
* 0.75 <= CPI < 1 → TaskCostPoint = 60 \* wcost
* 1 <= CPI < 1.25 → TaskCostPoint = 80 \* wcost
* 1.25 <= CPI < 1.5 → TaskCostPoint = 90 \* wcost
* 1.5 <= CPI → TaskCostPoint = 100 \* wcost

Lý do tại sao CPI nằm trong khoảng từ 1 đến 1.25 mới chỉ được 80 điểm, bởi vì một công việc ở mức đó mới chỉ là ở mức hoàn thành đúng chỉ tiêu chi phí. Công việc nào sử dụng chi phí thực càng ít hơn chi phí ước lượng bao nhiêu thì điểm sẽ càng tăng thêm trong khoảng bấy nhiêu.

Vậy ta có công thức đánh giá công việc kết hợp bốn yếu tố là: TaskPoint = TaskTimePoint + TaskQualityPoint + TaskDilligencePoint + TaskCostPoint.

#### Công thức đánh giá thành viên trong công việc

Các công việc trên hệ thống dxclan sử dụng mô hình trách nhiệm RACI. Vậy nên khi đánh giá điểm số thành viên trong công việc, ta sẽ quan tâm đến hai vai trò là R (người thực hiện) và A (người phê duyệt) vì đây là các vai trò chính trong quá trình thực hiện công việc. Đồng thời mỗi vai trò sẽ có trọng số wmember được thêm vào một số công thức. Trọng số wmember được xác định bởi:

* Nếu vai trò là R thì wmember = 0.8 / ∑ (thành viên R của công việc)
* Nếu vai trò là R thì wmember = 0.2 / ∑ (thành viên A của công việc)

Trong đó, 0.8 và 0.2 có thể được thay đổi trong quá trình cấu hình công việc. Hai giá trị trên chỉ là giá trị mặc định.

Yếu tố tiến độ công việc được xác định bởi công thức: MemberTimePoint = TaskTimePoint đã được nêu ở mục 5.2.2.1. Vì tiến độ công việc là tiến độ chung cho các thành viên trong công việc đó.

Yếu tố chất lượng công việc được xác định bởi công thức sau: MemberQualityPoint = MemberQualityPoint đã được nêu ở mục 5.2.2.1. Vì hoạt động trong công việc chỉ có thành viên R được tạo, còn thành viên A chỉ có vai trò đánh giá, nên chất lượng các hoạt động trong công việc sẽ là chất lượng chung của các thành viên trong công việc.

Yếu tố chuyên cần trong công việc được xác định bởi công thức: MemberDilligencePoint = (dtlog / dtest) \* 100 \* wdilligence. Trong đó:

* dmlog – tổng thời gian thành viên hiện tại bấm giờ cho công việc (đơn vị là milliseconds)
* dmest – tổng thời gian ước lượng cho thành viên hiện tại hoàn thành công việc (đơn vị là milliseconds). dmest = dtest \* wmember
* wdilligence – trọng số yếu tố chuyên cần (giá trị mặc định là 0.25)

Lưu ý nếu dtlog / dtest > 1 thì ta sẽ lấy giá trị 1.

Yếu tố chi phí trong công việc được xác định bởi tham số MemberCostPoint. Tuy nhiên, CostPoint sẽ được quy đổi qua tham số CPI là “chỉ số hiệu suất chi phí công việc” được xác định bởi công thức sau: CPI = (p \* cmest) / cmact. Trong đó:

* p – tiến độ công việc hiện tại (đơn vị là %)
* cmest – tổng chi phí ước lượng cho thành viên hoàn thành công việc. cmest = ctest \* wmember
* cmact – tổng chi phí thực cho thành viên hoàn thành công việc (người dùng sẽ nhập vào)

Công thức quy đổi từ CPI sang MemberCostPoint kèm với tham số wcost là trọng số yếu tố chi phí (giá trị mặc định là 0.25) tương tự như công thức quy đổi từ CPI sang TaskCostPoint được nêu ở mục 5.2.2.1.

Vậy ta có công thức đánh giá công việc kết hợp bốn yếu tố là: MemberPoint = MemberTimePoint + MemberQualityPoint + MemberDilligencePoint + MemberCostPoint.

## Báo cáo tự động và thống kê đánh giá

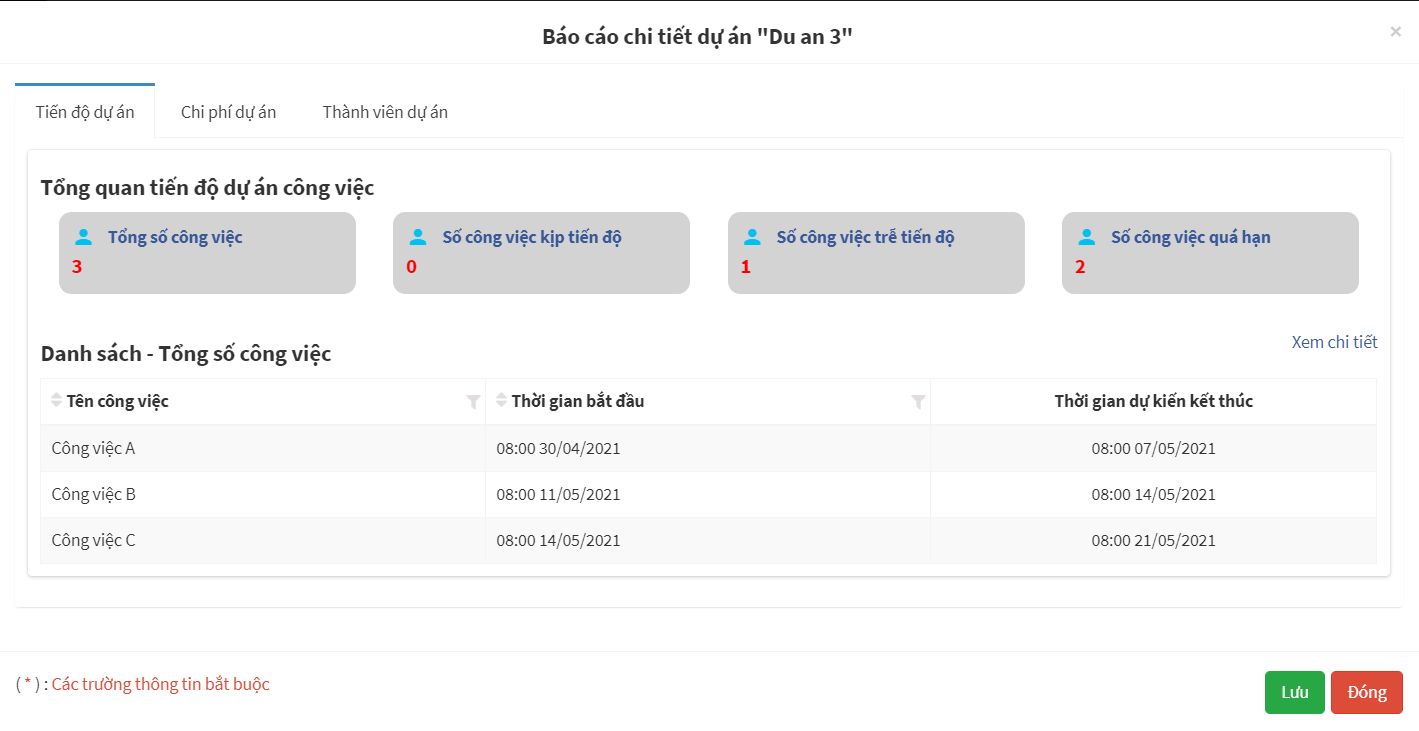
### Vấn đề

Báo cáo tình hình hiện tại của dự án là một tính năng mà bất kỳ phần mềm QLDA nào cũng cần phải thiết kế, tích hợp. Tuy nhiên, một số phần mềm nhỏ chỉ có chức năng tạo báo cáo bằng tay, khiến cho người quản lý mất nhiều thao tác, bên cạnh đó cũng không có yếu tố định lượng nào được xét đến trong báo cáo. Còn lại, với đa phần các công cụ, ứng dụng QLDA hiện nay, báo cáo đã mang tính tự động và có các tham số định lượng. Tuy nhiên, các phần mềm ấy chưa chú trọng các tham số định lượng một cách triệt để các yếu tố cấu thành nên dự án.

