

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



CÔNG NGHỆ PHẦN MỀM (CO3001)

ASSIGNMENT

Urban waste collection aid - UWC 2.0

GVHD:	Lê Đình Thuận	
SV thực hiện:	Nguyễn Hữu Hùng	2013364 - L04
	Ngô Vũ Anh Khoa	2011423 - L04
	Bùi Tiến Lộc	2013678 - L04
	Lê Trần Phúc Lộc	2013684 - L03
	Nguyễn Hữu Lượng	2013724 - L04
	Đậu Xuân Thành	2014486 - L04

Mục lục

1 Task 1: Requirement elicitation	2
1.1 Identify the context of this project. Who are relevant stakeholders? What are their current needs? What could be their current problem? In your opinion, what benefits UWC 2.0 will be for each stakeholder?	2
1.2 Describe all functional and non-functional requirements that can be inferred from the project description. Draw a use-case diagram for the whole system	3
1.3 For the Task assignment module, draw its use-case diagram and describe the use-case using a table format	5
2 Task 2: System modelling	23
2.1 Draw an activity diagram to capture the business process between systems and the stakeholders in Task Assignment module	23
2.2 Proposal a conceptual solution for the route planning task and draw a sequence diagram to illustrate it	25
2.3 Draw a class diagram of Task Assignment module as comprehensive as possible	26
3 Task 3: Architecture design	29
3.1 Describe an architectural approach you will use to implement the desired system. How many modules you plan for the whole WMC 2.0 system? Briefly describe input, output and function of each module	29
3.2 Draw an implementation diagram for Task Assignment module	42
4 Task 4: Implementation – Sprint 1	46
4.1 Setting up. The team creates an online repository (github, bitbucket, etc) for version control. folders this stage, no need for a database to store all menu items, customers, etc. Data can be hard coded in code files.	46
4.2 Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files	47
4.3 Implement MVP1 – design an interface of either a Desktop-view central dashboard for Task Management for back-officers OR a Mobile-view Task assignment for Janitors and Collectors. Decide yourself what to include in the view. Design use a wireframe tool. . . .	47

1 Task 1: Requirement elicitation

1.1 Identify the context of this project. Who are relevant stakeholders? What are their current needs? What could be their current problem? In your opinion, what benefits UWC 2.0 will be for each stakeholder?

Xác định bối cảnh của dự án

Dựa trên tình hình thực tế thu gom rác thải tại thành phố Hồ Chí Minh, khảo sát hình thức thu gom rác thải bằng phương tiện chạy theo lịch trình thu trước tại từng điểm thu và hình thức thu theo từng hộ gia đình. Theo kết quả khảo sát thì đa số người dân chọn hình thức thu gom rác thải bằng phương tiện chạy theo lịch trình thu trước tại từng điểm thu. Lý do được giải thích bởi vì tại một địa điểm thu gom nhất định đem lại nhiều lợi ích so với hình thức thu gom rác theo từng hộ gia đình. Thứ nhất, giảm thiểu được chi phí nhân công so với thu gom rác thải thủ công tại từng hộ gia đình. Thứ hai, giảm thiểu được thời gian thu gom và chi phí nhiên liệu khi di chuyển theo lịch trình đã được tính toán trước. Thứ ba, giúp cho người dân thuận tiện hơn trong việc đưa rác thải ra địa điểm thu gom. Nhà cung cấp dịch vụ hiện đang cung cấp dịch vụ dọn rác theo hình thức trên với một hệ thống quản lý thông tin UWC 1.0. Tuy nhiên hệ thống UWC 1.0 hiện tại có hiệu suất thu gom chưa cao. Vì vậy nhà cung cấp dịch vụ Y ký hợp đồng với tổ chức X để phát triển một hệ thống quản lý thông tin gọi là UWC 2.0 với mục đích để cải thiện hiệu năng thu gom rác thải.

Xác định các bên liên quan

- Back officer
- Collector
- Janitor

Nhu cầu hiện tại của họ

Phát triển một hệ thống quản lý thông tin gọi là UWC 2.0 với mục đích để cải thiện hiệu năng thu gom rác thải.

- Back officer: Cần có cái nhìn tổng quan về lịch làm việc của collectors, janitors, phân công công việc một cách dễ dàng. Có thể kiểm soát, điều phối các phương tiện, nhân công.
- Collectors, janitors: Cần biết tổng quan lịch làm việc mình và các phương tiện nếu cần sử dụng.

Các vấn đề họ gặp phải

- Collectors và janitors làm việc quá sức vì lịch làm việc không hợp lý.
- Không biết được khi nào thì các điểm thu thập chính (MCPs) được nạp đầy.
- Chi phí di chuyển giữa các tuyến đường vượt qua ngân sách dự kiến.
- Không có phương thức giao tiếp hiệu quả giữa back officers, janitors và collectors.

Theo quan điểm của nhóm, UWC 2.0 sẽ mang lại những lợi ích sau đây

- Back officers có thể điều phối collectors và janitors một cách hợp lý với các tuyến đường đã được tối ưu hóa.
- Các điểm thu gom chính luôn được xử lý khi dung tích đầy. Điều này giúp cải thiện môi trường sống cho cư dân.
- UWC 2.0 giảm chi phí kinh doanh cho nhà cung cấp dịch vụ Y.
- UWC 2.0 cải thiện các hạn chế so với UWC 1.0
 - Cải thiện tốc độ thao tác với giao diện, tối ưu hóa giao diện thân thiện với người dùng hơn.
 - Bổ sung thêm các tính năng như: Ngôn ngữ Tiếng Anh,
 - Tối ưu hóa thuật toán sắp xếp lịch trình di chuyển, thu gom.
 - Bổ sung thêm khả năng sử dụng trên đa nền tảng (Bổ sung giao diện sử dụng trên thiết bị di động.).

1.2 Describe all functional and non-functional requirements that can be inferred from the project description. Draw a use-case diagram for the whole system

Các yêu cầu chức năng

Back officer

- Có cái nhìn tổng quan về janitors và collectors và lịch làm việc của họ.
- Có cái nhìn tổng quan về phương tiện di chuyển và các chi tiết kỹ thuật của nó (cân nặng, sức chứa, mức tiêu hao nhiên liệu, ...)
- Có cái nhìn tổng quan về tất cả các điểm thu thập chính (MCPs) và thông tin về sức chứa của chúng.
- Có thể chỉ định các phương tiện cho janitors và collectors.
- Có thể chỉ định janitors và collectors cho MCPs (công việc).
- Có thể tạo một tuyến đường cho mỗi collector.
- Có khả năng nhắn tin cho collectors và janitors.
- Có thể tạo/chỉnh sửa lịch hàng tuần của hệ thống.
- Có thể quan sát trạng thái hoạt động/không hoạt động của janitors và collectors.

Collector và janitor

- Có cái nhìn tổng quan về lịch làm việc của họ
- Có chi tiết về công việc hàng ngày và hằng tuần của họ.
- Có khả năng giao tiếp với collectors, các janitors khác và back officers.
- Có thể check in/check out công việc hằng ngày.
- Có thể được thông báo nếu các MCPs đầy.
- Có thể xin nghỉ việc (khi xin nghỉ việc thì back officers được thông báo)
- Có thể nhập số lít xăng (nguyên liệu sử dụng) trong 1 task làm việc (để báo cáo cho back-officer).

Các yêu cầu phi chức năng

Yêu cầu về sản phẩm

- Thông tin về sức chứa của các MCPs sẽ được cập nhật từ MCPs mỗi 15 phút trong tối thiểu 95% thời gian vận hành của nó.
- Các tin nhắn giữa các collectors, các janitors khác và các back officers sẽ được chuyển tiếp trong thời gian thực với độ trễ thấp hơn 1 giây. WC 2.0 được kỳ vọng có thể nhập và sử dụng các dữ liệu có sẵn từ UWC1.0.
- Hệ thống phải sẵn sàng để sử dụng trong thời gian làm việc của Back officers, Collectors và janitors (Từ 7h-22h từ thứ hai đến chủ nhật).
- Tốc độ phản hồi phải dưới 1 giây cho từng thao tác.
- Ứng dụng dùng ít hơn 500MB bộ nhớ RAM trong suốt phiên làm việc.
- Thời gian hệ thống sập trong thời gian làm việc phải ít hơn 5 giây cho mỗi lần.
- Thời gian khởi động lại hệ thống không quá 1 phút.
- Quá trình gửi dữ liệu giữa các người sử dụng không quá 3 giây.

- Hệ thống có khả năng xử lý dữ liệu trong thời gian thực từ ít nhất 1000 MCPs ở hiện tại và 10.000 MCPs trong 5 năm.
- Tuyến đường được tối ưu hóa về mặt tiêu thụ nhiên liệu và khoảng cách di chuyển.
- Back officer, collector, janitor có thể thành thạo sử dụng các hoạt động hệ thống sau khi được hướng dẫn 15 phút.

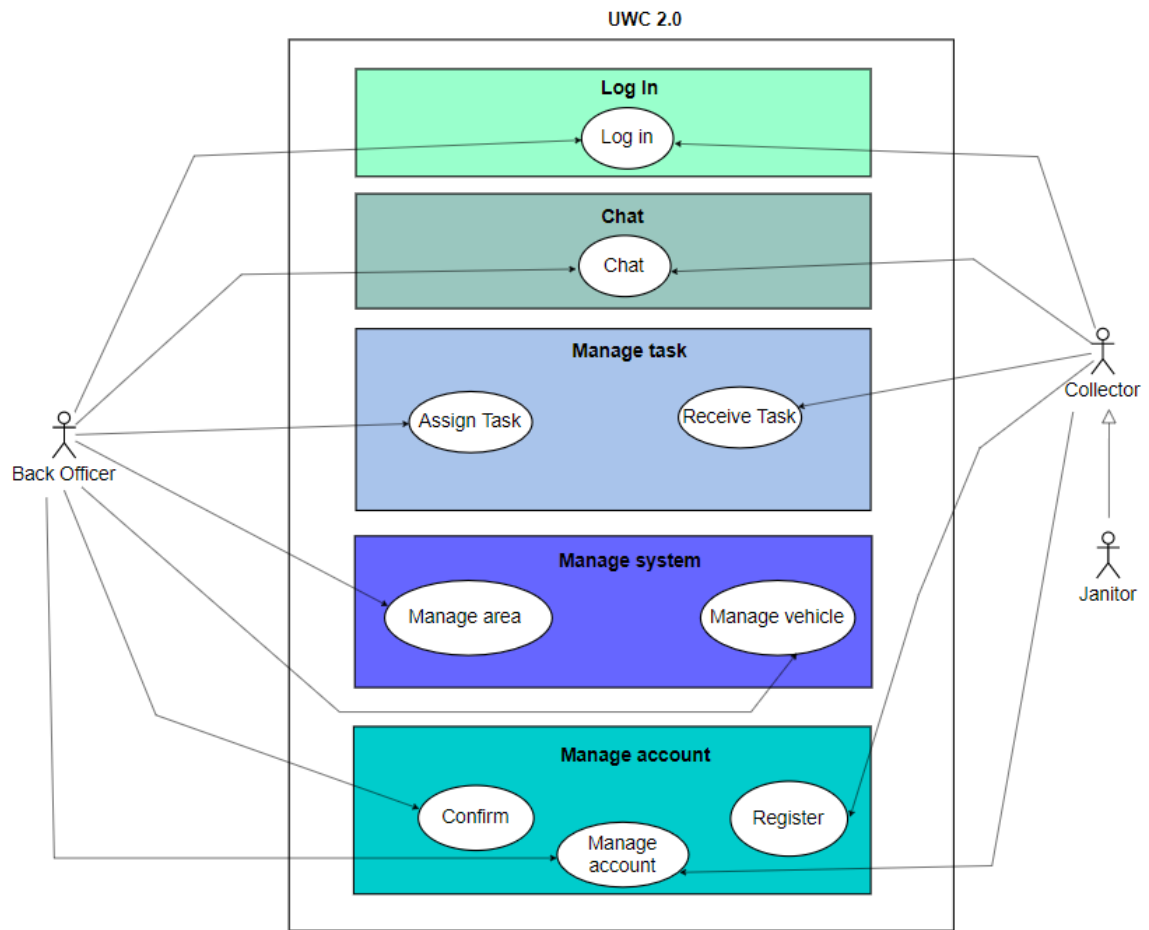
Các yêu cầu tổ chức

- Không có các tác vụ trùng nhau (Mỗi người chỉ có thể thực hiện 1 tác vụ trong 1 thời điểm => đưa ra cảnh cáo khi thời gian bị trùng)
- Tất cả thông tin về các tác vụ của collectors và janitors trong ngày và tuần sẽ được hiển thị trên một trang (không trượt xuống) - giống như Google calendar
- Giao diện hệ thống UWC 2.0 sử dụng Tiếng Việt, có khả năng chuyển đổi qua Tiếng Anh trong tương lai.
- Hệ thống có thể được sử dụng một cách hiệu quả trên điện thoại di động (Android, IOS), máy tính bảng hoặc máy tính xách tay (Windows, Linux, Mac) với trình duyệt (Chrome, Firefox, Safari, Opera)

Các yêu cầu bên ngoài

- Quản lý tác vụ của MCPs có thể hoạt động được với UWC 1.0 nhiều nhất có thể.
- Phần mềm bảo đảm thông tin của khách hàng được bảo mật trước khi các khách hàng khác, nhân viên, quản lý của nhà cung cấp dịch vụ Y, người khác tham gia vào phát triển phần mềm của X và Y.
- Phần mềm phải tuân theo luật địa phương và các điều lệ nơi mà ứng dụng hoạt động.

