

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN

CÔNG NGHỆ PHẦN MỀM (CO3001)

UWC 2.0

Giảng viên hướng dẫn: Bùi Công Tuấn
Mai Đức Trung
Lê Đình Thuận

Sinh viên thực hiện: Nguyễn Đăng Anh Khoa – 2010339
Nguyễn Đức An – 2010102
Trần Phúc Anh – 2010133
Trần Chí Công – 2010170
Cao Trần Anh Khoa – 2010335
Nguyễn Quang Huy – 1916081
Võ Thái Toàn – 2010709



Mục lục

I REQUIREMENT ELICITATION	9
1 Mô tả dự án	9
1.1 Ngữ cảnh	9
1.2 Quy trình nghiệp vụ	9
1.3 Giả định dự án	10
1.4 Stakeholders	11
1.5 Tầm vực dự án	11
1.6 User Story	12
1.7 Yêu cầu người dùng và Yêu cầu hệ thống	13
2 Xác định yêu cầu	19
2.1 Yêu cầu chức năng	19
2.2 Non-functional Requirements	21
3 Use-case diagram	24
3.1 Use-case diagram of whole system	24
3.2 Use-case và đặc tả cho Task Assignment (Ca - nhiệm vụ)	25
3.3 Use-case chi tiết cho một số module khác trong hệ thống	49
II SYSTEM MODELING	54
1 Activity Diagram	54
1.1 Task Assignment Module	54
1.2 Tạo task	57
2 Sequence Diagram	59
2.1 Conceptual Solution for the route planning task	59
2.2 Sequence diagram	61
2.3 Solution evaluation	63
3 Class Diagram	65
III ARCHITECTURE DESIGN	84
1 Architectural approach and Modules for the whole WMC 2.0 system	84
1.1 Architectural approach	84
1.2 List of Modules	86
2 Component Diagram - Task Assignment	88
2.1 Bản vẽ Component Diagram	88
2.2 Mô tả Component Diagram	88



3 Deployment diagram	94
3.1 Bản vẽ Deployment Diagram	94
3.2 Mô tả Deployment Diagram	94
IV IMPLEMENTATION - SPRINT 1	96
1 Github for Version Control	96
2 Revisions or Changelogs	96
2.1 Hình ảnh	96
2.2 Mô tả	98
3 Implement MVP1 - Figma & Mockup	101
3.1 Link Figma & Mockup	101
3.2 Mô tả	101
V IMPLEMENTATION - SPRINT 2	120
1 Implement MVP2 - Hiện thực thiết kế giao diện	120
2 Thuyết trình dự án	126
VI TỔNG KẾT	128
VII TÀI LIỆU THAM KHẢO	130



Danh mục hình ảnh

1	Use case Hệ thống	24
2	Use case Task assignment	25
3	Use case Quản lý nhân sự	50
4	Use case Đăng nhập - Đăng ký	51
5	Use case Quản lý MCP	51
6	Use case Quản lý phương tiện	52
7	Task Assignment Activity Diagram	54
8	Phân Task Activity Diagram	57
9	Sequence Diagram	61
10	Task Assignment Class Diagram	65
11	Mô hình kiến trúc MVC	85
12	Component Diagram	88
13	Deployment Diagram	94
14	Giao diện đăng nhập	101
15	Giao diện Quản lý ca làm việc	102
16	Giao diện Quản lý nhiệm vụ	109
17	Giao diện xem nhiệm vụ và tạo route tối ưu	117
18	Hiện thực giao diện đăng nhập	120
19	Hiện thực giao diện quản lý danh sách ca làm việc	121
20	Dialog chứa biểu mẫu tạo ca mới	121
21	Hệ thống thông báo khi tạo ca thành công	122
22	Dialog chứa biểu mẫu chỉnh sửa ca	122
23	Hiện thực giao diện quản lý danh sách nhiệm vụ và quản lý tiến độ nhiệm vụ trong ca - 1	123
24	Hiện thực giao diện quản lý danh sách nhiệm vụ và quản lý tiến độ nhiệm vụ trong ca - 2	123
25	Hiện thực giao diện phân nhiệm vụ mới - 1	124
26	Hiện thực giao diện phân nhiệm vụ mới - 2	124
27	Hệ thống thông báo khi phân nhiệm vụ mới trùng nhân sự hoặc phương tiện với nhiệm vụ đã tạo cùng ca	125
28	Hiện thực giao diện xem chi tiết nhiệm vụ	125
29	Hiện thực giao diện hiển thị lộ trình tối ưu	126
30	Hệ thống hiển thị đề xuất lộ trình dạng mô tả	126



History Log

Task I: REQUIREMENT ELICITATION

STT	Thời gian	Công việc	Thành viên
1	5/9/2022	Meet: thảo luận đề tài	
2	7/9/2022	Meet: phần 1 Project Description	
		Phân công: Context, Business Process, Project Assumptions, Stakeholders	Cao Trần Anh Khoa
		Phân công: User Story, Project Scope, User and system requirements	Trần Chí Công
3	9/9/2022	Meet: phần 2 Requirements	
		Bổ sung: Phần 1 User story, user and system requirements	Nguyễn Đăng Anh Khoa
		Phân công: 2.1 Functional Requirements	Nguyễn Đức An
		Phân công: 2.2 Non - functional requirements	Võ Thái Toàn
4	13/9/2022	Meet: phần 3 Use-case diagram and describe	
		Phân công: Use case diagram of whole system, Task Assignment	Nguyễn Đăng Anh Khoa
		Phân công: Manage Employee, Login	Trần Phúc Anh
		Phân công: Manage MCP, Manage Vehicle	Nguyễn Quang Huy
5	16/9/2022	Meet: xem lại báo cáo	
		Bổ sung: 2.1 Functional requirements	Cao Trần Anh Khoa
		Chỉnh sửa: Use case Manage MCP	Trần Phúc Anh
6	18/9/2022	Meet: xem lại báo cáo	
		Chỉnh sửa: 3.3 Describe use case Manage Employee	Trần Chí Công
		Chỉnh sửa: 2.2 Non functional requirements	Nguyễn Quang Huy
		Bổ sung: 3.2 Task Assignment	Nguyễn Đức An
7	20/9/2022	Meet: xem lại báo cáo	
		Bổ sung: Describe use case Manage Vehicle	Trần Chí Công
		Chỉnh sửa: Describe use case Manage MCP	Nguyễn Đăng Anh Khoa
		Bổ sung: Use case whole system	Võ Thái Toàn
8	21/9/2022	Chỉnh sửa: Báo cáo	Cao Trần Anh Khoa
		Nộp bài	Nguyễn Đăng Anh Khoa

Task II: SYSTEM MODELING



STT	Thời gian	Công việc	Thành viên
1	24/9/2022	Meet: thảo luận phần 1 Activity Diagram	
		Phân công: 1.1 Task Assignment Module	Nguyễn Đức An, Nguyễn Quang Huy
		Phân công: 1.2 Tạo task	Cao Trần Anh Khoa, Võ Thái Toàn
2	28/9/2022	Meet: thảo luận phần 2 Sequence Diagram	
		Phân công: 2.1 Conceptual Solution	Cao Trần Anh Khoa, Trần Chí Công
		Phân công: 2.2 Sequence Diagram	Trần Phúc Anh, Trần Chí Công
		Phân công: 2.3 Solution evaluation	Nguyễn Quang Huy
		Chỉnh sửa: Task Assignment	Trần Phúc Anh
		Bổ sung: Tạo task	Nguyễn Quang Huy, Trần Chí Công
3	5/10/2022	Meet: thảo luận phần 3 Class diagram	
		Phân công: Class Diagram	Nguyễn Đức An, Nguyễn Đặng Anh Khoa
		Bổ sung: 2.2 Sequence Diagram	Võ Thái Toàn
4	11/10/2022	Meet: xem lại báo cáo	
		Chỉnh sửa: Class Diagram	Cao Trần Anh Khoa, Nguyễn Quang Huy, Nguyễn Đức An
		Chỉnh sửa: 2.3 Solution evaluation	Trần Chí Công
5	15/10/2022	Meet: xem lại báo cáo	
		Chỉnh sửa báo cáo	Trần Chí Công
		Nộp bài	Nguyễn Đặng Anh Khoa

Task III: ARCHITECTURE DESIGN

STT	Thời gian	Công việc	Thành viên
1	17/10/2022	Meet: thảo luận phần 1 Architectural approach and Modules	
		Phân công: Architectural approach	Cao Trần Anh Khoa



		Phân công: List of Modules	Nguyễn Quang Huy, Nguyễn Đức An
2	21/10/2022	Meet: thảo luận phần 2 Component Diagram - Task Assignment	
		Phân công: 2.1 Mô tả Component Diagram	Trần Chí Công
		Phân công: 2.2 Bản vẽ Component Diagram	Nguyễn Đặng Anh Khoa, Võ Thái Toàn
		Chỉnh sửa: 1.2 List of modules	Võ Thái Toàn
3	26/10/2022	Meet: thảo luận về phần 3 Deployment diagram	
		Phân công: 3.1 Bản vẽ Deployment Diagram	Nguyễn Đức An, Nguyễn Đặng Anh Khoa
		Phân công: Mô tả Deployment Diagram	Trần Phúc Anh, Nguyễn Đức An
4	2/11/2022	Meet: xem lại báo cáo	
		Chỉnh sửa: Bản vẽ Component Diagram	Nguyễn Quang Huy
		Chỉnh sửa: Mô tả Component Diagram	Cao Trần Anh Khoa
		Bổ sung: Bản vẽ Deployment Diagram	Trần Chí Công
		Bổ sung: 1.1 Architectural approach	Trần Phúc Anh
5	5/11/2022	Meet: xem lại báo cáo	
		Chỉnh sửa báo cáo	Nguyễn Đặng Anh Khoa
		Nộp bài	Nguyễn Đặng Anh Khoa

Task IV: IMPLEMENTATION - SPRINT 1

STT	Thời gian	Công việc	Thành viên
1	7/11/2022	Meet: thảo luận phần cài đặt git	
		Cài đặt git	
		Phân công: Revisions or Changlogs	Nguyễn Đức An, Võ Thái Toàn
2	9/11/2022	Meet: thảo luận phần Implementation	
		Phân công: Giao diện đăng nhập	Nguyễn Quang Huy, Cao Trần Anh Khoa
		Phân công: Giao diện quản lý ca làm việc	Trần Phúc Anh, Trần Chí Công



		Phân công: Giao diện quản lý nhiệm vụ	Cao Trần Anh Khoa, Nguyễn Đức An
		Phân công: Giao diện xem chi tiết nhiệm vụ và tạo route tối ưu	Nguyễn Đặng Anh Khoa, Trần Phúc Anh
3	16/11/2022	Meet: thảo luận và trình bày báo cáo	
		Chỉnh sửa: Giao diện đăng nhập	Nguyễn Đặng Anh Khoa
		Chỉnh sửa: Giao diện quản lý ca làm việc	Nguyễn Quang Huy
		Chỉnh sửa: Giao diện quản lý nhiệm vụ	Võ Thái Toàn, Nguyễn Đức An
		Bổ sung: Giao diện xem chi tiết nhiệm vụ và tạo route tối ưu	Trần Chí Công
4	19/11/2022	Meet: xem lại figma và báo cáo	
		Chỉnh sửa báo cáo	Nguyễn Quang Huy
		Nộp bài	Nguyễn Đức An

Task V: IMPLEMENTATION - SPRINT 2

STT	Thời gian	Công việc	Thành viên
1	20/11/2022	Meet: Thảo luận sprint 2	
		Set up môi trường	Nguyễn Đức An
		Add and edit task page	Nguyễn Quang Huy
		Employee page	Võ Thái Toàn
		Login page	Trần Phúc Anh
		MCP page	Nguyễn Đặng Anh Khoa
		Not found page	Nguyễn Đặng Anh Khoa
		Route planner page	Cao Trần Anh Khoa
		Shift page	Trần Chí Công
		Task page	Nguyễn Quang Huy
		Task detail page	Võ Thái Toàn
		Vehicle page	Nguyễn Đức An
		Chuẩn bị Data	Trần Phúc Anh, Võ Thái Toàn



2	Từ 21/11/2022 đến 4/12/2022	Các thành viên trong nhóm thực hiện bài làm và gửi lên trang git của nhóm	
3	7/12/2022	Meet: Họp, chỉnh sửa, tổng kết báo cáo (cập nhật và chỉnh sửa các diagram cùng mô tả, viết tổng kết)	
		Chuẩn bị slide thuyết trình	Cao Trần Anh Khoa, Trần Chí Công, Nguyễn Đức An
		Phân công thuyết trình	Cả nhóm
4	10/12/2022	Họp, tổng kết báo cáo	
		Nộp bài	Nguyễn Đặng Anh Khoa



PHẦN I

REQUIREMENT ELICITATION

1 Mô tả dự án

1.1 Ngữ cảnh

Quản lý chất thải đô thị là một trong những vấn đề quan trọng mà nhiều quốc gia trên thế giới phải đối mặt. Tuy nhiên, trong bối cảnh đô thị hóa hiện nay, việc quản lý chất thải rất tốn kém và không mang lại hiệu quả, từ đó dẫn đến nhiều vấn nạn liên quan đến con người và môi trường, đe dọa đến sự tồn vong của toàn nhân loại. Cải thiện việc thu gom và quản lý chất thải được các chính phủ và tổ chức chú trọng với mục tiêu tạo nên một thành phố xanh, sạch và tuyệt vời.

Một nhà cung cấp Dịch vụ quản lý chất thải chuyên nghiệp Y, với mong muốn cải thiện và nâng cao hiệu quả của việc quản lý và thu gom chất thải tại một thành phố, đã ký hợp đồng phát triển hệ thống quản lý thông tin có tên là UWC 2.0 với tổ chức X. Hệ thống UWC 2.0 này sẽ hỗ trợ quản lý việc thu gom chất thải trong một thành phố do nhà cung cấp Dịch vụ quản lý chất thải Y đang đảm nhiệm thông qua việc lập lịch trình và nhiệm vụ cho các nhân viên có liên quan làm việc hàng ngày.

Hệ thống UWC 2.0 lưu trữ và quản lý các thông tin về nhân viên gồm Back officer (Cán bộ phụ trách), Collector (Người điều khiển xe thu gom), Janitor (người gom rác ở MCP); thông tin về vị trí các Điểm thu gom (MCP); thông tin về phương tiện sử dụng; thông tin về thời gian các ca làm việc và thông tin các nhiệm vụ trong các ca. Bên cạnh đó, hệ thống sẽ cung cấp các tính năng dành cho từng loại người dùng cụ thể và giao diện người dùng thông qua truy cập trực tuyến trên website của nhà cung cấp Dịch vụ quản lý chất thải chuyên nghiệp Y.

1.2 Quy trình nghiệp vụ

Việc thu gom chất thải được chỉ định cho một tổ chức cung cấp dịch vụ quản lý chất thải chuyên nghiệp. Quy trình thu gom chất thải cơ bản bao gồm:



(1) Hàng tháng, Back officer chỉ định các phương tiện đưa vào sử dụng và quyết định các điểm thu gom (MCP) mỗi phương tiện đi qua trong lộ trình.

(2) Hàng tuần, Back officer chỉ định thời gian các ca làm việc trong tuần; tạo các nhiệm vụ trong ca, chọn phương tiện sử dụng và phân công Collector cùng các Janitors cho nhiệm vụ.

(3) Hàng ngày, trước mỗi ca làm việc của mình, Collector và Janitor tiến hành điểm danh.

(4) Sau khi điểm danh thành công, Collector và Janitor nhận nhiệm vụ.

Các Janitor thu gom rác thủ công từ các hộ gia đình và đưa đến MCP được chỉ định cho mình.

Sau khi Janitor đã hoàn thành việc thu gom ở các MCP thuộc lộ trình của phương tiện được sử dụng trong nhiệm vụ, Back Office tiến hành chọn ra các MCP cần được gom đi (đạt mức sức chứa quy định) để thiết lập lộ trình và gửi cho Collector.

Collector điều khiển phương tiện đi thu gom rác theo lộ trình và đưa rác đến các Depot.

(5) Sau khi hoàn thành nhiệm vụ, Collector và Janitor tiến hành điểm danh kết thúc công việc.

1.3 Giả định dự án

Một số giả định nghiệp vụ

Giả định 1. Hàng tháng, Back officer chỉ quyết định các điểm thu gom (MCP) cho mỗi phương tiện được đưa vào hoạt động. Lộ trình tối ưu cụ thể sẽ được hệ thống hỗ trợ back officer để lập ra cho tài xế của mỗi ca dựa trên tập hợp MCP đã được quyết định mỗi tháng của xe tương ứng.

Giả định 2. Back officer chỉ được giao tối đa 2 nhiệm vụ (tương ứng với 2 ca) cho 1 nhân viên trong 1 ngày.

Giả định 3. Một ca làm việc sẽ có khung thời gian cụ thể trong ngày. Trong một ca, back officer có thể phân nhiều nhiệm vụ.

Giả định 4. Ứng với mỗi nhiệm vụ, back officer cần chỉ định phương tiện được sử dụng, 1 collector điều khiển xe và phân công cụ thể janitor cho từng MCP của lộ trình xe đi qua (*tối thiểu 2 và tối đa 4 janitors cho 1 MCP*).

Giả định 5. Nhóm janitor được phân công vào chung một MCP tự phân công nhau để đi thu gom rác từ các hộ gia đình. Back officer chỉ định nhóm trưởng cho mỗi MCP. Nhóm trưởng có nhiệm vụ cập nhật tiến độ nhiệm vụ khi nhóm janitor của mình hoàn thành thu gom.

Giả định 6. Một ca làm việc bao gồm các trạng thái: Chưa diễn ra (Chưa đến thời gian



bắt đầu của ca), Đang diễn ra (Đang trong thời gian làm việc của ca), Kết thúc (Đã qua thời gian kết thúc của ca).

Giả định 7. Một nhiệm vụ bao gồm các trạng thái: Chưa bắt đầu, Đang thu gom (Janitor đang gom rác đến MCP), Vận chuyển (Collector đang chuyển rác từ các MCP chỉ định đến Depot), Hoàn thành.

Một số giả định hệ thống

Giả định 1. Back officer được cấp sẵn tài khoản có thể truy cập các chức năng cần thiết hệ thống cung cấp. Collector và Janitor cần đăng ký tài khoản và cần được cấp quyền để truy cập tài nguyên.

Giả định 2. Sức chứa của các MCP được xác định bởi các cảm biến, gửi thông tin đến hệ thống mỗi 15 phút (tối đa cho phép).

Giả định 3. Hệ thống lọc ra các MCP có sức chứa còn lại nhiều nhất 15%, kết hợp với địa điểm bắt đầu (nơi lấy phương tiện điều khiển) và địa điểm kết thúc lộ trình (các Depot - nơi xử lý rác) để tiến hành tạo lộ trình tối ưu.

1.4 Stakeholders

1. Internal Stakeholder

- **Back officers:** điều phối công việc của collectors và janitors.
- **Collectors:** lái phương tiện theo lộ trình được chỉ định bởi các back officers để thu gom rác tại các MCPs .
- **Janitors:** thu gom rác tại các hộ gia đình và vận chuyển đến MCP được chỉ định.

2. External Stakeholder

- **IT staff:** chịu trách nhiệm cài đặt và bảo trì hệ thống.
- **Organization X:** là tổ chức chịu trách nhiệm phát triển hệ thống, đảm bảo hệ thống được phát triển đúng theo nhu cầu và mong đợi của người dùng.
- **Service Provider Y:** là đơn vị sẽ sử dụng hệ thống UWC 2.0 để hỗ trợ quy trình thu gom rác thải đô thị, nhằm nâng cao hiệu suất, rút ngắn thời gian và tiết kiệm chi phí, nhiên liệu.
- Ngoài ra còn có các nhà cung cấp trang thiết bị xử lý và thu gom rác, nhà cung cấp nhiên liệu (xăng, dầu), nhà cung cấp dịch vụ sửa chữa phương tiện vận chuyển.

1.5 Tâm vực dự án

Về mục tiêu, dự án UWC 2,0 được lên kế hoạch và phát triển nhằm tạo ra môi trường làm việc:



- Thân thiện với người dùng, dễ sử dụng, thời gian phản hồi nhanh, dễ dàng cập nhật và bảo trì, tính bảo mật cao.
- Cho phép Back Officers có thể quản lý thông tin của Collectors và Janitors, các phương tiện thu gom và vận chuyển (Collecting Vehicle) và tình trạng của các điểm thu gom (MCPs).
- Cho phép Back Officers có thể phân công công việc và theo dõi tiến độ công việc cho Collectors và Janitors, quan sát được tình trạng sức chứa của các MCPs mà không cần đến trực tiếp các điểm thu gom, có thể tạo lộ trình làm việc thuận tiện, tiết kiệm chi phí đi lại, chi phí về nhiên liệu (xăng dầu).
- Cho phép Collectors và Janitors có thể xem và nắm rõ được lịch làm việc của mình, có thể dễ dàng báo cáo công việc của bản thân chỉ với một vài thao tác đơn giản.
- Cho phép các nhân viên trong hệ thống có thể giao tiếp với nhau nhanh chóng.
- Xét theo phương diện môi trường, hệ thống cho phép các Back Officers tạo và phân công lộ trình theo cách tối ưu hóa việc tiêu thụ nhiên liệu và độ dài đường đi, từ giảm lượng khí thải của phương tiện vận chuyển thải ra môi trường, giảm chi phí sửa chữa, bảo dưỡng phương tiện thu gom.

Về thời gian hoàn thành, dự án sẽ được thực hiện trong 12 tuần bao gồm 5 nhiệm vụ chính (5 tasks):

- Task 1 được thực hiện trong 2 tuần đầu.
- Task 2 được thực hiện trong tuần thứ 3 và 4.
- Task 3 được thực hiện trong tuần thứ 5, 6 và 7.
- Task 4 được thực hiện trong tuần thứ 8 và 9.
- Task 5 được thực hiện trong tuần thứ 10, 11, 12.

1.6 User Story

- **Back officers**

- Là back officers, tôi muốn quản lý được thông tin về lịch làm việc và tiến độ công việc của tất cả collectors và janitors trong hệ thống.
- Là back officers, tôi muốn xem được thông tin và tình trạng của các phương tiện thu gom (khối lượng, sức chứa, mức tiêu thụ nhiên liệu ...).



- Là back officers, tôi muốn xem được thông tin và tình trạng sức chứa của các điểm thu gom (MCPs).
- Là back officers, tôi muốn phân công công việc, thời gian làm việc và sắp xếp phương tiện, trang thiết bị cho collectors và janitors.
- Là back officers, tôi muốn được hệ thống đề xuất lộ trình tuyến đường thuận tiện và tiết kiệm nhiên liệu nhất cho collectors.

- **Janitors & Collectors**

- Là janitors/collectors, tôi muốn xem được thông tin chung về lịch làm việc của bản thân hàng ngày, hàng tuần (các ca làm việc và thông tin nhiệm vụ trong ca của tôi).
- Là janitors/collectors, tôi muốn giao tiếp qua tin nhắn với các collectors, janitors khác và back officers.
- Là janitors/collectors, tôi muốn điểm danh đầu ca làm việc (check in) và báo cáo hoàn thành ca làm việc (check out) cho back officers.
- Là janitors/collectors, tôi muốn biết những thông tin về các MCPs khi xảy ra tình trạng quá tải.
- Là collectors, tôi muốn xem thông tin của phương tiện được bàn giao cho tôi trong các ca làm việc.
- Là collectors, tôi muốn biết được tuyên đường tối ưu nhất để tiến hành đi thu thập rác từ các MCPs.
- Là janitors, tôi muốn biết địa điểm những điểm thu gom được chỉ định cho tôi.

1.7 Yêu cầu người dùng và Yêu cầu hệ thống

- **Back officers**



Yêu cầu người dùng

Xem tổng quan thông tin về các Collector, Janitor và lịch trình làm việc của họ.

Yêu cầu hệ thống

- Hệ thống hiển thị danh sách chứa thông tin tổng quan của các nhân viên làm việc cho tổ chức.
- Hệ thống cho phép xem thông tin chi tiết của nhân viên bất kỳ, cũng như gửi tin nhắn cho nhân viên bất kỳ.
- Hệ thống cung cấp chức năng gán quyền nhân viên cho tài khoản user đăng ký cũng như thu hồi quyền của tài khoản đó.
- Hệ thống cho phép xem tổng quan lịch làm việc của nhân viên bất kỳ.
- Hệ thống cho phép lọc danh sách theo các trường, sắp xếp theo thứ tự bảng chữ cái và có chức năng tìm kiếm.
- Hệ thống cho phép xuất danh sách dưới dạng tập tin .pdf, .xlsx và cho phép in danh sách.

Yêu cầu người dùng

Xem tổng quan về các phương tiện thu gom đang có và thông số kỹ thuật tương ứng.

Yêu cầu hệ thống

- Hệ thống hiển thị tổng quan dưới dạng một danh sách các phương tiện thu gom với các trường thông tin quan trọng.
- Hệ thống cho phép xem thông tin chi tiết của phương tiện.
- Hệ thống cho phép lọc danh sách theo các trường, sắp xếp theo thứ tự bảng chữ cái và có chức năng tìm kiếm.
- Hệ thống cho phép xuất danh sách dưới dạng tập tin .pdf, .xlsx và cho phép in danh sách.



Yêu cầu người dùng

Xem tổng quan về tất cả các Điểm thu gom.

Yêu cầu hệ thống

- Hệ thống hiển thị tổng quan dưới dạng một danh sách các Điểm thu gom với các trường thông tin như Mã Điểm thu gom, Vị trí, Sức chứa.
- Hệ thống cho phép xem thông tin chi tiết của Điểm thu gom.
- Hệ thống thông báo khi có Điểm thu gom đạt mức sức chứa được quy định.
- Hệ thống cho phép lọc danh sách theo các trường, sắp xếp theo thứ tự bảng chữ cái và có chức năng tìm kiếm.
- Hệ thống cho phép xuất danh sách dưới dạng tập tin .pdf, .xlsx và cho phép in danh sách.

Yêu cầu người dùng

Tạo và quản lý các ca làm việc.

Yêu cầu hệ thống

- Hệ thống cho phép tạo mới các ca làm việc.
- Hệ thống hiển thị tổng quan dưới dạng danh sách các ca làm việc đã tạo theo thứ tự thời gian từ gần nhất đến xa nhất.
- Hệ thống cho phép xem thông tin chi tiết của ca làm việc.
- Hệ thống cho phép lọc các ca làm việc theo ngày và theo trạng thái.



Yêu cầu người dùng

Phân công nhiệm vụ cho Collector và Janitor.

Yêu cầu hệ thống

- Hệ thống cho phép phân nhiệm vụ mới, chọn phương tiện sử dụng trong nhiệm vụ, chọn Collector điều khiển phương tiện và phân các nhóm Janitor vào các Điểm thu gom (MCP) trong lô trình của phương tiện.
- Hệ thống cho phép chỉnh sửa thông tin về phân công cho các nhiệm vụ, xóa nhiệm vụ nếu cần.
- Hệ thống cho phép theo dõi và xác nhận tiến độ cho các nhiệm vụ.

Yêu cầu người dùng

Tạo lô trình di chuyển tối ưu Collector.

Yêu cầu hệ thống

- Hệ thống hỗ trợ lọc ra các điểm thu gom đạt ngưỡng yêu cầu về sức chứa và tạo ra lô trình tối ưu cho Collector.
- Hệ thống cho phép gửi trực tiếp lô trình tối ưu cho Collector sau khi tạo dưới dạng bản đồ trực quan.



Yêu cầu người dùng

Gửi tin nhắn đến Collector hoặc Janitor.

Yêu cầu hệ thống

- Hệ thống cho phép nhắn tin với một người dùng, một nhóm người dùng hoặc toàn bộ người dùng của hệ thống.
- Tin nhắn phải được gửi đến người nhận với độ trễ nhỏ hơn 1 giây theo thời gian thực.
- Tin nhắn có thể là văn bản, hình ảnh, video, tệp tin đính kèm với dung lượng không quá 100 MB.

• Collectors & Janitors

Yêu cầu người dùng

Xem tổng quan và chi tiết về lịch làm việc của mình.