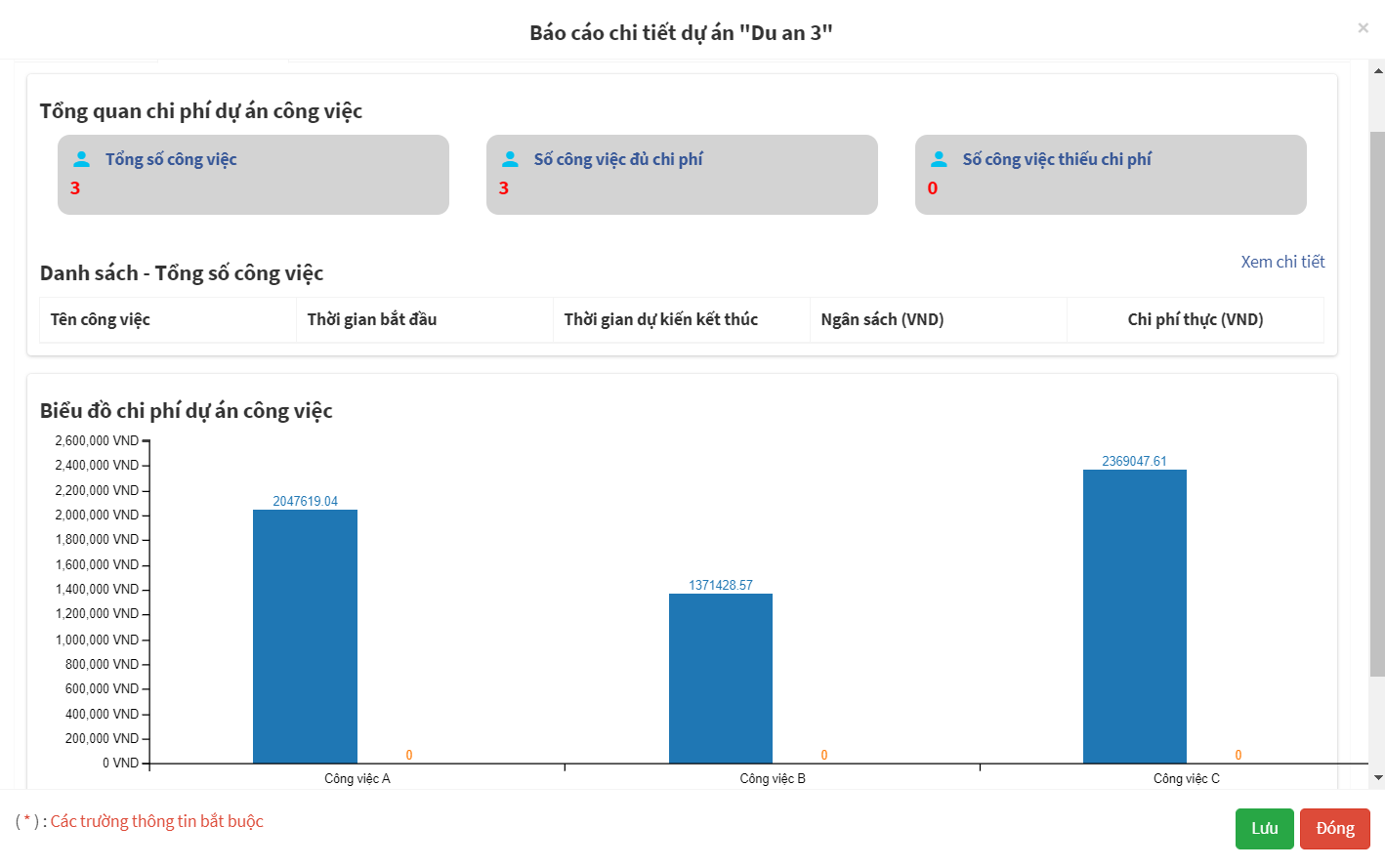
Bên cạnh việc báo cáo tình hình hiện tại, thống kê đánh giá theo tháng cũng là một chức năng quan trọng trong việc giám sát tiến độ dự án. Người quản lý có thể sử dụng tính năng này để so sánh thông số các tháng trong dự án, lấy các số liệu từng tháng làm cơ sở để đánh giá, khen thưởng, trả lương nhân viên theo tháng.

### Giải pháp

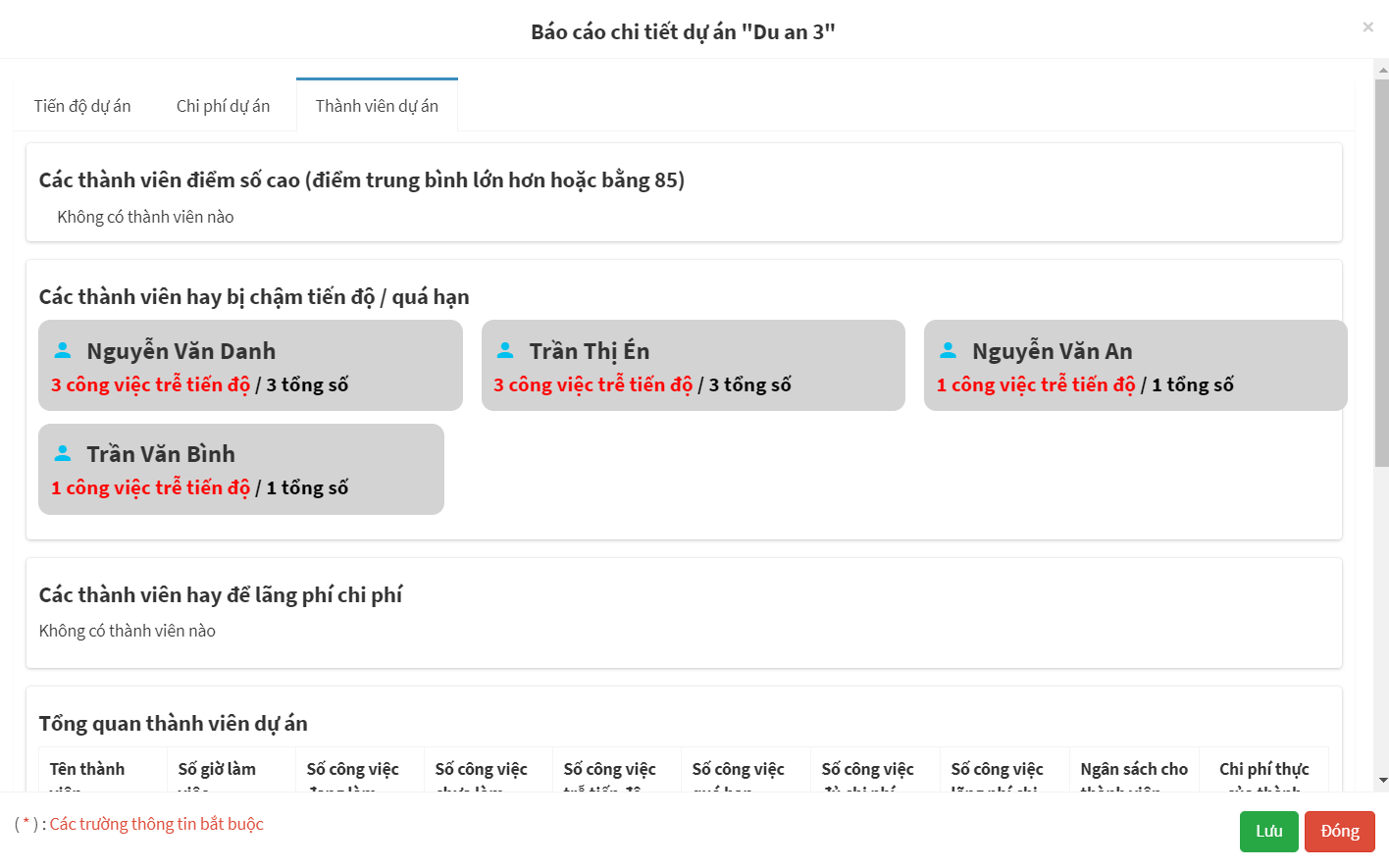
ĐATN của em đã đi theo xu hướng các phần mềm QLDA thông dụng hiện nay, giúp người quản lý có thể xem được báo cáo tự động dựa trên những thông số sẵn có. Báo cáo trong ĐATN của em trình bày được ba yếu tố quan trọng trong dự án là (i) tiến độ dự án, (ii) chi phí dự án và (iii) thành viên dự án.