Sơ đồ use-case cho toàn bộ hệ thống

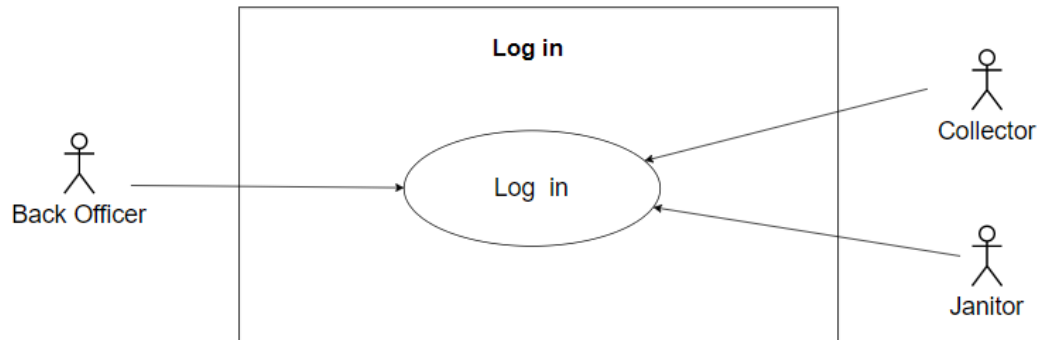


Bảng mô tả các usecase chính của hệ thống

Use case ID	Tên use case	Mô tả
1	Log in	Đăng nhập vào ứng dụng
2	Chat	Gửi, nhận tin nhắn giữa Back officer, Collector, Janitor
3	Assign Task	Quản lý công việc dưới góc nhìn của Back officer
4	Recieve Task	Quản lý công việc dưới góc nhìn của Collector, Janitor
5	Manage area	Quản lý tuyến đường và MCPS
6	Manage verhicle	Quản lý phương tiện
7	Register	Đăng ký tài khoản
8	Confirm	Xác nhận đăng ký tài khoản thành công
9	Manage account	Các hoạt động chỉnh sửa, xóa, xem thông tin tài khoản

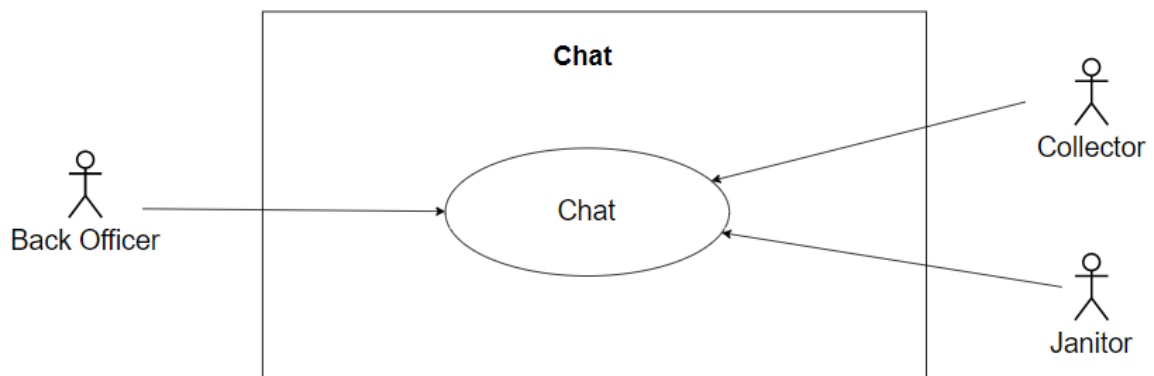
1.3 For the Task assignment module, draw its use-case diagram and describe the use-case using a table format

a) Tính năng đăng nhập (Log in)



Use-case name	Log in
Actor	Back officer, Collector, Janitor
Description	Người dùng muốn đăng nhập vào hệ thống
Preconditions	Không
Postconditions	Người dùng đăng nhập được vào hệ thống
Normal flow	1. Hệ thống cung cấp giao diện để người dùng cung cấp thông tin gồm 2 trường là tài khoản và mật khẩu 2. Người dùng nhập thông tin đăng nhập 3. Hệ thống kiểm tra thông tin đăng nhập 4. Hệ thống phản hồi người dùng. Nếu tài khoản và mật khẩu đúng thì hệ thống hiện thị giao diện. Nếu sai thì hệ thống sẽ hiển thị thông báo về lỗi sai
Exceptions	Người dùng nhập sai hoặc thiếu thông tin đăng nhập
Alternative flows	Không

b) Tính năng tương tác (Chat)



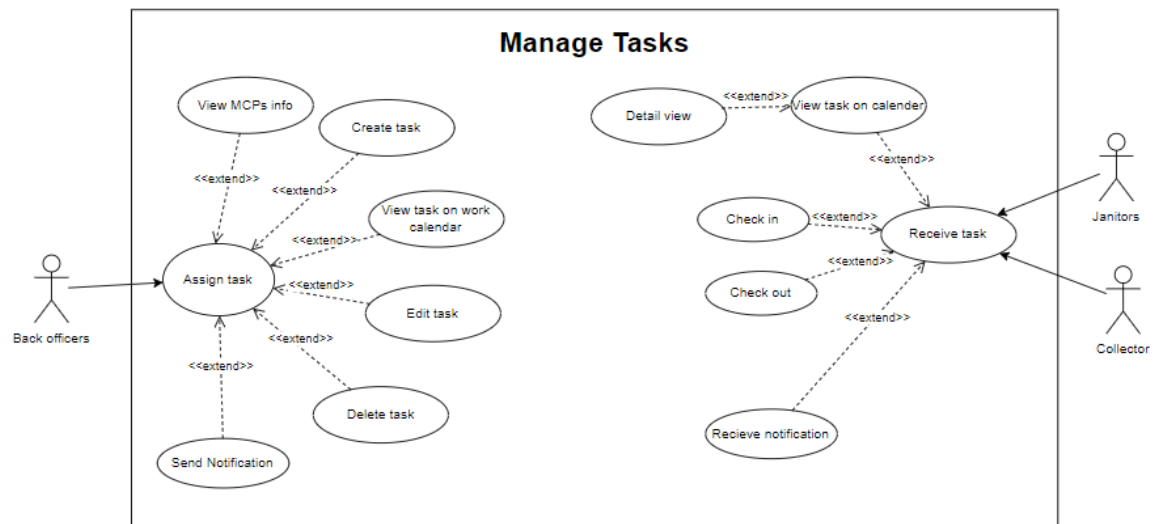
Nhận tin nhắn (Receive message)

Use-case name	Receive Message
Actor	Back officer, Collectors, Janitors
Description	Người dùng nhận được tin nhắn gửi tới mình
Preconditions	Người dùng phải đăng nhập vào hệ thống
Postconditions	Người dùng nhận được tin nhắn
Normal flow	<ol style="list-style-type: none"> 1. Khi người nào đó gửi tin nhắn tới người dùng, người dùng nhận được thông báo trên góc phải màn hình kèm âm thanh thông báo tin nhắn. 2. Người dùng nhấp vào thông báo sẽ chuyển tới hộp thoại nhận tin nhắn và hệ thống sẽ xác nhận tin nhắn đã được đọc
Exceptions	Không
Alternative flows	Không

Gửi tin nhắn (Send message)

Use-case name	Receive Message
Actor	Back officer, Collectors, Janitors
Description	Người dùng muốn gửi tin nhắn
Preconditions	Người dùng phải đăng nhập vào hệ thống
Postconditions	Người dùng gửi được tin nhắn
Normal flow	<ol style="list-style-type: none"> 1. Người dùng nhập tin nhắn và bấm gửi. Hệ thống sẽ gửi thông báo tới người nhận và đánh dấu nếu gửi thành công 2. Riêng Backofficer có khả năng gửi tin nhắn tới nhiều người
Exceptions	Không
Alternative flows	Không

c) Tính năng quản lý công việc (Manage Task)



Gán công việc (Assign Task)

Use-case name	Create task
Actor	Back officer
Description	Back-officer muốn tạo task
Preconditions	Người dùng phải đăng nhập vào hệ thống và người dùng phải là Back-officer
Postconditions	Back officer tạo công việc thành công cho nhân viên(collectors và janitors)
Normal flow	1. Back-officer tạo công việc dựa trên work-calendar để phân công việc theo thời gian hợp lý. 2. Hệ thống sẽ lưu lại công việc vừa tạo trên calendar và thông báo với Back-officer đã tạo thành công, đồng thời gửi thông báo về cho nhân viên tương ứng với công việc vừa tạo.
Exceptions	Không
Alternative flows	Không

Use-case name	Edit task
Actor	Back officer
Description	Back-officer muốn sửa task đã tạo
Preconditions	Người dùng phải đăng nhập vào hệ thống và người dùng phải là Back-officer
Postconditions	Back officer sửa thành công task của nhân viên (collectors và janitors)
Normal flow	1. Back-officer sửa công việc dựa trên work-calendar để phân công việc theo thời gian hợp lý. 2. Hệ thống sẽ lưu lại công việc vừa sửa trên calendar và thông báo sửa task thành công, đồng thời gửi thông báo về cho nhân viên tương ứng với công việc vừa sửa.
Exceptions	Không
Alternative flows	Không

Use-case name	Delete Task
Actor	Back officer, Collector, Janitor
Description	Back-officer muốn xóa task đã tạo
Preconditions	Người dùng phải đăng nhập vào hệ thống và người dùng phải là Back-officer
Postconditions	Người dùng xóa thành công công việc của nhân viên (collectors và janitors)
Normal flow	1. Back-officer xóa công việc trên work calendar. 2. Hệ thống sẽ xóa task trên calendar và thông báo nếu xóa task thành công.
Exceptions	Không
Alternative flows	1.1. Hệ thống yêu cầu xác nhận chắc chắn muốn xóa task. 1.2. Nếu người dùng chọn "Yes" thực hiện bước 2. 1.3. Nếu người dùng chọn "No" thì hệ thống không làm gì.

Use-case name	View Task
Actor	Back officer
Description	Người dùng quan sát tổng quan lịch làm việc của mình
Preconditions	Người dùng phải đăng nhập vào hệ thống.
Postconditions	Người dùng có thể đọc tất cả công việc trên work-calendar.
Normal flow	1. Sau các thao tác thêm sửa xóa của Back Officer, end-user có thể view ngắn gọn mình làm việc vào khung giờ nào ở trên work-calendar. 2. Người dùng có thể chọn chức năng xem chi tiết (đọc thêm ở bảng view detail)
Exceptions	Không
Alternative flows	Không

Use-case name	View work calendar
Actor	Management, Back officer
Description	Xem lịch công việc
Preconditions	Người dùng phải đăng nhập vào hệ thống và thuộc nhóm Manage hoặc Back officer
Normal flow	1. Hệ thống hiển thị giao diện lịch các công việc hiện tại.
Exceptions	Không
Alternative flows	Không.

Receive task

Use-case name	View task on calender
Actor	Collector, Janitor
Description	Collector, Janitor muốn xem thông tin về lịch trình công việc
Preconditions	Collector, Janitor đăng nhập vào hệ thống
Postconditions	Collector, Janitor biết được thông tin về công việc của mình
Normal flow	1. Hệ thống cung cấp các thông tin ngắn gọn về lịch làm việc trên work calendar 2. Người dùng có thể chọn chức năng xem chi tiết(đọc thêm ở bảng view detail)
Exceptions	Không
Alternative flows	Không

Use-case name	Detail view
Actor	Collector, Janitor
Description	Collector, Janitor muốn xem thông tin chi tiết về lịch trình công việc
Preconditions	Collector, Janitor đăng nhập vào hệ thống
Postconditions	Collector, Janitor biết được thông tin chi tiết về công việc của mình
Normal flow	1. Hệ thống cung cấp các thông tin chi tiết về lịch làm việc bao gồm: ca làm, công sự, địa điểm, phương tiện sử dụng
Exceptions	Không
Alternative flows	Không

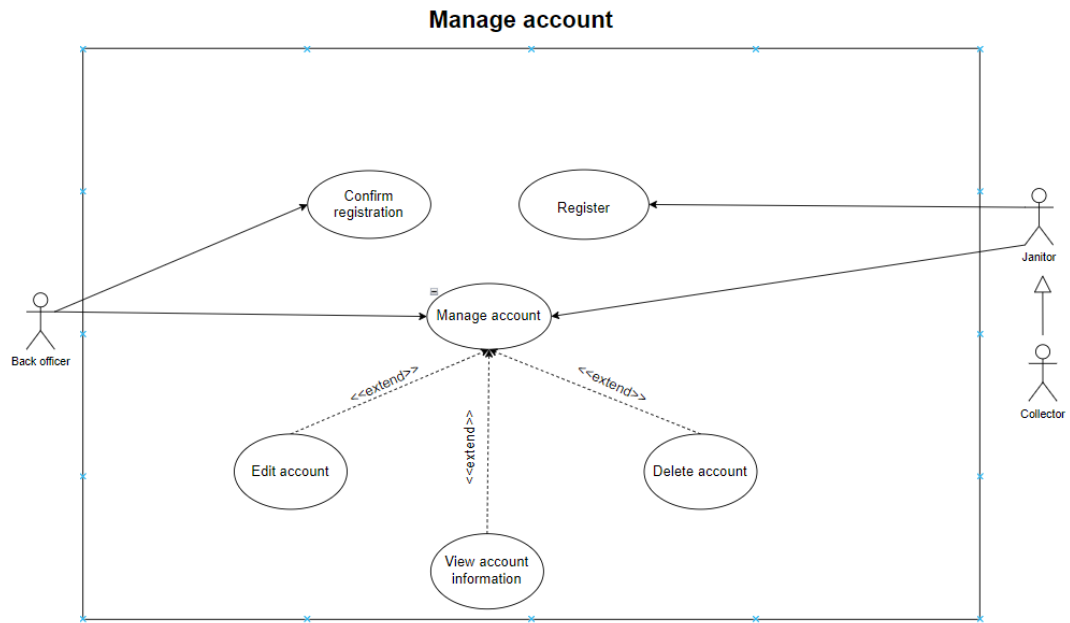
Use-case name	Check in
Actor	Collector, Janitor
Description	Collector, Janitor muốn báo cáo tiến độ công việc cho Back officer
Preconditions	Collector, Janitor đăng nhập vào hệ thống
Postconditions	Back officer biết được tiến độ công việc của Collector, Janitor
Normal flow	<ol style="list-style-type: none"> 1. Hệ thống cung cấp giao diện dạng tin nhắn để người dùng soạn thông tin 2. Người dùng soạn tin nhắn cần gửi 3. Hệ thống gửi tin nhắn đến Back officer 4. Nếu tin nhắn gửi thất bại hệ thống sẽ hiển thị thông báo
Exceptions	Không
Alternative flows	Không