Yêu cầu hệ thống

- Hệ thống hiển thị lịch tổng quan theo tuần và hỗ trợ lọc xem danh sách nhiệm vụ trong một ngày cụ thể.
- Hệ thống cho xem thông tin chi tiết của một ca làm việc như thời gian làm việc và thông tin nhiệm vụ được phân trong ca đó như vị trí các Điểm thu gom, Tiễn độ nhiệm vụ...
- Nếu có thay đổi từ Back officer, hệ thống phải được cập nhật thông tin với độ trễ không quá 2 giây.



Yêu cầu người dùng

Nhắn tin với các nhân viên khác.

Yêu cầu hệ thống

- Hệ thống cho phép nhắn tin với một người dùng, một nhóm người dùng.
- Tin nhắn phải được gửi đến người nhận với độ trễ nhỏ hơn 1 giây theo thời gian thực.
- Tin nhắn có thể là văn bản, hình ảnh, video, tệp tin đính kèm với dung lượng không quá 100 MB.

Yêu cầu người dùng

Điểm danh cho ca làm việc.

Yêu cầu hệ thống

- Hệ thống hiển thị thông báo điểm danh trong vòng 15 phút trước thời gian bắt đầu ca làm việc. Sau khoảng thời gian đó, hệ thống mặc định trạng thái làm việc của nhân viên là Vắng nếu nhân viên không điểm danh.
- Hệ thống cho phép điểm danh kết thúc ca làm việc (check out).
- Sau khi thực hiện điểm danh, hệ thống phải cập nhật trạng thái làm việc của nhân viên với độ trễ không quá 1 giây.

Yêu cầu người dùng

Được thông báo nếu Điểm thu gom đã đầy.

Yêu cầu hệ thống

- Khi có một Điểm thu gom đầy, hệ thống gửi thông báo “Điểm thu gom đã đầy” cho các Janitor đã được phân công trong nhiệm vụ.
- Từ thời điểm Điểm thu gom bắt đầu gửi thông tin về hệ thống đến thời điểm nhận được thông báo với độ trễ không quá 2 giây.



2 Xác định yêu cầu

2.1 Yêu cầu chức năng

General:

Requirement	Feature	Descriptions
G01	Login	User đăng nhập tài khoản cá nhân vào hệ thống và được cấp tài nguyên hệ thống theo vai trò. Ngoài ra còn có chức năng giúp user đặt lại mật khẩu khi quên.
G02	Logout	User đăng xuất tài khoản khỏi hệ thống.

Back Officers:

Requirement	Feature	Descriptions
B01	Manage employee	Back officers có thể thêm, xoá, điều chỉnh nhân sự cho hệ thống (collectors hoặc janitors).
B02	View employee information	Back officers có thể xem thông tin cá nhân, lịch làm việc và tìm kiếm thông tin nhân sự.
B03	View vehicles details	Back officers có thể xem các thông số kỹ thuật của các loại phương tiện bao gồm cân nặng, sức chứa, mức tiêu thụ nhiên liệu...
B04	View MCPs capacity	Back officers có thể xem được sức chứa của các MCPs trong hệ thống. Thông tin của các MCPs được cập nhật sau mỗi 15 phút.
B05	Manage shifts	Back officers có thể quản lý danh sách các ca làm việc (thêm, xóa, sửa): thông tin thời gian làm việc, theo dõi số nhiệm vụ đã hoàn thành trong ca.
B06	Assign tasks	Back officers có thể phân nhiệm vụ mới trong ca làm việc: chỉ định phương tiện sử dụng, Collector và các Janitor đảm nhận các MCP trong lộ trình phương tiện.



B07	Plan vehicles and routes	Back officers có thể quyết định các phương tiện được sử dụng cùng với tuyến đường tương ứng cho phương tiện đó hàng tháng.
B08	Confirm task progress	Back officers có thể xác nhận tiến độ cho các nhiệm vụ với thông báo về tiến độ được gửi đến từ nhân viên.
B09	Send messages	Back officers có thể gửi tin nhắn đến collectors và janitors.
B10	Send Optimized Route	Back officers có thể gửi lộ trình tối ưu được tạo đến Collector.

Collectors và Janitors:

Requirement	Feature	Descriptions
CJ01	View calendar	Collectors và Janitors có thể xem được lịch làm việc của họ trong ngày, tuần: danh sách các ca làm việc.
CJ02	View tasks	Collectors và Janitors có thể xem được mô tả cụ thể các nhiệm vụ được giao hàng ngày.
CJ03	Send messages	Collectors và Janitors có thể gửi tin nhắn đến back officers.
CJ04	Check in / check out	Collectors và Janitors có thể điểm danh chấm công cho các nhiệm vụ.
CJ05	View full MCPs	Collectors và Janitors có thể nhận được thông báo về các MCPs trong lộ trình đã đầy.
CJ06	Update progress	Collectors và Janitors có thể cập nhật tiến độ cho nhiệm vụ.
CJ07	Register	Collectors và Janitors có thể đăng ký tài khoản trên hệ thống và được cấp tài nguyên dựa trên vai trò.



System

Requirement	Feature	Descriptions
ST01	Suggest Optimized Route	Hệ thống có thể đề xuất lộ trình tối ưu cho Collector dựa vào sức chứa các MCP trong lộ trình phương tiện.

2.2 Non-functional Requirements

Performance

Requirement	Feature	Descriptions
P01	Handle real-time messages	Quá trình gửi tin nhắn giữa các người dùng chỉ được delay không quá 1 giây.
P02	Handle real-time data	Xử lý được dữ liệu với ít nhất 1000 MCPs trong cùng một thời điểm và đạt được mục tiêu 10.000 MCPs trong 5 năm tiếp theo.

Reliability

Requirement	Feature	Descriptions
R01	Provide high availability	Availability của thông tin về sức khỏe được cập nhật từ MCPs (mỗi 15 phút) phải đạt tối thiểu 95% trong suốt quá trình vận hành.
R02	Backup data	Hệ thống sẽ tự động lưu lại các logs vào mỗi ngày, admin và back officers có thể backup lại dữ liệu khi cần.

Ease of use



Requirement	Feature	Descriptions
E01	Provide clear interface	Giao diện đẹp, rõ ràng, đảm bảo người dùng có thể dễ dàng tập trung vào những thông tin quan trọng, có hỗ trợ mode sáng tối.
E02	Display one-page tasks	Các thông tin quan trọng của task được trình bày trong 1 trang màn hình (không cần phải scroll down)
E03	Provide clear feature	Các tính năng đơn giản, dễ sử dụng, đảm bảo mỗi chức năng thực hiện dưới 4 thao tác.
E04	Provide user guide	Hướng dẫn sử dụng trọng tâm, dễ hiểu, đảm bảo mọi đối tượng người dùng đều có thể sử dụng các thao tác cơ bản sau 5 phút hướng dẫn.
E05	Support multi language	Hỗ trợ hai ngôn ngữ tiếng Anh và tiếng Việt.
E06	Support multi platform	Trang web hỗ trợ responsive và đảm bảo sự tương thích trên nhiều thiết bị (computer, mobile, tablet), hệ điều hành (Android, iOS), browser (Chrome, MS Edge, Safari),....

Security

Requirement	Feature	Descriptions
SE01	Strong password	Mật khẩu của mỗi tài khoản được yêu cầu phải ít nhất 8 chữ cái, bao gồm cả chữ in hoa, in thường, số và ký tự đặc biệt.
SE02	Auto block	Trong trường hợp người dùng đăng nhập sai 5 lần sẽ tự động khóa tài khoản, để mở khóa cần liên hệ với quản trị viên để xác nhận thông tin.
SE03	Secure website	Trang web sử dụng một số công nghệ như TSL, SSL, JWT để bảo mật cho website, ngăn ngừa tấn công từ bên ngoài.



Maintainability

Requirement	Feature	Descriptions
M01	Get user feedback	Hệ thống ghi nhận feedback từ người dùng về các vấn đề trên hệ thống, nhằm sửa chữa lỗi (nếu có) và update tính năng trong các lần cập nhật tiếp theo.
M02	Quarter maintenance	Hệ thống sẽ được bảo trì định kỳ theo quý (3 tháng), thời gian bảo trì không được trùng với giờ hành chính, thời gian bảo trì tối đa 3 tiếng.

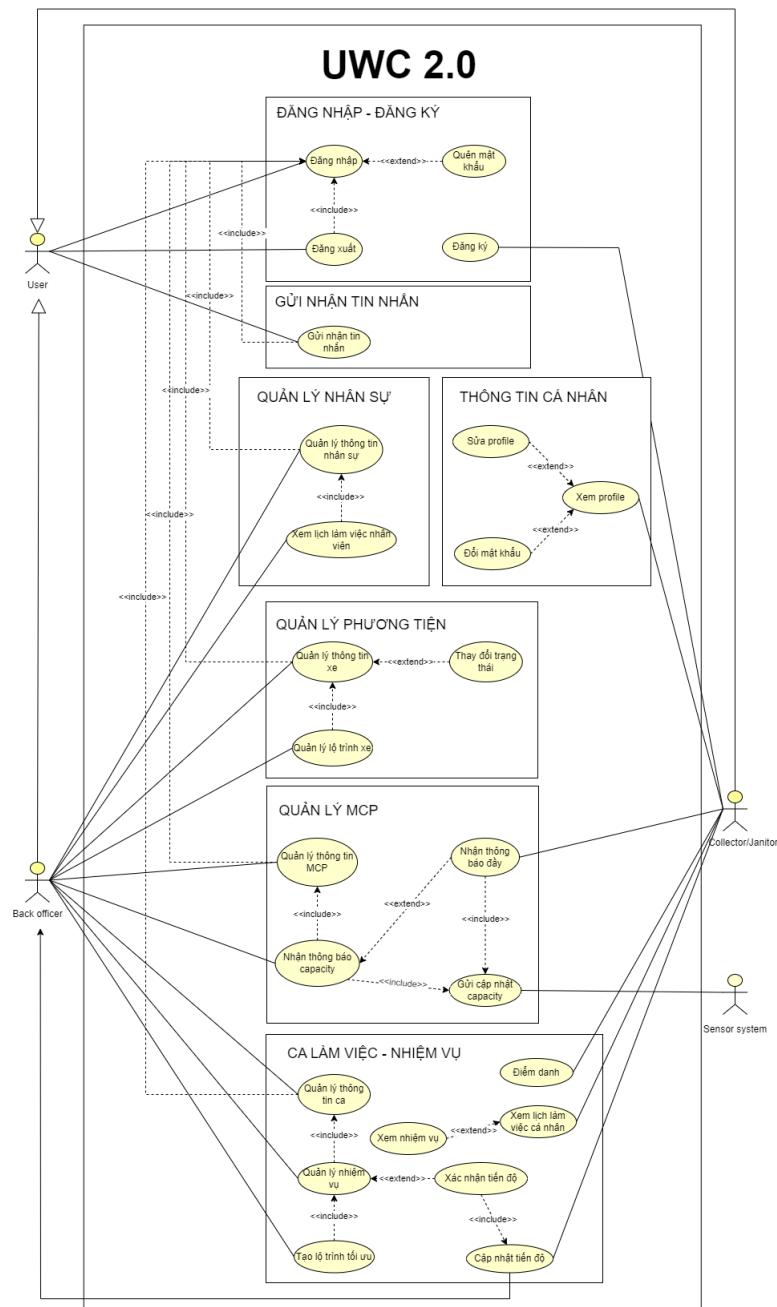
Scalability

Requirement	Feature	Descriptions
S01	Connect UWC 1.0	Kết nối được với cơ sở dữ liệu của ứng dụng UWC 1.0
S02	Extend Compatibility	Hệ thống tương thích với nhiều tổ chức quản lý rác khác nhau, có thể mở rộng quy mô ra toàn cầu.

3 Use-case diagram

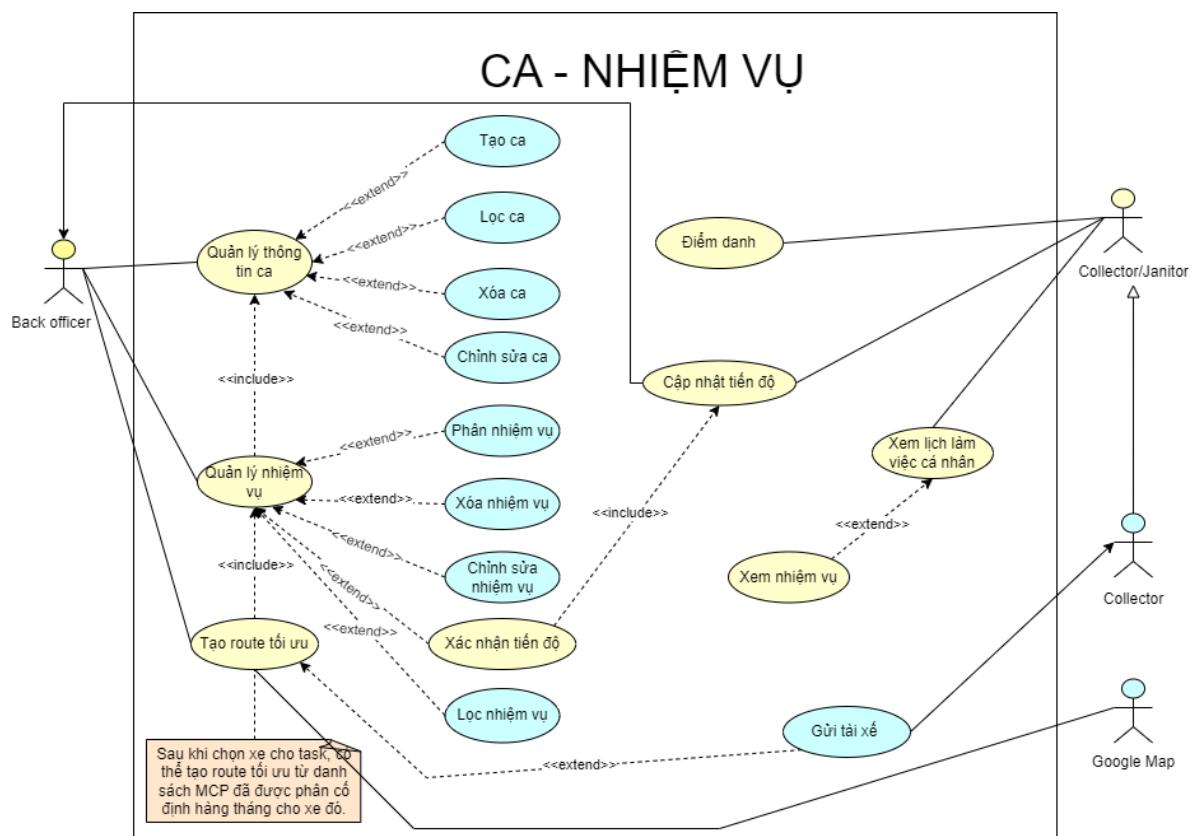
Xem các bản vẽ Use-case diagram FULL HD của nhóm tại https://drive.google.com/drive/folders/1-eux6bosRaMz1_3jPxYIF5uX-SSXgEpe?usp=sharing

3.1 Use-case diagram of whole system



Hình 1: Use case Hệ thống

3.2 Use-case và đặc tả cho Task Assignment (Ca - nhiệm vụ)



Hình 2: Use case Task assignment

Quản lý thông tin ca

Use case name	Quản lý thông tin ca
Actors	Back officer
Description	Là một back officer, tôi muốn quản lý danh sách các ca làm việc một cách tổng quan hoặc chi tiết.
Trigger	Back officer chọn mục “Ca và nhiệm vụ” trong menu chính hiển thị màn hình thiết bị.
Preconditions	<ul style="list-style-type: none"> Back officer đã đăng nhập hệ thống và có quyền truy cập chức năng “Ca và nhiệm vụ”. Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.



Postconditions	Dữ liệu thay đổi do back officer thực hiện các thao tác quản lý (nếu có) được cập nhật trong database và hiển thị lại trên màn hình.
Normal Flow	<ol style="list-style-type: none">1. Back officer chọn mục “Ca và nhiệm vụ” trong menu chính.2. Hệ thống mở cửa sổ “Ca và nhiệm vụ”.3. Hệ thống lấy dữ liệu danh sách các ca.4. Hệ thống hiển thị danh sách tất cả các ca. Extension points: Lọc ca, Xóa ca, Chỉnh sửa ca, Tạo ca mới.5. Back officer chọn xem cụ thể một ca trong danh sách.6. Hệ thống lấy thông tin chi tiết của ca được chọn và hiển thị.
Alternative Flow	Không có
Exception Flow	<p>E1: Tại bước 4.</p> <p>4.1 Hệ thống lấy thông tin từ database thất bại và hiển thị thông báo.</p> <p>Use case dừng lại.</p> <p>E2: Tại bước 6.</p> <p>6.1 Hệ thống thất bại trong việc lấy thông tin chi tiết của ca và báo lỗi.</p> <p>Use case dừng lại.</p>
Constraint	<ul style="list-style-type: none">– Màn hình giao diện có thể hiển thị 5 ca, 10 ca hoặc 25 ca theo thứ tự từ mới nhất đến cũ nhất trên một trang.– Những thông tin quan trọng của ca như thời gian, trạng thái, số nhiệm vụ đã hoàn thành phải được hiển thị tóm lược trong 1 hàng của danh sách.



Lọc ca

Use case name	Lọc ca
Actors	Back officer
Description	Là một back officer, tôi muốn lọc ra các ca theo ngày và/hoặc theo trạng thái.
Trigger	Trong lúc ở màn hình giao diện Ca và nhiệm vụ, Back officer muốn lọc ra chỉ các ca với ngày và/hoặc trạng thái cụ thể.
Preconditions	<ul style="list-style-type: none">– Thiết bị back officer đang ở cửa sổ “Ca và nhiệm vụ”.– Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	Màn hình hiển thị danh sách các ca được lọc thỏa mãn thông tin tại các trường lọc.
Normal Flow	<ol style="list-style-type: none">1. Back officer chọn ngày muốn lọc.2. Back officer chọn trạng thái muốn lọc.3. Back officer nhấn nút “Lọc”.4. Hệ thống lấy dữ liệu các ca đã tạo từ database.5. Hệ thống lọc ra các ca thỏa mãn thông tin tại các trường lọc.6. Hệ thống hiển thị ra màn hình kết quả lọc.
Alternative Flow	<p>A1: Tại bước 1.</p> <p>1.1 Back officer bỏ trống trường "Ngày".</p> <p>Tiếp tục tại bước 2 trong Normal Flow.</p> <p>A2: Tại bước 2.</p> <p>2.1 Back officer bỏ trống trường "Trạng thái".</p> <p>Tiếp tục tại bước 3 trong Normal Flow.</p>



Exception Flow	<p>E1: Tại bước 6.</p> <p>6.1 Hệ thống thông báo không tìm thấy ca thỏa mãn. Use-case dừng lại.</p>
Constraint	Không có.

Xóa ca

Use case name	Xóa ca
Actors	Back officer
Description	Là một back officer, tôi muốn xóa một ca ra khỏi danh sách.
Trigger	Back officer click vào button “Xóa” nằm bên phải hàng của ca tương ứng muốn xóa.
Preconditions	<ul style="list-style-type: none">– Thiết bị back officer đang ở cửa sổ “Ca và nhiệm vụ”.– Hàng chứa thông tin ca có trong danh sách hiển thị trên màn hình giao diện.– Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	Hàng chứa ca vừa xóa không còn xuất hiện trong danh sách ca. Đồng thời dữ liệu về ca cũng bị xóa khỏi database hệ thống.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống hiển thị ra 1 dialog để xác nhận xóa ca.2. Back officer chọn xác nhận xóa.3. Hệ thống xác nhận và thực hiện xóa ca ra khỏi danh sách.4. Hệ thống thông báo xóa ca thành công.
Alternative Flow	Không có.



Exception Flow	<p>E1: Tại bước 2.</p> <p>2.1 Back officer chọn Hủy xóa. 2.2 Hệ thống thoát khỏi dialog và trở lại cửa sổ “Ca và nhiệm vụ”. Use-case dừng lại.</p>
Constraint	Không có.

Tạo ca

Use case name	Tạo ca
Actors	Back officer
Description	Là một back officer, tôi muốn tạo một ca làm việc mới.
Trigger	Back officer click vào button “Tạo ca mới”.
Preconditions	<ul style="list-style-type: none">– Thiết bị back officer đang ở cửa sổ “Ca và nhiệm vụ”.– Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	Hàng chứa ca vừa tạo xuất hiện ở trong danh sách ca.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống hiển thị ra 1 dialog chứa mẫu nhập thông tin ca.2. Back officer nhập/chọn thông tin cho ca tại các ô nhập/chọn (ngày, giờ bắt đầu, giờ kết thúc, trạng thái).3. Back officer click nút “Lưu lại”.4. Hệ thống thông báo tạo ca thành công và hiển thị thông tin của ca vừa được tạo trên danh sách.



Alternative Flow	<p>A1: Tại bước 3.</p> <p>3.1 Back officer click nút "Đặt lại".</p> <p>3.2. Hệ thống xóa thông tin đang có trên các trường của form nhập.</p> <p>Tiếp tục tại bước 2 trong Normal Flow.</p>
Exception Flow	<p>E1: Tại bước 2.</p> <p>2.1 Back officer chọn Hủy tạo.</p> <p>2.2 Hệ thống thoát khỏi dialog và trở lại cửa sổ "Ca và nhiệm vụ".</p> <p>Use-case dừng lại.</p> <p>E2: Tại bước 4.</p> <p>4.1 Hệ thống thông báo lỗi "Không thể tạo ca. Khung thời gian trùng với ca đã được tạo!".</p> <p>4.2 Hệ thống thoát dialog và trở lại cửa sổ "Ca và nhiệm vụ".</p> <p>Use-case dừng lại.</p>
Constraint	Ca mới được tạo phải nằm ở hàng đầu tiên trong danh sách ca.

Chỉnh sửa ca

Use case name	Chỉnh sửa ca
Actors	Back officer
Description	Là một back officer, tôi muốn chỉnh sửa một ca làm việc đã tạo.
Trigger	Back officer click vào nút "Chỉnh sửa ca" nằm bên phải hàng của ca tương ứng muốn chỉnh sửa.



Preconditions	<ul style="list-style-type: none">- Thiết bị back officer đang ở cửa sổ “Ca và nhiệm vụ”.- Hàng chứa thông tin ca muôn chỉnh sửa có trong danh sách hiển thị trên màn hình giao diện.- Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	Thông tin của ca vừa chỉnh sửa được cập nhật trong hệ thống và trên giao diện.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống lấy thông tin của ca làm việc được chọn từ database.2. Hệ thống hiển thị ra 1 dialog chứa mẫu nhập thông tin ca với các trường chứa dữ liệu ca muôn chỉnh sửa.3. Back officer chỉnh sửa thông tin cho ca tại các ô nhập/chọn (ngày, giờ bắt đầu, giờ kết thúc, trạng thái).4. Back officer click nút “Lưu lại”.5. Hệ thống thông báo chỉnh sửa ca thành công.
Alternative Flow	Không có.



Exception Flow	<p>E1: Tại bước 3.</p> <p>3.1 Back officer chọn Hủy bỏ.</p> <p>3.2 Hệ thống thoát khỏi dialog và trở lại cửa sổ “Ca và nhiệm vụ”.</p> <p>Use-case dừng lại.</p> <p>E2: Tại bước 5.</p> <p>5.1 Hệ thống thông báo lỗi “Không thể chỉnh sửa ca. Khung thời gian trùng với ca đã được!”.</p> <p>4.2 Hệ thống thoát dialog và trở lại cửa sổ “Ca và nhiệm vụ”.</p> <p>Use-case dừng lại.</p>
Constraint	Không có.

Quản lý nhiệm vụ

Use case name	Quản lý nhiệm vụ
Actors	Back officer
Description	Là một back officer, tôi muốn quản lý danh sách các nhiệm vụ trong ca cụ thể và xem thông tin chi tiết của nhiệm vụ.
Trigger	Back officer click vào button “Nhiệm vụ” nằm bên phải hàng của ca tương ứng.
Preconditions	Thiết bị back officer đang ở cửa sổ “Ca và nhiệm vụ”. Hàng chứa thông tin ca có trong danh sách hiển thị trên giao diện. Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	Dữ liệu thay đổi do back officer thực hiện các thao tác quản lý nhiệm vụ (nếu có) được cập nhật trong database và hiển thị lại trên màn hình.



Normal Flow	<ol style="list-style-type: none">1. Hệ thống mở cửa sổ quản lý nhiệm vụ của ca tương ứng được chọn.2. Hệ thống lấy dữ liệu danh sách các nhiệm vụ của ca.3. Hệ thống hiển thị danh sách tất cả các nhiệm vụ của ca. Extension points: Lọc nhiệm vụ, Phân nhiệm vụ, Xóa nhiệm vụ, Chỉnh sửa nhiệm vụ, Xác nhận tiến độ.4. Back officer chọn xem cụ thể một nhiệm vụ trong danh sách.5. Hệ thống lấy thông tin chi tiết của nhiệm vụ được chọn và hiển thị (thông tin phân công Collector, Janitors, phương tiện...).
Alternative Flow	Không có.
Exception Flow	<p>E1: Tại bước 3.</p> <p>3.1 Hệ thống lấy thông tin từ database thất bại và hiển thị thông báo lỗi.</p> <p>Use case dừng lại.</p> <p>E2: Tại bước 5.</p> <p>5.1 Hệ thống thất bại trong việc lấy thông tin chi tiết của nhiệm vụ và báo lỗi.</p> <p>Use case dừng lại.</p>
Constraint	<ul style="list-style-type: none">- Màn hình giao diện có thể hiển thị 5, 10 hoặc 25 nhiệm vụ (tùy lựa chọn) theo thứ tự từ mới nhất đến cũ nhất trên một trang.- Những thông tin quan trọng của nhiệm vụ như mã phương tiện, mã lộ trình, tên Collector và trạng thái của nhiệm vụ phải được hiển thị ở hàng của nhiệm vụ đó trong danh sách.



Lọc nhiệm vụ

Use case name	Lọc nhiệm vụ
Actors	Back officer
Description	Là một back officer, tôi muốn lọc ra nhiệm vụ theo Collector và/hoặc theo trạng thái nhiệm vụ.
Trigger	Trong lúc ở màn hình giao diện danh sách nhiệm vụ của một ca nào đó, Back officer muốn lọc ra chỉ các nhiệm vụ có Collector và/hoặc trạng thái cụ thể.
Preconditions	<ul style="list-style-type: none">– Thiết bị back officer đang ở cửa sổ hiển thị danh sách nhiệm vụ của một ca nào đó.– Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	Màn hình hiển thị danh sách các nhiệm vụ được lọc thỏa mãn thông tin tại các trường lọc.
Normal Flow	<ol style="list-style-type: none">1. Back officer nhập tên Collector muốn lọc.2. Back officer chọn trạng thái muốn lọc.3. Back officer nhấn nút “Lọc”.4. Hệ thống lấy dữ liệu các nhiệm vụ đã tạo trong ca cụ thể từ database.5. Hệ thống lọc ra các nhiệm vụ thỏa mãn thông tin tại các trường lọc.6. Hệ thống hiển thị ra màn hình kết quả lọc.



Alternative Flow	<p>A1: Tại bước 1.</p> <p>1.1 Back officer bỏ trống trường "Collector".</p> <p>Tiếp tục tại bước 2 trong Normal Flow.</p> <p>A2: Tại bước 2.</p> <p>2.1 Back officer bỏ trống trường "Trạng thái".</p> <p>Tiếp tục tại bước 3 trong Normal Flow.</p>
Exception Flow	<p>E1: Tại bước 6.</p> <p>6.1 Hệ thống thông báo không tìm thấy nhiệm vụ thỏa mãn.</p> <p>Use-case dừng lại.</p>
Constraint	Không có.

Tạo route tối ưu

Use case name	Tạo route tối ưu
Actors	Back officer (Primary), Google Map (Secondary)
Description	Là một back officer, tôi muốn tạo ra route tối ưu chỉ đi qua các MCP có capacity đạt ngưỡng từ danh sách MCP đã được phân cố định mỗi tháng cho xe trong task để gửi cho tài xế sau khi janitors hoàn thành việc thu gom rác giao đến các MCP.
Trigger	Back officer click button “Tạo route tối ưu” trên giao diện cửa sổ của một nhiệm vụ nào đó đã được tạo.
Preconditions	Hệ thống đang mở giao diện hiển thị thông tin chi tiết của nhiệm vụ. Nhiệm vụ đã được phân công phương tiện sử dụng. Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	Route tối ưu hiển thị trực quan trên Google Map được tích hợp.