**Hình 5.2** Báo cáo tiến độ hiện tại của dự án

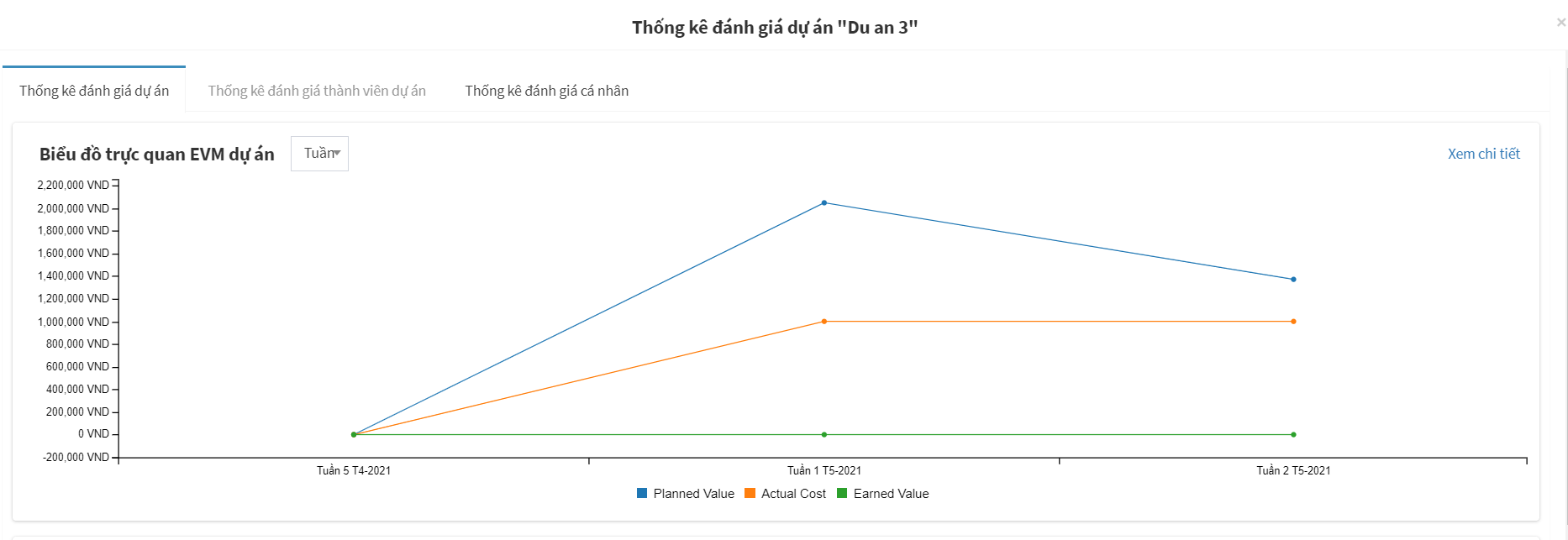


**Hình 5.3** Báo cáo chi phí hiện tại của dự án

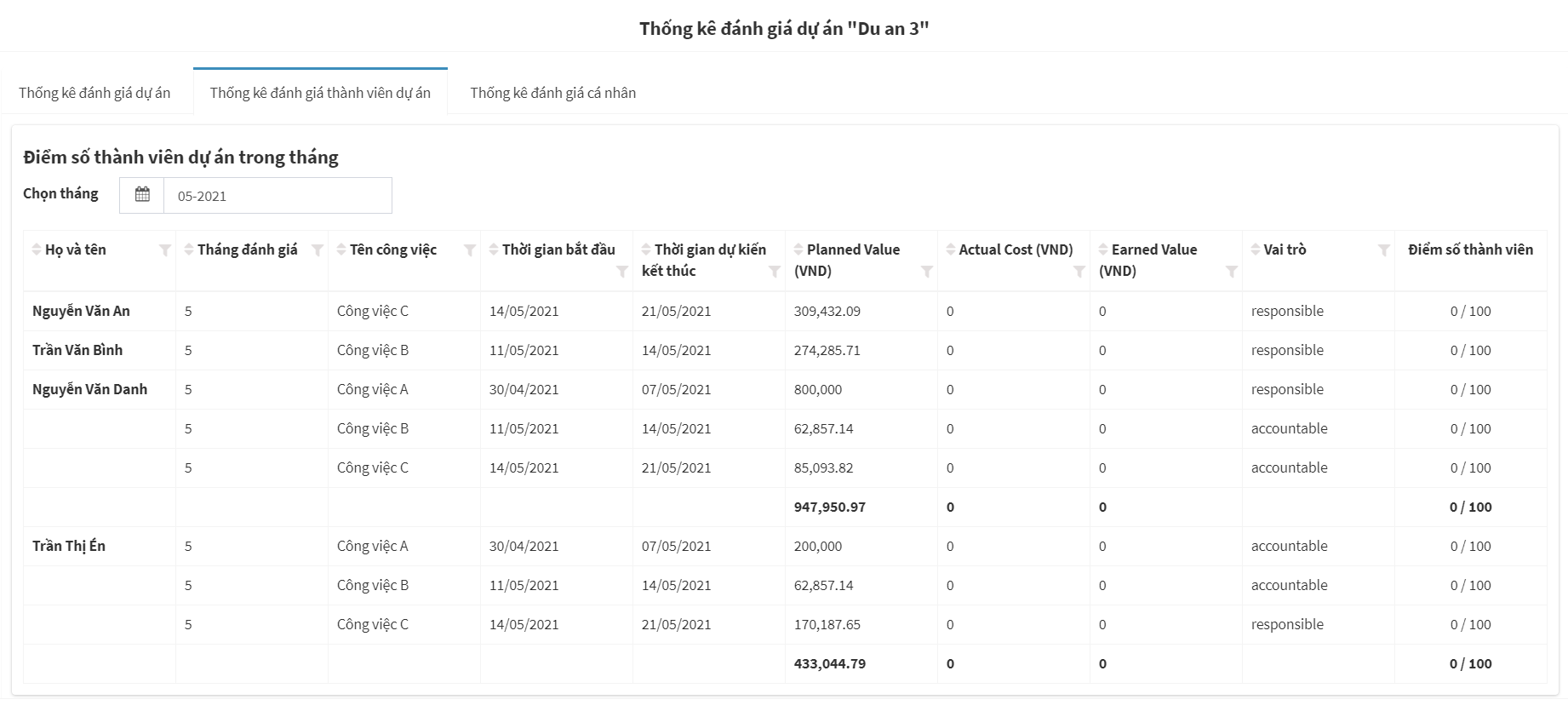


**Hình 5.4** Báo cáo tình hình thành viên hiện tại của dự án

Bên cạnh đó, với chức năng “Thống kê đánh giá”, ĐATN của em tập trung vào ba yếu tố chính là thống kê thông số dự án, thống kê thông số các thành viên của dự án và thống kê thông số cá nhân. Trong đó, chức năng “thông số dự án” có một biểu đồ trực quan EVM để biểu thị các dữ liệu Planned Value, Actual Cost và Earned Value theo từng đoạn thời gian (mặc định là Tuần hoặc Tháng). Chức năng “thành viên dự án” và “cá nhân” sẽ giúp người quản lý và bản thân thành viên dự án kiểm tra được trong tháng này mình được bao nhiêu điểm, số công việc theo tiến độ, số công việc theo chi phí. Những dữ liệu trực quan và mình bạch này sẽ làm cơ sở để khen thưởng và trả lương thành viên trong dự án.