Use-case name	Check out
Actor	Collector, Janitor
Description	Collector, Janitor muốn báo cáo công việc đã hoàn thành cho Back officer
Preconditions	Người dùng đăng nhập vào hệ thống
Postconditions	Back officer biết được các công việc mà Collector, Janitor đã hoàn thành
Normal flow	<ol style="list-style-type: none"> 1. Hệ thống cung cấp giao diện dạng check box bao gồm các công việc của Collector, Janitor 2. Người dùng chọn công việc mà mình đã hoàn thành 3. Hệ thống sẽ cập nhật bảng check box của Back officer 4. Nếu việc cập nhật bị thất bại hệ thống sẽ hiển thị thông báo
Exceptions	Không
Alternative flows	Không

Use-case name	Recieve notification
Actor	Collector, Janitor
Description	Collector, Janitor muốn nhận thông báo từ Back officer khi MCP đầy
Preconditions	Người dùng đăng nhập vào hệ thống
Postconditions	Collector, Janitor xử lý kịp thời khi các MCP đầy
Normal flow	<ol style="list-style-type: none"> 1. Hệ thống sẽ gửi tin nhắn từ Back officer đến Collector, Janitor về MCP đầy 2. Collector và Janitor phản hồi tin nhắn với nội dung có thể đến MCP đầy hay không 3. Hệ thống sẽ gửi tin nhắn phản hồi đến Back officer
Exceptions	Không
Alternative flows	Không

d) Tính năng quản lý tài khoản(Account management)

Register



Use-case name	Register
Actor	Collector, Janitor
Description	Người dùng tạo tài khoản
Preconditions	Người dùng truy cập vào hệ thống
Post-conditions	Người dùng hoàn thành đăng ký tài khoản
Normal flow	1.Hệ thống hiển thị giao diện cho người dùng nhập các thông tin cần thiết 2.Người dùng nhập các thông tin hệ thống yêu cầu 3.Hệ thống xác minh tính hợp lệ của thông tin người dùng cung cấp 4.Hệ thống hiển thị thông tin tài khoản cho người dùng 5.Hệ thống hiển thị giao diện Account management
Exceptions	Người dùng nhập thông tin thiếu hoặc không hợp lệ
Alternative flows	Hiện thông báo yêu cầu người dùng nhập lại thông tin

Confirm registration

Use-case name	Confirm registration
Actor	Back officer
Description	Người dùng xác nhận đăng ký tài khoản
Preconditions	Người dùng truy cập vào hệ thống
Post-conditions	Người dùng hoàn thành xác nhận đăng ký
Normal flow	1.Hệ thống hiển thị giao diện cho người dùng xem các yêu cầu đăng ký mới 2.Hệ thống hiển thị thông tin đăng ký cho người dùng 3.Người dùng xác nhận thông tin đăng ký 4.Hệ thống hiển thị thông tin tài khoản cho người dùng 5.Hệ thống hiển thị giao diện Account management
Exceptions	Không
Alternative flows	Không

Manage account

Use-case name	Manage account
Actor	Back officer, Collector, Janitor
Description	Người dùng quản lý tài khoản
Preconditions	Người dùng đăng nhập vào hệ thống
Post-conditions	Người dùng hoàn thành quản lý thông tin tài khoản
Normal flow	1.Hệ thống hiện thị giao diện các thông tin người dùng có thể quản lý 2.Người dùng lựa chọn tác vụ cần quản lý 3.Hệ thống xác nhận việc quản lý của người dùng 4.Hệ thống trở về hiển thị giao diện Account management
Exceptions	Không
Alternative flows	Tại bước 2: Người dùng chọn chỉnh sửa tài khoản thì tiếp tục với use-case "Edit account" Người dùng chọn xem thông tin tài khoản thì tiếp tục với use-case "View account information" Người dùng chọn xóa tài khoản thì tiếp tục với use-case "Delete account"

Edit account

Use-case name	Edit account
Actor	Back officer, Collector, Janitor
Description	Người dùng chỉnh sửa thông tin tài khoản
Preconditions	Người dùng đăng nhập vào hệ thống
Post-conditions	Thông tin tài khoản được sửa đổi trên hệ thống.
Normal flow	1.Người dùng chọn chức năng chỉnh sửa thông tin tài khoản 2.Hệ thống hiển thị các thông tin người dùng có thể chỉnh sửa 3. Người dùng nhấn xác nhận chỉnh sửa 4.Hệ thống hiện thị xác nhận lại yêu cầu của người dùng 5.Hệ thống chuyển sang giao diện gốc
Exceptions	Không
Alternative flows	Không

View account information

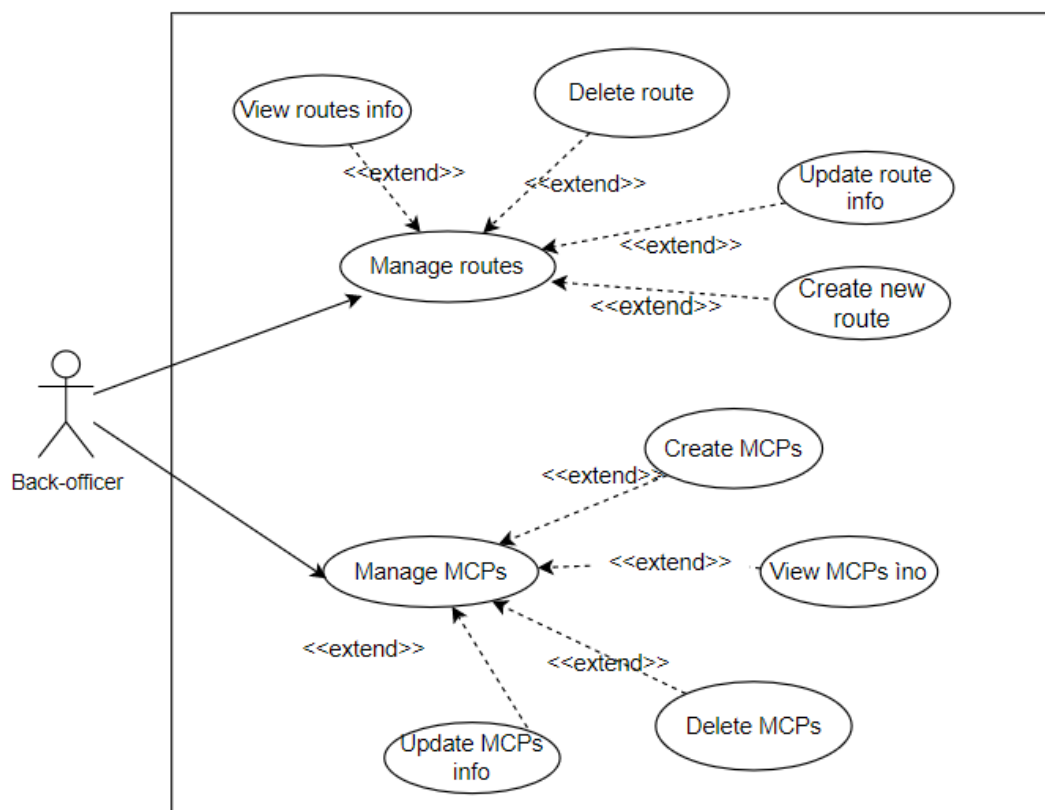
Use-case name	View account information
Actor	Back officer, Collector, Janitor
Description	Người dùng xem thông tin tài khoản
Preconditions	Người dùng đăng nhập vào hệ thống
Post-conditions	Người dùng xem được thông tin tài khoản
Normal flow	1.Người dùng chọn chức năng xem thông tin tài khoản 2.Hệ thống hiện thị thông tin người dùng 3.Hệ thống trở về giao diện gốc
Exceptions	Không
Alternative flows	Không

Delete account

Use-case name	Delete account
Actor	Back officer, Collector, Janitor
Description	Người dùng xóa tài khoản
Preconditions	Người dùng đăng nhập vào hệ thống
Post-conditions	Tài khoản được xóa khỏi hệ thống
Normal flow	1. Người dùng chọn chức năng xóa tài khoản 2. Hệ thống hiện thị xác nhận lại yêu cầu của người dùng 3. Hệ thống chuyển sang giao diện gốc
Exceptions	Không
Alternative flows	Không

e) **Tính năng manage area**

Manage Area



Use case name	Manage Area
Actor	Back-officer
Description	Back-officer muốn quản lý được thông tin của khu vực cần thu gom rác và tuyến đường thu gom đến MCPs của khu vực đó.
Preconditions	Back-officer phải log in vào hệ thống bằng tài khoản được cấp quyền quản lý.
Postconditions	Thông tin về khu vực được quản lý dễ dàng.
Normal flow	<ol style="list-style-type: none"> Hệ thống hiển thị giao diện cho việc quản lý khu vực và đưa ra 4 option: <ul style="list-style-type: none"> Quản lý tuyến đường thu gom rác (routes). Quản lý địa điểm thu gom rác (MCPs). Back-officer lựa chọn một trong hai option. Hệ thống chuyển sang giao diện một trong hai option. Back-officer thực hiện quản lý khu vực thu gom rác. Hệ thống trở về giao diện chính.
Exceptions	Không
Alternative flows	Không.

Use-case name	Manage MCPs
Actor	Back-officer
Description	Back-officer muốn quản lý thông tin về địa điểm thu gom rác (MCPs).
Preconditions	Back-officer phải log in vào hệ thống bằng tài khoản hợp lệ được cấp quyền quản lý thông tin.
Postconditions	Back-officer quản lý thông tin MCPs thành công.
Normal flow	<ol style="list-style-type: none"> Hệ thống chuyển sang giao diện hiển thị cho việc quản lý MCPs và hiển thị ra 4 option <ul style="list-style-type: none"> Cập nhật thông tin địa điểm. Xem thông tin địa điểm. Xóa thông tin địa điểm. Thêm thông tin địa điểm. Back-officer click vào một trong bốn option. Hệ thống hiển thị giao diện tương ứng với một trong bốn option. Back-officer thực hiện quản lý thông tin của địa điểm thu gom rác. Hệ thống trở về giao diện chính.
Exception	Không.
Alternative flows	Không

Use-case name	Manage routes
Actor	Back officer
Description	Back officer muốn quản lý thông tin về tuyến đường thu gom rác.
Precondition	Back officer phải log in thành công bằng tài khoản được cấp quyền quản lý.
Postcondition	Back officer thực hiện quản lý thông tin tuyến đường thành công.
Normal flow	<ol style="list-style-type: none"> Hệ thống hiển thị giao diện cho việc quản lý routes và hiện ra 3 option. <ul style="list-style-type: none"> -Xem thông tin tuyến đường. -Cập nhật thông tin tuyến đường. -Xóa thông tin tuyến đường. Back officer click chọn một trong ba option. Hệ thống chuyển sang giao diện một trong ba option tương ứng. Back officer thực hiện quản lý thông tin tuyến đường. Hệ thống trở về giao diện chính.
Exception	Không.
Alternative Flow	Không.

c.1) Tính năng manage MCPs

Use-case name	Update MCPs)
Actor	Back officer
Description	Back officer muốn cập nhật thông tin của MCPs trong hệ thống.
Preconditions	Back officer phải log in vào hệ thống với tài khoản được cấp quyền quản lý tài nguyên công ty.
Normal flow	<ol style="list-style-type: none"> Hệ thống hiển thị thông tin của MCPs (địa chỉ, sức chứa, tình trạng MCPs,...). Back officer có thể cập nhật các mục thông cần thay đổi của MCPs. Để xác nhận cập nhật, Back officer click vào nút "Cập nhật thông tin". Hệ thống hiển thị thông báo "Xác nhận cập nhật thông tin". Nếu Back officer click vào nút "Có", hệ thống sẽ tự động thay đổi những thông tin mới cập nhật. Hệ thống hiển thị thông báo "Cập nhật thông tin thành công". Back officer click vào nút "OK". Hệ thống trở lại giao diện ban đầu.

Use-case name	Delete MCPs
Actor	Back-officer
Description	Back-officer muốn xóa thông tin của MCPs ra khỏi hệ thống.
Preconditions	Back officer phải log in vào hệ thống bằng tài khoản hợp lệ được cấp quyền quản lý thông tin tài nguyên công ty.
Exceptions	Không
Normal flows	<ol style="list-style-type: none">1. Hệ thống hiển thị danh sách tập các MCPs trong hệ thống.2. Back officer chọn ra những MCPs cần xóa thông tin bằng cách click vào checkbox của từng MCPs.3. Sau khi chọn xong, Back officer click vào nút "Xóa" ở cuối danh sách.4. Hệ thống hiển thị danh sách các MCPs mà Back officer muốn xóa.5. Back officer có thể điều chỉnh danh sách xóa bằng cách click vào nút "Xóa" để xóa MCPs ra khỏi danh sách xóa.6. Để xác nhận xóa thì Back officer click tiếp vào nút "Xóa" bên dưới danh sách MCPs muốn xóa.7. Hệ thống hiển thị thông báo "Bạn chắc chắn muốn xóa những MCPs này ra khỏi hệ thống?".8. Nếu Back officer click vào nút "Có", hệ thống sẽ tự động xóa thông tin MCPs.9. Hệ thống hiển thị thông báo "Xóa thông tin MCPs thành công".10. Back officer click vào nút "OK".11. Hệ thống trở về giao diện ban đầu.
Alternative flow	<p>Alternative Flow 1: tại bước 4:</p> <ol style="list-style-type: none">1.1 Nếu Back officer không muốn xóa MCPs nữa thì nhấn nút "Hủy".1.2 Hệ thống trở về giao diện hiển thị danh sách ở bước 3. <p>Alternative Flow 2: tại bước 8:</p> <ol style="list-style-type: none">2.1 Nếu Back officer muốn chỉnh sửa lại danh sách xóa thì click vào nút "Hủy".2.2 Hệ thống hiển thị lại danh sách xóa, ở bước 4.