Normal Flow	<ol style="list-style-type: none">Back officer click nút “Tạo route tối ưu”.Hệ thống hiển thị dialog thông báo đang xử lý.Hệ thống lấy danh sách các MCP đã phân công định tháng này cho xe phân trong task.Hệ thống lọc ra chỉ các MCP có capacity còn nhiều nhất 15% (giả định).Hệ thống tạo lộ trình tối ưu từ các MCP được lọc.Hệ thống thông báo tạo route tối ưu thành công.Hệ thống đề xuất route tối ưu hiển thị trên Google Map và mô tả cụ thể hướng đi của route và địa chỉ các điểm đi qua theo thứ tự.
Alternative Flow	Không có.
Exception Flow	<p>E1: Tại bước 6.</p> <ol style="list-style-type: none">Hệ thống thông báo tạo route tối ưu thất bại.Hệ thống thoát dialog quay lại cửa sổ của nhiệm vụ. <p>Use case dừng lại.</p>
Assumption	Hệ thống lọc ra các MCP có capacity còn nhiều nhất 15%, kết hợp với địa điểm bắt đầu (công ty) và địa điểm kết thúc lộ trình (nơi xử lý rác) để tiến hành tạo route tối ưu.
Constraint	Thời gian chờ xử lý tạo route tối ưu của hệ thống không quá 3 phút.

Xóa nhiệm vụ

Use case name	Xóa nhiệm vụ
Actors	Back officer
Description	Là một back officer, tôi muốn xóa 1 nhiệm vụ ra khỏi ca.



Trigger	Back officer click vào button “Xóa nhiệm vụ” nằm bên phải hàng của nhiệm vụ tương ứng muốn xóa.
Preconditions	<ul style="list-style-type: none">– Thiết bị back officer đang ở cửa sổ quản lý nhiệm vụ của ca tương ứng.– Hàng của nhiệm vụ muốn xóa có trong phần danh sách đang hiển thị.– Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	Hàng của nhiệm vụ vừa xóa không còn xuất hiện trong danh sách nhiệm vụ của ca.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống hiển thị ra 1 dialog để xác nhận xóa nhiệm vụ.2. Back officer chọn xác nhận xóa.3. Hệ thống xóa nhiệm vụ ra khỏi danh sách nhiệm vụ của ca và thông báo xóa thành công.
Alternative Flow	Không có.
Exception Flow	<p>E1: Tại bước 2.</p> <p>2.1 Back officer chọn Hủy bỏ.</p> <p>2.2 Hệ thống thoát khỏi dialog và trở lại cửa sổ quản lý nhiệm vụ của ca.</p> <p>Use-case dừng lại.</p>
Constraint	Không có.



Chỉnh sửa nhiệm vụ

Use case name	Chỉnh sửa nhiệm vụ
Actors	Back officer
Description	Là một back officer, tôi muốn thay đổi phương tiện hoặc nhân sự đã phân cho một nhiệm vụ.
Trigger	Back officer click vào button “Chỉnh sửa nhiệm vụ” nằm bên phải hàng của nhiệm vụ tương ứng muốn chỉnh sửa.
Preconditions	<ul style="list-style-type: none">– Thiết bị back officer đang ở cửa sổ quản lý nhiệm vụ của ca tương ứng.– Hàng của nhiệm vụ muốn chỉnh sửa có trong phần danh sách đang hiển thị.– Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	Dữ liệu thay đổi được cập nhật và hiển thị mới cho nhiệm vụ được chỉnh sửa.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống lấy thông tin của nhiệm vụ được chọn từ database.2. Hệ thống hiển thị ra 1 trang chứa thông tin chi tiết của nhiệm vụ cho phép chỉnh sửa.3. Back officer thay đổi phân công xe hoặc nhân sự cho nhiệm vụ đó tại các trường nhập/chọn.4. Back officer chọn “Lưu lại” để xác nhận thay đổi.5. Hệ thống xác nhận chỉnh sửa thành công và lưu lại thay đổi.6. Hệ thống cập nhật mới dữ liệu về nhiệm vụ được chỉnh sửa trong danh sách.
Alternative Flow	Không có.



Exception Flow	<p>E1: Tại bước 3.</p> <p>3.1 Back officer chọn Quay lại.</p> <p>3.2 Hệ thống thoát khỏi trang Chính sửa và trở lại cửa sổ quản lý nhiệm vụ của ca.</p> <p>Use-case dừng lại.</p> <p>E2: Tại bước 5.</p> <p>5.1 Hệ thống báo lỗi xe đã được phân cho task khác trong cùng ca.</p> <p>5.2 Hệ thống thoát khỏi dialog và trở lại cửa sổ quản lý task của ca.</p> <p>Use-case dừng lại.</p> <p>E3: Tại bước 5.</p> <p>5.1 Hệ thống báo lỗi nhân sự đã được phân cho task khác trong cùng ca.</p> <p>5.2 Hệ thống thoát khỏi dialog và trở lại cửa sổ quản lý task của ca.</p> <p>Use-case dừng lại.</p> <p>E4: Tại bước 5.</p> <p>5.1 Hệ thống báo lỗi nhân sự đã được phân đủ task trong ngày.</p> <p>5.2 Hệ thống thoát khỏi dialog và trở lại cửa sổ quản lý task của ca.</p> <p>Use-case dừng lại.</p>
Assumption	Back officer chỉ được giao tối đa 2 task/1 nhân viên/1 ngày.
Constraint	Không có.



Phân nhiệm vụ

Use case name	Phân nhiệm vụ
Description	Là một back officer, tôi muốn tạo mới một nhiệm vụ cho ca được chọn và có thể chỉ định phương tiện và phân công nhân sự (collector và janitors) cho nhiệm vụ đó.
Trigger	Back officer click vào button “Phân nhiệm vụ”.
Preconditions	<ul style="list-style-type: none">– Thiết bị Back officer đang ở cửa sổ quản lý task của ca tương ứng.– Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	Hàng chứa task vừa tạo xuất hiện trong danh sách task của ca.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống hiển thị ra 1 trang mới chứa mẫu nhập thông tin nhiệm vụ.2. Back officer chọn(nhập) phương tiện và chọn(nhập) collector điều khiển xe.3. Back officer chọn "Tiếp theo" để chuyển sang trang kế tiếp của mẫu Phân nhiệm vụ.4. Tại trang kế tiếp, hệ thống hiển thị danh sách các MCP trong lộ trình (được phân hàng tháng cho xe).5. Ứng với mỗi MCP, back officer chọn(nhập) tên các janitor (nhóm trưởng và các thành viên).6. Back officer chọn button “Lưu lại”.7. Hệ thống xác nhận thêm nhiệm vụ thành công và lưu thông tin của nhiệm vụ vừa được tạo.



Alternative Flow	<p>A1: Tại bước 7.</p> <p>7.1 Hệ thống báo lỗi janitor đã được phân cho task khác trong cùng ca.</p> <p>7.2 Hệ thống trả lại trang kè trước của mẫu form Phân nhiệm vụ.</p> <p>Trở lại bước 3 của Normal Flow.</p> <p>A2: Tại bước 7.</p> <p>7.1 Hệ thống báo lỗi janitor đã được phân đủ task khác trong ngày.</p> <p>7.2 Hệ thống trả lại trang kè trước của mẫu form Phân nhiệm vụ.</p> <p>Trở lại bước 3 của Normal Flow.</p> <p>A3: Tại bước 7.</p> <p>7.1 Hệ thống báo lỗi phân janitor cho 2 MCP trong cùng ca.</p> <p>7.2 Hệ thống trả lại trang kè trước của mẫu form Phân nhiệm vụ.</p> <p>Trở lại bước 3 của Normal Flow.</p> <p>A4: Tại bước 5.</p> <p>5.1 Back officer nhấn nút "Trở lại" tại trang thứ 2 của mẫu form Phân nhiệm vụ.</p> <p>5.2 Hệ thống trả lại trang kè trước của mẫu form Phân nhiệm vụ.</p> <p>Trở lại bước 2 của Normal Flow.</p> <p>A5: Tại bước 4.</p> <p>4.1 Hệ thống báo xe/collector đã được phân cho task khác trong cùng ca.</p> <p>Trở lại bước 2 của Normal Flow.</p>
------------------	--



Exception Flow	<p>E1: Tại bước 2.</p> <p>2.1 Back officer chọn Quay lại.</p> <p>2.2 Hệ thống thoát khỏi form Phân nhiệm vụ và trả lại cửa sổ quản lý task của ca.</p> <p>Use-case dừng lại.</p> <p>E2: Tại bước 7.</p> <p>7.1 Hệ thống báo lỗi thêm task thất bại.</p> <p>Use-case dừng lại.</p>
Assumption	Back officer chỉ được giao tối đa 3 task/1 nhân viên/1 ngày.
Constraint	<ul style="list-style-type: none">- Task mới được tạo phải nằm ở hàng đầu tiên trong danh sách task.- Trường để chọn(nhập) xe và nhân sự là các trường bắt buộc.- Hệ thống chỉ cho phép phân tối đa 1 collector cho mỗi task, tối thiểu 2 janitors và tối đa 4 janitors với mỗi MCP.

Xác nhận tiến độ

Use case name	Xác nhận tiến độ
Actors	Back officer
Description	Là một back officer, tôi muốn xác nhận tiến độ cho task đã được cập nhật tiến độ gửi từ janitors/collector và đang trong trạng thái chờ xác nhận.
Trigger	Back officer muốn xác nhận tiến độ cho một nhiệm vụ đang chờ xác nhận.



Preconditions	<ul style="list-style-type: none">Thiết bị back officer đang ở cửa sổ quản lý nhiệm vụ của một ca.Thông báo tiền độ của nhiệm vụ muôn xác nhận tiền độ phải có trong phần danh sách các thông báo đang hiển thị (được Collector/Janitor gửi trước đó).Thiết bị của Back officer có kết nối internet và kết nối với hệ thống.
Postconditions	<ul style="list-style-type: none">Trạng thái của task được cập nhật mới trên hàng tương ứng.Hệ thống gửi thông báo “Tiền độ task đã xác nhận thành công” đến thiết bị của collectors/janitors.
Normal Flow	<ol style="list-style-type: none">Back officer click vào button “Xác nhận” của một thông báo cập nhật tiền độ trong danh sách các thông báo tiền độ ở trang quản lý nhiệm vụ của một ca.Hệ thống hiển thị dialog thông báo xác nhận thành công.Hệ thống cập nhật lại hiển thị trạng thái mới cho task tương ứng và đồng thời xóa thông báo được xác nhận ở danh sách thông báo.
Alternative Flow	Không có.
Exception Flow	<p>E1: Tại bước 2.</p> <p>2.1 Hệ thống thông báo xác nhận thất bại. Use-case dừng lại.</p>
Constraint	Không có.



Cập nhật tiến độ

Use case name	Cập nhật tiến độ
Actors	Collector/Janitor (Primary), Back officer (Secondary)
Description	Là collector/janitor, tôi muốn cập nhật tiến độ của task (ca) và nhận xác nhận cập nhật từ back officer.
Trigger	Collector/Janitor muốn cập nhật tiến độ cho ca làm việc.
Preconditions	<ul style="list-style-type: none">– Collector/Janitor đã đăng nhập hệ thống và đã được cấp quyền xem lịch làm việc và cập nhật tiến độ.– Thiết bị collector/janitor đang ở cửa sổ “Xem lịch làm việc tuần”.– Button “Cập nhật tiến độ” của task đang enable.– Thiết bị của collector/janitor có kết nối internet và kết nối với hệ thống.
Postconditions	Task được cập nhật tiến độ thành công và chuyển sang trạng thái tiếp theo.



Normal Flow	<ol style="list-style-type: none">1. Collector/Janitor chọn một ca trong lịch làm việc để xem chi tiết.2. Hệ thống lấy thông tin chi tiết của ca được chọn và hiển thị ở cửa sổ mới.3. Collector/Janitor xem trạng thái hiện tại của task và click button “Cập nhập tiến độ”.4. Hệ thống hiển thị dialog thông báo confirm lại việc cập nhật.5. Collector/Janitor chọn “OK”.6. Hệ thống thông báo Collector/Janitor chờ xác nhận từ back officer và disable button “Cập nhật tiến độ”.7. Thông báo yêu cầu xác nhận tiến độ cho nhiệm vụ tương ứng được thêm vào danh sách các thông báo có trong giao diện quản lý các nhiệm vụ của ca ở phía Back Officer.8. Use case tiếp tục use case Xác nhận tiến độ.
Alternative Flow	Không có.



Exception Flow	<p>E1: Tại bước 2.</p> <p>2.1 Hệ thống lấy thông tin chi tiết của ca thất bại và thông báo lỗi.</p> <p>Use-case dừng lại.</p> <p>E2: Tại bước 5.</p> <p>5.1 Collector/Janitor chọn “Hủy thao tác”</p> <p>5.2 Hệ thống thoát khỏi dialog và trở lại cửa sổ của ca.</p> <p>Use-case dừng lại.</p> <p>E3: Tại bước 6.</p> <p>6.1 Hệ thống thông báo cập nhật thất bại.</p> <p>Use-case dừng lại.</p>
Constraint	<ul style="list-style-type: none">– Hệ thống có khả năng nhận cập nhật tiến độ của tất cả nhân viên trong ca tại cùng thời điểm.– Hệ thống hiển thị thông báo cho các task ở phía back officer kể từ khi Collector/Janitor gửi cập nhật tiến độ không quá 10s.

Xem lịch làm việc cá nhân

Use case name	Xem lịch làm việc cá nhân
Actors	Collector/Janitor
Description	Là một Collector/Janitor, tôi muốn theo dõi các ca làm việc của mình trong ngày/tuần.
Trigger	Collector/Janitor chọn mục “Xem lịch” trên menu chính.



Preconditions	<ul style="list-style-type: none">– Collector/Janitor đã đăng nhập hệ thống và có quyền truy cập vào xem lịch cá nhân.– Thiết bị của Collector/Janitor có kết nối internet và kết nối với hệ thống.
Postconditions	Không có.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống lấy dữ liệu danh sách các ca làm việc trong tuần của nhân viên tương ứng.2. Hệ thống hiển thị danh sách các ca làm việc.
Alternative Flow	<p>A1: Tại bước 2.</p> <p>2.1 Collector/Janitor chọn một ngày cụ thể trong tuần.</p> <p>2.2 Hệ thống lọc ra và chỉ hiển thị các ca của ngày được chọn.</p>
Exception Flow	<p>E1: Tại bước 2.</p> <p>2.1 Hệ thống lấy thông tin từ database thất bại và hiển thị thông báo.</p> <p>Use case dừng lại.</p>
Constraint	Không có.

Xem nhiệm vụ

Use case name	Xem nhiệm vụ
Actors	Collector/Janitor
Description	Là một Collector/Janitor, tôi muốn theo thông tin chi tiết của nhiệm vụ được phân cho mình trong ca làm việc.



Trigger	Collector/Janitor click vào một ca làm việc trong khi đang xem lịch làm việc cá nhân.
Preconditions	<ul style="list-style-type: none">- Collector/Janitor đã đăng nhập hệ thống và có quyền truy cập vào xem lịch cá nhân.- Thiết bị của Collector/Janitor có kết nối internet và kết nối với hệ thống.
Postconditions	Không có.
Normal Flow	<ol style="list-style-type: none">1. Hệ thống lấy dữ liệu của ca và nhiệm vụ phân cho nhân viên tương ứng từ database (ngày, giờ, đồng nghiệp, địa điểm làm việc...).2. Hệ thống hiển thị thông tin chi tiết nhiệm vụ ở trang mới.
Alternative Flow	Không có
Exception Flow	<p>E1: Tại bước 2.</p> <p>2.1 Hệ thống lấy thông tin từ database thất bại và hiển thị thông báo.</p> <p>Use case dừng lại.</p>
Constraint	Không có.

Điểm danh

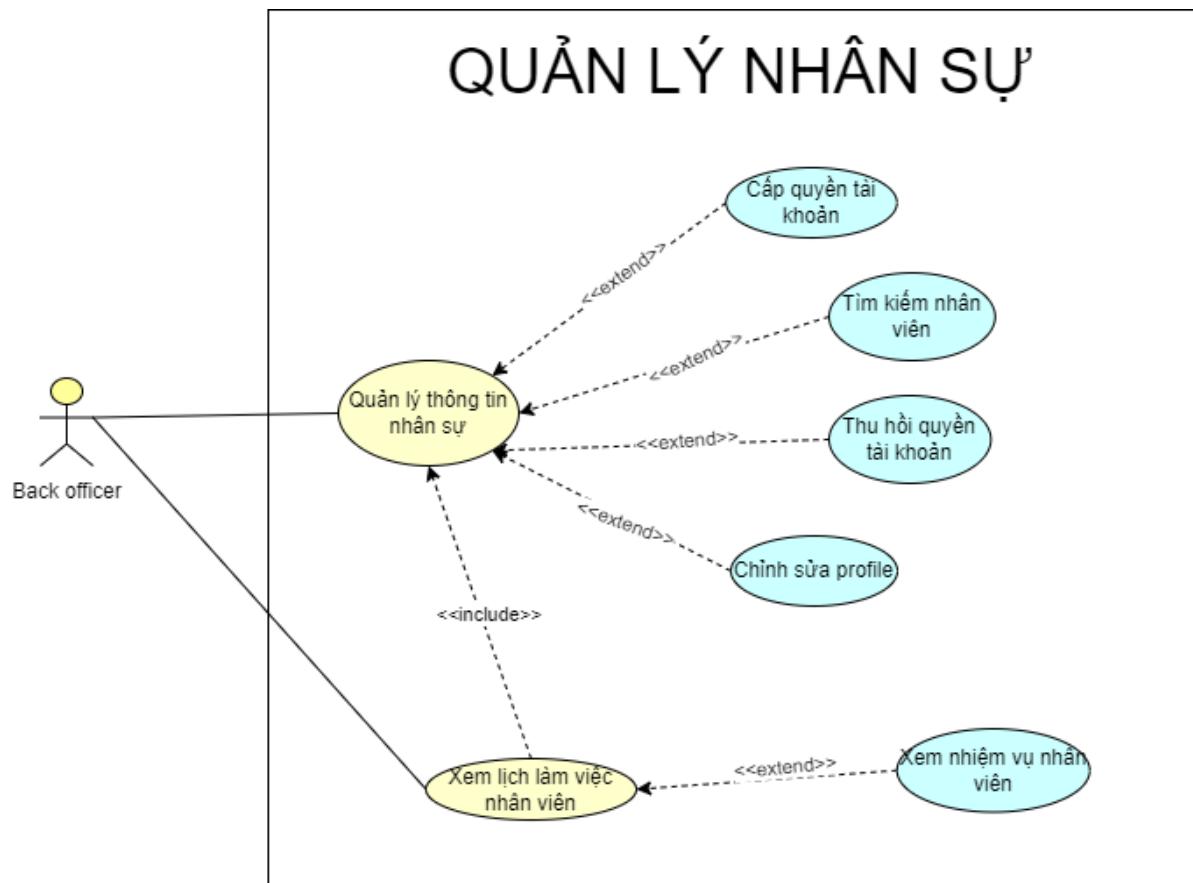
Use case name	Điểm danh
Actors	Collector/Janitor
Description	Là một Collector/Janitor, tôi muốn điểm danh cho một ca làm việc.
Trigger	Hệ thống thông báo yêu cầu điểm danh cho ca làm việc trong 15 phút trước giờ bắt đầu ca.



Preconditions	<ul style="list-style-type: none">- Collector/Janitor đã đăng nhập hệ thống.- Thiết bị của Collector/Janitor có kết nối internet và kết nối với hệ thống.
Postconditions	Collector/Janitor điểm danh thành công và thông tin được cập nhật vào database.
Normal Flow	<ol style="list-style-type: none">1. Collector/Janitor vào mục "Điểm danh" trên ứng dụng.2. Collector/Janitor nhấn nút "Điểm danh" trong vòng 15 phút trước giờ bắt đầu ca.3. Hệ thống tiến hành kiểm tra so khớp wi-fi kết nối ở thiết bị nhân viên và wi-fi công ty.4. Hệ thống thông báo điểm danh thành công cho ca làm việc.
Alternative Flow	Không có
Exception Flow	<p>E1: Tại bước 2.</p> <p>2.1 Collector/Janitor nhấn nút "Điểm danh" sau giờ bắt đầu ca.</p> <p>2.2 Hệ thống thông báo điểm danh trễ.</p> <p>E1: Tại bước 4.</p> <p>4.1 Hệ thống kiểm tra không thành công và thông báo nhân viên điểm danh không hợp lệ.</p>
Constraint	Không có.

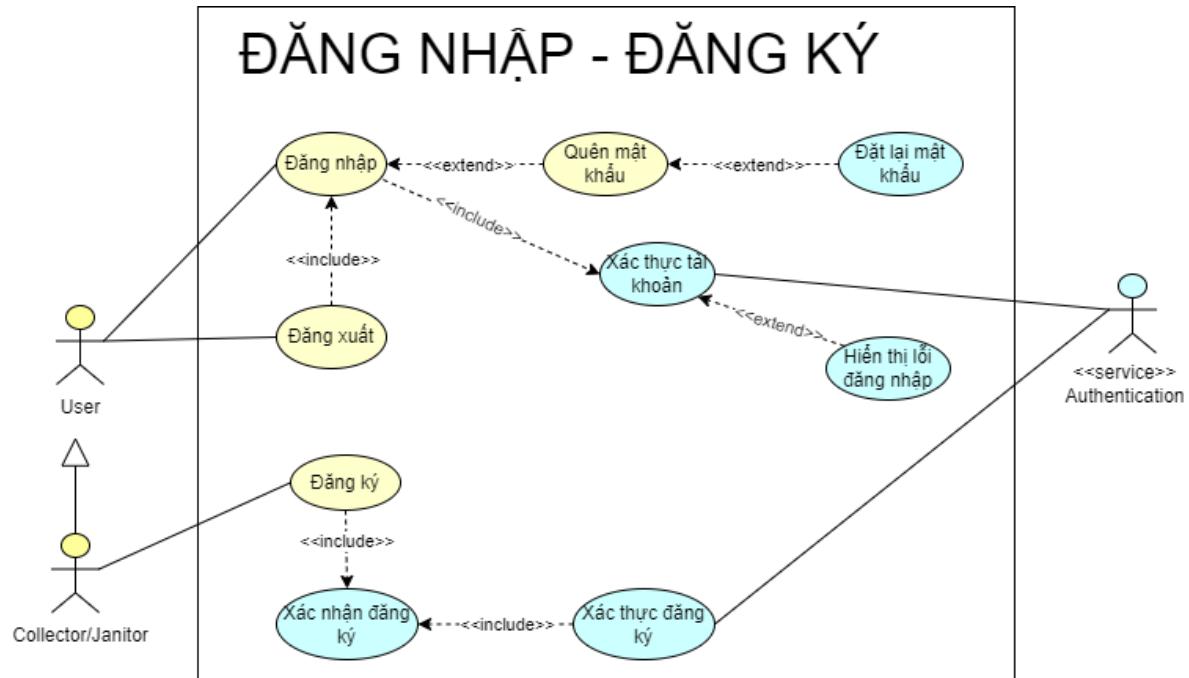
3.3 Use-case chi tiết cho một số module khác trong hệ thống