**Hình 5.5** Thống kê thông số dự án với biểu đồ EVM theo tuần



**Hình 5.6** Thống kê thông số các thành viên trong tháng

## Thuật toán đệ quy xác định tầm ảnh hưởng một công việc

### Vấn đề

Sau khi lập kế hoạch dự án, ta có thể thấy trong số danh sách các công việc sẽ có một số chuỗi công việc có sự gắn kết nhất định với nhau. Ta xét một ví dụ gồm bảng danh sách công việc dưới đây:

**Bảng 5.4** Bảng danh sách công việc dùng cho thuật toán đệ quy

|  |  |  |
| --- | --- | --- |
| **Hoạt động** | **Mô tả** | **Hoạt động tiền nhiệm** |
| A | Công việc A | - |
| B | Công việc B | - |
| C | Công việc C | A |
| D | Công việc D | A, B |
| E | Công việc E | B |
| F | Công việc F | C |
| G | Công việc G | C |
| H | Công việc H | E, F |
| I | Công việc I | D |
| J | Công việc J | H, G |

Giả sử đề bài yêu cầu ta tìm những công việc kế nhiệm có liên quan tới công việc C, ta sẽ giải quyết bài toán này như thế nào? Trước hết nhìn bằng mắt thường, ta có thể thấy có 2 công việc F, G có công việc tiền nhiệm là C. Vậy danh sách các công việc bị ảnh hưởng gồm [F, G] Tuy nhiên, công việc H có công việc kế nhiệm là F, và công việc J có công việc kế nhiệm là G. Nên danh sách sẽ được bổ sung thêm thành [F, G, H, J]. Bên cạnh đó, công việc J cũng có công việc kế nhiệm là H. Tuy nhiên J đã ở trong danh sách nên ta bỏ qua và kết thúc công việc tính toán bằng tay.

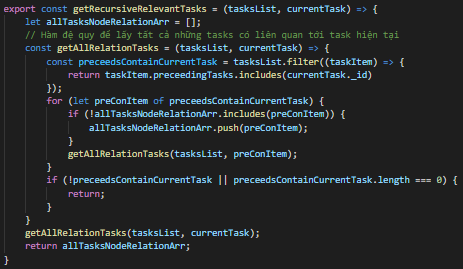
Để giải quyết bài toán này bằng thuật toán, điều đầu tiên ta nghĩ đến là dùng thuật toán vòng lặp. Tuy nhiên, vòng lặp có 2 nhược điểm lớn. (i) Vòng lặp sẽ lặp tất cả các phần tử trong mảng, nên nếu mảng công việc càng nhiều thì hiệu năng càng giảm và thời gian thực hiện thuật toán sẽ tăng lên theo cấp số nhân. (ii) Nếu bài toán chỉ yêu cầu tìm các công việc liên quan “trực tiếp” tới C thì ta có thể dùng vòng lặp là ra được mảng [F, G]. Tuy nhiên ngoài liên quan “trực tiếp” ta còn có sự liên quan “gián tiếp” như các công việc H, J, vậy nên số lần để thực hiện bước lặp là bất khả định. Kết luận lại, thuật toán vòng lặp không thể sử dụng cho bài toán đã nêu trên.

### Giải pháp

ĐATN của em đã sử dụng thuật toán đệ quy để giải quyết bài toán được nêu trên. Các bước trong thuật toán đệ quy bao gồm:

* Tìm kiếm những công việc kế nhiệm liên quan trực tiếp tới công việc i hiện tại.
* Với mỗi công việc trong mảng công việc trên, ta thực hiện lại hàm tìm kiếm những công việc kế nhiệm liên quan trực tiếp tới công việc j hiện tại.
* Nếu mảng công việc của j không có phần tử, quay trở về trạng thái trước đó của đệ quy.

Thuật toán đệ quy đã giúp em xử lý triệt để vấn đề tìm kiếm danh sách các công việc liên quan. Đồng thời thuật toán này đảm bảo hiệu năng và thời gian thực thi để có thể thực hiện được các tính năng khác trong ĐATN.



**Hình 5.7** Hình ảnh minh hoạ thuật toán bằng code

## Yêu cầu thay đổi có yếu tố định lượng về chi phí – thời gian

### Vấn đề

Các phần mềm QLDA hiện nay như Jira, Zoho có chức năng tạo yêu cầu thay đổi trong dự án. Tuy nhiên, những yêu cầu thay đổi này mới chỉ dừng lại ở mức độ báo cáo tạo bằng tay, người dùng phải tự liệt kê ra danh sách những yếu tố bị ảnh hưởng trong yêu cầu thay đổi này mà không có một sự trợ giúp mang tính định lượng nào. Ví dụ khi thay đổi thông số một công việc, những công việc phía sau công việc đó sẽ bị ảnh hưởng như thế nào? Tiến độ và chi phí của các công việc ấy sẽ tăng giảm như thế nào? Các bên liên quan sẽ bị tác động nhiều hay ít? Điều đó ảnh hưởng tới toàn bộ dự án ra sao? v.v.

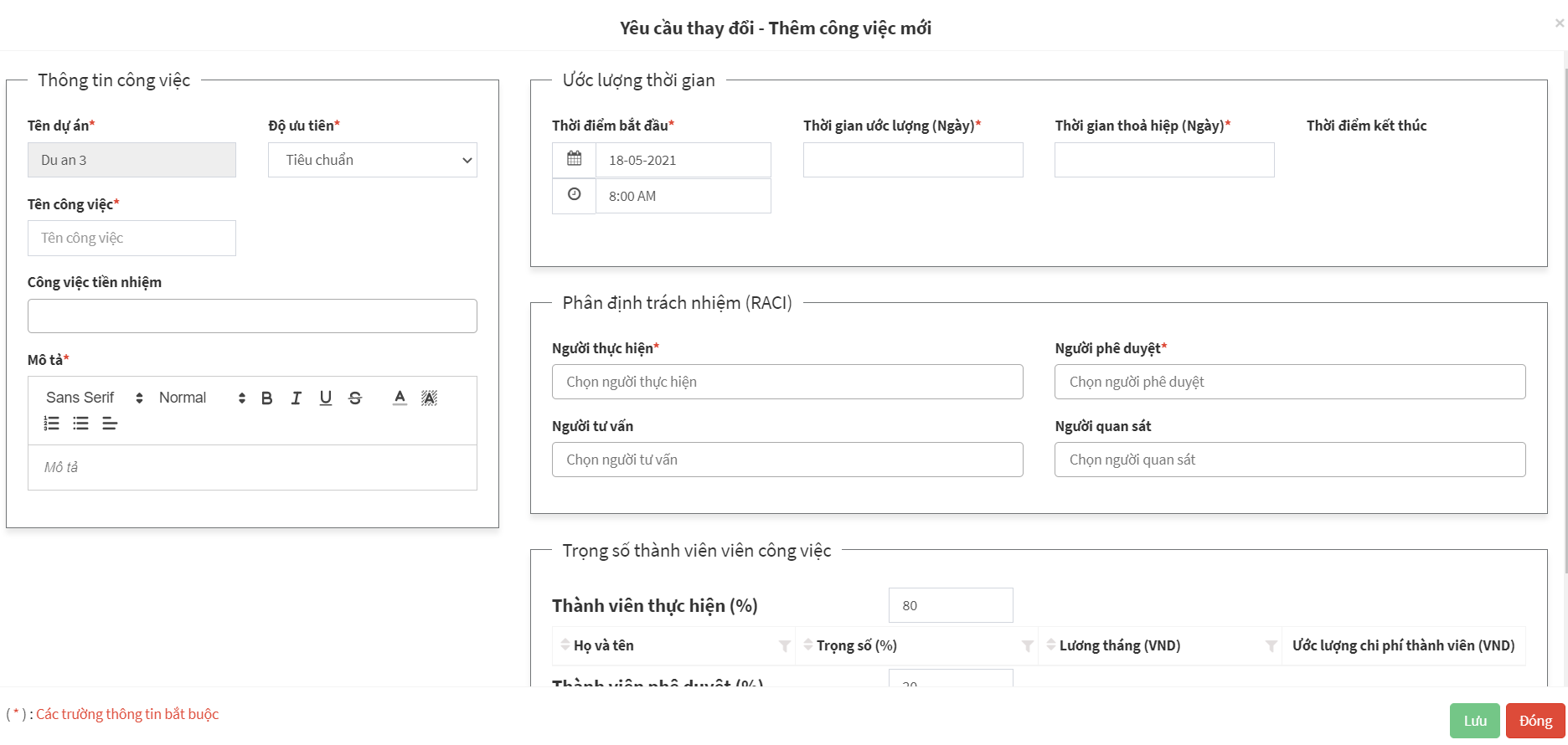
Vấn đề phát sinh trong dự án của doanh nghiệp có tất cả năm loại chính. Thứ nhất là loại kỹ thuật bao gồm những vấn đề phát sinh liên quan đến kỹ thuật, máy móc. Thứ hai là loại quy trình doanh nghiệp, liên quan đến việc thiết kế phạm vi, mục tiêu và quy trình thực hiện công việc, các vấn đề về giao tiếp và hỗ trợ trong công việc. Thứ ba là loại thay đổi quan lý, liên quan đến vấn đề quản lý công việc và khách hàng. Thứ tư là loại bên thứ ba, là các rủi ro và vấn đề phát sinh từ nhà cung cấp, các yếu tố bên ngoài như pháp luật và nhóm người bên ngoài doanh nghiệp. Cuối cùng là loại nguồn lực, liên quan đến nhân lực, chi phí và thời gian trong công việc. Mỗi loại sẽ đòi hỏi quy trình giải quyết khác nhau và loại yêu cầu thay đổi tương ứng. Trong phạm vi đồ án này, em xin phép được xét đến yêu cầu thay đổi liên quan đến loại nguồn lực.