Use-case name	Create MCPs
Actor	Back-officer
Description	Back-officer muốn thêm thông tin về MCP mới vào hệ thống.
Precondition	Back-officer phải log in vào hệ thống bằng tài khoản hợp lệ được cấp quyền quản lý tài nguyên công ty.
Normal flow	<ol style="list-style-type: none">1. Hệ thống yêu cầu back-officer phải nhập đầy đủ thông tin của MCPs với đúng định dạng bao gồm: địa chỉ, sức chứa, tình trạng,...2. Back-officer phải điền đầy đủ các thông tin bắt buộc với đúng định dạng.3. Sau khi điền xong, back-officer click vào nút "Thêm MCPs vào hệ thống".4. Hệ thống hiển thị tin nhắn "Bạn có muốn thêm MCPs này vào hệ thống?".5. Back officer click vào nút "Có" để xác nhận thêm.6. Hệ thống hiển thị thông báo "Thêm MCPs thành công vào hệ thống".7. Back-officer click vào nút OK.8. Hệ thống trở về giao diện ban đầu.
Alternative flow	<p>Alternative flow 1: tại bước 5:</p> <ol style="list-style-type: none">5.1 Nếu back-officer không muốn thêm MCPs mới vào hệ thống nữa thì chọn nút "Hủy".5.2 Hệ thống trở về giao diện điền thông tin của MCPs mới.

Use-case name	View MCPs info
Actor	Back officer
Description	Back officer muốn xem thông tin chi tiết của địa điểm thu gom rác trong hệ thống.
Preconditions	Back officer phải log in vào hệ thống bằng tài khoản hợp lệ được cấp quyền quản lý thông tin.
Normal flow	1. Hệ thống hiển thị thông tin chi tiết của địa điểm thu gom rác (địa chỉ, hình ảnh, sức chứa, tình trạng hiện tại...). 2. Sau khi xem xong, Back officer click nút "Hoàn thành xem" để trở lại giao diện chính.
Exceptions	Không.
Alternative flow	Không.

c.2) **Tính năng manage routes.**

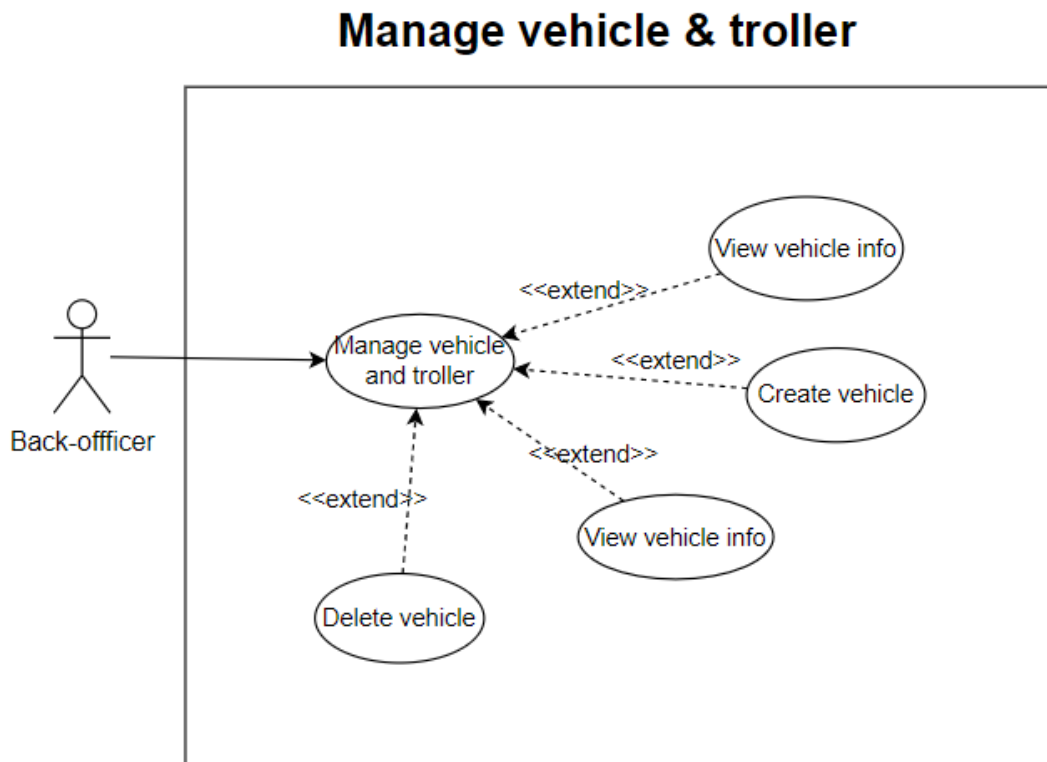
Use-case name	Update routes)
Actor	Back-officer
Description	Back-officer muốn cập nhật thông tin của tuyến đường thu gom rác trong hệ thống.
Preconditions	Back-officer phải log in vào hệ thống với tài khoản được cấp quyền quản lý thông tin.
Normal flow	1. Hệ thống hiển thị thông tin của tuyến đường (tên, lộ trình, chiều dài đường đi, thời gian...). 2. Back-officer có thể cập nhật các mục thông cần thay đổi của đường đi. 3. Để xác nhận cập nhật, Back officer click vào nút "Cập nhật thông tin". 4. Hệ thống hiển thị thông báo "Xác nhận cập nhật thông tin". 5. Nếu Back officer click vào nút "Có", hệ thống sẽ tự động thay đổi những thông tin mới cập nhật. 6. Hệ thống hiển thị thông báo "Cập nhật thông tin thành công". 7. Back-officer click vào nút "OK". 8. Hệ thống trở lại giao diện ban đầu.
Exception	Exception 1. 1.1. Nếu back-officer không nhập đầy đủ thông tin tại những mục bắt buộc thì hệ thống hiển thị thông báo "Hãy nhập đầy đủ thông tin đường ". 1.2 Back-officer phải nhập đầy đủ thông của nhân viên. Exception 2. 2.1 Nếu back-officer không nhập đúng định dạng thông tin thì hệ thống hiển thị "Hãy nhập đúng định dạng: [kiểu định dạng]". 2.2 Back-officer phải nhập đúng định dạng thông tin.
Alternative flow	Alternative flow 1: tại bước 3 3.1 Nếu back-officer nhập sai thông tin cần chỉnh sửa và muốn thay đổi lại thì click vào nút "Khôi phục thông tin cũ của đường đi". 3.2 Hệ thống khôi phục lại thông tin cũ trước khi nhập. Alternative flow 2: tại bước 5 5.1 Nếu back-officer không muốn cập nhật thông tin của đường đi nữa thì click vào nút "Hủy". 5.2 Hệ thống trở về giao diện chỉnh sửa như bước 2.

Use-case name	Delete route
Actor	Back-officer
Description	Back-officer muốn xóa thông tin của tuyến đường thu gom rác ra khỏi hệ thống.
Preconditions	Back-officer phải log in vào hệ thống bằng tài khoản hợp lệ được cấp quyền quản lý.
Postconditions	Back-officer xóa thành công thông tin của tuyến đường ra khỏi hệ thống.
Normal flows	<ol style="list-style-type: none">1. Hệ thống hiển thị danh sách tập các MCPs trong hệ thống.2. Back officer chọn ra những tuyến đường cần xóa thông tin bằng cách click vào checkbox của từng tuyến đường.3. Sau khi chọn xong, Back officer click vào nút "Xóa" ở cuối danh sách.4. Hệ thống hiển thị danh sách các tuyến đường mà back-officer muốn xóa.5. Back officer có thể điều chỉnh danh sách xóa bằng cách click vào nút "Xóa" để xóa tuyến đường ra khỏi danh sách xóa.6. Để xác nhận xóa thì Back officer click tiếp vào nút "Xóa" bên dưới danh sách tuyến đường muốn xóa.7. Hệ thống hiển thị thông báo "Bạn chắc chắn muốn xóa những tuyến đường này ra khỏi hệ thống?".8. Nếu Back officer click vào nút "Có", hệ thống sẽ tự động xóa thông tin tuyến đường.9. Hệ thống hiển thị thông báo "Xóa thông tin tuyến đường thành công".10. Back officer click vào nút "OK".11. Hệ thống trở về giao diện ban đầu.
Alternative flow	<p>Alternative Flow 1: tại bước 4:</p> <ol style="list-style-type: none">4.1 Nếu Back officer không muốn xóa MCPs nữa thì nhấn nút "Hủy".4.2 Hệ thống trở về giao diện hiển thị danh sách ở bước 3. <p>Alternative Flow 2: tại bước 8:</p> <ol style="list-style-type: none">8.1 Nếu Back officer muốn chỉnh sửa lại danh sách xóa thì click vào nút "Hủy".8.2 Hệ thống hiển thị lại danh sách xóa, ở bước 4.

Use-case name	Create new route
Actor	Back-officer
Description	Back-officer muốn thêm thông tin về tuyến đường thu gom rác mới vào hệ thống.
Precondition	Back-officer phải log in vào hệ thống bằng tài khoản hợp lệ được cấp quyền quản lý.
Postcondition	Back-officer thêm thông tin về tuyến đường thu gom rác mới vào hệ thống thành công.
Normal flow	<ol style="list-style-type: none">1. Hệ thống yêu cầu back-officer phải nhập đầy đủ thông tin của tuyến đường với đúng định dạng bao gồm: tên đường đi, chiều dài, thời gian hoạt động, ...2. Back-officer phải điền đầy đủ các thông tin bắt buộc với đúng định dạng.3. Sau khi điền xong, back-officer click vào nút "Thêm tuyến đường vào hệ thống".4. Hệ thống hiển thị tin nhắn "Bạn có muốn thêm tuyến đường này vào hệ thống?".5. Back officer click vào nút "Có" để xác nhận thêm.6. Hệ thống hiển thị thông báo "Thêm tuyến đường thành công vào hệ thống".7. Back-officer click vào nút OK.8. Hệ thống trở về giao diện ban đầu.
Alternative flow	Alternative flow 1: tại bước 5: 5.1 Nếu back-officer không muốn thêm tuyến đường mới vào hệ thống nữa thì chọn nút "Hủy". 5.2 Hệ thống trở về giao diện điền thông tin của tuyến đường mới.

Use-case name	View routes info
Actor	Back-officer
Description	Back-officer muốn xem thông tin chi tiết của tuyến đường thu gom rác trong hệ thống.
Preconditions	Back-officer phải log in vào hệ thống bằng tài khoản hợp lệ được cấp quyền quản lý thông tin.
Normal flow	<ol style="list-style-type: none">1. Hệ thống hiển thị thông tin chi tiết của tuyến đường thu gom rác (lộ trình, chiều dài đường đi, thời gian,...).2. Sau khi xem xong, Back officer click nút "Hoàn thành xem" để trở lại giao diện chính.
Exceptions	Không.
Alternative flow	Không.

c.3) Tính năng manage vehicle and troller



Use-case name	Manage Vehicle
Actor	Back-officer
Description	Back-officer muốn quản lý thông tin tài nguyên về xe rác và thùng rác trong hệ thống.
Preconditions	Back-officer phải log in vào hệ thống với tài khoản được cấp quyền quản lý.
Normal flow	<ol style="list-style-type: none"> Hệ thống hiển thị giao diện gồm 4 option để quản lý thông tin vehicle: <ul style="list-style-type: none"> -Xóa xe rác (thùng rác) ra khỏi hệ thống. -Thêm xe rác (thùng rác) vào hệ thống. -Xem thông tin xe rác (thùng rác). -Cập nhật thông tin xe rác (thùng rác). Back-officer click vào 1 trong 4 option. Hệ thống chuyển sang giao diện của 1 trong 4 option vừa chọn. Back-officer thực hiện xong công việc quản lý thông tin. Hệ thống trở về giao diện chính.
Exception	Không
Alternative flow	Không

Use-case name	Update vehicle info
Actor	Back officer
Description	Back officer muốn cập nhật thông tin của xe rác (thùng rác) trong hệ thống.
Preconditions	Back officer phải log in vào hệ thống với tài khoản được cấp quyền quản lý.
Normal flow	<ol style="list-style-type: none">1. Hệ thống hiển thị thông tin của vehicle (số xe, tải trọng, loại xe,...).2. Back officer có thể cập nhật các mục thông cần thay đổi của vehicle.3. Để xác nhận cập nhật, back officer click vào nút "Cập nhật thông tin".4. Hệ thống hiển thị thông báo "Xác nhận cập nhật thông tin".5. Nếu Back officer click vào nút "Có", hệ thống sẽ tự động thay đổi những thông tin mới cập nhật.6. Hệ thống hiển thị thông báo "Cập nhật thông tin thành công".7. Back officer click vào nút "OK".8. Hệ thống trở lại giao diện ban đầu.