Quản lý nhân sự



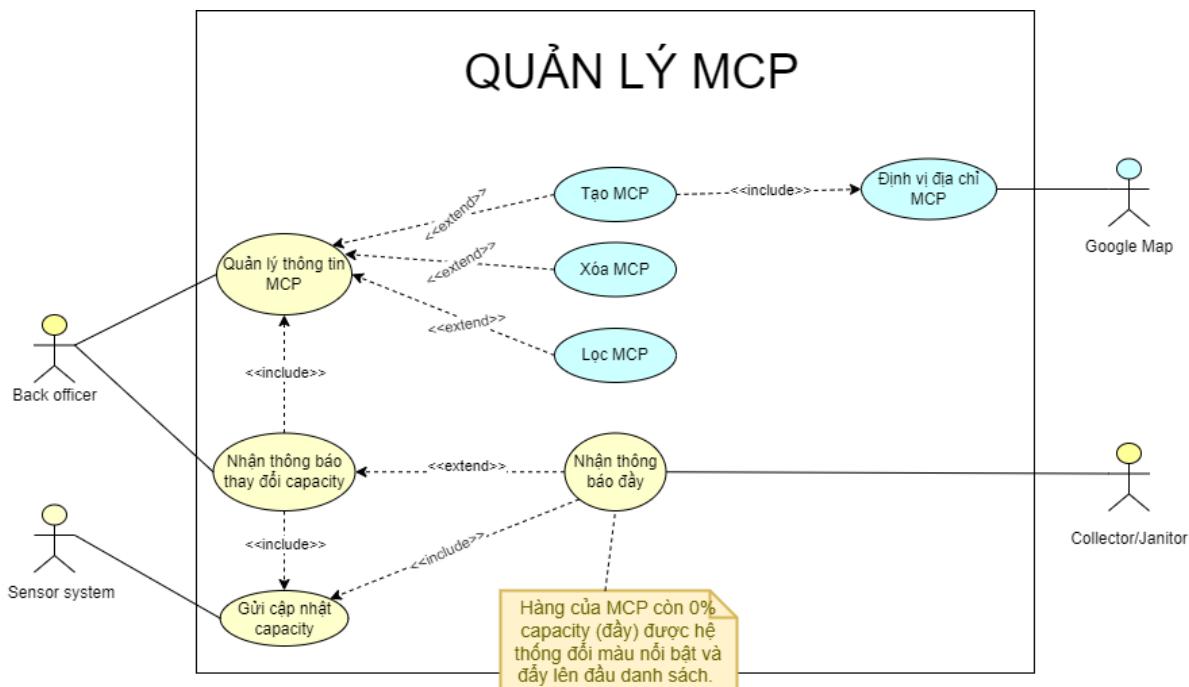
Hình 3: Use case Quản lý nhân sự

Đăng nhập - Đăng ký



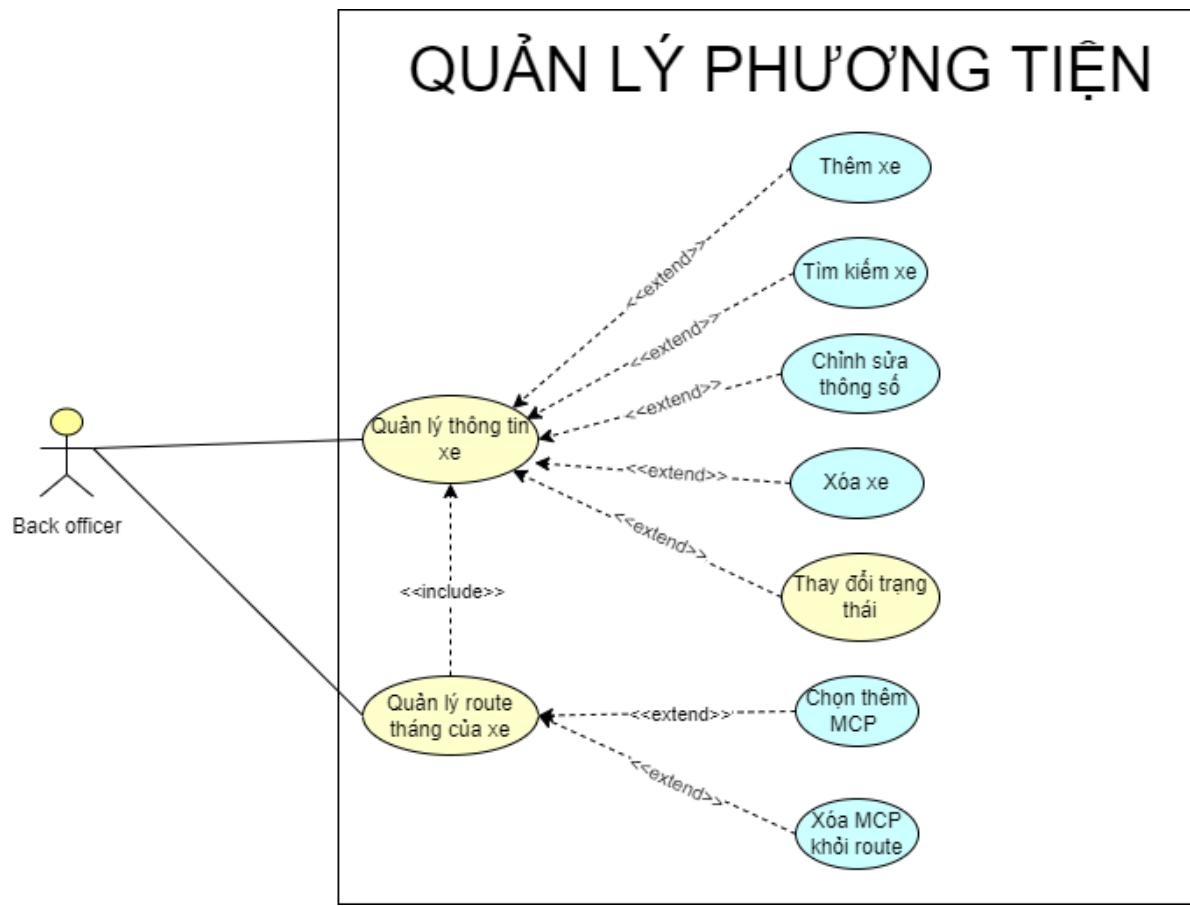
Hình 4: Use case Đăng nhập - Đăng ký

Quản lý Điểm thu gom



Hình 5: Use case Quản lý MCP

Quản lý phương tiện



Hình 6: Use case Quản lý phương tiện

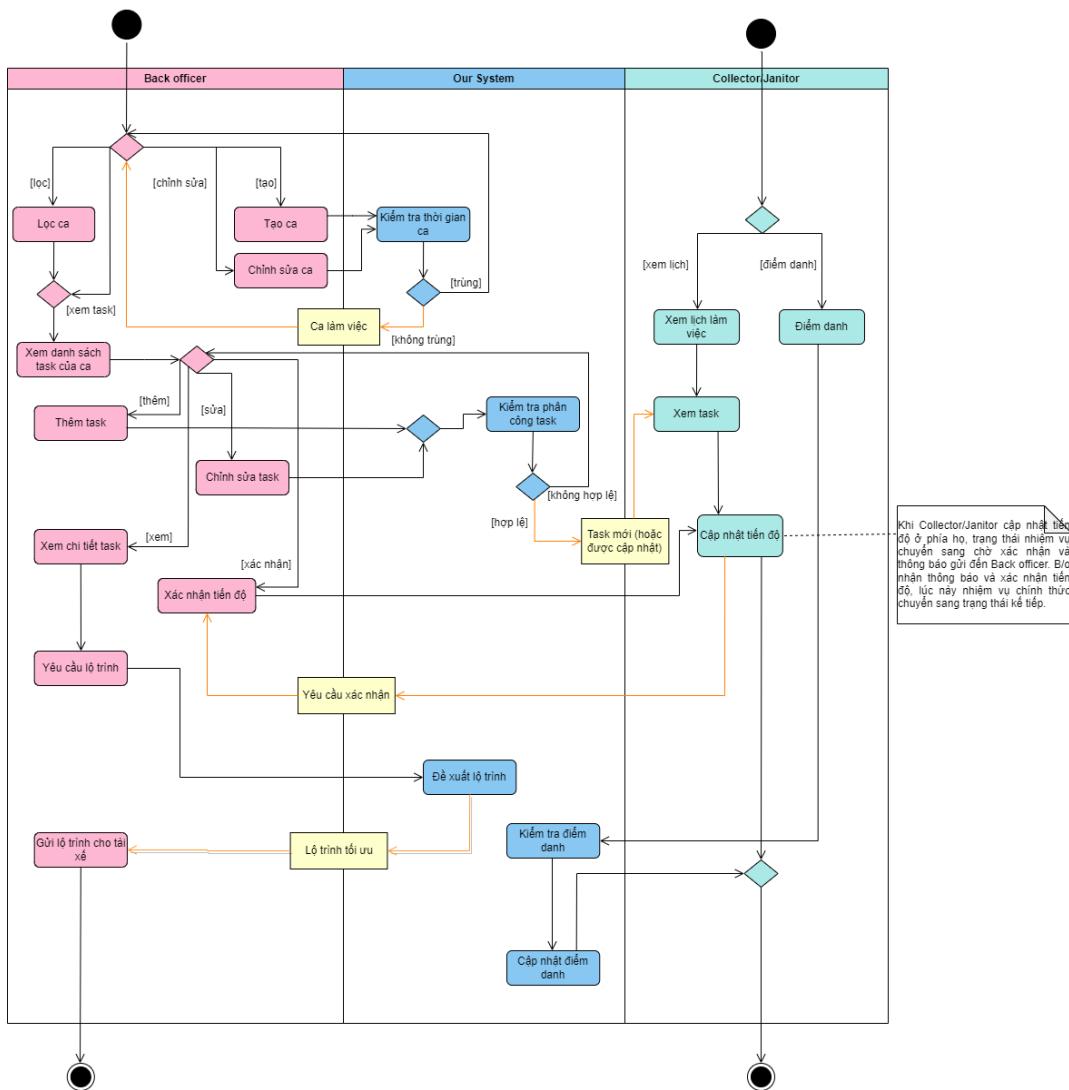


PHẦN II

SYSTEM MODELING

1 Activity Diagram

1.1 Task Assignment Module



Hình 7: Task Assignment Activity Diagram



Xem các bản vẽ Activity Diagram của nhóm tại link

<https://drive.google.com/file/d/1t4ZjdPat769yxcGSitHU-T6pNWY1rYby/view>

Mô tả:

Back officer đăng nhập và truy cập giao diện quản lý ca, danh sách các ca hiện có được hiển thị trên màn hình:

- Nếu Back officer chọn tạo ca hoặc chỉnh sửa ca, hệ thống sẽ tiến hành kiểm tra thời gian của ca vừa tạo/được chỉnh sửa có trùng với những ca đã có không, nếu có hệ thống sẽ đưa người dùng quay về giao diện quản lý ca, còn trong trường hợp không trùng, hệ thống sẽ tiến hành tạo ca mới/ cập nhật sửa đổi ca làm việc, sau đó đưa người dùng quay về giao diện quản lý ca.
- Nếu Back officer chọn lọc ca, hệ thống sẽ tiếp nhận thông tin muốn lọc của người dùng, sau đó hiển thị thông tin các ca thỏa bộ lọc lên màn hình.

Khi back officer chọn xem danh sách nhiệm vụ trong một ca cụ thể, hệ thống sẽ hiển thị danh sách các nhiệm vụ của ca lên màn hình.

- Khi Back officer chọn xem chi tiết một nhiệm vụ, hệ thống sẽ hiển thị lên màn hình chứa thông tin chi tiết của nhiệm vụ đó. Ở đây, Back officer có thể yêu cầu hệ thống gợi ý lộ trình tối ưu. Khi đó, hệ thống đề xuất lộ trình tối ưu và hiển thị lên màn hình và Back officer có thể gửi đến tài xế.
- Khi Back officer chọn chỉnh sửa một nhiệm vụ hoặc chọn thêm nhiệm vụ mới, hệ thống sẽ kiểm tra xem nhiệm vụ vừa thêm/chỉnh sửa có hợp lệ không, nếu không hệ thống sẽ đưa người dùng quay về lại màn hình xem danh sách nhiệm vụ của ca; còn nếu có, hệ thống sẽ tiến hành thêm nhiệm vụ mới/cập nhật sửa đổi cho nhiệm vụ và đưa Back officer quay về giao diện xem danh sách nhiệm vụ ban đầu. Đồng thời, phía Collector/Janitor cũng có thể nhận thông tin về nhiệm vụ này.
- Ở đây, Back Officer cũng có thể kiểm tra các thông báo cập nhật nhiệm vụ trong ca gửi đến từ Collector/Janitor và tiến hành xác nhận tiến độ cho nhiệm vụ đó.

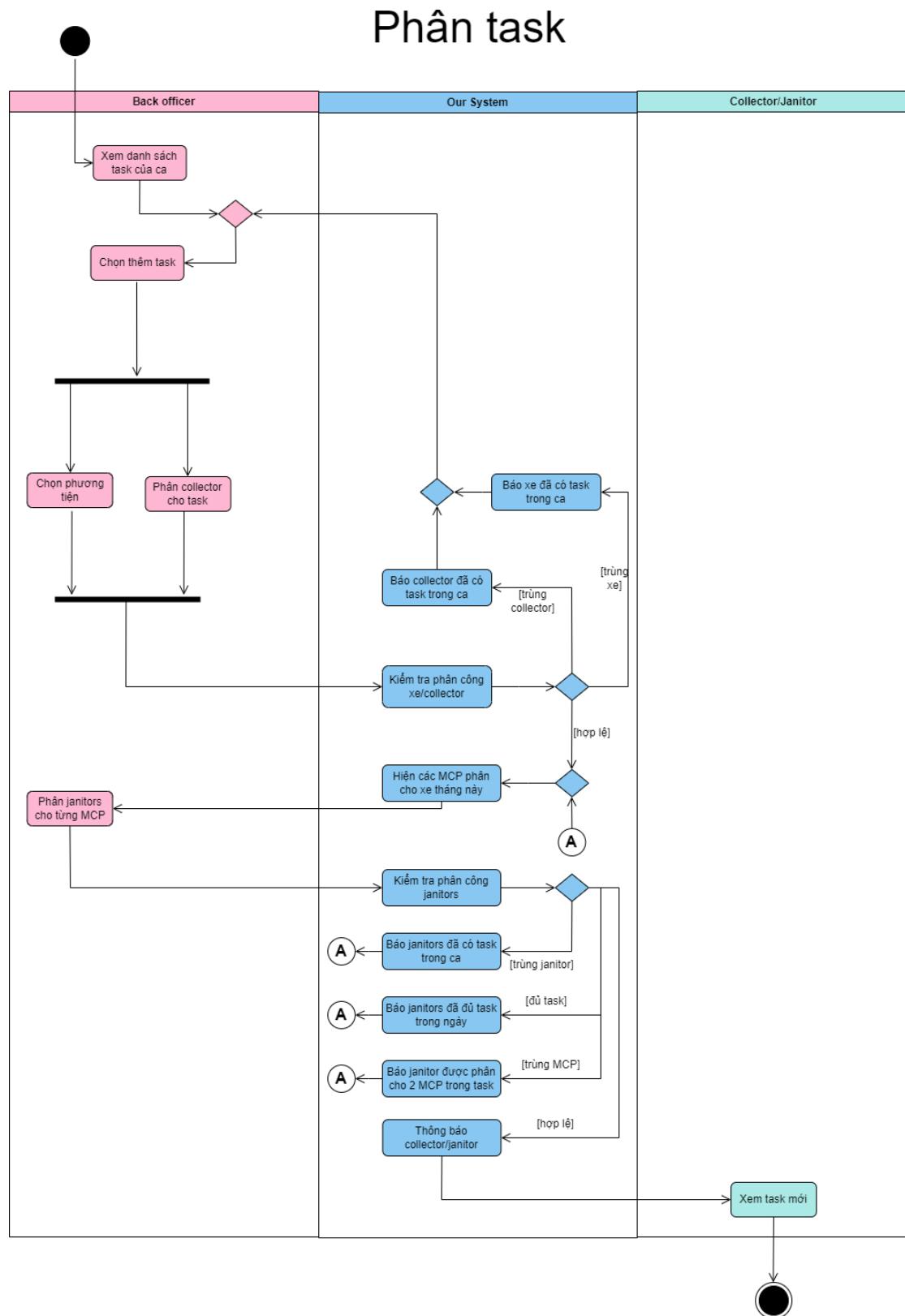
Collector/Janitor đăng nhập và có thể truy cập vào giao diện điểm danh hoặc xem lịch làm việc.

- Collector/Janitor truy cập vào giao diện điểm danh để điểm danh trước lúc bắt đầu ca làm việc của họ. Sau khi điểm danh, hệ thống tiến hành kiểm tra rồi cập nhật trạng thái điểm danh.



- Collector/Janitor truy cập vào giao diện xem lịch làm việc để xem danh sách các ca làm việc của mình trong ngày/tuần. Ở đây, Collector/Janitor chọn một ca để xem chi tiết nhiệm vụ của mình trong ca đó. Họ có thể cập nhật tiến độ cho nhiệm vụ. Khi Collector/Janitor cập nhật tiến độ ở phía họ, trạng thái nhiệm vụ chuyển sang chờ xác nhận và thông báo được gửi đến Back officer. Back officer nhận thông báo và xác nhận tiến độ, lúc này nhiệm vụ chính thức chuyển sang trạng thái kế tiếp.

1.2 Tạo task



Hình 8: Phân Task Activity Diagram



Mô tả:

Back officer đầu tiên sẽ xem danh sách task của ca, sau đó chọn thêm task. Ở đây trước tiên cần làm hai việc là chọn phương tiện và phân collector cho task. Sau khi làm hai việc trên sẽ tiến hành kiểm tra việc phân công xe và collector có hợp lệ hay không.

Tại đây sẽ xảy ra hai tình huống:

1. Nếu hợp lệ:

Hệ thống hiển thị các MCP đã phân cho xe tháng này, sau đó back officer có thể phân janitors cho từng MCP và hệ thống sẽ kiểm tra phân công janitors. Tại đây sẽ xảy ra các tình huống sau:

- Nếu janitor đã được phân cho một task khác trong cùng ca (*trùng janitor*), hệ thống báo janitor đã có task trong ca.
- Nếu janitor đã được phân đủ task trong ngày (*đủ task*), hệ thống báo janitors đã đủ task trong ngày.
- Nếu janitor được phân nhiều hơn một MCP trong task hiện tạo (*trùng MCP*), hệ thống báo janitors được phân cho 2 MCP trong task.
- Nếu hợp lệ, hệ thống tiến hành thông báo task mới đến collector, janitor và sau đó chuyển về giao diện thêm task mới ở giao diện của back officer.

2. Nếu không hợp lệ:

- Không hợp lệ do trùng xe, lập tức báo xe đã có task trong ca.
- Không hợp lệ do trùng collector, báo collector đã có task trong ca

2 Sequence Diagram

2.1 Conceptual Solution for the route planning task

Quy trình nghiệp vụ và một số giả định liên quan đến nhiệm vụ lập lộ trình di chuyển

1. Hàng tháng, Back officer sẽ chỉ chọn các điểm thu gom (MCPs) cụ thể và gán cho mỗi phương tiện được đưa vào hoạt động.
2. Hàng tuần, Back officer sẽ phân công nhiệm vụ và lịch trình cho các đội bao gồm Collector và Janitor.
3. Hàng ngày, các cán bộ phụ trách sẽ tạo lộ trình di chuyển và thông báo đến Collector. Hệ thống sẽ hỗ trợ các Back officer lập ra lộ trình cụ thể cho mỗi ca làm việc của Collector dựa trên tập hợp các MCPs đã được quyết định mỗi tháng cho xe tương ứng.
4. Việc tạo lộ trình sẽ được thực hiện dựa trên cơ sở tối ưu hóa tiêu thụ nhiên liệu và rút ngắn đường đi. Vì thế, hệ thống sẽ lọc ra các MCP có sức chứa còn lại ít hơn hoặc bằng 15% sức chứa tối đa của MCP, kết hợp với địa điểm bắt đầu (công ty) và địa điểm kết thúc lộ trình để tiến hành tạo route tối ưu.
5. Việc lập kế hoạch lộ trình được Back officer thực hiện cho tất cả các Collector làm việc trong ngày. Mỗi ca làm việc sẽ có ít nhất 1 Collector đi thu gom chất thải tại các MCP. Sau khi các Janitor thực hiện thu gom và chuyển chất thải đến các MCP xong, gửi thông báo về tiến độ cho Back officer thì Back officer sẽ dựa vào tình trạng của các MCP hiện tại để tạo route tối ưu và gửi lộ trình di chuyển đến cho các Collector đang làm việc tại ca đó. Để tránh lãng phí thời gian (thời gian chờ các janitor thực hiện công việc) cũng như đúng tiến độ, các Collector sẽ được gửi lộ trình và bắt đầu di chuyển sau 30 phút kể từ thời điểm bắt đầu ca làm việc.
6. Các MCPs sẽ được cập nhật trạng thái sức chứa bằng các hệ thống cảm biến (sensor system) được đặt tại cái MCPs, và sẽ được cập nhật sau mỗi 15 phút, khi MCPs quá tải sẽ có thông báo đến các officer để có hướng xử lý. Tuy nhiên, hệ thống được triển khai và phát triển hướng đến mục đích giảm đến mức tối đa tình trạng này diễn ra.

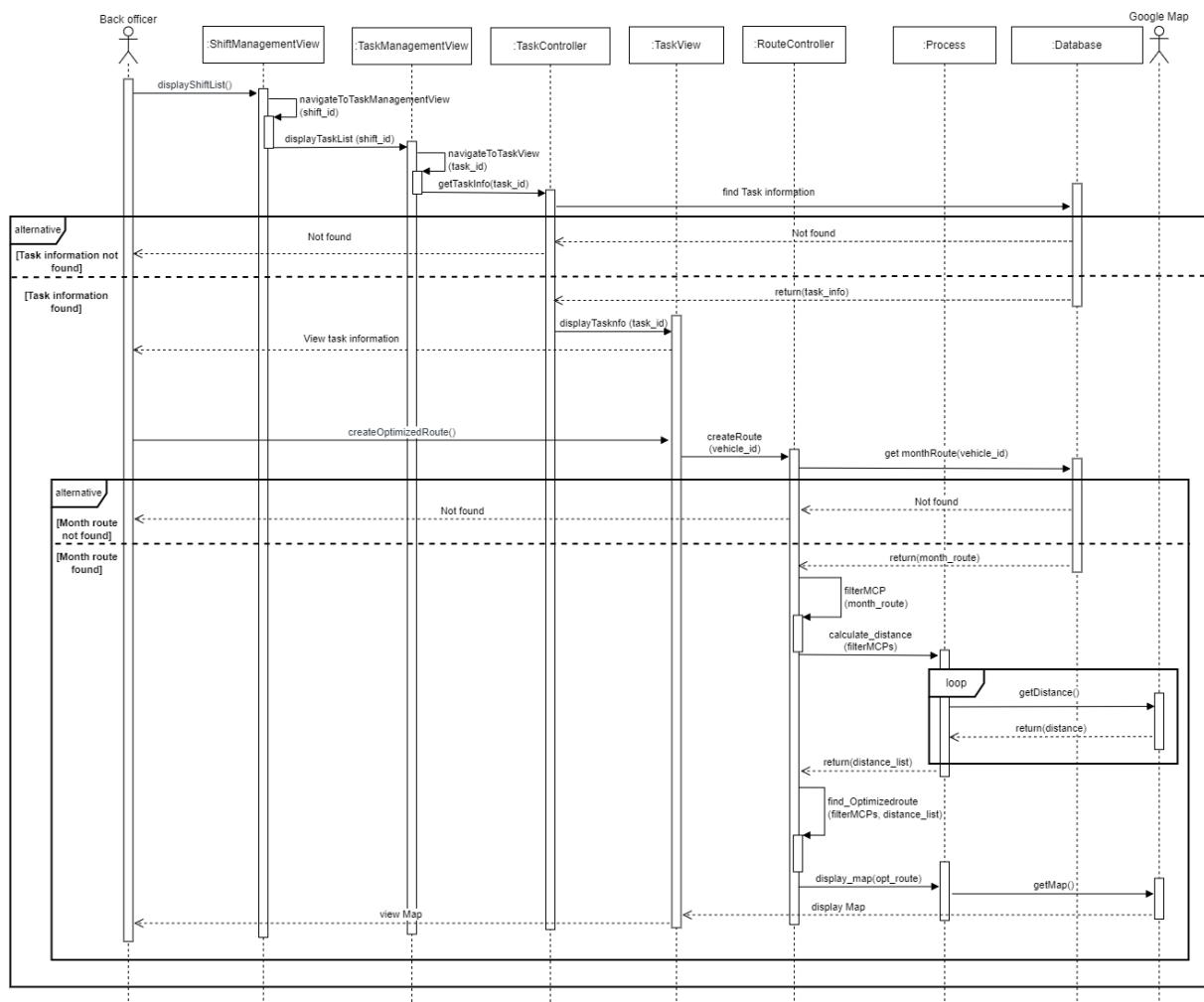
Giải pháp ý niệm được đề xuất cho nhiệm vụ lập lộ trình di chuyển cho Collector bởi Back officer

1. Để tiến hành tạo lập kế hoạch lộ trình (lập route), trước hết Back officer cần vào được trang Shift Management để xem thông tin các ca làm việc, mỗi ca sẽ bao gồm nhiều task và có thể xem chi tiết thông tin của task mong muốn tạo route.



2. Back officer nhấn nút “Tạo route tối ưu” tại trang thông tin chi tiết của task để yêu cầu hệ thống thực hiện tạo route tối ưu cho các MCPs đã được lưu cho task tương ứng.
3. Lúc này, dựa trên danh sách các MCPs có được từ thông tin task, hệ thống tiến hành lựa chọn các MCPs trong số các MCPs đó để tạo route tối ưu (phải chọn ít nhất 80% MCPs trong số các MCPs đã được phân công cho task) theo nguyên tắc như sau
 - Ưu tiên lựa chọn tất cả các MCPs có sức chứa còn lại nhiều nhất là 15% (tức sức chứa MCP đã được sử dụng lớn hơn hoặc bằng 85%) để tạo route, điều này được thực hiện để tránh tình trạng quá tải tại các MCPs.
 - Nếu số lượng MCPs được chọn chưa đủ 80% số MCPs của ca làm việc đó, hệ thống sẽ tiến hành chọn thêm các MCPs theo sự ưu tiên về sức chứa còn lại, sức chứa còn lại càng ít sẽ được chọn để tạo route.
 - Sẽ có những ngày số MCPs được chọn nhiều hơn 80% số MCPs của ca đó, do số lượng rác thải sinh hoạt tăng đột biến, có thể vào những dịp lễ hay sự kiện đặc biệt, khi nhu cầu sử dụng hàng hóa của người dân lớn, đồng nghĩa lượng rác thải sinh hoạt thải ra môi trường nhiều, khiến các MCPs nhanh đầy hơn bình thường. Do đó, số lượng MCPs được chọn thoả mãn không vượt quá sức chứa tối đa của Collecting Vehicle của Collector.
4. Sau khi thực hiện lựa chọn các MCPs, kết hợp với điểm đầu (cố định tại công ty) và điểm cuối (cố định tại nơi xử lý rác), hệ thống sẽ tiến hành tạo đường đi ngắn nhất
 - Hệ thống được tích hợp Google Map. Với sự hỗ trợ từ công cụ định vị này, hệ thống dễ dàng tính được khoảng cách giữa các MCPs với nhau và với điểm đầu, điểm cuối. Khoảng cách này mang tính tương đối.
 - Sau khi có được khoảng cách giữa các điểm MCPs cùng với điểm đầu và điểm cuối, hệ thống thực hiện tìm và tạo đường đi ngắn nhất. Từ đó, kết quả cuối cùng sẽ là lộ trình (route) tối ưu là đường đi ngắn nhất từ công ty, thông qua các MCPs đến điểm cuối cùng là nơi xử lý rác.
5. Cuối cùng, khi đã xác định được lộ trình, Back officer có thể quan sát được route tối ưu hiển thị trên Google Map.

2.2 Sequence diagram



Hình 9: Sequence Diagram

Xem chi tiết ảnh: [Here!](#)

Mô tả các bước để thực hiện nhiệm vụ tạo route tối ưu từ Sequence Diagram

1. **Back officer** đang ở trang **ShiftManagementView**, tại đó muốn xem danh sách quản lý các Task: gọi hàm `navigateToTaskManagementView(shift_id: Integer): void` của nó để điều hướng đến trang quản lý nhiệm vụ của một ca cụ thể **TaskManagementView**, trong đó truyền vào `shift_id` là ID của ca làm việc bao gồm task mà Back officer mong muốn tạo route.
2. Ở trang **TaskManagementView**, Back officer nhấn vào task mong muốn để xem thông tin chi tiết: gọi hàm `navigateToTaskView(task_id: Integer): void` của nó để điều hướng đến trang thông tin một nhiệm vụ cụ thể **TaskView**, trong đó truyền vào `task_id` tương ứng với task được click vào xem thông tin.



3. Sau đó, **TaskController** yêu cầu lấy thông tin của task từ database: thực hiện hàm `getTask(task_id)`. Tại đây sẽ có 2 trường hợp xảy ra:

- Nếu không tìm thấy thông tin ca tại database, hệ thống hiển thị thông báo lỗi trên màn hình và kết thúc tạo route do task mong muốn tạo route tìm thông tin không thành công.
- Ngược lại, kết quả `task_info` sẽ được trả về, **TaskView** thực hiện hàm `displayTaskInfo(task_info)` để hiển thị thông tin chi tiết của task cho Back officer có thể quan sát và tiếp tục thực hiện các thao tác tiếp theo.

`task_info` sẽ bao gồm các thông tin liên quan của task: ID, mô tả, ca của task, phương tiện, các collector và janitor được phân công, thông tin của collecting vehicle, trạng thái của task.

4. Khi đã xem được thông tin chi tiết của một task nào đó, **Back officer** nhấn nút tạo route để yêu cầu tạo route tối ưu: gọi hàm `createOptimizedRoute(): void`.

5. **RouteController** thực hiện hàm `createRoute()` và truyền vào `vehicle_id` là ID của phương tiện đã được gán cho task và thông tin vehicle được lấy trong thông tin task có được trước đó.

6. **RouteController** yêu cầu lấy thông tin và các thông tin liên quan của monthRoute từ database thông qua `vehicle_id`, `monthRoute` là một danh sách các MCPs được gán cho mỗi vehicles hàng tháng, mỗi MCP có thông tin về ID, sức chứa, vị trí. Tại đây cũng sẽ có 2 trường hợp:

- Nếu không tìm thấy thông tin `monthRoute` tại database có thể do Back officer quên chưa phân công MCPs cho vehicle hoặc do một số lỗi không mong muốn, hệ thống hiển thị thông báo lỗi trên màn hình và kết thúc tạo route do task mong muốn tạo route lấy thông tin về MCPs không thành công.
- Ngược lại, kết quả `monthRoute` sẽ được trả về, **RouteController** tiếp tục các bước sau.

7. Lúc này **Route Controller** tiến hành lọc ra các MCPs để tiến hành tạo route: gọi hàm `filterMCP(monthRoute)` thực hiện những công việc sau:

- Như giải pháp đề xuất đã được trình bày, mỗi task có ít nhất 80% các MCPs của task được chọn để tạo route. Đầu tiên, tiến hành lọc ra những MCPs có `capacity <= 15%`, tức những MCPs gần đầy, nếu số MCPs sau khi lọc $>= 80\%$ số MCPs của task, hàm kết thúc và trả về danh sách các MCPs đã được lọc.



- Nếu sau khi lọc các MCPs còn chưa đủ yêu cầu, tiến hành sort dựa theo capacity của các MCPs còn lại và lấy các MCPs có capacity từ thấp đến cao cho đến khi đủ 80% số MCPs, sau đó mới kết thúc và trả về danh sách các MCPs được lọc.
8. Sau khi có được danh sách các MCPs cần có để tạo route, **Route Controller** gọi hàm calculate_distance() để tạo route và truyền vào filterMCPs là danh sách MCPs đã lọc được trước đó để tính khoảng cách giữa các MCPs với nhau:
- Khai báo một mảng hai chiều dùng để lưu khoảng cách giữa các MCPs với nhau (index của mảng hai chiều tương ứng với index của MCPs trong danh sách filterMCPs).
 - Như đã đề cập trước đó, hệ thống sẽ được tích hợp **Google Map** để hỗ trợ việc tạo lộ trình được tối ưu và quan sát kết quả được trực quan hơn. Vì thế, hệ thống sẽ thực hiện vòng lặp để tính khoảng cách giữa các MCPs và gán giá trị vào các vị trí của mảng hai chiều, trong mỗi lần lặp, **Process** sẽ gọi getDistance() để lấy khoảng cách giữa từng cặp MCPs từ **Google Map**.
 - Như vậy, thực thi xong calculate_distance() sẽ trả về distance_list một mảng 2 chiều lưu trữ khoảng cách giữa các cặp MCPs trong danh sách các filterMCPs.
9. Tiếp theo, **Route Controller** gọi hàm find_Optimizedroute(filterMCPs, distance_list) để thực hiện tìm đường đi tối ưu (route) đi qua tất cả các MCPs.
10. Cuối cùng, **Route Controller** gọi hàm display_map() để hiển thị route tối ưu vừa tìm được trực quan trên map, **Process** thực hiện getMap() để lấy map từ hệ thống **Google Map** và hiển thị trên màn hình **Screen** của Back officer.