Vậy bài toán đặt ra ở đây là ĐATN của em cần giải quyết được vấn đề tạo yêu cầu thay đổi liên quan đến nguồn lực và có ảnh hưởng định lượng như thế nào tới các công việc khác trong dự án.

### Giải pháp

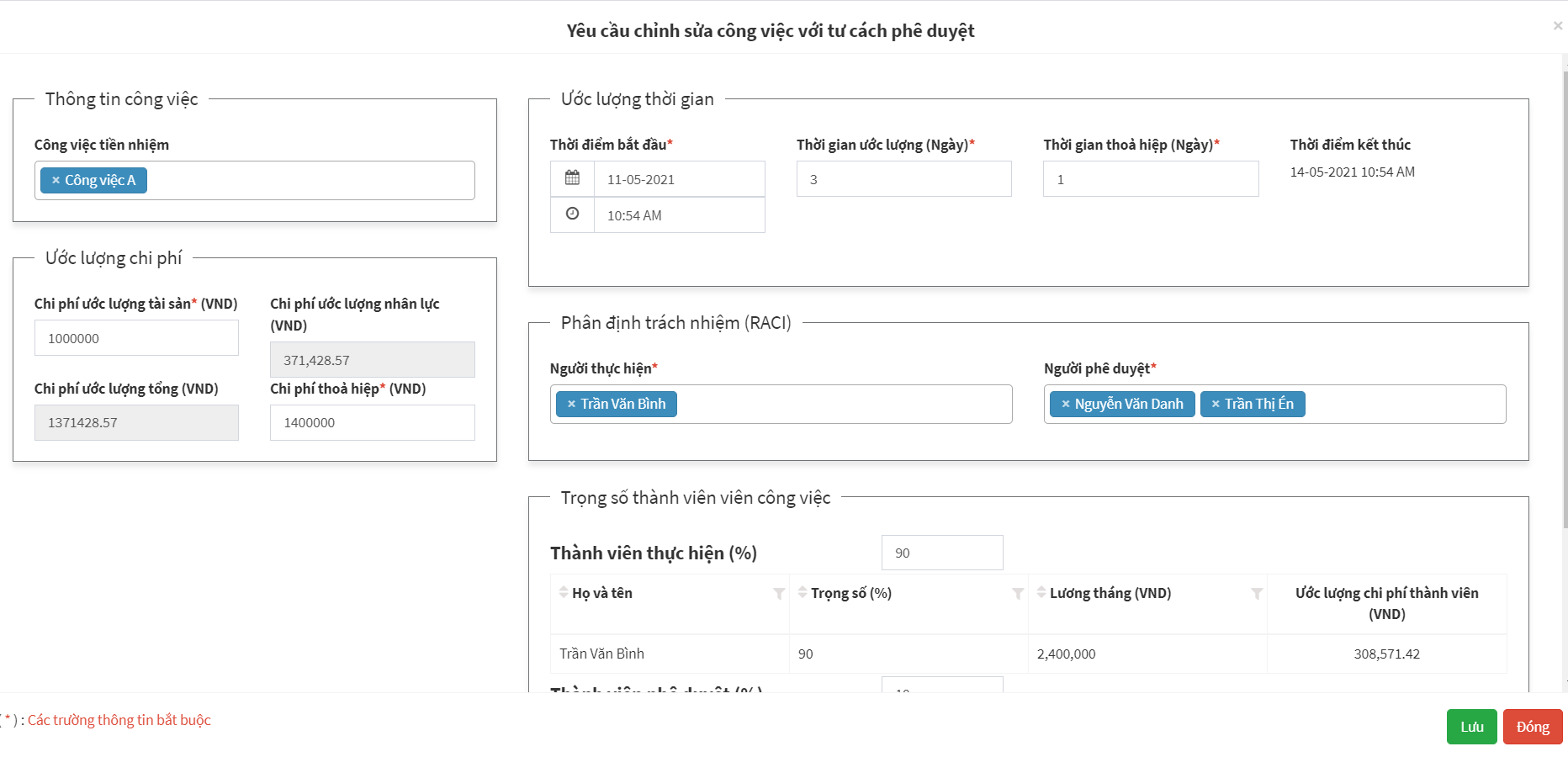
Từ lục lập kế hoạch dự án, danh sách các công việc đã được kết nối với nhau theo một trình tự và kết hợp các tham số nhất định gồm thời gian, chi phí và nhân lực. Vì thế, bất kỳ hành động gì tác động đến ba yếu tố trên trong dự án phải được biểu diễn dưới dạng một yêu cầu thay đổi. Trong ĐATN của mình, em xin được phép xét đến 2 trường hợp cụ thể là (i) tạo mới một công việc không theo lập kế hoạch và (ii) chỉnh sửa công việc sau khi đã lập kế hoạch.

Với chức năng “tạo mới một công việc không theo lập kế hoạch”, khi một công việc mới tạo ra, sẽ có sự thay đổi về mặt tổng thể chi phí ước lượng và tổng thể thời gian dự kiến kết thúc của dự án. Đa phần những công việc khác sẽ không ảnh hưởng gì nhiều do công việc tạo mới này không cần định nghĩa các công việc kế nhiệm.