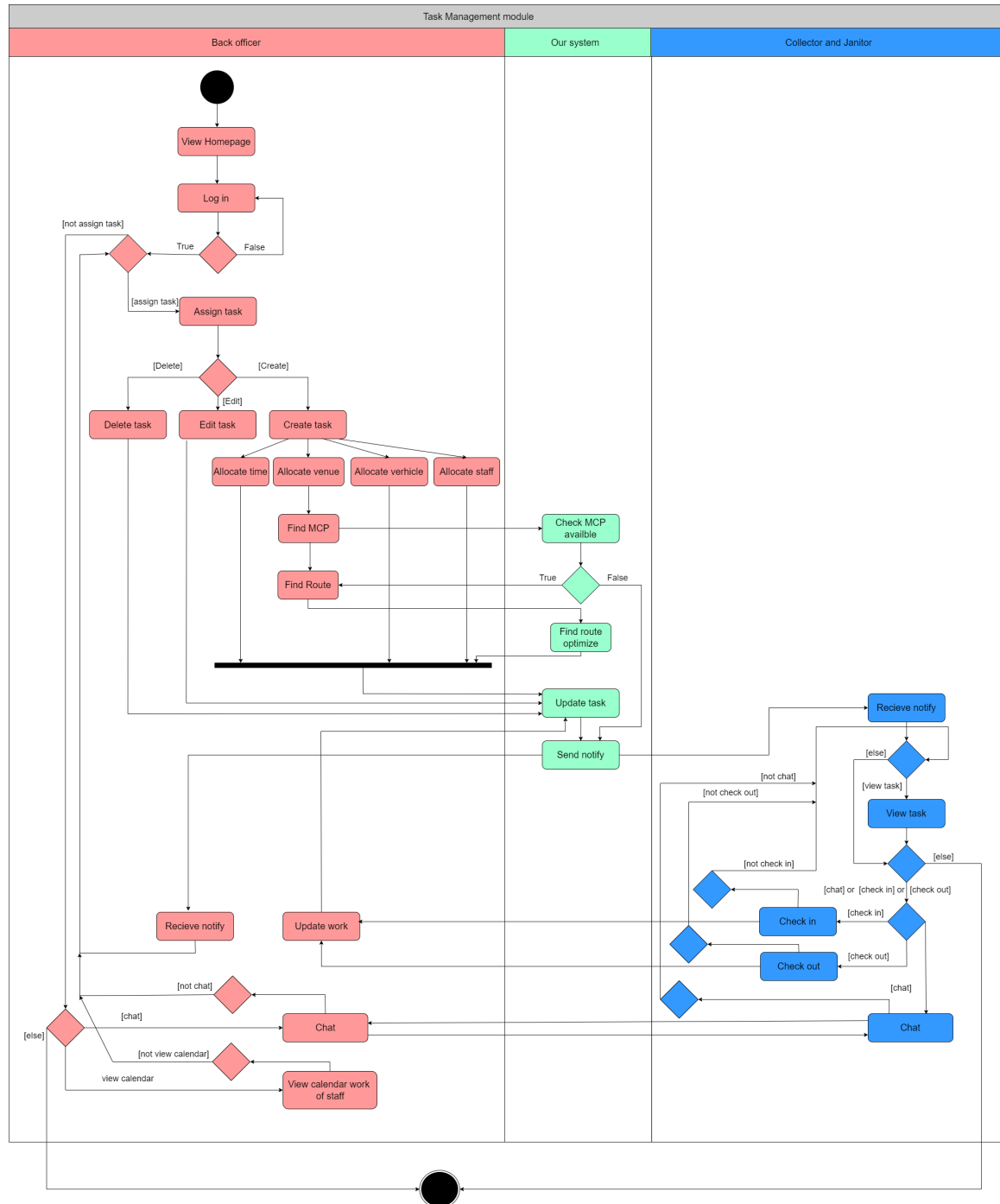
Use-case name	View routes info
Actor	Back-officer
Description	Back-officer muốn xem thông tin chi tiết của tuyến đường thu gom rác trong hệ thống.
Preconditions	Back-officer phải log in vào hệ thống bằng tài khoản hợp lệ được cấp quyền quản lý thông tin.
Normal flow	<ol style="list-style-type: none">1. Hệ thống hiển thị thông tin chi tiết của tuyến đường thu gom rác (lộ trình, chiều dài đường đi, thời gian,...).2. Sau khi xem xong, Back officer click nút "Hoàn thành xem" để trở lại giao diện chính.
Exceptions	Không.
Alternative flow	Không.

Use-case name	Delete vehicle
Actor	Back officerr
Description	Back officer muốn xóa thông tin của xe rác (thùng rác) ra khỏi hệ thống.
Preconditions	Back officer phải log in vào hệ thống bằng tài khoản hợp lệ được cấp quyền quản lý thông tin tài nguyên công ty.
Exceptions	Không
Normal flows	<ol style="list-style-type: none">1. Hệ thống hiển thị danh sách tập các xe rác (thùng rác) trong hệ thống.2. Back officer chọn ra những vehicle cần xóa thông tin bằng cách click vào checkbox của từng phương.3. Sau khi chọn xong, back officer click vào nút "Xóa" ở cuối danh sách.4. Hệ thống hiển thị danh sách các vehicle mà back-officer muốn xóa.5. Back-officer có thể điều chỉnh danh sách xóa bằng cách click vào nút "Xóa" để xóa vehicle ra khỏi danh sách xóa.6. Để xác nhận xóa thì Back officer click tiếp vào nút "Xóa" bên dưới danh sách vehicle muốn xóa.7. Hệ thống hiển thị thông báo "Bạn chắc chắn muốn xóa những vehicle này ra khỏi hệ thống?".8. Nếu Back-officer click vào nút "Có", hệ thống sẽ tự động xóa thông tin vehicle.9. Hệ thống hiển thị thông báo "Xóa thông tin phương tiện thành công".10. Back-officer click vào nút "OK".11. Hệ thống trở về giao diện ban đầu.
Alternative flow	<p>Alternative Flow 1: tại bước 4:</p> <ol style="list-style-type: none">1.1 Nếu Back-officer không muốn xóa vehicle nữa thì nhấn nút "Hủy".1.2 Hệ thống trở về giao diện hiển thị danh sách ở bước 3. <p>Alternative Flow 2: tại bước 8:</p> <ol style="list-style-type: none">2.1 Nếu back-officer muốn chỉnh sửa lại danh sách xóa thì click vào nút "Hủy".2.2 Hệ thống hiển thị lại danh sách xóa, ở bước 4.

Use-case name	Create vehicle
Actor	Y's Back officer
Description	Back officer muốn thêm thông tin về MCP mới vào hệ thống.
Precondition	Back officer phải log in vào hệ thống bằng tài khoản hợp lệ được cấp quyền quản lý tài nguyên công ty.
Normal flow	<ol style="list-style-type: none">1. Hệ thống yêu cầu Back officer phải nhập đầy đủ thông tin của vehicle với đúng định dạng bao gồm: số xe, loại xe, tải trọng,...2. Back officer phải điền đầy đủ các thông tin bắt buộc với đúng định dạng.3. Sau khi điền xong, Back officer click vào nút "Thêm vehicle vào hệ thống".4. Hệ thống hiển thị tin nhắn "Bạn có muốn thêm vehicle này vào hệ thống?".5. Back officer click vào nút "Có" để xác nhận thêm.6. Hệ thống hiển thị thông báo "Thêm vehicle thành công vào hệ thống".7. Back officer click vào nút OK.8. Hệ thống trở về giao diện ban đầu.
Alternative flow	<p>Alternative flow 1: tại bước 5:</p> <ol style="list-style-type: none">1.1 Nếu Back officer không muốn thêm vehicle vào hệ thống nữa thì chọn nút "Hủy".1.2 Hệ thống trở về giao diện điền thông tin của vehicle mới.

2 Task 2: System modelling

2.1 Draw an activity diagram to capture the business process between systems and the stakeholders in Task Assignment module



Link diagram: <https://app.diagrams.net/#G18ej9PUIOD8sJmzp6tkr3NTYe7TG1CP-M>

Mô tả

Ta chọn start point là ở góc nhìn của **Back Officer**

- Người sau khi truy cập vào ứng dụng sẽ phải đăng nhập để sử dụng ứng dụng.
- Ở bước đăng nhập (Log in), nếu người dùng nhập đúng thông tin được lưu trong hệ thống, người dùng sẽ tùy chọn các tính năng: *Assign task*, *Chat* và *View Calendar*.
- Tiếp theo, nếu người dùng chọn chức năng *Assign task* - giao nhiệm vụ, sẽ có 3 lựa chọn:

- *Create task* - tạo task mới, chức năng này bao gồm gán yêu cầu thời gian (allocate time), gán địa điểm (venue), phương tiện (vehicle), nhân viên (staff).

Ở tính năng gán địa điểm (allocate venue), yêu cầu hệ thống tính toán, tìm được các MCPs cần thu gom, và từ đó tìm ra đường đi tối ưu để giảm thiểu chi phí.

Nếu hệ thống tìm được available MCPs → tìm ra đường đi tối ưu. Nếu không có MCPs nào, thì hệ thống gửi thông báo về cho **Back Officer** và trở về trang chủ.

- *Create task* - sửa task (thay đổi người thực hiện, thay đổi vị trí của MCPs,...)
- *Delete task* - xóa task

Ở cả 3 tùy chọn trên, nếu thực hiện thành công, hệ thống sẽ tự động update task, calendar. Và gửi thông báo về cho những người liên quan (Back officer, collector và janitor). Sau đó, hệ thống sẽ đưa người dùng trở về trang chủ. Người dùng có thể tiếp tục lựa chọn các tính năng khác như *Chat*, *View Calendar* hoặc rời đi.

- *Chat* - người dùng vào tính năng này để giao tiếp bằng tin nhắn.
- *View Calendar* - người dùng xem lịch làm việc của mình (đối với Janitor, Collector) và xem lịch làm việc của nhân viên (Back Officer).

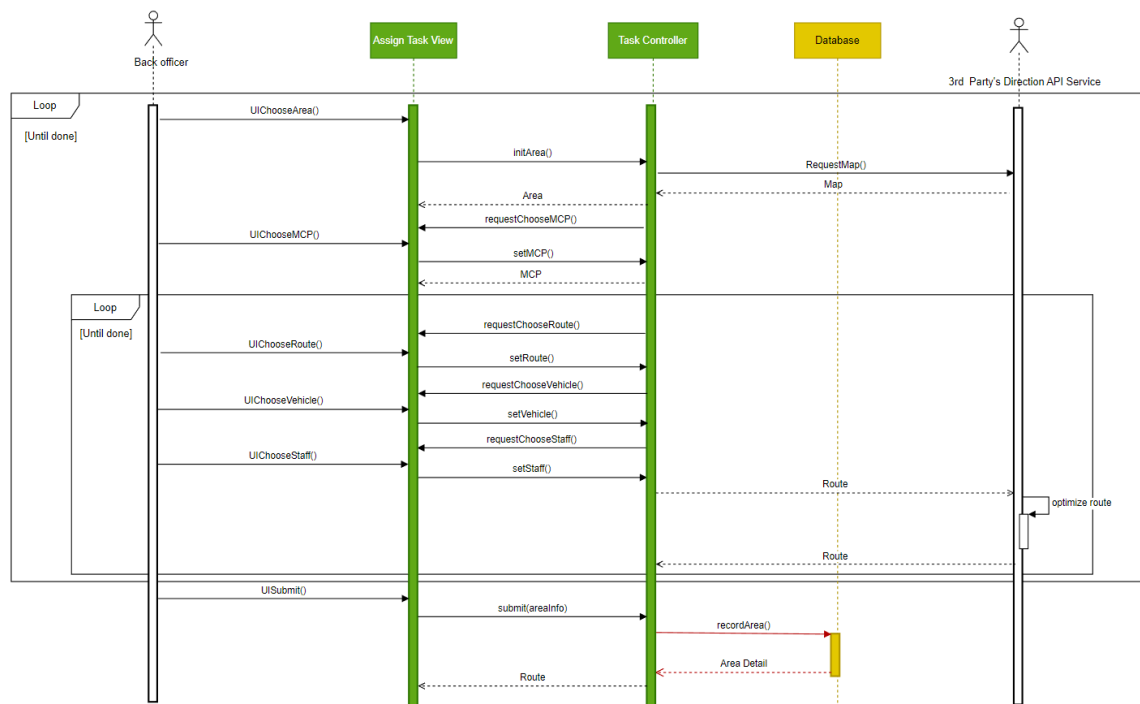
Đối với **Janitor** và **Collector**, sau khi được *Assign task*, họ sẽ nhận được thông báo và họ có thể xem sau khi đã đăng nhập vào hệ thống. Từ đó, người dùng có các lựa chọn:

- Họ có thể xem task mà mình đã được phân công từ bảng thông báo.
- Thực hiện lựa chọn các tính năng khác như *Chat*, *Check in*, *Check out*
- Hoặc thoát khỏi chương trình.

→ Từ bảng thông báo, người dùng cũng có thể vào tính năng *Check in*, *Check out* để thực hiện Check in hoặc Check out công việc đã được phân công.

→ Nếu *Check in*, *Check out* thành công, hệ thống sẽ update tình trạng của task tự động → từ đó update calendar.