2.3 Solution evaluation

Ưu điểm

- Trọng tâm giải quyết các MCP có sức chứa còn lại thấp nên tránh tình trạng MCP bị đầy, đồng thời giảm thiểu ô nhiễm khu vực do có thể chất thải đã tồn tại từ lâu.
- Số lượng MCP được chọn là gần 80% để các MCP không bị đầy đồng thời. Từ đó, giảm thiểu sự quá tải trong việc thu gom rác.
- Lựa chọn các MCP trong lộ trình di chuyển thay vì đi qua hết các MCP đã được chỉ định nhằm tiết kiệm nhiên liệu phương tiện, rút ngắn thời gian di chuyển nhưng vẫn đảm bảo sự ổn định và hiệu quả.



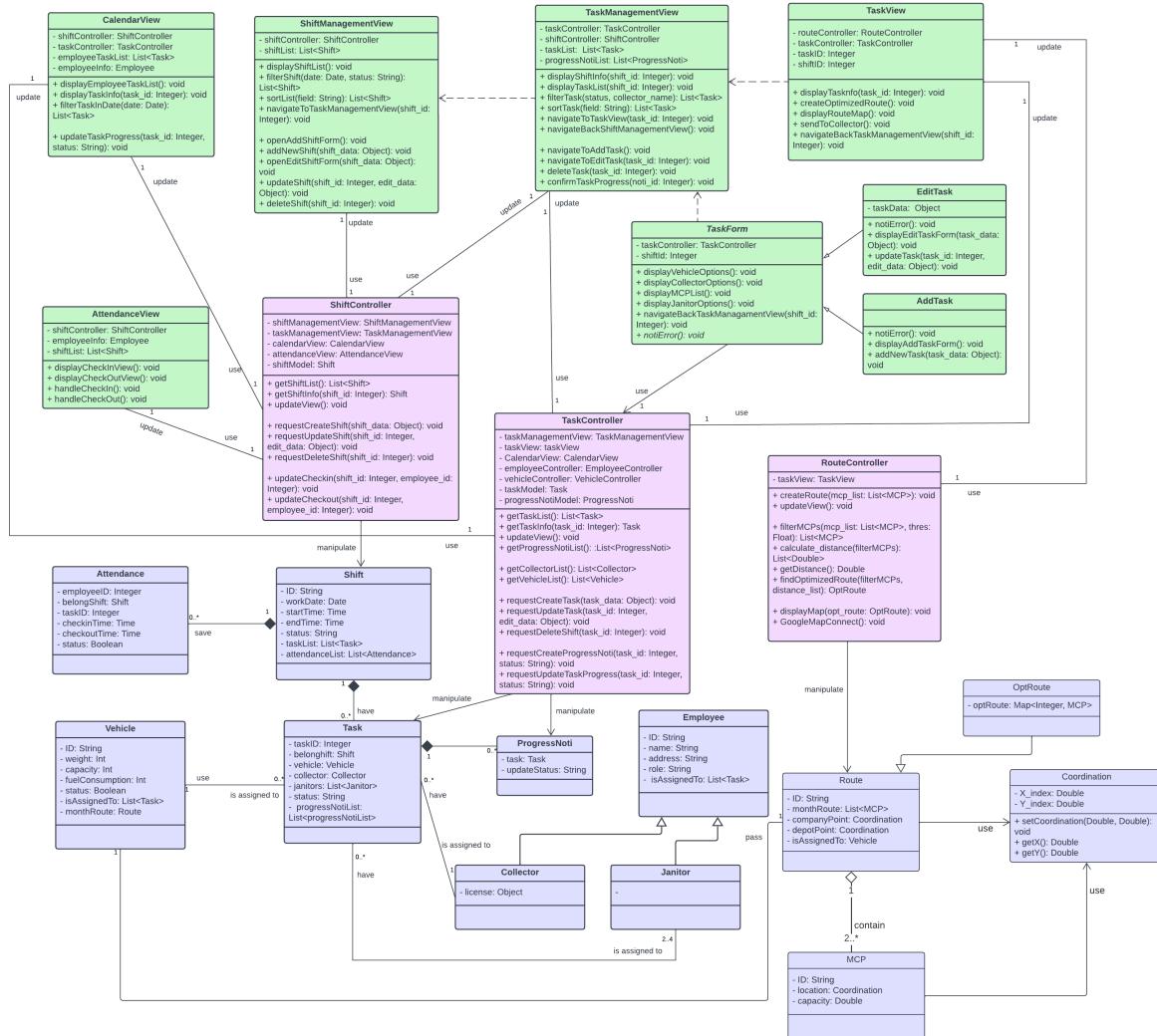
- Hệ thống được tích hợp Google Map giúp tạo lộ trình một cách trực quan hỗ trợ cho các Collector có thể dễ dàng hình dung được lộ trình di chuyển, tránh tình trạng lạc đường, tiếp cận sai các MCP. Bên cạnh đó Google Map cũng là công cụ hữu ích trong việc tìm đường đi ngắn nhất giữa 2 vị trí bất kỳ, giúp rút ngắn khoảng cách di chuyển.

Nhược điểm

- Tập các MCP được chỉ định trong tháng có thể không đầy do lượng chất thải tại các khu vực đó không cao. Từ đó, việc thu gom tại các điểm này là không hiệu quả do tồn tại các MCP khác có lượng chất thải cao hơn nhưng chưa được thu gom.
- Các MCP được lựa chọn theo sức chứa còn lại có thể không là phương án tối ưu nhất (chưa xem xét vị trí tương đối của các MCP,...).
- Xây dựng giải thuật nhằm tìm kiếm lộ trình tối ưu qua các điểm MCP là vô cùng khó khăn. Bên cạnh đó, việc tạo lộ trình còn phụ thuộc vào nhiều yếu tố bên ngoài (đường xá, giao thông,...).
- Dữ liệu được gửi từ hệ thống cảm biến sau mỗi 15 phút nên có thể gây nên sự sai lệch thông tin sức chứa còn lại đã được ghi nhận so với thực tế. Ngoài ra, tồn tại các vấn đề về hư hỏng.

3 Class Diagram

Xem các bản vẽ Class Diagram của nhóm tại link <https://drive.google.com/file/d/1ZXFVFTdyb8Et4qCDupt5aUlZXzXn-i6U/view?usp=sharing>



Hình 10: Task Assignment Class Diagram

Class Diagram của module Task Assignment được nhóm thiết kế theo mô hình kiến trúc **MVC** (Model-Controller-View).

Mỗi tầng đảm nhiệm mỗi nhóm chức năng trong hệ thống:

- **Tầng View:** lắng nghe sự kiện, thu nhận dữ liệu và tương tác người dùng để chuyển đến Controller xử lý; hiển thị dữ liệu do Controller chuyển giao từ Model lên, hiển thị các giao diện, biểu mẫu xác nhận, thông báo, nhập liệu.
- **Tầng Controller:** xử lý sự kiện, điều khiển luồng dữ liệu, cập nhật giao diện.



- **Tầng Model:** lưu trữ dữ liệu hệ thống và cung cấp phương thức cơ bản xử lý logic nghiệp vụ.

Bảng mô tả các Class và quan hệ giữa chúng:

Class	Mô tả	Mối quan hệ
ShiftManagementView	<p>Chức năng chính: hiển thị giao diện quản lý danh sách ca làm việc; nhận các thao tác thêm, xóa, sửa ca; hỗ trợ lọc ca, sắp xếp ca.</p> <p>Các thuộc tính:</p> <p>shiftController: ShiftController</p> <p>shiftList: List<Shift> (danh sách ca làm việc được lấy bởi ShiftController)</p> <p>Các phương thức:</p> <p>displayShiftList(): void (hiển thị danh sách ca làm việc)</p> <p>filterShift(date: Date, status: String): List<Shift> (lọc danh sách theo trường ngày hoặc trường trạng thái ca)</p> <p>sortList(field: String): List<Shift> (sắp xếp danh sách theo trường dữ liệu)</p> <p>navigateToTaskManagementView (shift_id: Integer): void (chuyển đến trang quản lý danh sách nhiệm vụ của ca cụ thể)</p> <p>openAddShiftForm(): void (hiển thị mẫu thêm ca làm việc)</p> <p>addNewShift(shift_data: Object): void (nhận dữ liệu ca mới và truyền cho ShiftController xử lý)</p> <p>openEditShiftForm(shift_data: Object): void (hiển thị mẫu chỉnh sửa ca làm việc)</p> <p>updateShift(shift_id: Integer, edit_data: Object): void (nhận dữ liệu chỉnh sửa của ca và truyền cho ShiftController xử lý)</p> <p>deleteShift(shift_id: Integer): void (nhận sự kiện xóa một ca và gọi ShiftController xử lý)</p>	<p>Với ShiftController: (association) gọi các chức năng điều khiển luồng dữ liệu ca làm việc (lấy, thêm, xóa, sửa) từ ShiftController.</p>



TaskManagementView	<p>Chức năng chính: hiển thị giao diện quản lý danh sách nhiệm vụ của một ca cụ thể, quản lý danh sách các thông báo cập nhật tiến độ cho nhiệm vụ của ca từ nhân viên; nhận các thao tác thêm, xóa, sửa nhiệm vụ; hỗ trợ lọc nhiệm vụ, sắp xếp nhiệm vụ.</p> <p>Các thuộc tính:</p> <ul style="list-style-type: none">taskController: TaskControllershiftController: ShiftControllertaskList: List<Task> (danh sách nhiệm vụ trong ca được lấy bởi TaskController)progressNotiList: List<ProgressNoti> (danh sách các thông báo cập nhật tiến độ cho nhiệm vụ của ca từ nhân viên) <p>Các phương thức:</p> <ul style="list-style-type: none">displayShiftInfo(shift_id: Integer): void (hiển thị thông tin tổng quan của ca như thời gian, tổng số nhiệm vụ, trạng thái ca)displayTaskList(shift_id: Integer): void (hiển thị danh sách nhiệm vụ trong ca)filterTask(status, collector_name): List<Task> (Lọc nhiệm vụ theo trường trạng thái hoặc tên tài xế)sortTask(field: String): List<Task> (sắp xếp danh sách theo trường dữ liệu)navigateToTaskView(task_id: Integer): void (chuyển tiếp đến trang thông tin một nhiệm vụ cụ thể)navigateBackShiftManagementView(): void (trở lại trang quản lý danh sách ca)navigateToAddTask(): void (chuyển đến trang thêm nhiệm vụ)navigateToEditTask(task_id: Integer): void (chuyển đến trang chỉnh sửa nhiệm vụ)	<p>Với TaskController: (association) gọi chức năng điều khiển lấy, xóa nhiệm vụ từ TaskController.</p> <p>Với ShiftManagementView: (dependency) được ShiftManagementView cung cấp ID ca để hiển thị danh sách nhiệm vụ của ca đó.</p>
--------------------	---	---



	<p><code>deleteTask(task_id: Integer): void</code> (nhận sự kiện xóa một nhiệm vụ và gọi TaskController xử lý)</p> <p><code>confirmTaskProgress(noti_id: Integer): void</code> (nhận sự kiện xác nhận tiến độ cho thông báo cập nhật nhiệm vụ và gọi TaskController xử lý)</p>	<p>Với ShiftController: (association): gọi chức năng điều khiển lấy thông tin ca cụ thể từ ShiftController.</p>
TaskForm	<p>Chức năng chính: là Abstract Class, cung cấp các chức năng hiển thị các lựa chọn về phương tiện và nhân viên phân công cho nhiệm vụ; có abstract function cho chức năng thông báo lỗi.</p> <p>Các thuộc tính:</p> <p>taskController: TaskController</p> <p>shiftId: Integer</p> <p>Các phương thức:</p> <p><code>displayVehicleOptions(): void</code>: hiển thị các lựa chọn về phương tiện.</p> <p><code>displayCollectorOptions(): void</code>: hiển thị các lựa chọn về Collector.</p> <p><code>displayMCPLList(): void</code>: hiển thị danh sách các MCP trong lộ trình được phân hằng tháng của phương tiện được chọn.</p> <p><code>displayJanitorOptions(): void</code>: hiển thị các lựa chọn về Janitor.</p> <p><code>navigateBackTaskManagementView(shift_id: Integer): void</code>: trở lại trang quản lý danh sách nhiệm vụ của ca.</p> <p><code>notiError(): void</code>: abstract function dùng cho thông báo lỗi khi thêm/sửa.</p>	<p>Với TaskManagementView: (dependency) được cung cấp ID ca để điều hướng trở lại trang danh sách nhiệm vụ của ca đó; được cung cấp dữ liệu nhiệm vụ dùng cho chỉnh sửa.</p> <p>Với EditTask: (inheritance) là lớp cha của nó.</p> <p>Với AddTask: (inheritance) là lớp cha của nó.</p>



EditTask	<p>Chức năng chính: nhận các thao tác chỉnh sửa nhiệm vụ.</p> <p>Các thuộc tính:</p> <p>taskData: Object (dữ liệu của nhiệm vụ cần được sửa)</p> <p>Các phương thức:</p> <p><code>notiError(): void</code> (hiện thực abstract function của lớp cha, hiển thị thông báo lỗi khi sửa)</p> <p><code>displayEditTaskForm(task_data: Object): void</code>: (hiển thị mẫu chỉnh sửa nhiệm vụ)</p> <p><code>updateTask(task_id: Integer, edit_data: Object): void</code> (nhận dữ liệu chỉnh sửa của nhiệm vụ và truyền cho TaskController xử lý)</p>	<p>Với TaskForm: (inheritance) là lớp con của nó.</p>
AddTask	<p>Chức năng chính: nhận các thao tác thêm nhiệm vụ.</p> <p>Các thuộc tính:</p> <p>Các phương thức:</p> <p><code>notiError(): void</code> (hiện thực abstract function của lớp cha, hiển thị thông báo lỗi khi thêm)</p> <p><code>displayAddTaskForm(): void</code> (hiển thị mẫu thêm nhiệm vụ)</p> <p><code>addNewTask(task_data: Object):void</code> (nhận dữ liệu nhiệm vụ mới và truyền cho TaskController xử lý)</p>	<p>Với TaskForm: (inheritance) là lớp con của nó.</p>



TaskView	<p>Chức năng chính: hiển thị thông tin chi tiết của nhiệm vụ, danh sách phân công; nhận sự kiện yêu cầu tạo lộ trình tối ưu cho nhiệm vụ và hỗ trợ hiển thị trực quan bản đồ lộ trình.</p> <p>Các thuộc tính:</p> <ul style="list-style-type: none">routeController: RouteControllertaskController: TaskControllertaskID: IntegershiftID: Integer <p>Các phương thức:</p> <ul style="list-style-type: none"><code>displayTaskInfo(task_id: Integer): void</code>: hiển thị thông tin nhiệm vụ (thời gian, trạng thái, danh sách phân công, lộ trình...)<code>createOptimizedRoute(): void</code> (nhận yêu cầu tạo lộ trình và gọi phương thức tạo lộ trình của RouteController xử lý)<code>displayRouteMap(): void</code> (hiển thị trực quan bản đồ lộ trình tích hợp Google Map)<code>sendToCollector(nhận sự kiện yêu cầu gửi lộ trình cho Collector của nhiệm vụ): void</code><code>navigateBackTaskManagementView(shift_id: Integer): void</code> (điều hướng trở lại trang quản lý nhiệm vụ của ca làm việc chứa nó)	<p>Với TaskManagementView: (dependency) được cung cấp ID ca để cho việc điều hướng trở lại, được cung cấp ID nhiệm vụ dùng cho việc truy xuất thông tin nhiệm vụ.</p> <p>Với TaskController: (association) gọi chức năng điều khiển lấy thông tin nhiệm vụ cụ thể từ TaskController.</p> <p>Với RouteController: (association) gọi chức năng tạo lộ trình và lấy thông tin lộ trình cho hiển thị từ RouteController.</p>
----------	---	---



CalendarView	<p>Chức năng chính: hiển thị lịch làm việc ở giao diện cho Collector/Janitor; hỗ trợ lọc nhiệm vụ theo ngày; nhận sự kiện cập nhật tiến độ nhiệm vụ.</p> <p>Chức thuộc tính:</p> <ul style="list-style-type: none">shiftController: ShiftControllertaskController: TaskControlleremployeeTaskList: List<Task> (danh sách nhiệm vụ của nhân viên đó)employeeInfo: Employee (thông tin nhân viên dùng cho việc truy xuất nhiệm vụ của nhân viên đó) <p>Các phương thức:</p> <ul style="list-style-type: none"><code>displayEmployeeTaskList(): void</code> (hiển thị danh sách nhiệm vụ được phân cho nhân viên)<code>displayTaskInfo(task_id: Integer): void</code> (hiển thị thông tin của một nhiệm vụ cụ thể)<code>filterTaskInDate(date: Date): List<Task></code> (lọc nhiệm vụ theo ngày)<code>updateTaskProgress(task_id: Integer, status: String): void</code> (nhận thao tác cập nhật tiến độ cho nhiệm vụ và gọi TaskController xử lý)	<p>Với ShiftController: (association) gọi chức năng lấy thông tin của ca làm việc gắn với nhiệm vụ trong ShiftController.</p> <p>Với TaskController: (association) gọi chức năng lấy danh sách nhiệm vụ và chức năng tạo thông báo cập nhật tiến độ nhiệm vụ trong TaskController.</p>
--------------	---	--



AttendanceView	<p>Chức năng chính: hiển thị giao diện để điểm danh cho ca làm việc; nhận các sự kiện check in/check out.</p> <p>Các thuộc tính:</p> <ul style="list-style-type: none">shiftController: ShiftControlleremployeeInfo: EmployeeshiftList: List<Shift> (danh sách ca để lấy thời gian dùng cho việc thiết lập thời gian điểm danh) <p>Các phương thức:</p> <ul style="list-style-type: none"><code>displayCheckInView(): void</code> (hiển thị giao diện điểm danh trước ca - check in)<code>displayCheckOutView(): void</code> (hiển thị giao diện điểm danh kết thúc ca - check out)<code>handleCheckIn(): void</code> (nhận sự kiện check in và gọi phương thức từ ShiftController xử lý)<code>handleCheckOut(): void</code> (nhận sự kiện check out và gọi phương thức từ ShiftController xử lý)	<p>Với ShiftController: (association) gọi các phương thức kiểm tra và cập nhật trạng thái check in/check out ca làm việc cho nhân viên cụ thể từ ShiftController.</p>
----------------	---	--



ShiftController	<p>Chức năng chính: xử lý, điều khiển luồng dữ liệu ca làm việc; kiểm tra và cập nhật trạng thái điểm danh ca của nhân viên; cập nhật giao diện.</p> <p>Các thuộc tính:</p> <ul style="list-style-type: none">shiftManagementView: ShiftManagementViewtaskManagementView: TaskManagementViewcalendarView: CalendarViewattendanceView: AttendanceViewshiftModel: Shift <p>Các phương thức:</p> <ul style="list-style-type: none">getShiftList(): List<Shift> (gọi truy vấn của shiftModel để lấy danh sách ca)getShiftInfo(shift_id: Integer): Shift (gọi truy vấn của shiftModel để lấy thông tin ca)updateView(): void (cập nhật các giao diện liên quan)requestCreateShift(shift_data: Object): void (gọi truy vấn của shiftModel để thêm ca mới)requestUpdateShift(shift_id: Integer, edit_data: Object): void (gọi truy vấn của shiftModel để cập nhật thông tin chỉnh sửa ca)requestDeleteShift(shift_id: Integer): void (gọi truy vấn của shiftModel để xóa một ca)updateCheckin(shift_id: Integer, employee_id: Integer): void (kiểm tra và cập nhật trạng thái check in cho nhân viên trong ca)updateCheckout(shift_id: Integer, employee_id: Integer): void (kiểm tra và cập nhật trạng thái check out cho nhân viên trong ca)	<p>Với CalendarView: (association) cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện.</p> <p>Với ShiftManagementView: (association) cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện.</p> <p>Với AttendanceView: (association) cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện.</p> <p>Với TaskManagementView: (association) cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện.</p> <p>Với Shift: (association) gọi các truy vấn thao tác với dữ liệu ca làm việc từ Shift.</p>
-----------------	---	---



TaskController	<p>Chức năng chính: xử lý, điều khiển luồng dữ liệu nhiệm vụ; lấy danh sách thông báo cập nhật tiến độ, xử lý sự kiện yêu cầu cập nhật tiến độ nhiệm vụ; cập nhật giao diện.</p> <p>Các thuộc tính:</p> <p>taskManagementView: TaskManagementView taskView: taskView CalendarView: CalendarView employeeController: EmployeeController vehicleController: VehicleController taskModel: Task progressNotiModel: ProgressNoti</p> <p>Các phương thức:</p> <p><code>getTaskList(): List<Task></code> (gọi truy vấn của taskModel để lấy danh sách nhiệm vụ trong ca) <code>getTaskInfo(task_id: Integer): Task</code> (gọi truy vấn của taskModel để lấy thông tin nhiệm vụ) <code>updateView(): void</code> (cập nhật các giao diện liên quan) <code>getProgressNotiList(): List<ProgressNoti></code> (gọi truy vấn của progressNotiModel để lấy danh sách các thông báo cập nhật tiến độ gửi từ nhân viên trong ca) <code>getCollectorList(): List<Collector></code> (lấy danh sách Collector từ API do employeeController cung cấp) <code>getCollectorList(): List<Collector></code> (lấy danh sách Janitor từ API do employeeController cung cấp) <code>getVehicleList(): List<Vehicle></code> (lấy danh sách Vehicle từ API do vehicleController cung cấp) <code>requestCreateTask(task_data: Object): void</code> (gọi truy vấn của taskModel để thêm nhiệm vụ mới) <code>requestUpdateTask(task_id: Integer, edit_data: Object): void</code> (gọi truy vấn của taskModel để cập nhật thông tin chỉnh sửa)</p>	<p>Với CalendarView: (association) cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện.</p> <p>Với TaskManagementView: (association) cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện.</p> <p>Với TaskView: (association) cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện.</p> <p>Với Task: (association) gọi các truy vấn thao tác với dữ liệu nhiệm vụ từ Task.</p>
----------------	--	--



	<p><code>requestDeleteShift(task_id: Integer): void</code> (gọi truy vấn của taskModel để xóa nhiệm vụ)</p> <p><code>requestCreateProgressNoti(task_id: Integer, status: String): void</code> (gọi truy vấn của progressNotiModel để tạo một thông báo cập nhật tiến độ khi nhân viên cập nhật tiến độ cho nhiệm vụ)</p> <p><code>requestUpdateTaskProgress(task_id: Integer, status: String): void</code> (gọi truy vấn của taskModel để cập nhật trạng thái cho nhiệm vụ, đồng thời gọi truy vấn của progressNotiModel để xóa một thông báo cập nhật sau khi Back officer đã xác nhận tiến độ)</p>	
--	--	--



RouteController	<p>Chức năng chính: hỗ trợ tạo lộ trình tối ưu; điều khiển luồng dữ liệu lộ trình; cập nhật giao diện.</p> <p>Các thuộc tính: taskView: TaskView</p> <p>Các phương thức:</p> <ul style="list-style-type: none"><code>createRoute(mcp_list: List<MCP>):</code> (tạo lộ trình tối ưu từ danh sách MCP)<code>filterMCPs(mcp_list: List<MCP>, thres: Float): List<MCP></code> (lọc ra các MCP thỏa mãn ngưỡng sức chứa)<code>calculate_distance(filterMCPs): List<Double></code> (tính toán và trả về danh sách khoảng cách giữa các MCP sau khi lọc)<code>getDistance(): Double</code> (lấy khoảng cách tương đối giữa 2 MCP)<code>findOptimizedRoute(filterMCPs, distance_list): OptRoute</code> (cung cấp thuật toán để tìm lộ trình tối ưu từ danh sách MCP và khoảng cách giữa chúng)<code>GoogleMapConnect(): void</code> (kết nối với GoogleMap qua API để hiển thị trực quan)	<p>Với TaskView: (association) cung cấp chức năng điều khiển luồng dữ liệu và cập nhật giao diện hiển thị lộ trình.</p> <p>Với Route: (association) gọi các truy vấn thao tác với dữ liệu lộ trình.</p>
-----------------	---	---



Attendance	<p>Chức năng chính: cung cấp cách truy vấn đến dữ liệu Điểm danh của nhân viên trong ca làm việc.</p> <p>Các thuộc tính:</p> <ul style="list-style-type: none">employeeID: IntegerbelongShift: ShifttaskID: IntegercheckinTime: TimecheckoutTime: Timestatus: Boolean <p>Các truy vấn:</p> <ul style="list-style-type: none">insertAttendance(data: object): voidupdateAttendance(data: object): voiddeleteAttendance(ID: interger): voidselectAttendance(): void	<p>Với Shift: (composition) một Shift gồm 0..* Attendance. Một Attendance thuộc 1 Shift.</p>
Shift	<p>Chức năng chính: cung cấp cách truy vấn đến dữ liệu Ca</p> <p>Các thuộc tính:</p> <ul style="list-style-type: none">ID: StringworkDate: DatestartTime: TimeendTime: Timestatus: StringtaskList: List<Task>attendanceList: List<Attendance> <p>Các truy vấn:</p> <ul style="list-style-type: none">insertShift(data: object): voidupdateShift(data: object): voiddeleteShift(ID: interger): voidselectShift(): void	<p>Với Attendance: (composition) một Shift gồm 0..* Attendance. Một Attendance thuộc 1 Shift.</p> <p>Với ShiftController: (association) cung cấp truy vấn lấy, thêm, sửa, xóa dữ liệu ca cho Shift-Controller.</p> <p>Với Task: (composition) một Shift gồm 0..* Task. Một Task thuộc 1 Shift.</p>



Vehicle	<p>Chức năng chính: cung cấp cách truy vấn đến dữ liệu Phương tiện</p> <p>Các thuộc tính:</p> <ul style="list-style-type: none">ID: Stringweight: Intcapacity: IntfuelConsumption: Intstatus: BooleanisAssignedTo: List<Task>monthRoute: Route <p>Các truy vấn:</p> <ul style="list-style-type: none">insertVehicle(data: object): voidupdateVehicle(data: object): voiddeleteVehicle(ID: interger): voidselectVehicle(): void	<p>Với Route: (association) một Vehicle đi qua một Route cố định trong tháng.</p> <p>Với Task: một Vehicle có thể được chỉ định cho nhiều Task. Một Task chỉ sử dụng một Vehicle.</p>
---------	--	---



Task	<p>Chức năng chính: cung cấp cách truy vấn đến dữ liệu Nhiệm vụ</p> <p>Các thuộc tính:</p> <ul style="list-style-type: none">taskID: Integerbelongshift: Shiftvehicle: Vehiclecollector: Collectorjanitors: List<Janitor>status: String <p>Các truy vấn:</p> <ul style="list-style-type: none"><code>insertTask(data: object): void</code><code>updateTask(data: object): void</code><code>deleteTask(ID: interger): void</code><code>selectTask(): void</code>	<p>Với TaskController: (association) cung cấp truy vấn lấy, thêm, sửa, xóa dữ liệu Nhiệm vụ cho TaskController.</p> <p>Với Shift: (composition) một Shift gồm 0..* Task. Một Task thuộc 1 Shift.</p> <p>Với Vehicle: một Vehicle có thể được chỉ định cho nhiều Task. Một Task chỉ sử dụng một Vehicle.</p> <p>Với Collector: một Collector có thể được chỉ định cho nhiều Task. Một Task chỉ gồm một Collector.</p> <p>Với Janitor: một Janitor có thể được chỉ định cho nhiều Task. Một Task chỉ gồm 2 đến 4 Janitor.</p>
------	--	--