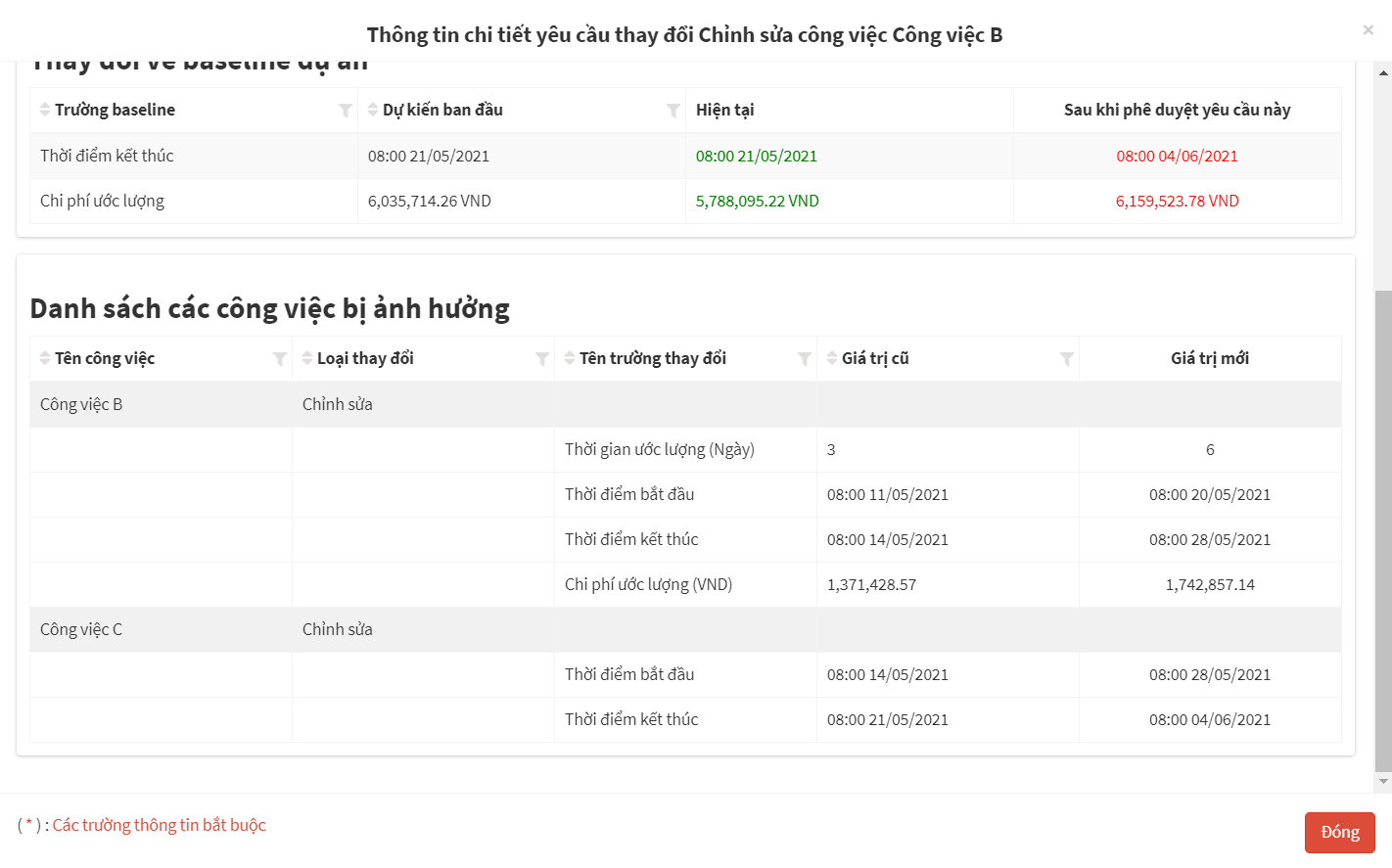
**Hình 5.8** Biểu mẫu yêu cầu tạo công việc mới không theo lập kế hoạch

Với chức năng “chỉnh sửa một công việc sau khi lập kế hoạch”, khi một công việc được chỉnh sửa, sẽ có sự thay đổi về mặt tổng thể chi phí ước lượng và tổng thể thời gian dự kiến kết thúc. Bên cạnh đó, nếu công việc hiện tại có những công việc liên quan kế nhiệm, ta cần phải xét xem yêu cầu thay đổi này sẽ tác động như thế nào tới thời gian, chi phí và nhân lực của các công việc đó. Thuật toán tìm kiếm danh sách công việc có liên quan tới công việc hiện tại được đề cập ở mục 5.4 là yếu tố quyết định để thực hiện được chức năng này.



**Hình 5.9** Biểu mẫu yêu cầu chỉnh sửa công việc sau khi đã lập kế hoạch

Sau khi đã tạo yêu cầu, ta có thể xem được chi tiết những yếu tố bị ảnh hưởng của từng yêu cầu thay đổi. Đó là những dữ kiện về thông số gốc ban đầu, thông số hiện tại và thông số sau khi yêu cầu được phê duyệt. Tất cả số liệu đều mang tính định lượng, minh bạch, trực quan, giúp cho người dùng có được cái nhìn khách quan hơn và giúp người quản lý có thể ra quyết định sáng suốt hơn khi phê duyệt hoặc từ chối yêu cầu thay đổi này.



**Hình 5.10** Chi tiết yêu cầu thay đổi công việc

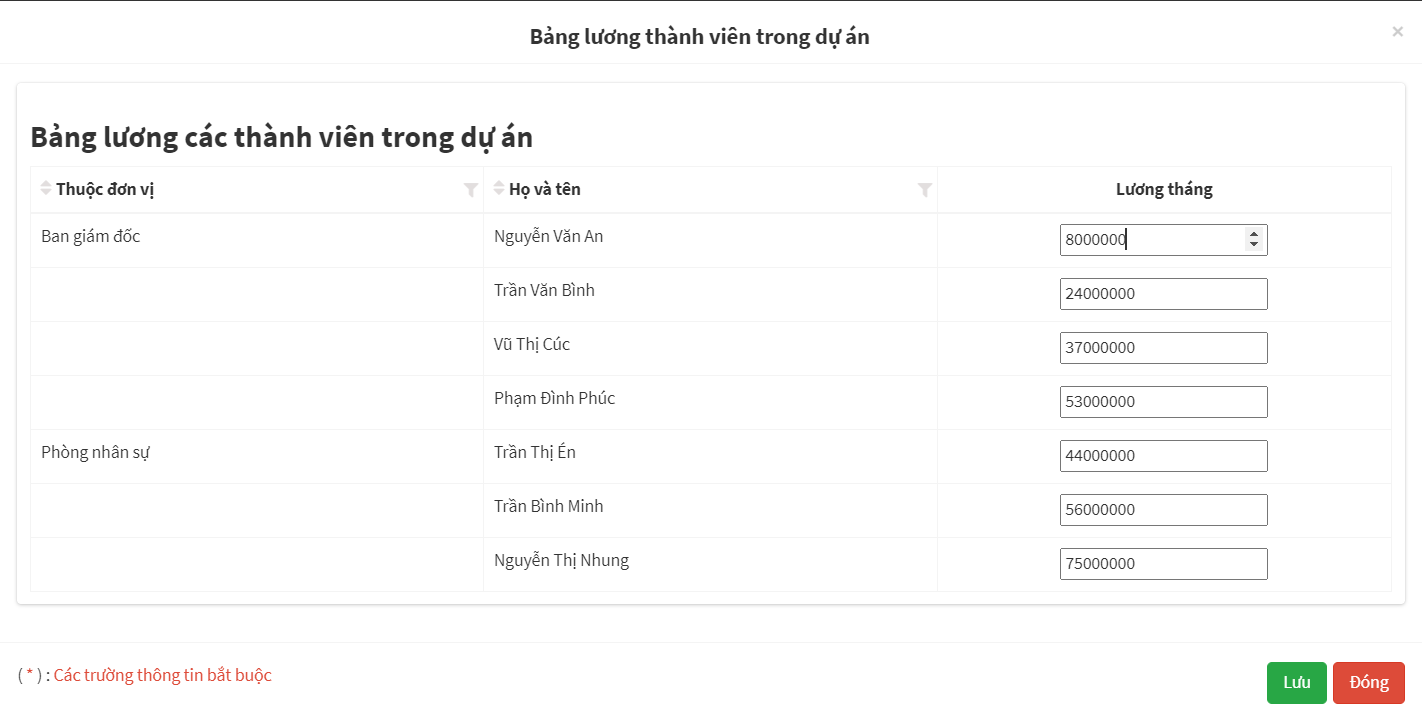
## Tích hợp với module lương nhân viên

### Vấn đề

Khi tạo một dự án mới, các thành viên sẽ có phần lương cứng mỗi tháng của bản thân. Sau mỗi tháng đánh giá, thành viên dựa vào kết quả công việc và điểm số bản thân mà được trả thêm hoặc bị trừ đi số tiến vào lương đó. Trong quá trình tạo dự án mới, việc nhập từng số lương cho mỗi thành viên sẽ gây khó khan và mất thời gian cho người quản trị dự án. Vậy nên bài toán đặt ra ở đây là cần làm giảm thao tác nhập lương của từng thành viên cho người quản lý.