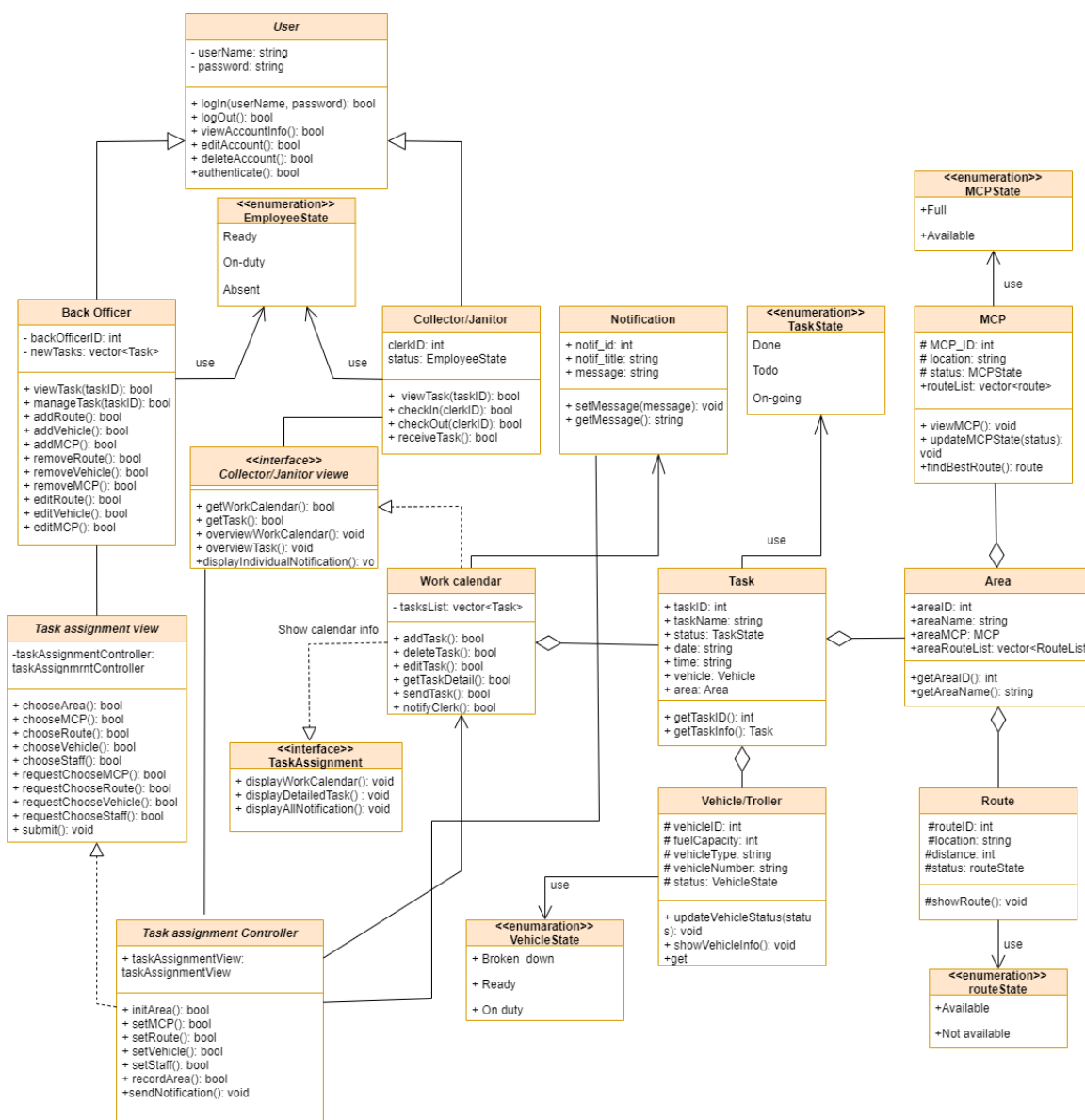
2.2 Proposal a conceptual solution for the route planning task and draw a sequence diagram to illustrate it



Mô tả:

- Một khu vực sẽ bao gồm các thông tin: địa điểm khu vực công việc, các vị trí MCP trong khu vực, các tuyến đường (được tối ưu thông qua hệ thống thứ ba), danh sách nhân viên tham gia công việc.
- Người dùng chọn khu vực xác định công việc thông qua bản đồ điện tử của bên thứ ba cung cấp. Khi đó hệ thống khởi tạo một khu vực rộng thông tin và yêu cầu người dùng chọn một MCP (Major Collecting Point).
- Người dùng chọn điểm MCP thông qua bản đồ điện tử. Hệ thống sẽ cập nhật thông tin MCP, sau đó yêu cầu các thông tin về tuyến đường, các phương tiện vận chuyển và các nhân viên tham gia vào công việc. Sau đó hệ thống sẽ gửi thông tin tọa độ của các MCP qua hệ thống thứ ba để tính toán tối ưu đường đi.
- Hệ thống thứ ba gửi thông tin tuyến đường đã tối ưu về cho hệ thống quản lý công việc. Hệ thống sẽ cập nhật thông tin tuyến đường của khu vực đã chọn. Việc chọn tuyến đường được lặp lại đến khi xong.
- Sau cùng, người dùng thông qua giao diện submit, bấm nút submit. Thông tin chi tiết của area sẽ lưu vào database ở phía server và gửi các message tương ứng.

2.3 Draw a class diagram of Task Assignment module as comprehensive as possible



Link diagram: https://drive.google.com/file/d/16tQafK0AQ0X4cEHqD2Em772F_R33cF9d/view?usp=sharing

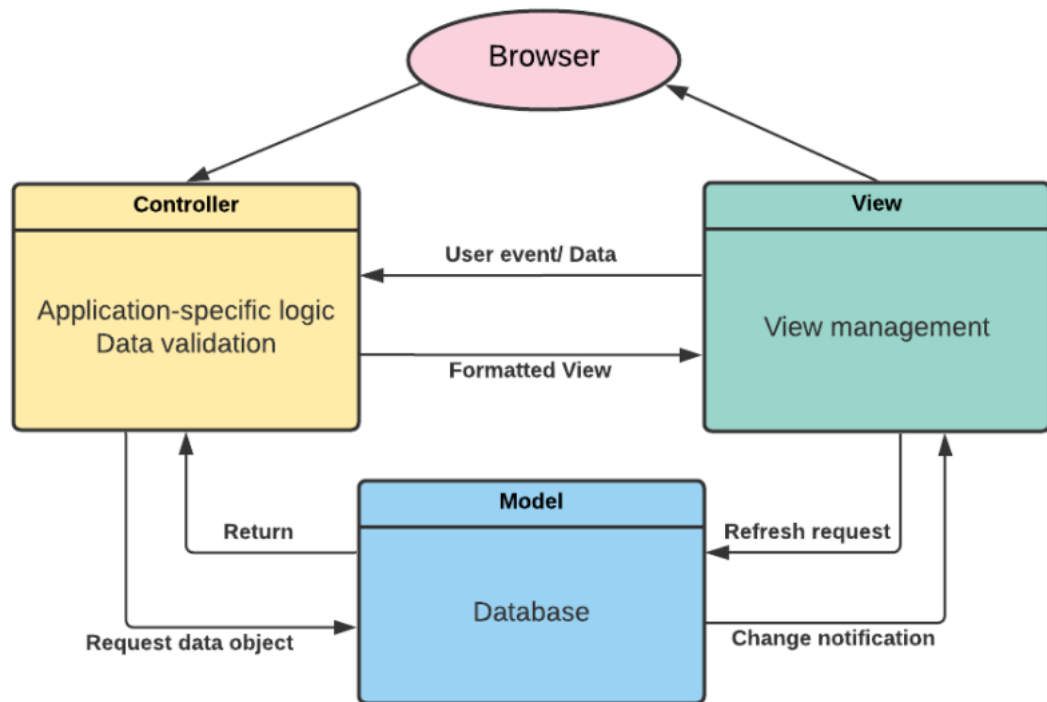
Mô tả:

- Giao diện chính của UWC2.0 có trang đăng nhập. **Backofficer**, **collector** và **janitor** đã đăng kí sẽ có *userName* và *password*, và một vài method khác như: *login*, *logout*,... *userName* và *password* tương ứng. Sau khi người dùng nhập thông tin, thông tin sẽ được gửi đến **Server**,
- **Lịch làm việc** (*work calendar*) sẽ lưu trữ danh sách các công việc (*task*) được tạo ra và đưa lên bởi **Back officer**. Khi người dùng muốn thao tác với các công việc sẽ chuyển sang giao diện của lịch làm việc.
- **Back officer** có thể tạo ra công việc mới trên lịch làm việc, sau khi công việc được tạo được đưa lên lịch làm việc. Thông tin công việc đưa lên lịch làm việc sẽ lưu các thông tin như *mã số công việc*, *thời gian*, *ngày*, *tháng*. Ngoài ra, nội dung chi tiết của công việc lưu các thông tin chi tiết về *phương tiện*, *tuyến đường cần làm việc* và *các điểm thu gom chính* (MCPs). Lúc này, công việc mới đưa lên lịch làm việc sẽ ở trạng thái *To do*. Trong các công việc, **Back officer** có thể quản lý các thông tin về *phương tiện*, *tuyến đường cần làm việc* và *các điểm thu gom chính*.
 - **Back officer** có thể tạo mới, xóa, sửa thông tin về *tuyến đường* (route). Các thông tin về tuyến đường như *mã số*, *địa điểm*, và *quãng đường*.
 - **Back officer** có thể tạo mới, xóa, sửa thông tin về các *điểm thu thập chính* (MCPs). Các thông tin về điểm thu thập như *mã số*, *địa điểm*, và *trạng thái*. Các trạng thái của *điểm thu thập* có thể là đầy hay trống. Trạng thái sẽ được cập nhật sau khi hoàn thành các công việc.
 - **Back officer** có thể tạo mới, xóa, sửa thông tin về các *phương tiện* (Vehicle). Các thông tin về phương tiện như *mã số phương tiện*, *tên phương tiện*, *số hiệu phương tiện*, *tình trạng nhiên liệu* và *trạng thái của phương tiện*. Khi phương tiện thuộc công việc được nhận, sẽ chuyển sang trạng thái *On duty*, khi không được sử dụng phương tiện sẽ ở trạng thái *Ready*. Đối với các phương tiện đang được sửa chữa sẽ ở trạng thái *Broken down*.
 - Các thao tác mà back-officer có thể thực hiện trên Task assignment View như: chọn địa điểm (choose Area), chọn nhân viên (Choose employee) sẽ được Task assignmet controller thao tác trực tiếp với dữ liệu trên Work Calendar với những hàm như: thay đổi MCP (set MCP),... Đồng thời, mỗi khi một task được tạo ra trên Work Calendar, hệ thống tự động tạo ra 1 notification cho back-officer thông qua Task Assignemnt Controller và hiển thị một thông báo trên Task Assignment View.
- Janitor/Collector cũng tương tác với Work Calendar qua interface Janitor/Collector. Qua đó, mọi thông tin về task đều được hiển thị lên trang lịch trình riêng của mỗi Janitor/Collect. Đồng thời, Khi có công việc mới được đưa lên bởi back-officer, lịch làm việc sẽ tạo ra *thông báo* với nội dung chi tiết về việc công việc, sau đó gửi thông báo cho **collector/janitor**, khi có thông tin thay đổi về chi tiết công việc đến từ **Back officer**, **collector/janitor** sẽ nhận được chi tiết về thay đổi đó trên lịch làm việc.
- Mỗi ngày, **collector/janitor** vào lịch làm việc để điểm danh và nhận công việc, sau khi nhận công việc, trạng thái của công việc sẽ chuyển sang *On-going*.
- Khi nhận công việc, **collector/janitor** sẽ được chuyển sang giao diện thông tin chi tiết công việc. Tại đây, **collector/janitor** có thể quan sát thông tin chi tiết về *tuyến đường*, *điểm thu thập chính* và *phương tiện* mà họ sử dụng. Khi bắt đầu công việc, *phương tiện* được sử dụng trong công việc sẽ chuyển sang trạng thái *On duty* và về lại trạng thái *Ready* sau khi hoàn thành. Khi kết thúc công việc, *điểm thu thập chính* trong công việc sẽ chuyển sang trạng thái *Available*.
- Vào cuối ngày, sau khi hoàn thành công việc, **collector/janitor** thực hiện *check out* trên lịch làm việc, khi đó, trạng thái công việc sẽ chuyển sang Done.

3 Task 3: Architecture design

3.1 Describe an architectural approach you will use to implement the desired system. How many modules you plan for the whole WMC 2.0 system? Briefly describe input, output and function of each module

- Mô tả kiến trúc MVC



Hình 1: Mô hình kiến trúc

Kiến trúc MVC (Model-View-Controller) là mô hình kiến trúc chia một ứng dụng thành 3 component chính về mặt logic: Model, View, Controller. Mỗi phần sẽ đảm nhiệm một tính năng riêng của hệ thống. Đây là một trong những kiến trúc phổ biến nhất được sử dụng trong công nghiệp vì tính chất linh hoạt và dễ dàng mở rộng của nó.

– Model component

- * Xác định ứng dụng sẽ có dữ liệu gì. Model component sẽ gửi thông điệp cho View khi giao diện có sự tương tác (frontend) và gửi thông điệp cho Controller khi có sự thay đổi về mặt dữ liệu của hệ thống (backend).
- * Trong app UWC, Model sẽ mô tả những thông cần lưu trữ về nhân viên (back-officer, janitor/collector), phương tiện (trolley, truck), MCPs, Route.

– View component

- * Xác định phần UI của hệ thống sẽ được hiển thị như thế nào.
- * Trong app UWC, View sẽ mô tả hệ thống cần render những phần nào cho người dùng như: lịch trình làm việc (work calendar) cho back-officer (dưới góc độ quản lý) và janitor/collector (dưới góc độ sử dụng), thông tin về những công việc cần làm (task info),...

– Controller component

- * Đảm nhiệm vai trò xử lý logic từ những request của người dùng và cập nhật những thông tin bên dưới database, đứng giữa View và Model.
- * Trong app UWC, khi người dùng có yêu cầu nào đó cho hệ thống, ví dụ: yêu cầu chỉnh sửa thông tin chi tiết của một task; thì phần Controller sẽ thực hiện cả hai tác vụ:
 - Lấy input từ người dùng và cập nhật hiển thị lên phần UI cho người dùng thấy thông tin nào cần chỉnh sửa.
 - Thực hiện cập nhật thông tin bên dưới phần database (nếu có yêu cầu chỉnh sửa).