Employee	<p>Chức năng chính: cung cấp cách truy vấn đến dữ liệu Nhân viên</p> <p>Các thuộc tính:</p> <p>ID: String name: String address: String role: String isAssignedTo: List<Task></p> <p>Các truy vấn:</p> <p>insertEmployee(data: object): void updateEmployee(data: object): void deleteEmployee(ID: interger): void selectEmployee(): void</p>	<p>Với Collector: là lớp cha</p> <p>Với Janitor: là lớp cha</p>
Collector	<p>Các thuộc tính:</p> <p>license: Object</p>	<p>Với Employee: là lớp con</p> <p>Với Task: một Collector có thể được chỉ định cho nhiều Task. Một Task chỉ gồm một Collector.</p>
Janitor		<p>Với Employee: là lớp con</p> <p>Với Task: một Janitor có thể được chỉ định cho nhiều Task. Một Task chỉ gồm 2 đến 4 Janitor.</p>



Route	<p>Chức năng chính: cung cấp cách truy vấn đến dữ liệu Lộ trình</p> <p>Các thuộc tính:</p> <ul style="list-style-type: none">ID: StringmonthRoute: List<MCP>companyPoint: CoordinationdepotPoint: CoordinationisAssignedTo: Vehicle <p>Các truy vấn:</p> <ul style="list-style-type: none">insertRoute(data: object): voidupdateRoute(data: object): voiddeleteRoute(ID: interger): voidselectRoute(): void	<p>Với RouteController: (association) cung cấp truy vấn lấy, thêm, sửa, xóa dữ liệu Lộ trình cho RouteController.</p> <p>Với Vehicle: (association) một Vehicle đi qua một Route cố định trong tháng.</p> <p>Với OptRoute: là lớp cha</p> <p>Với Coordination: (association) sử dụng như thuộc tính</p> <p>Với MCP: (aggregation) 1 Route đi qua từ 2 MCP. Một MCP thuộc một Route.</p>
OptRoute	<p>Chức năng chính: cung cấp cách truy vấn đến dữ liệu Lộ trình tối ưu</p> <p>Các thuộc tính:</p> <ul style="list-style-type: none">optRoute: Map<Integer, MCP>	<p>Với Route: là lớp con</p>



Coordination	<p>Chức năng chính: cung cấp cách truy vấn đến dữ liệu Tọa độ</p> <p>Các thuộc tính:</p> <p>X_index: Double Y_index: Double</p> <p>Các truy vấn:</p> <p>insertCoordination(data: object): void updateCoordination(data: object): void deleteCoordination(ID: interger): void selectCoordination(): void</p>	<p>Với Route, MCP: được sử dụng.</p>
MCP	<p>Chức năng chính: cung cấp cách truy vấn đến dữ liệu MCP</p> <p>Các thuộc tính:</p> <p>ID: String location: Coordination capacity: Double</p> <p>Các truy vấn:</p> <p>insertMCP(data: object): void updateMCP(data: object): void deleteMCP(ID: interger): void selectMCP(): void</p>	<p>Với Route: (aggregation) 1 Route đi qua từ 2 MCP. Một MCP thuộc một Route.</p> <p>Với Coordination: (association) sử dụng như thuộc tính.</p>



ProgressNoti	<p>Chức năng chính: cung cấp cách truy vấn đến dữ liệu Thông báo cập nhật tiến độ</p> <p>Các thuộc tính:</p> <p>task: Task</p> <p>updateStatus: String</p> <p>Các truy vấn:</p> <p>insertProgressNoti(data: object): void</p> <p>updateProgressNoti(data: object): void</p> <p>deleteProgressNoti(ID: interger): void</p> <p>selectProgressNoti(): void</p>	<p>Với Task:</p> <p>Với TaskController: (association) cung cấp truy vấn lấy, thêm, sửa, xóa dữ liệu Thông báo cập nhật tiến độ cho TaskController.</p>
--------------	--	--



PHẦN III

ARCHITECTURE DESIGN

1 Architectural approach and Modules for the whole WMC 2.0 system

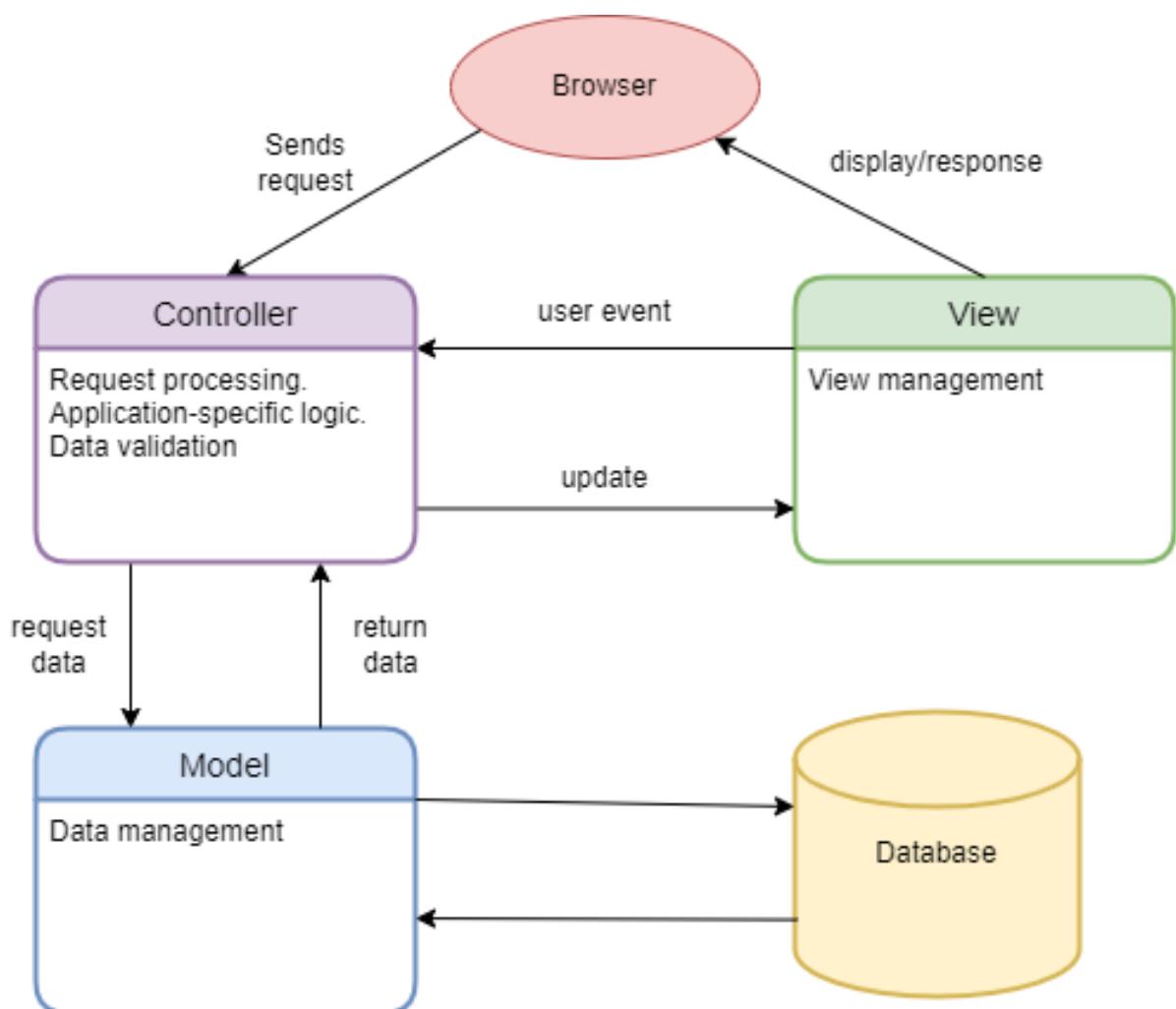
1.1 Architectural approach

Để hiện thực hệ thống như mong muốn, nhóm chọn hướng tiếp cận kiến trúc theo mô hình kiến trúc MVC (Model - Controller - View) như sau Trong đó:

- Người dùng truy cập và sử dụng hệ thống trên trình duyệt (**browser**).
- **View**: có thể xem đây là phần giao diện dành cho người sử dụng, là nơi quản lý về “phần nhìn” (View management), tức thực hiện việc hiển thị các đối tượng trong hệ thống, các biểu mẫu, thông báo,... để người dùng có thể quan sát và tương tác được. Ngoài ra, đây còn là nơi lắng nghe các sự kiện và thu nhận dữ liệu đầu vào từ người dùng (input) khi họ tương tác với hệ thống, sau đó chuyển giao cho **Controller** xử lý, rồi thực hiện hiển thị ra màn hình những kết quả/ dữ liệu được **Controller** xử lý và cập nhật.
- **Controller**: là nơi điều khiển sự tương tác giữa hai thành phần **Model** và **View**.
 - Request processing: **Controller** thu nhận và xử lý các yêu cầu từ người dùng được đưa đến thông qua **View**, nhận vào input rồi thực hiện các cập nhật tương ứng.
 - Application-specific logic: **Controller** là nơi chứa các phương thức xử lý chính của hệ thống, thực hiện các chức năng cụ thể để phục vụ cho hệ thống.
 - Data validation: kiểm tra tính hợp lệ của dữ liệu nhập vào (tên tài khoản, mật khẩu, thông tin tài khoản, ..).
- **Model**: là nơi quản lý dữ liệu của hệ thống, làm việc với database (nơi lưu trữ dữ liệu hệ thống). Mỗi model là dữ liệu được sử dụng bởi chương trình. **Model** cũng cung cấp những phương thức cơ bản xử lý logic nghiệp vụ.

Ví dụ: Ở module **Task Assignment**:

- Khi **Back Officer** muốn xem thông tin về các ca làm việc đang được lưu trữ trong hệ thống. Lúc này yêu cầu của **Back Officer** sẽ thông qua tầng **View** (Shift Management View) đến tầng **Controller** (Shift Controller) để xử lý. Tại đây, **Shift Controller** sẽ thực hiện tổng hợp các dữ liệu từ **Working Shift** (Model) sau đó cập nhật và hiển thị ra màn hình thông qua **Shift Management View**.
- Tương tự, khi **Back Officer** mong muốn tạo một ca làm việc mới, hay cập nhật thông tin của ca cũ, yêu cầu cũng như input sẽ được **Shift Management View** tiếp nhận và gửi về cho **Shift Controller** xử lý, tại đây dữ liệu mới sẽ được xác thực, xử lý rồi cập nhật cho model **Working Shift**.



Hình 11: Mô hình kiến trúc MVC

Xem chi tiết ảnh: [Here!](#)



1.2 List of Modules

1. Module đăng ký

Input:

- Thông tin cá nhân (Họ và tên, Mã CCCD, Ngày sinh,...) của người dùng X, tên đăng nhập, mật khẩu do người dùng X cung cấp.
- Vai trò của người dùng X (Collector hay Janitor).

Output:

- Một tài khoản x với một mã định danh (được hệ thống cấp), vai trò tương ứng với vai trò của người dùng X, tên đăng nhập, mật khẩu tương ứng với tên đăng nhập, mật khẩu do người dùng X cung cấp, và chứa các thông tin cá nhân của người dùng X.

2. Module xác thực

Input:

- Người dùng X.
- Tài khoản x.

Output:

- Xác định tài khoản x có phải thuộc về người dùng X không.
- Vai trò của người dùng X tương ứng với vai trò được chỉ định trong tài khoản x.

3. Module quản lý nhân sự

Input:

- Back officer X.
- Nhân sự Y (Collector, Janitor hoặc Back officer) (hoặc nhiều hơn).

Output:

- Thông tin của nhân sự Y (hoặc nhiều hơn) đã được tạo mới / cập nhật hoặc xoá bỏ.
- Các thông kê liên quan đến nhân sự Y (hoặc nhiều hơn).

4. Module quản lý phương tiện

Input:

- Back officer X.



- Phương tiện Y (hoặc nhiều hơn).

Output:

- Thông tin của phương tiện Y (hoặc nhiều hơn) đã được tạo mới / cập nhật hoặc xoá bỏ.
- Lộ trình di chuyển theo tháng của phương tiện Y (hoặc nhiều hơn).
- Trạng thái của phương tiện Y (Hoạt động hay ngừng hoạt động).
- Các thông kê liên quan đến phương tiện Y (hoặc nhiều hơn).

5. Module quản lý MCP

Input:

- Back officer X.
- MCP Y (hoặc nhiều hơn).

Output:

- Thông tin của MCP Y (hoặc nhiều hơn) đã được tạo mới / cập nhật hoặc xoá bỏ.
- Trạng thái của MCP Y (Đang sử dụng, Không còn sử dụng, Vô hiệu hoá).
- Sức chứa còn lại của MCP Y.
- Các thông kê liên quan đến MCP Y (hoặc nhiều hơn).

6. Module quản lý nhiệm vụ

Input:

- Back officer X.
- Tập các nhân sự Y (Collector, Janitor hoặc Back officer).
- Tập các phương tiện Z.
- Tập các MCP T.

Output:

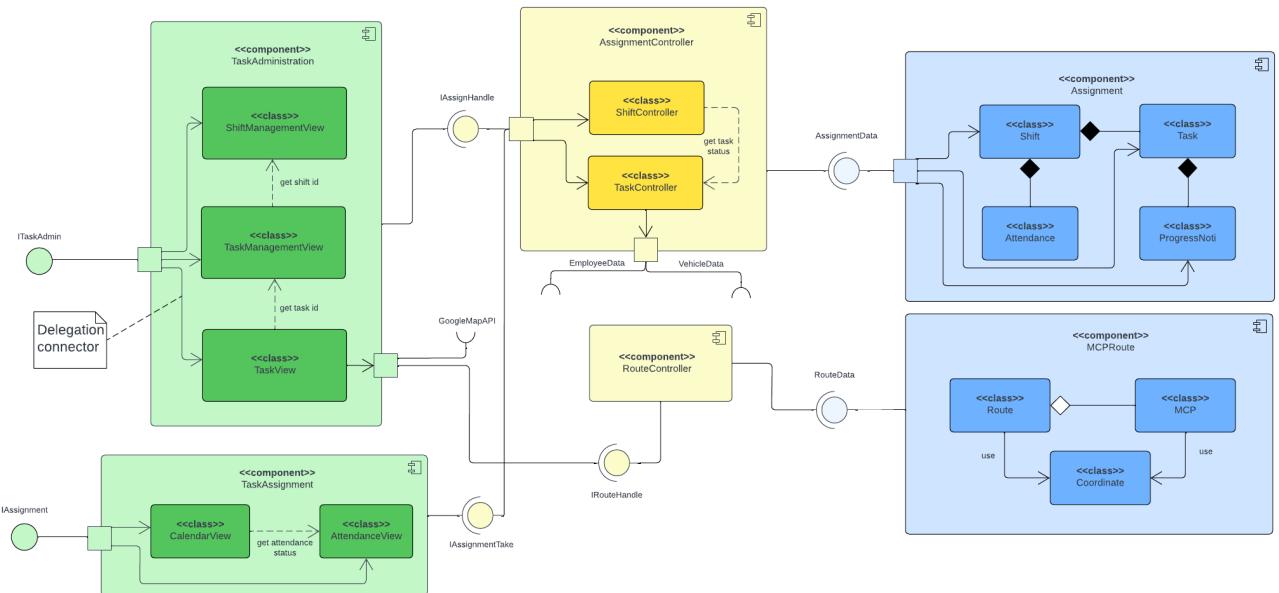
- Thời gian và vị trí làm việc tương ứng của các nhân sự trong tập các nhân sự Y
- Thực thể đi kèm tương ứng với vai trò của các nhân sự trong tập các nhân sự Y (Collector tương ứng với 1 phương tiện phân biệt trong tập các phương tiện Z, Janitor tương ứng với một số các MCP trong tập các MCP T).
- Lộ trình di chuyển trong thời gian làm việc tương ứng với phương tiện và nhân sự đã được gán.
- Các thông kê liên quan đến các ca làm việc.

2 Component Diagram - Task Assignment

2.1 Bản vẽ Component Diagram

Xem bản vẽ Component Diagram của nhóm tại

<https://drive.google.com/file/d/1xb8HKLsqHXh3U5lUmWZStRiQtGieiV7n/view?usp=sharing>



Hình 12: Component Diagram

2.2 Mô tả Component Diagram

Tầng View gồm có 2 component chính:

1. TaskAdministration:

- Chức năng chính: hiện thực các class cung cấp giao diện quản lý ca làm việc và nhiệm vụ.
- Gồm các class: **ShiftManagementView**, **TaskManagementView**, **TaskView**. Class **ShiftManagementView** cung cấp giao diện quản lý ca làm việc (xem danh sách, thêm, xóa, sửa ca). Class **TaskManagementView** cung cấp giao diện xem thông tin của một ca và quản lý danh sách nhiệm vụ của ca đó (xem danh sách nhiệm vụ, phân nhiệm vụ mới, xóa, sửa nhiệm vụ); theo dõi danh sách các thông báo cập nhật tiến độ nhiệm vụ gửi từ Collector/Janitor của ca để xác nhận cho



chúng. Class **TaskView** cung cấp giao diện hiển thị thông tin cụ thể của một nhiệm vụ và nhận yêu cầu tạo lộ trình tối ưu cho nhiệm vụ để hiển thị lên Bản đồ lộ trình và gửi Collector.

- Mối quan hệ giữa các thành phần: Class **TaskManagementView** dựa vào ID ca được cung cấp bởi **ShiftManagementView** để yêu cầu thông tin ca và danh sách nhiệm vụ của ca đó. Class **TaskView** dựa vào ID nhiệm vụ được cung cấp bởi **TaskManagementView** để yêu cầu thông tin chi tiết của nhiệm vụ (bao gồm lộ trình cố định của phương tiện sử dụng bởi nhiệm vụ dùng cho việc yêu cầu tạo lộ trình tối ưu).
- **ShiftManagementView**, **TaskManagementView**, **TaskView** hiện thực (realize) interface *ITaskAdmin*.
- Cả component yêu cầu (require) interface *IAssignHandle*. **TaskView** yêu cầu thêm các interface liên quan đến tạo và hiển thị lộ trình gồm *GoogleMapAPI* và *IRouteHandle*.

2. TaskAssignment:

- Chức năng chính: hiện thực các class cung cấp giao diện xem lịch làm việc cá nhân và giao diện điểm danh cho ca làm việc.
- Gồm các class: **CalendarView**, **AttendanceView**. **CalendarView** cung cấp giao diện xem lịch làm việc cá nhân: xem danh sách ca làm việc và chi tiết nhiệm vụ được giao cho mình trong ca, nhận yêu cầu cập nhật tiến độ cho nhiệm vụ khi hoàn thành công việc. **AttendanceView** cung cấp giao diện giúp cá nhân điểm danh (checkin, checkout) cho ca làm việc.
- Mối quan hệ giữa các thành phần: Class **CalendarView** hiển thị trạng thái điểm danh cho ca làm việc trên lịch sau thời gian điểm danh cho ca.
- Các thành phần trong component hiện thực interface *IAssignment*.
- Các thành phần trong component yêu cầu interface *IAssignmentTake*

Tầng **Controller** gồm có 2 component chính:

1. AssignmentController:

- Chức năng chính: hiện thực các class cung cấp phương thức xử lý sự kiện và điều khiển luồng dữ liệu liên quan đến phân công công việc (ca, nhiệm vụ, thông báo cập nhật tiến độ).
- Gồm các class: **ShiftController**, **TaskController**. Class **ShiftController** cung cấp các phương thức điều khiển luồng dữ liệu ca làm việc, kiểm tra và cập nhật



trạng thái điểm danh của nhân viên trong ca. Class **TaskController** cung cấp các phương thức điều khiển luồng dữ liệu nhiệm vụ và thông báo cập nhật tiến độ nhiệm vụ.

- Mỗi quan hệ giữa các thành phần: Class **ShiftController** dựa trên trạng thái các nhiệm vụ của nó được cập nhật bởi **TaskController** để cập nhật số lượng nhiệm vụ hoàn thành trong ca.
- Các thành phần trong component hiện thực interface *IAssignHandle*.
- **TaskController** yêu cầu các interface liên quan đến việc cung cấp danh sách các lựa chọn về nhân viên và phương tiện để phân công cho nhiệm vụ: *EmployeeData*, *VehicleData*.

2. **RouteController:**

- Chức năng chính: hiện thực các phương thức điều khiển dữ liệu về Lộ trình và tìm lộ trình tối ưu dựa trên lộ trình cố định hàng tháng của phương tiện và sức chứa hiện tại của các MCP trong lộ trình.
- Component hiện thực interface *IRouteHandle*.
- Component yêu cầu interface *RouteData*.

Tầng **Model** gồm có 2 component chính:

1. **Assignment:**

- Chức năng chính: lưu trữ và thực hiện truy vấn dữ liệu liên quan đến phân công công việc (ca, nhiệm vụ, thông báo cập nhật tiến độ).
- Gồm các class: **Shift** (Ca), **Task** (Nhiệm vụ), **Attendance**(Điểm danh), **ProgressNoti** (Thông báo cập nhật tiến độ).
- **Shift**, **Task**, **ProgressNoti** trong component hiện thực interface *AssignmentData*.

2. **MCPRoute:**

- Chức năng chính: lưu trữ và thực hiện truy vấn dữ liệu liên quan đến MCP và lộ trình.
- Gồm các class: **Route** (Lộ trình), **MCP** (Điểm thu gom), **Coordinate**(Tọa độ).
- Component hiện thực interface *RouteData*.



Bảng mô tả các Interface:

Interface	Chức năng	Phương thức
ITaskAdmin	Cung cấp giao diện giúp Back Officer quản lý phân công công việc.	renderShiftManagementView(); renderTaskManagementView(shift_id); renderTaskView(task_id); renderAddTaskView(shift_id); renderEditTaskView(task_id);
GoogleMapAPI	Cung cấp Google Map API giúp tích hợp bản đồ trực quan hiển thị lộ trình tối ưu.	displayGoogleMap(opt_route: Route)
IAssignment	Cung cấp giao diện giúp Collector và Janitor xem lịch làm việc và thực hiện điểm danh.	renderCalendarView(); renderAttendanceView();



IAssignHandle	Cung cấp các phương thức giúp điều khiển luồng dữ liệu phân công công việc liên quan đến các tác vụ quản lý của Back officer.	getShiftList(): List<Shift> getShiftInfo(shift_id: Integer): Shift requestCreateShift(shift_data: Object): void requestUpdateShift(shift_id: Integer, edit_data: Object): void requestDeleteShift(shift_id: Integer): void getProgressNotiList(): List<ProgressNoti> getTaskList(): List<Task> getTaskInfo(task_id: Integer): Task getAssignOption(): Object requestCreateTask(task_data: Object): void requestUpdateTask(task_id: Integer, edit_data: Object): void requestDeleteTask(task_id: Integer): void requestUpdateTaskProgress(task_id: Integer, status: String): void
IAssignmentTake	Cung cấp các phương thức giúp điều khiển luồng dữ liệu công việc liên quan đến các tác vụ xem và cập nhật tiến độ của Collector/Janitor.	getShiftList(): List<Shift> getShiftInfo(shift_id: Integer): Shift getTaskList(): List<Task> getTaskInfo(task_id: Integer): Task updateCheckin(shift_id: Integer, employee_id: Integer): void updateCheckout(shift_id: Integer, employee_id: Integer): void requestCreateProgressNoti(task_id: Integer, status: String): void
IRouteHandle	Cung cấp phương thức cho tác vụ yêu cầu tạo lộ trình tối ưu.	createRoute(mcp_list: List<MCP>): void



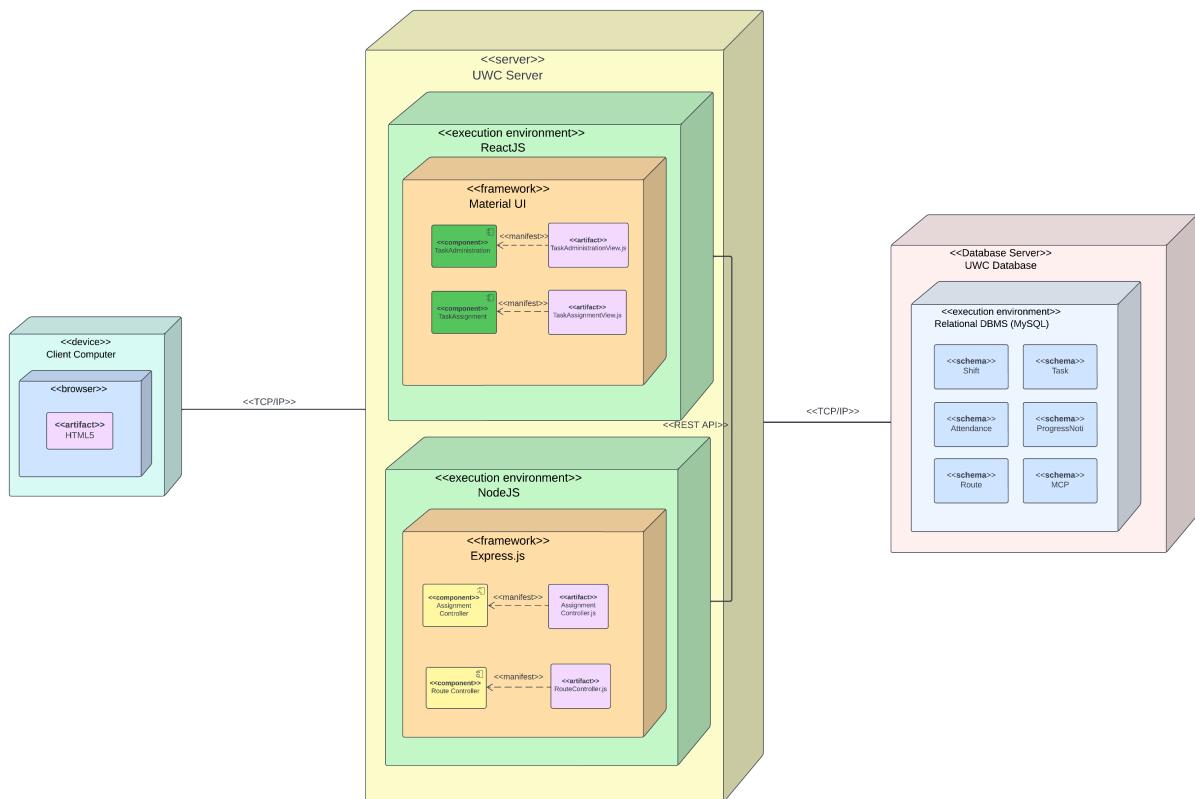
EmployeeData	Cung cấp API lấy dữ liệu nhân viên để phân công cho nhiệm vụ	<code>getJanitorList(): List<Janitor></code> <code>getCollectorList(): List<Collector></code>
VehicleData	Cung cấp API lấy dữ liệu phương tiện để chỉ định cho nhiệm vụ	<code>getVehicleList(): List<Vehicle></code>
AssignmentData	Cung cấp API truy vấn dữ liệu liên quan đến phân công công việc	<code>insertShift(data: object): void</code> <code>updateShift(data: object): void</code> <code>deleteShift(ID: interger): void</code> <code>selectShift(): void</code> <code>insertTask(data: object): void</code> <code>updateTask(data: object): void</code> <code>deleteTask(ID: interger): void</code> <code>selectTask(): void</code> <code>selectProgressNoti(): void</code> <code>insertProgressNoti(data: object): void</code> <code>deleteProgressNoti(ID: interger): void</code>
RouteData	Cung cấp API lấy dữ liệu về lộ trình và các MCP	<code>selectRoute(): void</code> <code>selectMCP(): void</code>

3 Deployment diagram

3.1 Bản vẽ Deployment Diagram

Xem bản vẽ Deployment Diagram của nhóm (PDF) tại

<https://drive.google.com/file/d/1AJCQVfrEmKHQE7D2uKoxzjZY01mTTDiX/view?usp=sharing>



Hình 13: Deployment Diagram

3.2 Mô tả Deployment Diagram

- Hệ thống là một ứng dụng Web, được phát triển theo kiến trúc MVC, bao gồm 3 máy chủ chính: Client, Server và Database Server.
- Máy chủ Client sẽ kết nối với Server thông qua giao thức TCP/IP, dữ liệu nhận được từ Server là các file hiện thực giao diện và sẽ được hiển thị trên browser website tại máy tính người dùng (Client) thông qua trình đọc HTML5 được nhúng trong trình duyệt.
- Máy chủ Server bao gồm môi trường thực thi (execution environment) các module chức năng chính của hệ thống. Môi trường thực thi phía **back-end** là *NodeJS*, sử dụng



Framework là *Express.js*, bao gồm 2 component chính là **AssignmentController** và **RouteController** đảm nhận các tác vụ xử lý chính và điều khiển luồng dữ liệu liên quan đến phân công công việc và lộ trình. Môi trường thực thi phía **front-end** là *ReactJS*, sử dụng Framework là *Material UI*, bao gồm 2 component chính là **TaskAdministration** và **TaskAssignment** đảm nhận việc hiện thực các giao diện người dùng liên quan đến phân công công việc. Các component sẽ được hiện thực từ các file mã nguồn JavaScript tương ứng. Thông qua các REST API, front-end và back-end có thể giao tiếp được với nhau để cập nhật, hiển thị giao diện và nhận phản hồi các tác vụ người dùng.