### Giải pháp

Hệ thống dxclan đã có module lương của nhân viên theo từng tháng. ĐATN của em sẽ lấy lương của từng nhân viên theo tháng hiện tại đang tạo dự án. Nếu thành viên nào không có lương thì giá trị mặc định sẽ bằng 0. Qua đó khi tạo bảng lương thành viên dự án, nhà quản lý chỉ cần chỉnh sửa lương với những người cần thiết phải thay đổi đề phù hợp với dự án.



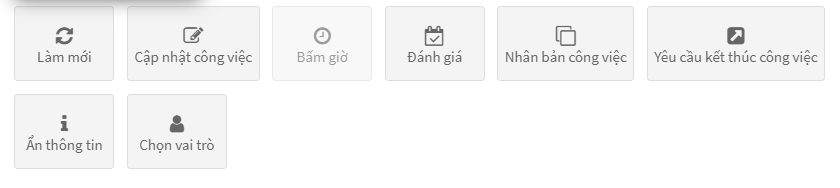
**Hình 5.11** Bảng lương gợi ý cho người quản lý khi tạo mới dự án

## Công việc bình thường và công việc theo dự án

### Vấn đề

Hiện tại, trên hệ thống không gian làm việc số dxclan.com[[9]](#footnote-9), module quản lý công việc đã được thiết lập và triển khai sử dụng. Tuy nhiên, những công việc được quản lý hiện tại đều ở dạng công việc chung chung, bình thường và không có ràng buộc chặt chẽ. Loại công việc này được thiết kế để có thể sử dụng chung trong các module khác như quản lý mẫu công việc, quản lý công việc theo đơn vị, quản lý công việc theo quy trình. Đặc điểm chung của chung của chúng là (i) có thể chỉnh sửa thông tin dựa theo vai trò RACI (ii) có thể tạo các hoạt động để thành viên công việc dễ dàng nắm bắt tiến độ công việc và đánh giá (iii) có thể bấm giờ (iv) có thể được nhân bản (v) có thể được yêu cầu kết thúc công việc và (vi) có thể được đánh giá.

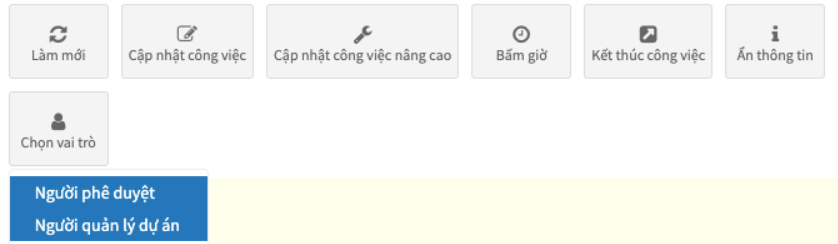
Tuy nhiên, trong dự án, một công việc sẽ có những tham số, yếu tố ràng buộc nhất định nên việc chỉnh sửa thông tin công việc có thể tạo ra những yêu cầu thay đổi như đã nêu ở mục 5.5. Bên cạnh đó, công việc bình thường có thể kéo dài trong nhiều tháng, vậy nên việc đánh giá sẽ được tính điểm theo từng tháng một. Nhưng công việc trong dự án được trả lương theo phương thức “khoán”, nghĩa là công việc cần được thực hiện xong mới được đánh giá. Cuối cùng, công việc bình thường đánh giá theo tháng nên có thể lúc đánh giá thì công việc chưa kết thúc, còn công việc với dự án thì chỉ đánh giá khi công việc có khả năng kết thúc được. Vì thế việc tách hai nút “Đánh giá” và “Yêu cầu kết thúc công việc” của công việc dự án là không cần thiết.



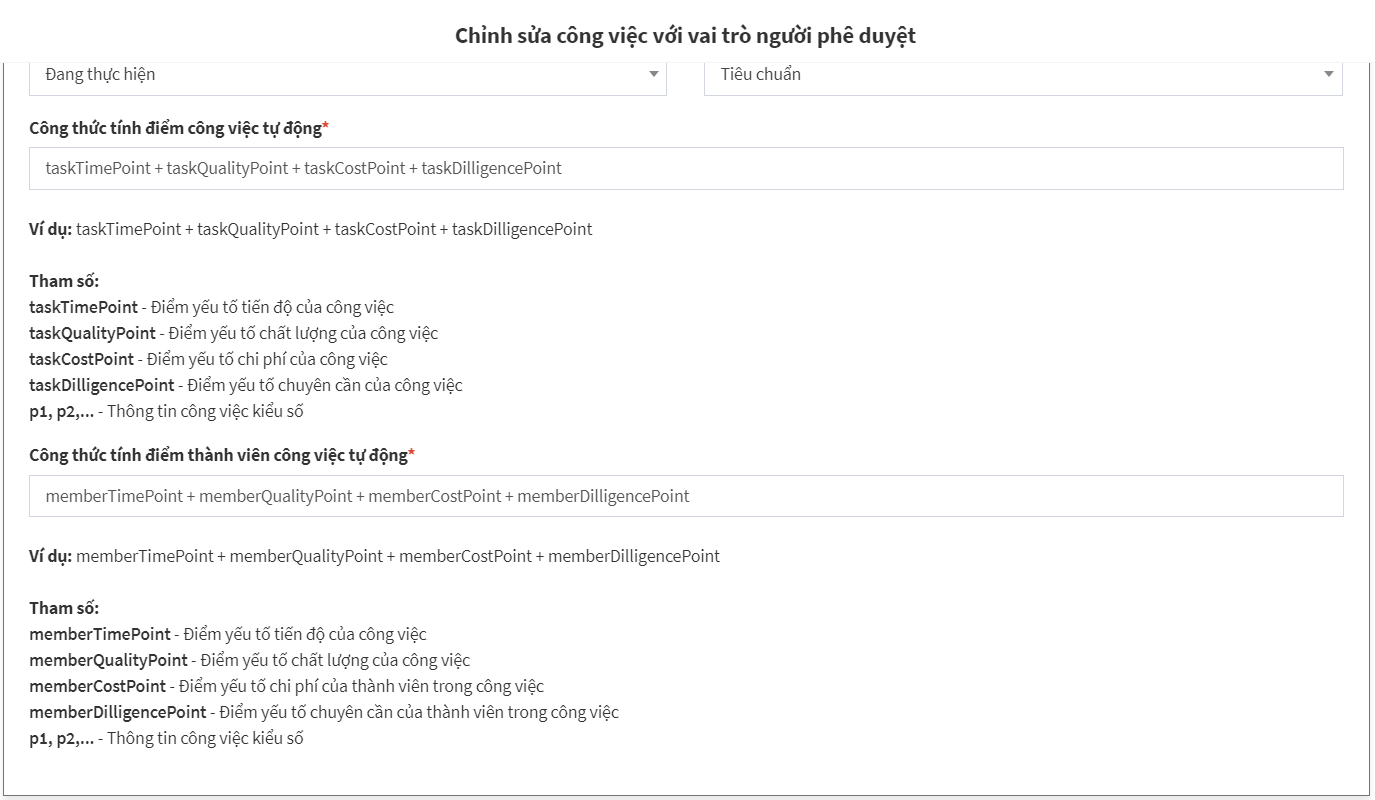
**Hình 5.12** Giao diện các nút trong công việc bình thường

### Giải pháp

Sau khi nghiên cứu và đưa ra những hạn chế trong giao diện của công việc bình thường, module QLDA của em đã cải tiến giao diện của công việc trong dự án như sau (i) thêm một nút “Cập nhật công việc nâng cao” để phục vụ cho việc thay đổi chỉnh sửa thông tin có ràng buộc tới tham số dự án (ii) gộp hai nút “Đánh giá” và “Yêu cầu kết thúc công việc” vào thành một nút “Yêu cầu kết thúc công việc” để người dùng không cần thao tác nhiều và (iii) công thức đánh giá mặc định của công việc bình thường được thay đổi thành công thức đánh giá mặc định của công việc dự án.