● Modules

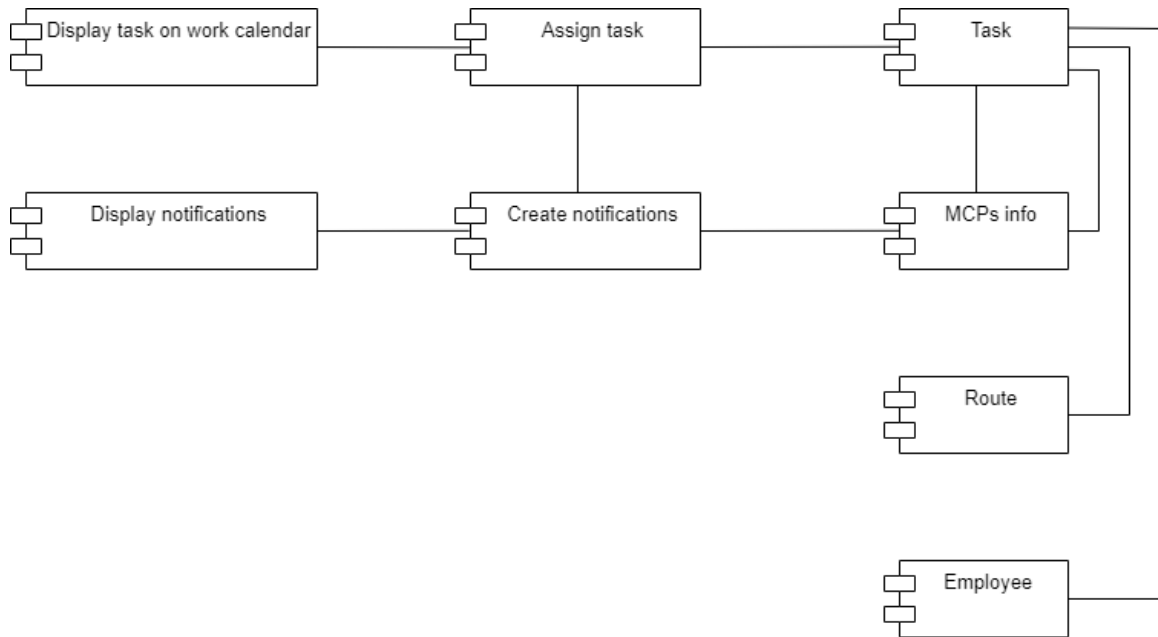
1. Task Assignment

Task Assignment có 8 module:

- Display task on work calendar

- Display notifications
- Assign task
- Create notifications
- Task
- MCPs info
- Route
- Employee

Các module được thể hiện như sau:



Mô tả input, output, function của mỗi module

Assign task

Task assignment result interface có

- Hàm Assign_result_info với đầu vào là thông tin công việc (task_info) và đầu ra là nội dung gán công việc
- Hàm Display_result_info với đầu vào nội dung gán công việc và đầu ra kiểu muốn hiển thị cho nội dung gán công việc

<<Interface>> Task assignment result	
+ Task_info: Object	
+ Assign_result_info(task_info: object): object	
+ Display_result_info(assign_task_info: object): object	

Assignment result interface có

- Hàm Assign_result trả về kết quả gán công việc có thành công hay không(True or False)

<<Interface>> Assignment result	
+ Assign_task_result: object	
+ Assign_result(): bool	

Create notifications

Task notification interface có

- Hàm Notify_content với đầu vào là kết quả gán công việc . Hàm trả về kiểu hiển thị thông báo tương ứng với kết quả đó

<<Interface>> Task notification
+ Notify_content (assign_result: bool): object

Employee

Employee shift list interface có

- Hàm Get_employ_shift trả về danh sách ca làm việc của công nhân

<<Interface>> Employee shift list
+ Employ_shift_list: list
+ Get_employ_shift(): object

Route

Route list interface có

- Hàm Get_route_list trả về danh sách các tuyến đường

<<Interface>> Route list
+ Route_list: list
+ Get_route_list(): object

MCPs info

MCP's info list interface có

- Hàm Get_mcp_list trả về danh sách các MCP

<<Interface>> MCP's info list
+ MCP_list: list
+ Get_mcp_list(): object

Task

Task info interface có

- Hàm Task_detail trả về thông tin công việc cần gán dựa trên các thuộc tính danh sách ca làm việc của nhân viên, danh sách tuyến đường, danh sách MCP

<<Interface>> Task_info
+ Employ_shift_list: list
+ Route_list: list
+ MCP_list: list
+ Task_detail(): object

New state interface có

- Hàm Update_employ_shift với đầu vào là danh sách nhân viên sau khi gán công việc(New_employ_shift_list), trả về danh sách nhân viên sau khi cập nhật
- Hàm Update_route với đầu vào là danh sách tuyến đường sau khi gán công việc(New_route_list), trả về danh sách tuyến đường sau khi cập nhật
- Hàm Update_MCP với đầu vào là danh sách MCP sau khi gán công việc(New_MCP_list), trả về danh sách MCP sau khi cập nhật

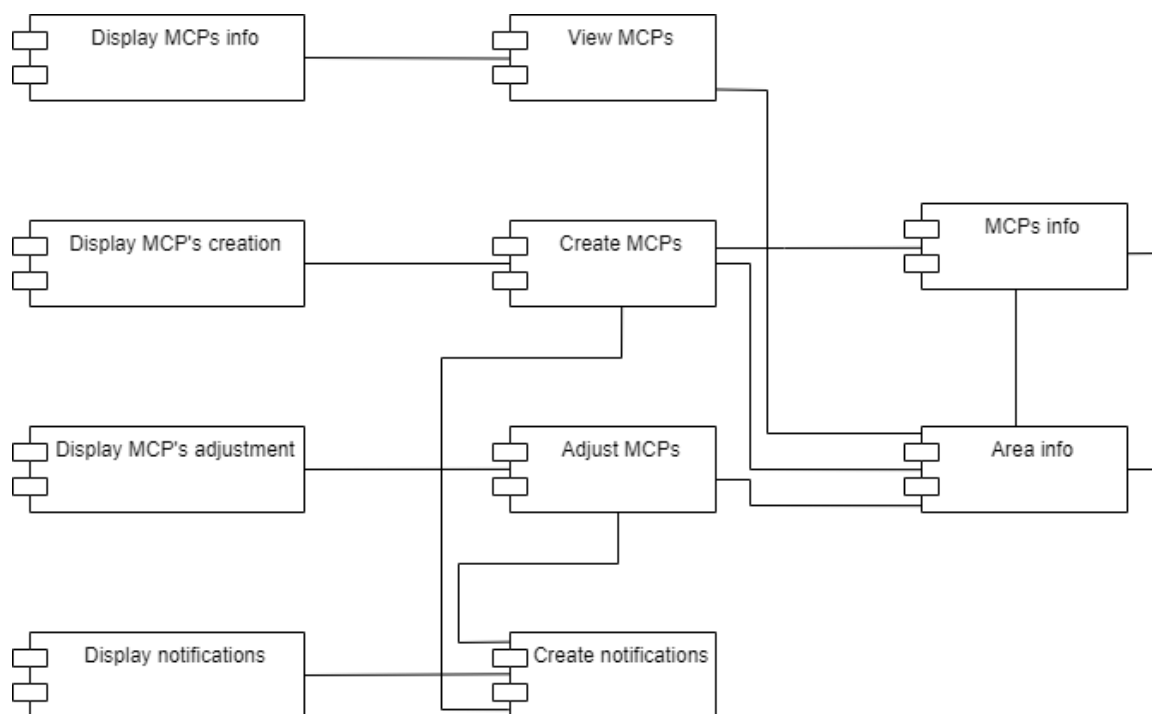
<<Interface>> New state
+ Employ_shift_list: list
+ Route_list: list
+ MCP_list: list
+ Update_employ_shift(New_employ_shift_list: list): void
+ Update_route(New_route_list: list): void
+ Update_MCP(New_MCP_list: list): void

2. MCP management

MCP management có 10 module:

- Display MCPs info
- Display MCP's creation
- Display MCP's adjustment
- Display notifications
- View MCPs
- Create MCPs
- Adjust MCPs
- Create notifications
- MCPs info
- Area info

Các module được thể hiện như sau:

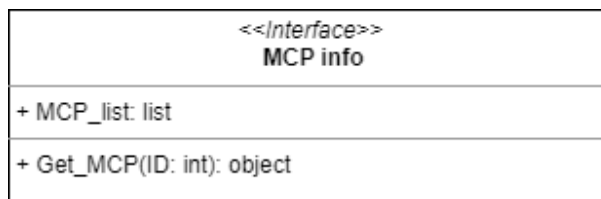


Mô tả input, output, function của mỗi module

Area info

MCP info interface có

- Hàm Get_MCP với đầu vào là ID của MCP. Hàm trả về MCP ứng với ID đầu vào



Empty MCPs list interface có

- Hàm `Get_empty_MCP_list` trả về danh sách MCP trống danh sách các MCP

MCPs info

Empty MCP info interface có

- Hàm `Get_empty_MCP` trả về MCP trống trong danh sách các MCP trống

View MCPs

MCP info view có

- Hàm `Display_MCP_info` trả về kiểu hiện hiển thị thông tin MCP

Create MCPs

MCP result interface có

- Hàm `MCP_create_result` với đầu vào là MCP rỗng. Hàm cập nhật trạng thái của MCP sang bận rộn và trả về MCP đó
- Hàm `MCP_modify_result` với đầu vào là MCP. Hàm điều chỉnh MCP và trả về MCP đó

Result interface có

- Hàm `Result` với đầu vào là MCP sau khi thao tác. Hàm trả về kết quả thao tác MCP có thành công hay không (True or False)

MCP creation result có

- Hàm `Display_MCP_create` trả về kiểu hiện hiển thị nội dung MCP sau khi tạo

**<<Interface>>
Empty MCPs list**

+ MCP_list: list

+ Get_empty_MCP_list(): list

**<<Interface>>
Empty MCP info**

+ Empty_MCP_list: list

+ Get_empty_MCP(): object

**<<Interface>>
MCP info view**

+ MCP: object

+ Display_MCP_info(): object

**<<Interface>>
MCP result**

+ MCP_create_result(MCP: object): object

+ MCP_modify_result(MCP: object): object

**<<Interface>>
Result**

+ Result(MCP_result: object): bool

**<<Interface>>
MCP creation result**

+ MCP_create: object

+ Display_MCP_create(): object

Adjust MCPs

MCP result interface có

- Hàm MCP_create_result với đầu vào là MCP rỗng. Hàm cập nhật trạng thái của MCP sang bận rộn và trả về MCP đó
- Hàm MCP_modiy_result với đầu vào là MCP. Hàm điều chỉnh MCP và trả về MCP đó

Result interface có

- Hàm Result với đầu vào là MCP sau khi thao tác. Hàm trả về kết quả thao tác MCP có thành công hay không(True or False)

MCP adjustment result có

- Hàm Display_MCP_adjust trả về kiểu hiển thị nội dung MCP sau khi điều chỉnh

MCP notification

MCP adjustment result có

- Hàm Display_MCP_adjust với đầu vào là kết quả tạo hoặc chỉnh sửa MCP . Hàm trả về kiểu hiển thị thông báo tương ứng với kết quả đó

<<Interface>> MCP result
+ MCP_create_result(MCP: object): object + MCP_modify_result(MCP: object): object

<<Interface>> Result
+ Result(MCP_result: object): bool

<<Interface>> MCP adjustment result
+ MCP_adjust: object
+ Display_MCP_adjust(): object

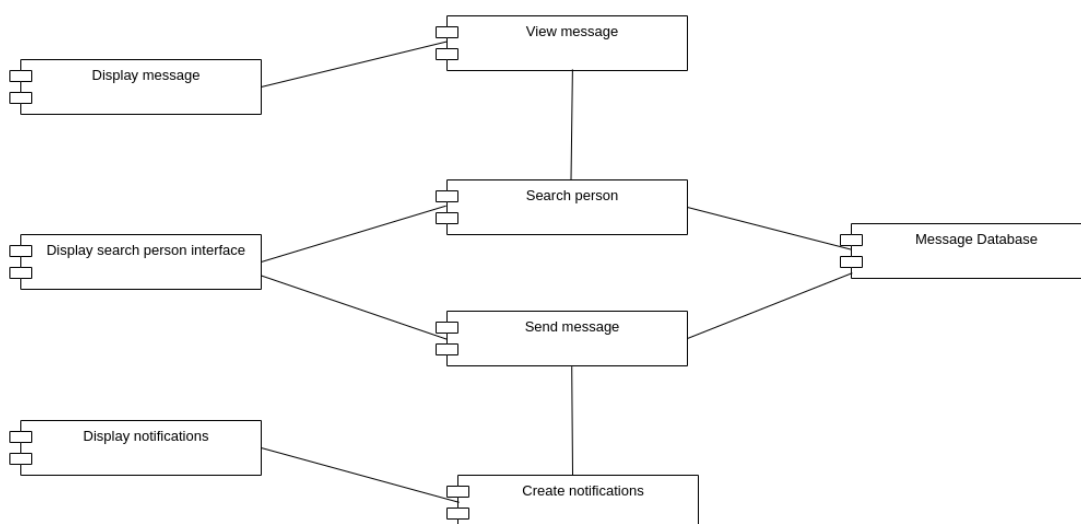
<<Interface>> MCP notification
+ Notify_content (MCP_result: bool): object

3. Communication

Communication có 8 module:

- Display message
- View message
- Search person
- Display search person interface
- Send message
- Message database
- Display notification
- Create notifications

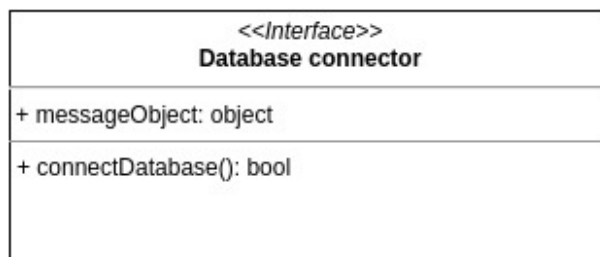
Các module được thể hiện như sau:



Mô tả input, output, function của mỗi module

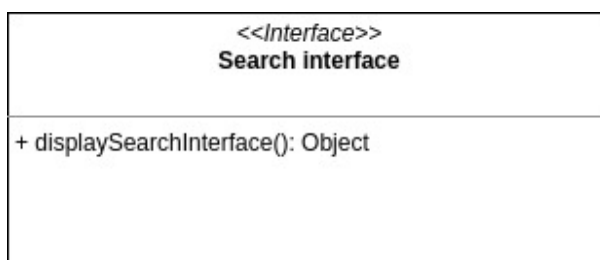
Message database

Message Database cung cấp Interface "**Database connector**" - cho phép truy cập vào database, có hàm connect-Database(), giúp kết nối với dữ liệu của hệ thống, hàm trả về true nếu kết nối thành công, và false nếu ngược lại.