- Máy chủ Server sẽ kết nối với Database Server thông qua giao thức TCP/IP, để truy xuất dữ liệu hiển thị lên giao diện hoặc cập nhật lại dữ liệu mỗi khi người dùng tương tác với ứng dụng, sau đó dữ liệu sau khi cập nhật sẽ được truyền qua giao thức TCP/IP sang Server và truyền sang máy Client để cập nhật lại giao diện.
- Máy chủ Database bao gồm môi trường thực thi là DBMS MySQL. Trong phạm vi module Task Assignment, máy chủ Database lưu trữ các thông tin bao gồm: Shift, Task, Attendance, ProgressNoti, MCP, Route ở dạng bảng có cấu trúc và các mối quan hệ với nhau.
- Trong phạm vi module Task Assignment, hệ thống cung cấp các chức năng chính sau:
 - Đối với **Back Officer**: xem danh sách, thêm, xóa, sửa các ca làm việc; xem danh sách, xem chi tiết, phân công, xóa, sửa các nhiệm vụ của ca; thực hiện xác nhận cho các cập nhật tiến độ gửi từ nhân viên; tạo lộ trình tối ưu và gửi cho tài xế nhiệm vụ được hỗ trợ bởi hệ thống.
 - Đối với **Janitor** và **Collector**: điểm danh ca làm việc; xem danh sách nhiệm vụ, xem chi tiết nhiệm vụ và cập nhật tiến độ nhiệm vụ.
 - Hệ thống đảm bảo 1000 người sử dụng, 100-200 người truy xuất cùng lúc.



PHẦN IV

IMPLEMENTATION - SPRINT 1

1 Github for Version Control

- Link github: <https://github.com/anduc146khmt/uwc-bit-by-bit-deploy.git>

2 Revisions or Changelogs

2.1 Hình ảnh

- Hình ảnh tất cả commits tính đến giai đoạn upload Documents và Mockup.



master

Commits on Nov 19, 2022

- update README.md description
anduc146khmt committed 16 minutes ago
- refactor documents folder
anduc146khmt committed 18 minutes ago
- refactor figma folder
anduc146khmt committed 19 minutes ago
- update mockup.txt
anduc146khmt committed 23 minutes ago
- complete employee view
anduc146khmt committed 1 hour ago
- update employee view
anduc146khmt committed 1 hour ago
- complete working shift view
anduc146khmt committed 1 hour ago
- complete task view
anduc146khmt committed 1 hour ago
- update working shift page for task management view
anduc146khmt committed 1 hour ago
- update tracking task view for back officer
anduc146khmt committed 1 hour ago
- update figma task assignement page for back officer
anduc146khmt committed 1 hour ago
- update figma login page
anduc146khmt committed 1 hour ago

Commits on Nov 9, 2022

- create documents folder
anduc146khmt committed 11 days ago



– Ví dụ thông tin chi tiết cho commit **upload mockup.txt**.

```
v 2 Figma/mockup.txt
...
... 00 -0,0 +1,2 00
1 + Link Figma: https://www.figma.com/file/y121I4X46wCG8b6QUvruAd/Mock-up-Software-Engineering-221?node-id=0%3A1
2 + Link Mockup: https://www.figma.com/proto/y121I4X46wCG8b6QUvruAd/Mock-up-Software-Engineering-221?node-id=25%3A71&scaling=scale-down&page-id=0%3A1&starting-point-node-id=25%3A71

v 18 README.md
...
... 00 -0,0 +1,18 00
1 + # Software Engineering Project - HK221
2 +
3 + A UWC 2.0 used to provide solution to Urban Waste Collection Problems. More description will be
4 + found in our report.
5 +
6 + # Contributor
7 + We are students studying Computer Science come from group **BIT BY BIT** at Bach Khoa University
8 + (HCMUT)
9 +
10 + Our member of the team:
11 + * Nguyễn Đăng Anh Khoa - 2010339
12 + * Nguyễn Đức An - 2010382
13 + * Trần Phúc Anh - 2010333
14 + * Trần Chí Công - 2010170
15 + * Cao Trần Anh Khoa - 2010335
16 +
17 + # Contact
18 + If you have any questions, please contact me via email: ducan1406@gmail.com
```

2.2 Mô tả

Chạy lệnh **git log > log.txt**

Ta được thông tin mô tả chi tiết git log như sau:

```
commit 37afae3ad0dce7b2dd70be872bbac59d2a560a96
Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>
Date:   Sat Nov 19 17:10:10 2022 +0700

    update final report

commit 97ed513382640afb4dc932232b3326953c132f83
Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>
Date:   Sat Nov 19 15:53:54 2022 +0700

    update README.md description

commit 3996bdb01e5e3c6935d3004c6fc003427269942f
Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>
Date:   Sat Nov 19 15:52:22 2022 +0700

    refactor documents folder

commit 531007676f8c4ea274bfa488683e6c4e89deaeca
Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>
Date:   Sat Nov 19 15:50:55 2022 +0700
```



```
refactor figma folder
```

```
commit efb0a24860dfdf248b4f7ab9d0d0e61c9d36c69c
```

```
Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>
```

```
Date: Sat Nov 19 15:47:43 2022 +0700
```

```
update mockup.txt
```

```
commit 676baaa0a7b131daa122783f45fe280ccdcce37e4
```

```
Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>
```

```
Date: Sat Nov 19 15:14:02 2022 +0700
```

```
complete employee view
```

```
commit c6f55f0a195845c8d41e703f41217bd6653a6bdf
```

```
Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>
```

```
Date: Sat Nov 19 15:12:28 2022 +0700
```

```
update employee view
```

```
commit 87dff739884c1a6976dc759f8539337c8aa2151e
```

```
Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>
```

```
Date: Sat Nov 19 15:11:27 2022 +0700
```

```
complete working shift view
```

```
commit 1399c4edc0ae41daf967009ee4955fdeb55e7756
```

```
Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>
```

```
Date: Sat Nov 19 15:09:59 2022 +0700
```

```
complete task view
```

```
commit 43d13d78cb393b55f8fc0ff611508b85100cf33e
```

```
Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>
```

```
Date: Sat Nov 19 14:59:17 2022 +0700
```

```
update working shift page for task management view
```

```
commit e6c0bca6fdb0d01fa9cd222955c9dc972d5dc062
```



Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>

Date: Sat Nov 19 14:56:16 2022 +0700

update tracking task view for back officer

commit 04649aecdad868dd58380ba221020941898eaf1b

Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>

Date: Sat Nov 19 14:54:00 2022 +0700

update figma task assignement page for back officer

commit cc348efd073e7976d531a759c5176e1834fca6f8

Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>

Date: Sat Nov 19 14:51:50 2022 +0700

update figma login page

commit 2462976a46dcbe66d2837fa3960e38e2e042595f

Author: Nguyen Duc An <an.nguyenduc1406@hcmut.edu.vn>

Date: Wed Nov 9 00:25:41 2022 +0700

create documents folder

3 Implement MVP1 - Figma & Mockup

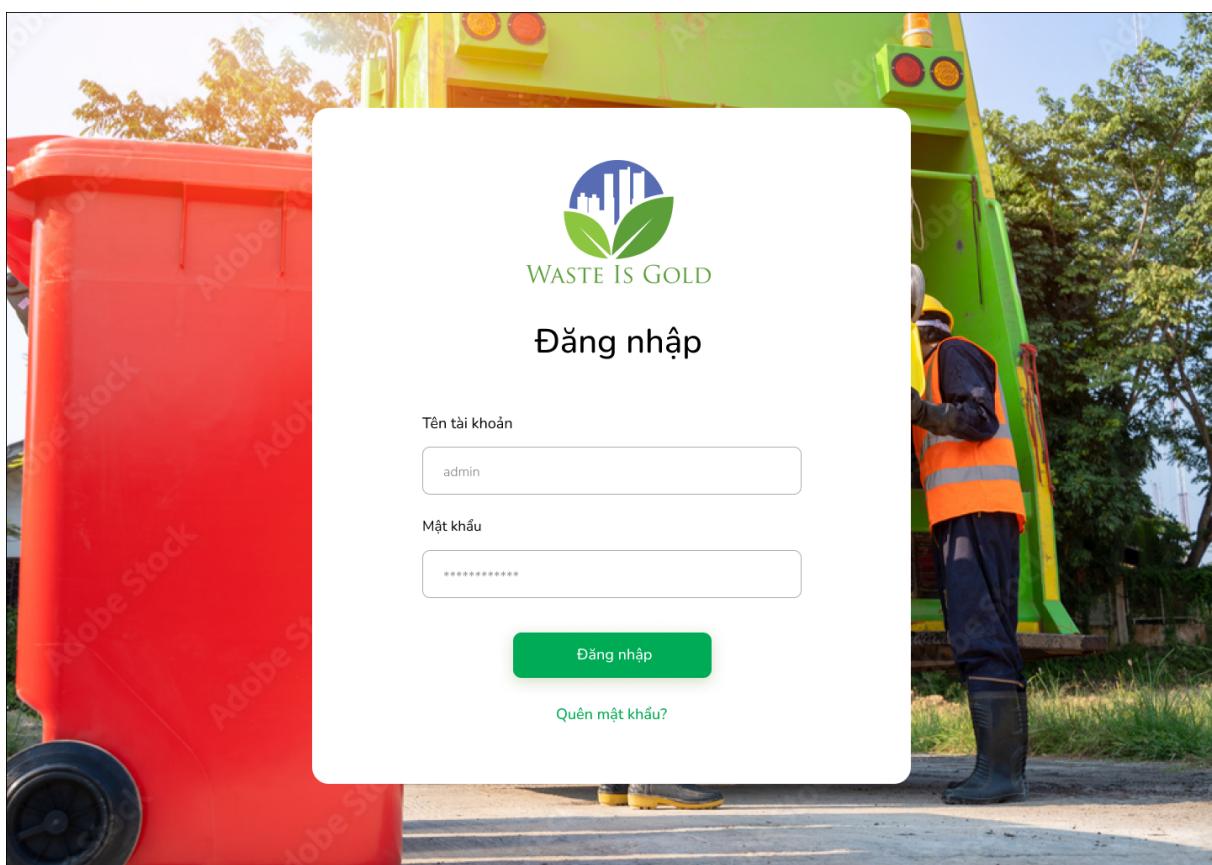
3.1 Link Figma & Mockup

- Link Figma: <https://www.figma.com/Figma-SoftwareEngineering-221/BITBYBIT>
- Link Mockup: <https://www.figma.com/Mockup-SoftwareEngineering-221/BITBYBIT>

3.2 Mô tả

1. Giao diện đăng nhập

- Back Officer nhập tên tài khoản và mật khẩu để đăng nhập hệ thống.



Hình 14: Giao diện đăng nhập

2. Giao diện quản lý ca làm việc

- Back Officer có thể xem danh sách các ca làm việc đã tạo. Mỗi hàng trong bảng danh sách gồm những thông tin của ca làm việc đó: ID ca, ngày, giờ bắt đầu, giờ kết thúc, trạng thái ca (Chưa diễn ra, Đang diễn ra, Kết thúc) và số nhiệm vụ trong ca đã hoàn thành.



- Back Officer có thể lọc ca làm việc theo ngày, theo trạng thái hoặc theo ID nhân viên có nhiệm vụ trong ca đó.

ID ca	Ngày	Giờ bắt đầu	Giờ kết thúc	Trạng thái	Đã xong
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4
12345	10/04/2022	6:00	7:30	Kết thúc	4/4
12345	10/04/2022	6:00	7:30	Chưa diễn ra	0/4
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4

Hình 15: Giao diện Quản lý ca làm việc

- Back Officer có thể nhấp vào nút “Tạo ca mới” để thêm một ca làm việc. Sau đó, chọn ngày, nhập giờ bắt đầu, giờ kết thúc và nhấn “Xác nhận” (nếu muốn xác nhận thêm ca mới) hoặc “Reset” (nếu muốn nhập lại thông tin).



The screenshot shows a mobile application interface for managing shifts. At the top, there is a search bar and a user profile icon for 'Khoa Nguyen'. Below the header, a navigation bar includes icons for 'Ca làm việc' (Shift Work), 'Nhân viên' (Employee), 'Phương tiện' (Means), and 'Điểm thu gom' (Collection Point). The main content area displays a list of shifts with columns for ID, Date, Start Time, End Time, Status, Tasks completed, and Task status. A modal dialog box titled 'Tạo ca mới' (Create New Shift) is open in the center. It contains fields for 'Giờ bắt đầu' (Start Time) set to 6:00, 'Ngày' (Date) set to 10/04/2022, and 'Giờ kết thúc' (End Time) set to 7:30. There are two buttons at the bottom of the modal: 'Xác nhận' (Confirm) in green and 'Reset' in red.

- Nếu thành công, hiện thông báo “Tạo ca thành công.”



Tìm kiếm Q

Khoa Nguyen X

✓ **Tạo ca thành công**

Danh sách ca Tạo ca mới

Filters ≡

Chọn ngày Filter Trạng thái Filter

ID nhân viên

ID ca	Ngày	Giờ bắt đầu	Giờ kết thúc	Trạng thái	Đã xong	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Kết thúc	4/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Chưa diễn ra	0/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮

- Nếu thất bại, hiện thông báo “Tạo ca thất bại”.



Tìm kiếm Q

Khoa Nguyen Khoa Nguyen ⋮

× **Tạo ca thất bại**

Danh sách ca Tạo ca mới

Filters ⋮

Chọn ngày Trạng thái ⋮ Filter Filter

ID nhân viên

ID ca	Ngày	Giờ bắt đầu	Giờ kết thúc	Trạng thái	Đã xong	⋮	
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Kết thúc	4/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Chưa diễn ra	0/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ	⋮

- Back Officer có thể nhấp vào biểu tượng 3 dấu chấm bên phải cùng mỗi hàng để hiển bảng chọn “Chỉnh sửa thời gian” và “Xóa ca”



Danh sách ca

Tạo ca mới

Filters

Chọn ngày

Trạng thái

ID nhân viên

ID ca	Ngày	Giờ bắt đầu	Giờ kết thúc	Trạng thái	Đã xong	
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ
12345	10/04/2022	6:00	7:30	Kết thúc	4/4	Chỉnh sửa thời gian
12345	10/04/2022	6:00	7:30	Chưa diễn ra	0/4	Xóa ca
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ

- Nhấp “Chỉnh sửa thời gian” khi muốn chỉnh sửa thời gian của ca.



The screenshot shows a software application interface for managing shifts. At the top, there's a search bar and a user profile for 'Khoa Nguyen'. Below the header, a button for 'Tạo ca mới' (Create new shift) is visible. A 'Filters' section includes 'Chọn ngày' (Select date) and 'Trạng thái' (Status). A green 'Filter' button is also present. A modal dialog titled 'Sửa thời gian ca' (Edit shift time) is open in the center. It contains fields for 'Giờ bắt đầu' (Start time) set to 6:00, 'Ngày' (Day) set to 10/04/2022, and 'Giờ kết thúc' (End time) set to 7:30. There are 'Xác nhận' (Confirm) and 'Reset' buttons. In the background, a table lists shifts for employee ID 12345 on April 10, 2022, from 6:00 to 7:30. The status for all shifts is 'Chưa diễn ra' (Not yet started), and they are marked as 'Đang diễn ra' (Currently running) in the modal. The table includes columns for Employee ID, Date, Start Time, End Time, Status, Completed tasks (3/4), and Task status.

Employee ID	Date	Start Time	End Time	Status	Completed tasks	Task status
12345	10/04/2022	6:00	7:30	Chưa diễn ra	0/4	Nhiệm vụ
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ
12345	10/04/2022	6:00	7:30	Đang diễn ra	3/4	Nhiệm vụ

- Nhập “Xóa ca” khi muốn xóa ca. Chọn “Xác nhận” để xác nhận xóa.



The screenshot shows a software application window titled "Danh sách ca" (List of shifts). On the left, there's a sidebar with icons for "Ca làm việc" (Work shift), "Nhân viên" (Employee), "Phương tiện" (Means of transport), and "Điểm thu gom" (Collection point). The main area displays a table of shift details. A modal dialog box is centered over the table, asking "Bạn có chắc chắn muốn xóa bỏ ca này?" (Are you sure you want to delete this shift?). It has two buttons: "Xác nhận" (Confirm) in green and "Hủy bỏ" (Cancel) in red.

3. Giao diện quản lý nhiệm vụ

- Back Officer có thể nhấp vào “Nhiệm vụ” bên phải của hàng trong danh sách ca để vào giao diện quản lý nhiệm vụ của ca đó.
- Giao diện gồm thông tin tóm lược của ca làm việc tương ứng, bảng danh sách nhiệm vụ và thông báo tiến độ của các nhiệm vụ trong ca.
- Back Officer có thể xem danh sách các nhiệm vụ đã tạo trong ca. Mỗi hàng trong bảng danh sách gồm những thông tin của nhiệm vụ đó: ID nhiệm vụ, trạng thái (Chưa bắt đầu, Đang thu gom, Vận chuyển, Hoàn thành), mã phương tiện, tài xế, mã lộ trình.
- Back Officer có thể lọc nhiệm vụ theo trạng thái hoặc ID nhân viên.
- Back Officer có thể xác nhận trạng thái được cập nhật cho ca (cập nhật bởi Collector và Janitor) để chuyển ca sang trạng thái kế tiếp ở mục Thông báo tiến độ.



Hình 16: Giao diện Quản lý nhiệm vụ

- Back Officer có thể nhấp vào nút “Thêm nhiệm vụ” để thêm nhiệm vụ mới cho ca. Sau đó, chuyển sang trang Phân nhiệm vụ, ở bước 1 có thể chọn phương tiện (cho tóm tắt tham khảo về payload capacity, cargo space, fuel consumption) và chọn tài xế (cho tham khảo về băng lái); ở bước 2 có thể phân người thu gom (nhóm trưởng, các thành viên) cho từng MCP trong lô trình ứng với phương tiện được chọn ở bước 1. Sau đó nhấp “Lưu lại”.



Tìm kiếm Q

Khoa Nguyen Khoa Nguyen ▼

← Phân nhiệm vụ

1 Phương tiện và tài xế 2 Người thu gom

Chọn phương tiện: Chọn tài xế:

SU-20-1234 Tên tài xế

SU-20-1234
Payload Capacity: 9.55 t
Cargo Space: 70 m³
Fuel Consumption: 35 l/100km

SU-20-1234
Payload Capacity: 9.55 t
Cargo Space: 70 m³
Fuel Consumption: 35 l/100km

SU-20-1234
Payload Capacity: 9.55 t
Cargo Space: 70 m³
Fuel Consumption: 35 l/100km

Tiếp theo

Tìm kiếm Q

Khoa Nguyen Khoa Nguyen ▼

← Phân nhiệm vụ

1 Phương tiện và tài xế 2 Người thu gom

Chọn phương tiện: Chọn tài xế:

SU-20-1234 Nguyễn Văn Anh

SU-20-1234
Payload Capacity: 9.55 t
Cargo Space: 70 m³
Fuel Consumption: 35 l/100km

Nguyễn Văn Anh
Loại bằng lái: C

Nguyễn Văn Anh
Loại bằng lái: B2

Nguyễn Văn Anh
Loại bằng lái: F

Nguyễn Văn Anh
Loại bằng lái: B2

Tiếp theo



Tìm kiếm Q

Khoa Nguyen Khoa Nguyen ▼

← Phân nhiệm vụ

1 Phương tiện và tài xế ————— 2 Người thu gom

Danh sách các Điểm thu gom trong lộ trình hàng tháng của phương tiện

MCP ID	Địa điểm	Nhóm trưởng	Thành viên
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Nguyễn Văn An ▼	Tên thành viên ▼
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Nguyễn Văn An Nguyễn Thị Bình	Tên thành viên ▼
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Lê Văn Đạt	Tên thành viên ▼
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng ▼	Tên thành viên ▼
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng ▼	Tên thành viên ▼
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng ▼	Tên thành viên ▼

Quay lại Lưu lại

Tìm kiếm Q

Khoa Nguyen Khoa Nguyen ▼

← Phân nhiệm vụ

1 Phương tiện và tài xế ————— 2 Người thu gom

Danh sách các Điểm thu gom trong lộ trình hàng tháng của phương tiện

MCP ID	Địa điểm	Nhóm trưởng	Thành viên
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Nguyễn Văn An ▼	Nguyễn Thị Bình Lê Văn Đạt ▼
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng ▼	Tên thành viên ▼
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng ▼	Tên thành viên ▼
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng ▼	Tên thành viên ▼
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng ▼	Tên thành viên ▼
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng ▼	Tên thành viên ▼

Quay lại Lưu lại



- Nếu gặp lỗi khi phân công, hiện thông báo lỗi tương ứng. Chẳng hạn, nếu nhân viên Nguyễn Thị Bình đã được phân cho nhiệm vụ khác cùng ca, hiện thông báo “**Nguyễn Thị Bình đã được phân làm nhiệm vụ khác trong ca này**”

The screenshot shows a mobile application interface for task assignment. On the left, there's a sidebar with icons for 'Ca làm việc' (Work Shift), 'Nhân viên' (Employee), 'Phương tiện' (Vehicle), and 'Điểm thu gom' (Collection Point). The main screen displays a search bar at the top. Below it, a red banner with a close button and the text "Nguyễn Thị Bình đã được phân làm nhiệm vụ khác trong ca này" (Nguyen Thi Binh has been assigned to do another task during this shift). A back arrow leads to the 'Phân nhiệm vụ' (Assign Task) screen. This screen shows two tabs: '1 Phương tiện và tài xế' (Vehicle and Driver) and '2 Người thu gom' (Collector). The 'Danh sách các Điểm thu gom trong lộ trình hàng tháng của phương tiện' (List of collection points in the monthly route of the vehicle) is displayed below. A table lists six collection points, each with dropdown menus for 'Tên nhóm trưởng' (Group Leader Name) and 'Tên thành viên' (Member Name). At the bottom right are 'Quay lại' (Back) and 'Lưu lại' (Save) buttons.

MCP ID	Địa điểm	Nhóm trưởng	Thành viên
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Nguyễn Văn An	Nguyễn Thị Bình, Lê Văn Đạt
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng	Tên thành viên
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng	Tên thành viên
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng	Tên thành viên
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng	Tên thành viên
12345	Tân Lập, Đông Hòa, Dĩ An, Bình Dương	Tên nhóm trưởng	Tên thành viên

- Nếu thành công, hiện thông báo “**Lưu nhiệm vụ thành công**”.



The screenshot shows a mobile application interface for managing tasks. On the left, there's a sidebar with a logo, a user profile for 'Khoa Nguyen' (@khoanguyendanganh), and several buttons: 'Ca làm việc' (Work Shift), 'Nhân viên' (Employee), 'Phương tiện' (Means), and 'Điểm thu gom' (Collection Point). The main area has a search bar at the top. Below it, a green button says 'Lưu nhiệm vụ thành công' (Save task successfully). The central part shows a task log for 'ID ca #12345' on 'Ngày: 10/4/2022'. It includes a progress bar from 'Đang diễn ra' (Ongoing) to '3/4'. To the right, a section titled 'Thông báo tiến độ' (Progress notification) lists three recent updates: '10 phút trước #12345' (Task updated), '7 phút trước #23456' (Task updated), and '5 phút trước #34567' (Task updated), each with a checkmark and the word 'Xác nhận' (Confirmed). Below this is a 'Danh sách nhiệm vụ' (List of tasks) with a 'Phân nhiệm vụ' (Assign task) button. A 'Filters' section allows filtering by 'ID nhân viên' (Employee ID) and 'Chưa diễn ra' (Not ongoing). The main list shows four tasks:

ID nhiệm vụ	Trạng thái	Mã phương tiện	Tài xế	Mã lô trình	Chi tiết	⋮
12345	Chưa bắt đầu	SU-20-1234	Nguyễn Văn Anh	Q345	Chi tiết	⋮
12345	Đang thu gom	SU-20-1234	Nguyễn Văn Anh	Q345	Chi tiết	⋮
12345	Vận chuyển	SU-20-1234	Nguyễn Văn Anh	Q345	Chi tiết	⋮
12345	(Hoàn thành)	SU-20-1234	Nguyễn Văn Anh	Q345	Chi tiết	⋮

– Nếu thất bại (thêm vào cơ sở dữ liệu), hiện thông báo “Lưu nhiệm vụ thất bại”



The screenshot shows a mobile application interface for managing waste collection tasks. On the left, there's a sidebar with a logo, a search bar, and a user profile for 'Khoa Nguyen'. Below the sidebar are four buttons: 'Ca làm việc' (Work Shift), 'Nhân viên' (Employee), 'Phương tiện' (Vehicle), and 'Điểm thu gom' (Collection Point). The main content area has a red banner at the top right with a close button and the text 'Lưu nhiệm vụ thất bại' (Save task failed). Below this, there's a section for 'ID ca #12345' showing the date 'Ngày: 10/4/2022', start time 'Giờ bắt đầu: 6:00', end time 'Giờ kết thúc: 7:30', and a progress bar at 3/4. The status 'Đang diễn ra' (Ongoing) is shown in orange. To the right, there's a 'Thông báo tiến độ' (Progress notification) section with three green boxes containing task details and checkmarks for 'Xác nhận' (Confirmation). At the bottom, there's a 'Danh sách nhiệm vụ' (Task list) with a 'Phân nhiệm vụ' (Assign task) button, filters for 'ID nhân viên' (Employee ID) and 'Chưa diễn ra' (Not ongoing), and a 'Filter' button. The task list table has columns for 'ID nhiệm vụ', 'Trạng thái' (Status), 'Mã phương tiện' (Vehicle ID), 'Tài xế' (Driver), 'Mã lô trình' (Route ID), and 'Chi tiết' (Details). The table contains four rows of task data.

ID nhiệm vụ	Trạng thái	Mã phương tiện	Tài xế	Mã lô trình	Chi tiết
12345	Chưa bắt đầu	SU-20-1234	Nguyễn Văn Anh	Q345	Chi tiết
12345	Đang thu gom	SU-20-1234	Nguyễn Văn Anh	Q345	Chi tiết
12345	Vận chuyển	SU-20-1234	Nguyễn Văn Anh	Q345	Chi tiết
12345	Hoàn thành	SU-20-1234	Nguyễn Văn Anh	Q345	Chi tiết

- Back Officer có thể nhấp vào biểu tượng 3 dấu chấm bên phải cùng mỗi hàng để hiển bảng chọn “Chỉnh sửa” và “Xóa nhiệm vụ”. Nhấp “Chỉnh sửa” khi muốn chỉnh sửa thông tin nhiệm vụ. Các giao diện tiếp theo tương tự khi phân nhiệm vụ.



Tìm kiếm Q

Khoa Nguyen Khoa Nguyen ▼

← Sửa nhiệm vụ

1 Phương tiện và tài xế ————— 2 Người thu gom

Chọn phương tiện:

Chọn tài xế:

SU-20-1234
Payload Capacity: 9.55 t
Cargo Space: 70 m³
Fuel Consumption: 35 l/100km

Nguyễn Văn Anh
ID: 12345

Tiếp theo

Ca làm việc

- Nhân viên
- Phương tiện
- Điểm thu gom

- Nhập “Xóa nhiệm vụ” khi muốn xóa nhiệm vụ. Chọn “Xác nhận” để xác nhận xóa.