**Hình 5.13** Giao diện các nút trong công việc dự án



**Hình 5.14** Giao diện công thức mặc định công việc dự án

# Kết luận và hướng phát triển

## Kết luận

Sau quá trình tìm hiểu, phân tích và xây dựng cùng với sự hướng dẫn tận tình của TS. Trịnh Tuấn Đạt, em đã xây dựng được cho ĐATN của bản thân một ứng dụng QLDA cho doanh nghiệp. Ứng dụng đã giải quyết được một số hạn chế của các phần mềm hiện nay như (i) cho phép tự động đề xuất phương án thoả hiệp khi lập kế hoạch dự án, (ii) thiết kế công thức đánh giá công việc và thành viên dự án dựa trên bốn yếu tố thời gian, chất lượng, chi phí và chuyên cần và (iii) quản lý yêu cầu thay đổi mang tính chất định lượng ở một số lĩnh vực.

Ứng dụng đã được xây dựng trên tinh thần hoàn thiện sản phẩm, tuy nhiên do thời gian ngắn hạn và kinh nghiệm lập trình web chưa có nhiều, nên sản phẩm sẽ còn nhiều hạn chế và thiếu sót. Tuy nhiên, ứng dụng vẫn tiếp tục lấy đánh giá và phản hồi từ phía người dùng để có thể xây dựng và phát triển thành phiên bản hoàn chỉnh hơn.

Trong quá trình làm ĐATN, em đã học hỏi và áp dụng những công nghệ mới mà trước đó em chưa từng sử dụng để xây dựng được một ứng dụng mang tính thực tế cao. Bên cạnh đó, việc đóng góp cho một hệ thống hoàn chỉnh và có thật trong thực tế khiến em cảm thấy bản thân cần phải trách nhiệm hơn trong từng khâu xử lý nghiệp vụ, từng dòng code và từng bản báo cáo mình viết ra. Trên tất cả, việc được vào vai từ đầu đến cuối của ĐATN từ khâu phân tích nghiệp vụ, đến khâu lập kế hoạch cho đồ án, đến khâu triển khai lập trình, rồi đến khâu thu hoạch và viết báo cáo, làm cho bản thân em học hỏi được nhiều thứ hơn không chỉ về kiến thức chuyên ngành, về công nghệ mà còn là phương pháp tư duy phản biện, suy nghĩ logic và khả năng nhìn vấn đề trên nhiều góc độ khác nhau. Đây sẽ là những kinh nghiệm quý báu, giúp ích cho sự phát triển của em trong những giai đoạn sau này.

## Hướng phát triển

Ứng dụng QLDA cho doanh nghiệp sẽ tiếp tục xây dựng để hoàn thiện hơn bao gồm các nội dụng (i) tiếp tục lấy đánh giá phản hồi từ khách hàng để cải thiện các chức năng và giao diện cho phù hợp, (ii) quản lý công việc trong dự án theo các cột mốc và có công thức đánh giá riêng cho từng cột mốc đó và (iii) quản lý vấn đề phát sinh và quản lý yêu cầu thay đổi cần được phát triển để mở rộng và bao quát cho nhiều lĩnh vực hơn nữa.

Hiện tại, hệ thống dxclan đã có ứng dụng di động trên Android. Tuy nhiên, các chức năng vẫn dừng lại ở mức cơ bản và đơn giản nhất. Trong thời gian sắp tới, hệ thống có thể chuyển hoá các chức năng của các module trên website thành các chức năng của ứng dụng di động để người sử dụng nói chung và các doanh nghiệp nói riêng có thể dễ dàng trải nghiệm và kiểm thử các chức năng ấy một cách nhanh chóng. Bên cạnh đó, việc triển khai ứng dụng di động trên phiên bản iOS cũng là một thách thức và cũng là sự mong chờ lớn tới sự phát triển của hệ thống dxclan này.

# Tài liệu tham khảo

1. Bodunwa, Oluwatoyin Kikelomo and Makinde, Jamiu Olalekan, Application of Critical Path Method (CPM) and Project Evaluation Review Techniques (PERT) in Project Planning and Scheduling, Journal of Mathematics and Statistical Science, 6, 1-8, 2020.
2. Chương 4 Điều hành dự án bằng PERT/CPM, <http://nguyenvanvuantvu.yolasite.com/resources/PP_%C4%91%E1%BB%8Bnh_l%C6%B0%E1%BB%A3ng_trong_KT/chuong4.PDF>, truy cập lần cuối 2021.
3. MongoDB, MongoDB Documentation, <https://docs.mongodb.com/>, truy cập lần cuối 2021.
4. Mongoose, Mongoose Documentation, <https://mongoosejs.com/>, truy cập lần cuối 2021.
5. Tutorialspoint, Node.js Tutorial, <https://www.tutorialspoint.com/nodejs/index.htm>, truy cập lần cuối 2021.
6. Mozilla, <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs>, truy cập lần cuối 2021.
7. ReactJS, React Documentation, <https://reactjs.org/>, truy cập lần cuối 2021.
8. ReduxJS, Redux Documentaion, <https://redux.js.org/>, truy cập lần cuối 2021.
9. Client-Server Model, <https://www.geeksforgeeks.org/client-server-model/>, truy cập lần cuối 2021.
10. Tutorialspoint, MVC framework, <https://www.tutorialspoint.com/mvc_framework/index.htm>, truy cập lần cuối 2021

1. <https://www.apm.org.uk/resources/what-is-project-management/>, truy cập lần cuối 2021 [↑](#footnote-ref-1)
2. <https://www.opendoorerp.com/the-standish-group-report-83-9-of-it-projects-partially-or-completely-fail/>, truy cập lần cuối 2021 [↑](#footnote-ref-2)
3. <https://hennyportman.wordpress.com/2021/01/06/review-standish-group-chaos-2020-beyond-infinity/> truy cập lần cuối 2021 [↑](#footnote-ref-3)
4. <https://dxclan.com/> Cung cấp các giải pháp quản lý doanh nghiệp như: Quản lý nhân sự, quản lý KPI, quản lý tài sản, quản lý sản xuất v.v. Truy cập lần cuối ngày 2021 [↑](#footnote-ref-4)
5. <https://expressmagazine.net/development/2324/so-sanh-hieu-suat-hoat-dong-cua-mongodb-va-sql-server-2008>, truy cập lần cuối 2021 [↑](#footnote-ref-5)
6. <http://dxclan.com/>, truy cập lần cuối 2021 [↑](#footnote-ref-6)
7. <http://dxclan.com/>, truy cập lần cuối 2021 [↑](#footnote-ref-7)
8. <https://www.damtaicap.net/2018/08/nhung-dieu-can-biet-ve-duong-gang-cua-du-an.html>, truy cập lần cuối ngày 17/05/2021 [↑](#footnote-ref-8)
9. <http://dxclan.com/>, truy cập lần cuối 2021 [↑](#footnote-ref-9)