The screenshot shows a mobile application interface for managing tasks. At the top, there is a header with a search bar, a notification icon, and a user profile for 'Khoa Nguyen'. Below the header, a navigation bar includes items like 'Ca làm việc' (Work Shift), 'Nhân viên' (Employee), 'Phương tiện' (Means), and 'Điểm thu gom' (Collection Point). A central card displays task details: ID ca #12345, Date: 10/4/2022, Status: Đang diễn ra (In Progress), Start time: 6:00, End time: 7:30. A modal dialog box in the center asks 'Bạn có chắc chắn muốn xóa nhiệm vụ này?' (Are you sure you want to delete this task?). It has two buttons: 'Xác nhận' (Confirm) in green and 'Hủy bỏ' (Cancel) in red. To the right of the main card, a sidebar titled 'Thông báo tiến độ' (Progress Report) lists recent task updates with timestamps, IDs, and status: '10 phút trước #12345 Nhiệm vụ cập nhật đã thu gom ✓ Xác nhận', '7 phút trước #23456 Nhiệm vụ cập nhật đã thu gom ✓ Xác nhận', and '5 phút trước #34567 Nhiệm vụ cập nhật đã thu gom ✓ Xác nhận'. Below the progress report is a pagination control with page 1 of 2. At the bottom, a table lists tasks with columns: ID nhiệm vụ, Trạng thái (Status), Mã phương tiện (Vehicle ID), Tài xế (Driver), and Mã lộ trình (Route ID). Each row includes a 'Chi tiết' (Detail) button and a more options button (three dots). The tasks listed are: 12345 (Chưa bắt đầu), 12345 (Đang thu gom), 12345 (Vận chuyển), and 12345 (Hoàn thành).

4. Giao diện xem chi tiết nhiệm vụ và tạo route tối ưu

- Back Officer có thể nhấp vào “Chi tiết” bên phải của hàng trong danh sách nhiệm vụ để vào giao diện xem chi tiết nhiệm vụ và tạo route tối ưu. Giao diện gồm thông tin tóm lược của nhiệm vụ tương ứng, bảng danh sách phân công người thu gom ở các MCP trong lộ trình của phương tiện.



The screenshot shows the Route Planner application interface. At the top right, there is a red button labeled "Lưu nhiệm vụ thất bại". Below it, a sidebar on the left lists categories: "Ca làm việc" (selected), "Nhân viên", "Phương tiện", and "Điểm thu gom". The main area displays task details: ID ca #12345, Ngày: 10/4/2022, Giờ bắt đầu: 6:00, Giờ kết thúc: 7:30, and a progress bar showing 3/4. To the right, a panel titled "Thông báo tiến độ" lists three tasks with status updates: "10 phút trước #12345", "7 phút trước #23456", and "5 phút trước #34567", all marked as "Xác nhận". Below these panels is a "Danh sách nhiệm vụ" table with columns: ID nhiệm vụ, Trạng thái, Mã phương tiện, Tài xế, and Mã lộ trình. The table contains four rows of data, each with a "Chi tiết" button and a more options button (three dots). The table header includes filters: "ID nhân viên" and "Chưa diễn ra".

ID nhiệm vụ	Trạng thái	Mã phương tiện	Tài xế	Mã lộ trình
12345	Chưa bắt đầu	SU-20-1234	Nguyễn Văn Anh	Q345
12345	Đang thu gom	SU-20-1234	Nguyễn Văn Anh	Q345
12345	Vận chuyển	SU-20-1234	Nguyễn Văn Anh	Q345
12345	Hoàn thành	SU-20-1234	Nguyễn Văn Anh	Q345

Hình 17: Giao diện xem nhiệm vụ và tạo route tối ưu

- Back Officer có thể nhập “Tạo route tối ưu” để yêu cầu tạo route dựa trên sức chứa hiện tại của các MCP trong lộ trình. Nếu thành công, chuyển sang trang Route Planner. Route Planner hiển thị bản đồ được tích hợp đánh dấu các MCP có thể đi qua theo thứ tự, hiển thị quãng đường ước lượng, danh sách địa chỉ từng MCP được chọn.



Tìm kiếm Quay lại

Khoa Nguyen Khoa Nguyen ...

Ca làm việc

Nhân viên

Phương tiện

Điểm thu gom

← Quay lại

Route Planner Gửi tài xế

10.5 km

Map data ©2019 Google Terms of Use Report a map error

- Tại đây, Back Officer có thể gửi đề xuất route tối ưu này cho tài xế của nhiệm vụ.
Nếu gửi thành công, hiện thông báo “Gửi tài xế thành công”



Khoa Nguyen
@khoanguyendanganh

Ca làm việc

Nhân viên

Phương tiện

Điểm thu gom

Q

Khoa Nguen

Gửi tài xế thành công

← Quay lại

Route Planner

Gửi tài xế

Báo cáo bài tập lớn - Công nghệ phần mềm (CO3001) - HK221

Trang 119/130

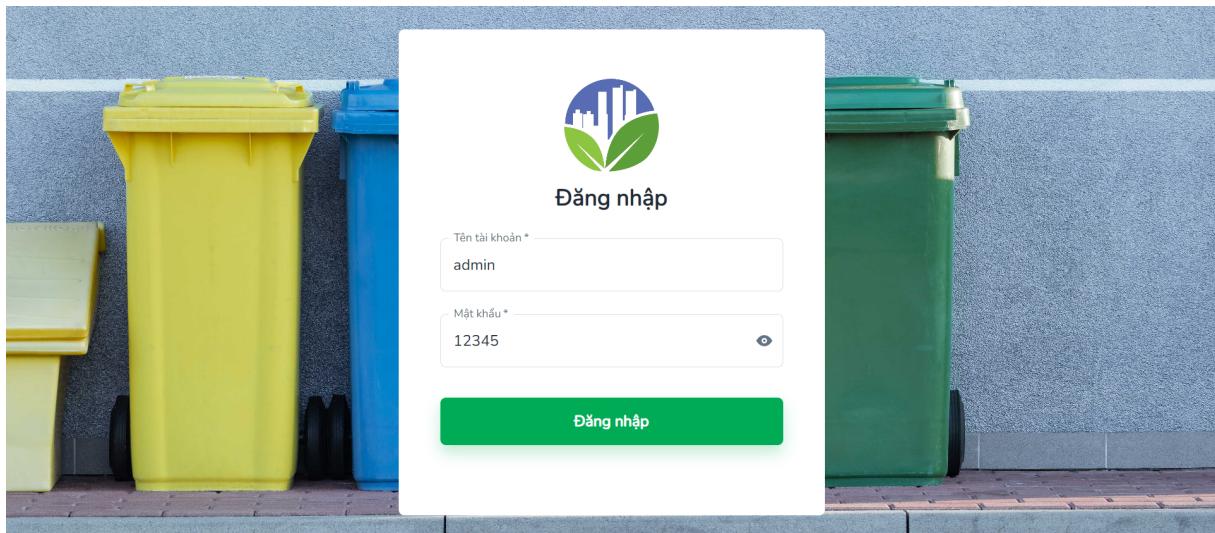
PHẦN V

IMPLEMENTATION - SPRINT 2

1 Implement MVP2 - Hiện thực thiết kế giao diện

Link web: <http://anduckhmt146.me/uwc-bit-by-bit-deploy/>

Screenflow của các giao diện đã được thể hiện ở Sprint 1 Task 4 (link figma). Trong báo cáo Sprint này, nhóm xin được liệt kê lại hình ảnh giao diện được hiện thực từ thiết kế wireframe.



Hình 18: Hiện thực giao diện đăng nhập

Giao diện quản lý ca cung cấp các chức năng xem danh sách, thêm, xóa, sửa, lọc ca theo ngày hoặc theo trạng thái, sắp xếp ca theo trường yêu cầu, điều hướng đến một ca cụ thể.



The screenshot shows a software application interface for managing shift schedules. On the left, there's a sidebar with a user profile picture and the name 'Khoa Dang @khoanguyendanganh'. Below the profile are four menu items: 'Ca và nhiệm vụ' (selected), 'Nhân viên', 'Phương tiện', and 'Điểm thu gom'. The main area is titled 'Danh sách ca' (Shift List). At the top right is a green button '+ Tạo ca mới' (Create new shift). Below the title is a 'Filter' section with two dropdown menus: 'Ngày' (Day) and 'Trạng thái' (Status). A large table lists shifts with columns: Ngày (Day), Bắt đầu (Start time), Kết thúc (End time), Trạng thái (Status), Số nhiệm vụ (Number of tasks), and Xong (Completed). The table contains three rows:

Ngày	Bắt đầu	Kết thúc	Trạng thái	Số nhiệm vụ	Xong
2022-11-28	06:00	08:00	Đang diễn ra	3	2
2022-11-28	14:00	16:00	Chưa diễn ra	3	0
2022-11-29	08:00	10:00	Đang diễn ra	3	1

Hình 19: Hiện thực giao diện quản lý danh sách ca làm việc

This screenshot shows the same software interface as Figure 19, but with a modal dialog box in the center titled 'Tạo ca mới' (Create new shift). The dialog has two input fields: 'Giờ bắt đầu' (Start time) and 'Ngày' (Day), each with a calendar icon. Below these are two buttons: a green 'Xác nhận' (Confirm) and a red 'Đặt lại' (Reset). The background of the main interface is dimmed.

Hình 20: Dialog chứa biểu mẫu tạo ca mới



The screenshot shows a software interface for managing shifts and tasks. On the left, there's a sidebar with a logo and the user's profile picture. The main area has a header "Danh sách ca" (Shift List) and a green button "+ Tạo ca mới" (Create new shift). Below the header is a "Filter" section with dropdowns for "Ngày" (Day) and "Trạng thái" (Status), and a "Filter" button. A table below lists shifts with columns: Ngày (Day), Bắt đầu (Start), Kết thúc (End), Trạng thái (Status), Số nhiệm vụ (Number of tasks), and Xong (Completed). One row is highlighted in yellow with the status "Chưa diễn ra" (Not yet started). At the bottom right of the table, there are buttons for "Nhiệm vụ" (Tasks) and more options. A green toast message at the bottom left says "Thao tác thành công!" (Operation successful!).

Hình 21: Hệ thống thông báo khi tạo ca thành công

This screenshot shows the same software interface as above, but with a modal dialog box in the center titled "Chỉnh sửa ca" (Edit shift). The dialog contains fields for "Giờ bắt đầu" (Start time) set to 06:00 and "Ngày" (Day) set to 2022/11/28, and "Giờ kết thúc" (End time) set to 08:00 and "Trạng thái" (Status) set to "Đang diễn ra" (In progress). There are "Xác nhận" (Confirm) and "Đặt lại" (Reset) buttons. The background shows the shift list with one shift currently selected.

Hình 22: Dialog chứa biểu mẫu chỉnh sửa ca

Giao diện quản lý nhiệm vụ của ca (chi tiết ca) cung cấp các chức năng xem danh sách nhiệm vụ và thông báo tiến độ các nhiệm vụ trong ca, thêm, xóa, sửa, lọc nhiệm vụ theo trạng thái hoặc tên tài xế, sắp xếp nhiệm vụ theo trường yêu cầu, điều hướng đến một nhiệm vụ cụ thể, xác nhận cho các yêu cầu cập nhật tiến độ.



The screenshot shows a software interface for managing tasks during a shift. On the left, there's a sidebar with a user profile (Khoa Dang) and navigation links: Ca và nhiệm vụ (selected), Nhân viên, Phương tiện, and Điểm thu gom. The main area has a search bar and a card for 'Thông tin ca #40001' with details: Ngày: 2022-11-28, Giờ bắt đầu: 06:00, Giờ kết thúc: 08:00, and a progress bar at 2/3. To the right, there's a 'Thông báo tiến độ' section with two notifications: '1 phút trước #50001' (Nhiệm vụ cập nhật đã thu gom) with a 'Xác nhận' button, and '2 phút trước #50002' (Nhiệm vụ cập nhật đã vận chuyển) with a 'Xác nhận' button. Below these are buttons for 'Danh sách nhiệm vụ' and 'Phản nhiệm vụ mới'.

Hình 23: Hiện thực giao diện quản lý danh sách nhiệm vụ và quản lý tiến độ nhiệm vụ trong ca - 1

The screenshot shows a task list interface. The sidebar includes a user profile (Khoa Dang) and navigation links: Ca và nhiệm vụ (selected), Nhân viên, Phương tiện, and Điểm thu gom. The main area features a filter section with 'Tên tài xế' and 'Trạng thái' dropdowns, and a 'Filter' button. Below is a table listing four tasks:

ID nhiệm vụ ↑	Trạng thái	Mã phương tiện	Tài xế	Mã lô trình	Chi tiết	⋮
50001	Đang thu gom	SU-20-0001	Trần Phúc Anh	Q716	<button>Chi tiết</button>	<button>⋮</button>
50002	Vận chuyển	TO-19-0002	Nguyễn Đức An	Q137	<button>Chi tiết</button>	<button>⋮</button>
50003	Đang thu gom	HY-20-0003	Võ Thái Toàn	Q457	<button>Chi tiết</button>	<button>⋮</button>
50004	Chưa bắt đầu	SU-21-0004	Nguyễn Quang Huy	Q689	<button>Chi tiết</button>	<button>⋮</button>

At the bottom, there are buttons for 'Rows per page: 5' and '1-4 of 4'.

Hình 24: Hiện thực giao diện quản lý danh sách nhiệm vụ và quản lý tiến độ nhiệm vụ trong ca - 2



← Quay lại

① Phương tiện và tài xế ② Người thu gom

Chọn phương tiện:

Mã phương tiện: SU-20-0001

Chọn tài xế:

Tên tài xế: Cao Trần Anh Khoa

Cao Trần Anh Khoa
ID tài xế: 30005
Loại bằng lái: E

Ca và nhiệm vụ
Nhân viên
Phương tiện
Điểm thu gom

Khoa Dang
@khoanguyendanganh

Hình 25: Hiện thực giao diện phân nhiệm vụ mới - 1

← Quay lại

✓ Phương tiện và tài xế ② Người thu gom

Danh sách Điểm thu gom trong lộ trình tháng của phương tiện

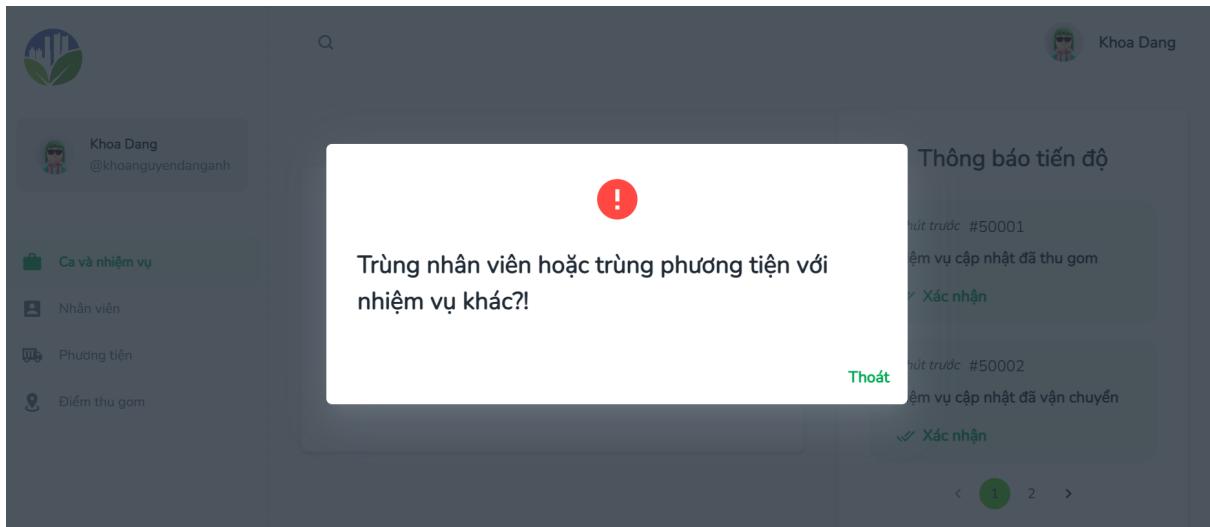
Q716

MCP ID	Địa điểm	Nhóm trưởng	Thành viên
60001	167 Phạm Hữu Lầu Phường Mỹ Phước Quận 7 TP. HCM	Nguyễn Quang Ân	Nguyễn Quang Bình Nguyễn Quang Anh
60002	69 Nguyễn Khắc Nhu Phường Cô Giang Quận 1 Tp. HCM	Nguyễn Đức Duy	Nguyễn Quang Cường

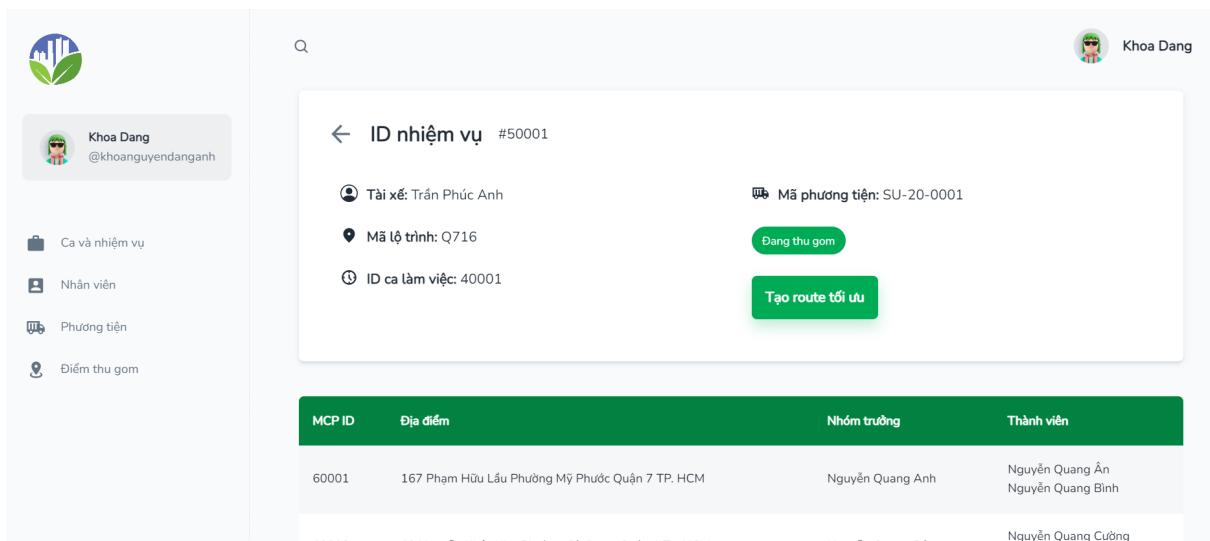
Ca và nhiệm vụ
Nhân viên
Phương tiện
Điểm thu gom

Khoa Dang
@khoanguyendanganh

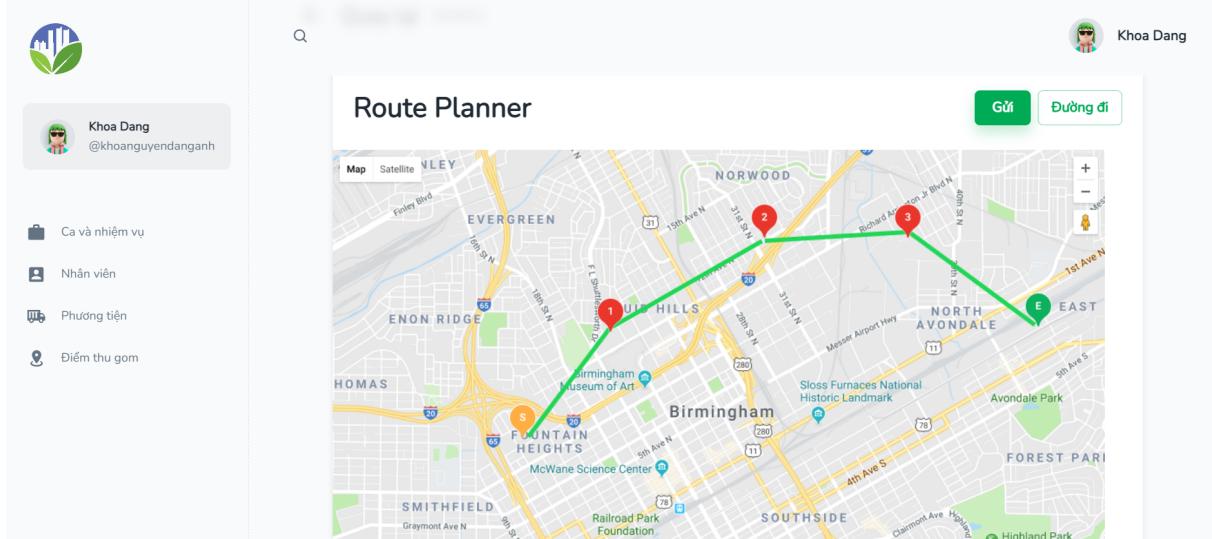
Hình 26: Hiện thực giao diện phân nhiệm vụ mới - 2



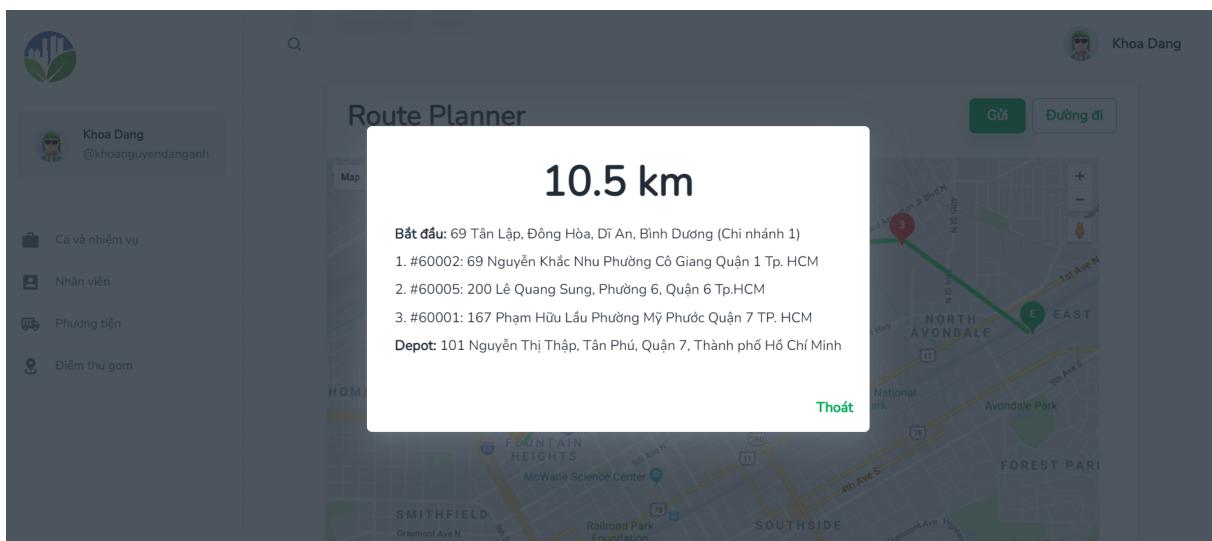
Hình 27: Hệ thống thông báo khi phân nhiệm vụ mới trùng nhân sự hoặc phương tiện với nhiệm vụ đã tạo cùng ca



Hình 28: Hiện thực giao diện xem chi tiết nhiệm vụ



Hình 29: Hiện thực giao diện hiển thị lộ trình tối ưu



Hình 30: Hệ thống hiển thị để xuất lộ trình dạng mô tả

2 Thuyết trình dự án

- Trello quản lý tiến độ công việc: <https://trello.com/w/uwc20project>
- Video thuyết trình: <https://youtu.be/632NU-4F7sI>
- Slide thuyết trình:
https://www.canva.com/design/DAFUVFgdJdQ/ti0HR0aUi2rXwMgRiaj0jQ/view?utm_content=DAFUVFgdJdQ&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink
- Link PDF Slide:



[https://drive.google.com/file/d/1cRxJR2XjS6vZ-7VERHxAX5ZM9hFn9wZr/
view?usp=sharing](https://drive.google.com/file/d/1cRxJR2XjS6vZ-7VERHxAX5ZM9hFn9wZr/view?usp=sharing)



PHẦN VI

TỔNG KẾT

Trong quá trình hiện thực bài tập lớn, nhóm đã thực hiện các giai đoạn trong quá trình phát triển một phần mềm, từ mô tả và xác định các yêu cầu của dự án đến việc hiện thực giao diện và những chức năng chính của một hệ thống hỗ trợ xử lý chất thải trong thành phố. Cụ thể hơn, nhóm đã xác định ngữ cảnh, quy trình nghiệp vụ, tầm vực của dự án và các yêu cầu từ người dùng, hệ thống, các yêu cầu chức năng và phi chức năng, sau đó tiếp tục đi vào mô hình hóa hệ thống và thiết kế kiến trúc.

Để làm rõ các thay đổi của hệ thống trong quá trình vận hành và mô tả cấu trúc cũng như nắm được quy trình xử lý chức năng, nhóm đã vẽ những UML Diagram như Use-case Diagram, Activity Diagram, Sequence Diagram, Class Diagram, Component Diagram và Deployment Diagram.

Tính đến nay, hệ thống mà nhóm hiện thực cũng đã tương đối hoàn chỉnh và đã được host để có thể truy cập trực tuyến. Phần hiện thực của nhóm đã đáp ứng cơ bản những chức năng cần có của một Back Officer đã được liệt kê ban đầu như quản lý ca, quản lý nhiệm vụ, xem chi tiết nhiệm vụ và tạo route tối ưu, cũng như đáp ứng một số yêu cầu phi chức năng như giao diện thân thiện, dễ sử dụng, xử lý dữ liệu thời gian thực,...

Bên cạnh những điểm đạt được, nhóm em cũng tự nhận thấy còn nhiều yếu điểm mà nhóm cần phải cải thiện, chủ yếu là trong khâu làm việc nhóm. Vì lượng công việc nhiều, kéo theo việc phân công công việc còn chưa đạt hiệu quả cao khiến nhóm gặp phải nhiều khó khăn, thách thức. Những vấn đề này đã phần nào ảnh hưởng đến chất lượng bài làm của nhóm tại thời điểm đó, thậm chí tạo ra mâu thuẫn giữa các thành viên. Tuy nhiên, nhóm đã không bỏ cuộc mà bình tĩnh tìm hướng giải quyết, tổ chức những lần họp mặt để cùng nhau tìm ra tiếng nói chung, giúp cho việc phối hợp cùng nhau trở nên ăn ý và gắn kết với nhau hơn.

Nhìn về mặt tích cực hơn thì các thành viên trong nhóm đã có thêm nhiều kinh nghiệm thông qua bài tập lớn lần này. Đó là kinh nghiệm làm việc nhóm, kinh nghiệm thuyết trình, kinh nghiệm trong việc giải quyết tình huống cũng như quản lý thời gian sao cho hiệu quả hơn. Và quan trọng không kém chính là kỹ năng làm việc nhóm để cùng nhau hoàn thiện một phần mềm. Có thể nói, đề tài này đã mang lại cho nhóm rất nhiều kiến thức chuyên ngành



quan trọng về quá trình phát triển một phần mềm, từ việc sử dụng UML Diagram sao cho hợp lý cho đến cách dùng github và tận dụng các công cụ như ReactJS, Redux và Material UI để hoàn thiện dự án.

Cuối cùng, nhóm Bit-By-Bit xin được gửi lời cảm ơn chân thành đến các thầy đã nhiệt tình hướng dẫn nhóm trong suốt quá trình thực hiện bài tập lớn lần này. Nhờ vào việc được các thầy góp ý, trả lời câu hỏi cũng như tổ chức những buổi sửa bài sau mỗi phần, bài làm của nhóm đã được cải thiện và ngày càng trở nên hoàn chỉnh hơn.



PHẦN VII

TÀI LIỆU THAM KHẢO

1. Learning UML 2.0, O'Reilly, Russ Miles Kim Hamilton, 2006.
2. Fundamentals of Software Architecture - An Engineering Approach, O'Reilly, Mark Richards, Neal Ford, 2020.