Search person

- Search interface cung cấp hàm display-SearchInterce() cung cấp giao diện tìm kiếm



- Personal message có hàm getConversation() nhận vào tên người dùng và trả về dữ liệu tin nhắn của người dùng ứng với username đó

View message

- Interface "Message Infor" có các thuộc tính userInfo chứa thông tin về người dùng của những tin nhắn này, messageInfo là nội dung tin nhắn, và hàm displayConversation() trả về cuộc hội thoại.

Display message

- Interface "Chat interface" có hàm addMessage() nhận vào một đoạn tin nhắn và trả về kết quả của về gửi tin nhắn, và hàm deleteMessage() nhận vào msgID và trả về kết quả của việc xóa một tin nhắn trong đoạn hội thoại.

Create notification

- Interface "Message notification" chứa các thuộc tính như senderName: tên của người gửi, time: thời gian gửi, và có hàm displayNotify() nhận vào trạng thái gửi tin nhắn.

Send message

- Interface "Send message status" chứa trạng thái của tin nhắn (đã được gửi thành công hay chưa), và hàm getMsgStatus() trả về trạng thái của tin nhắn.

- Interface "New database state" chứa thông tin về trạng thái của tin nhắn mới được gửi, và từ đó có thể update lại database

**<<Interface>>
Personal message**

+ getConversation(username: string): Object

**<<Interface>>
Message Infor**

+ userInfo: object
+ messageInfo: object

+ displayConversation(): Object

**<<Interface>>
Chat interface**

+ addMessage(msg: string): bool
+ deleteMessage(msgID): bool

**<<Interface>>
Message notification**

+ senderName: string
+ time: string

+ displayNotify(sendStatus: bool): Object

**<<Interface>>
Send message status**

+ msgStatus: Object

+ getMsgStatus(): Object

**<<Interface>>
New database state**

+ newMsgInfor: Object

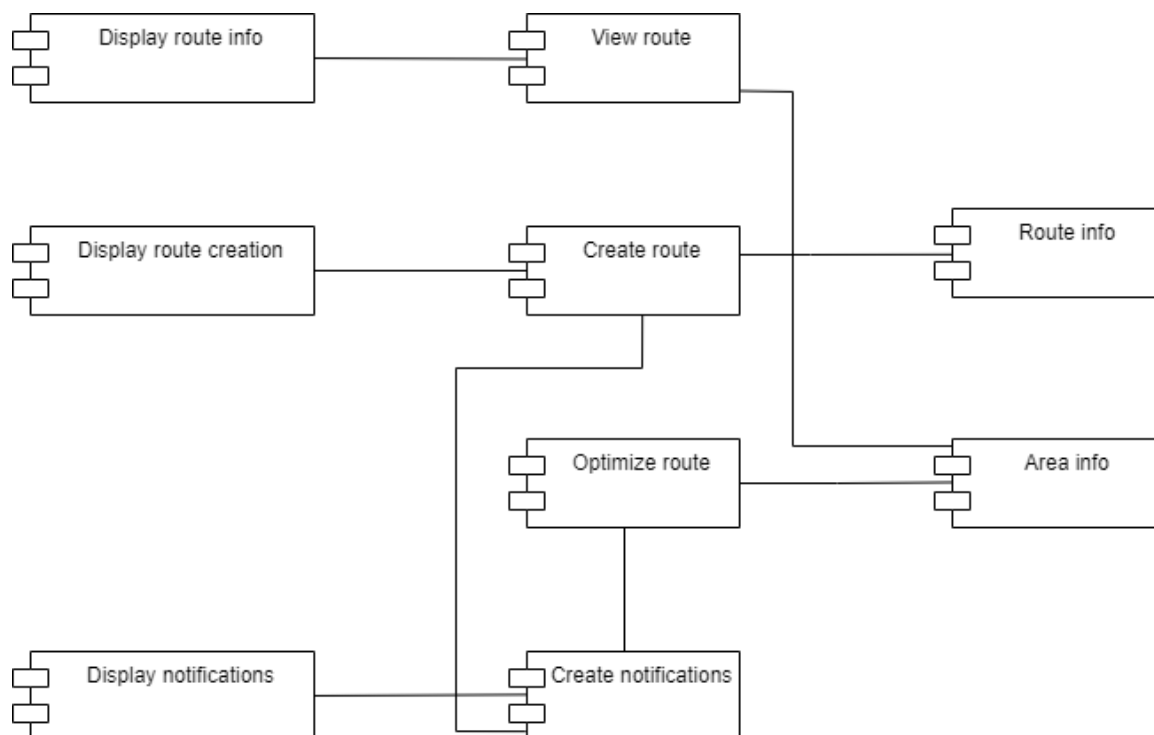
+ getNewMsgInfor(): Object

4. Route management

Route management có 9 modules:

- Display route info
- Display route creation
- Display notifications
- View route
- Create route
- Optimize route
- Create notifications
- Route info
- Area info

Các module được thể hiện như sau:

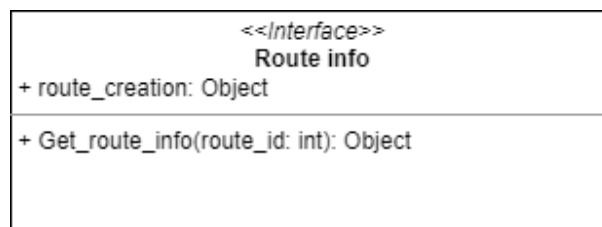


Mô tả input, output, function của mỗi module

Area info

Route info interface có

- Hàm `Get_route_info` với đầu vào là ID của route và trả về thông tin của route ứng với ID đó



Route list interface có
- Hàm Get_route_list trả về danh sách các tuyến đường

Route info

Route info interface có
- Hàm Get_route_info với đầu vào là ID của route đã được tối ưu và trả về thông tin của route ứng với ID đó

View route

Route info view có
- Hàm Display_route_info trả về kiểu hiển thị thông tin route

Create route

Route result interface có
- Hàm Result với đầu vào là Route sau khi được tạo. Hàm trả về kết quả tạo Route có thành công hay không(True or False)

Route creation result có
- Hàm Display_route_creation trả về kiểu hiển thị thông tin route sau khi tạo

Optimize route

Route optimization result interface có
- Hàm Get_optimized_route với đầu vào là các route chưa được tối ưu.Hàm trả về các route đã được tối ưu

<<Interface>> Route list
+ Route_list: list
+ Get_route_list(): object

<<Interface>> Route info
+ route_creation: Object
+ Get_route_info(route_id: int): Object

<<Interface>> Route info view
+ Route: Object
+ Display_route_info(): Object

<<Interface>> Route result
+ Result(Route_result: object): bool

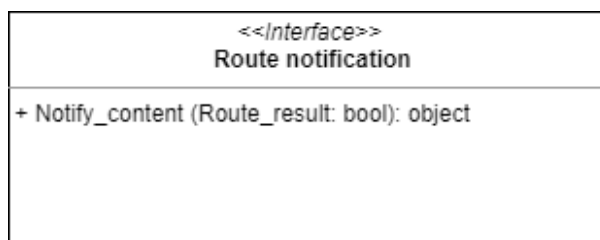
<<Interface>> Route creation result
+ route_creation: Object
+ display_route_creation(): Object

<<Interface>> Route optimization
+ Optimized_route: Object
+ Get_optimized_route(route:Object): Object

Route notifications

Route notifications có

- Hàm Notify_content với đầu vào là kết quả tạo. Hàm trả về hiển thị thông báo dựa trên kết quả đã nhận được

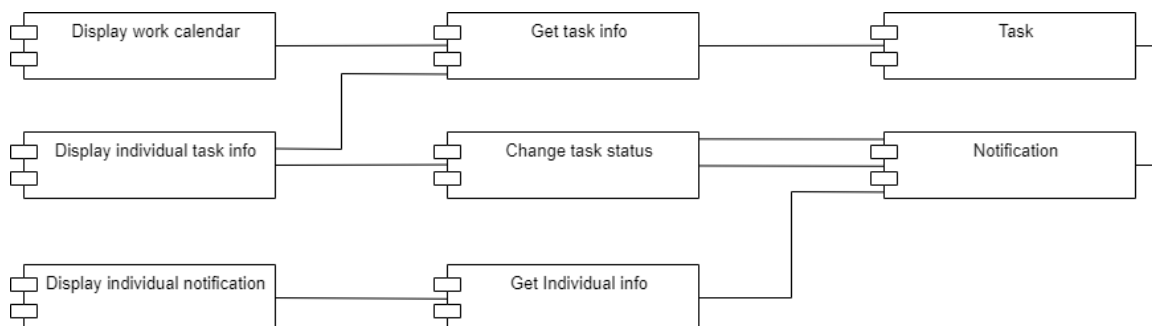


5. Work calendar

Route management có 8 modules:

- Display work calendar
- Display individual task info
- Display individual notifications
- Get task info
- Change task status
- Get notifications
- Task
- Notifications

Các module được thể hiện như sau:

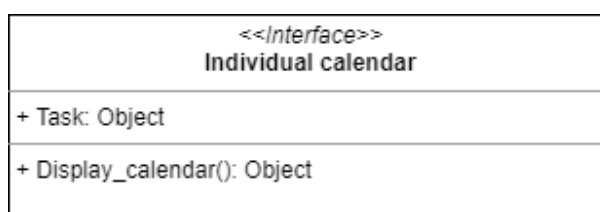


Mô tả input, output, function của mỗi module

Get task info

Individual calendar interface có

- Hàm Display_calendar_info trả về kiểu hiển thị thông tin chung của các task



Individual tasks info interface có

- Hàm Display_task_info với đầu vào ID của task. Hiện hiển thị thông báo của task có ID tương ứng

Task

Task info interface có

- Hàm Get_task với đầu vào là ID của task. Hàm trả về task ứng với ID đầu vào

Get notifications

Notifications có

- Hàm displayNotificationList có input là id của janitor/collector và output là danh sách các notification (có đánh dấu những notification nào là mới).

Notifications

Notification info có

- Hàm displayNewNotification trả về thông tin thông báo của việc thay đổi task.

Change task status

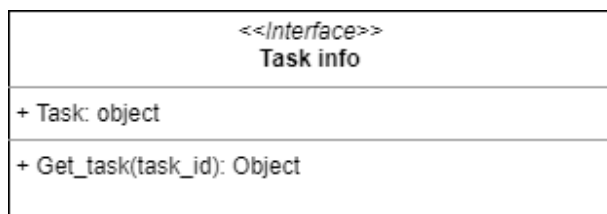
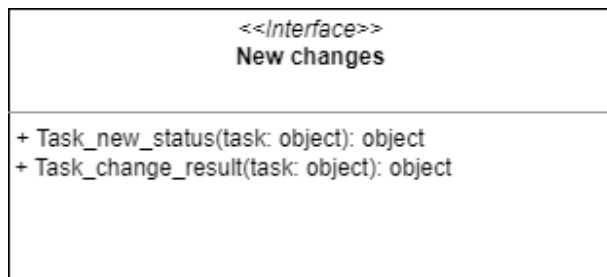
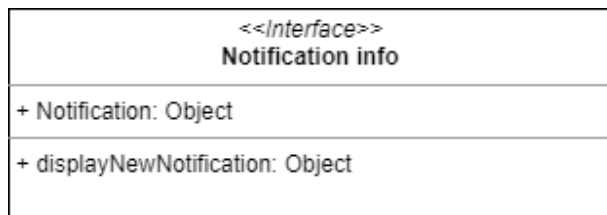
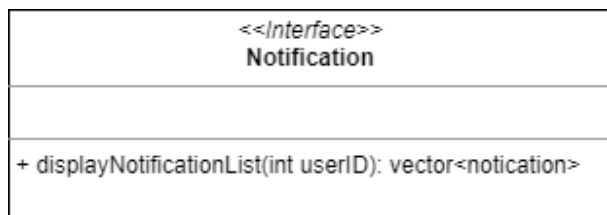
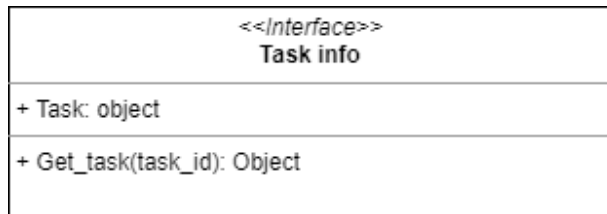
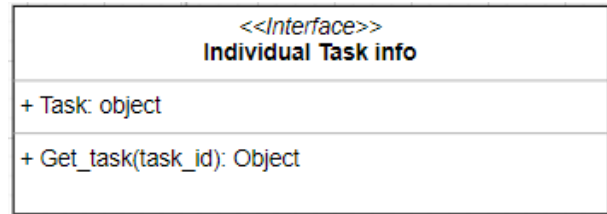
New status interface có

- Hàm Task_new_status với đầu vào là task với trạng thái Todo sau đó chuyển về trạng thái Ongoing và trả về.
- Hàm Task_change_result với đầu vào là task và trả về sau khi thay đổi.

Get individual info

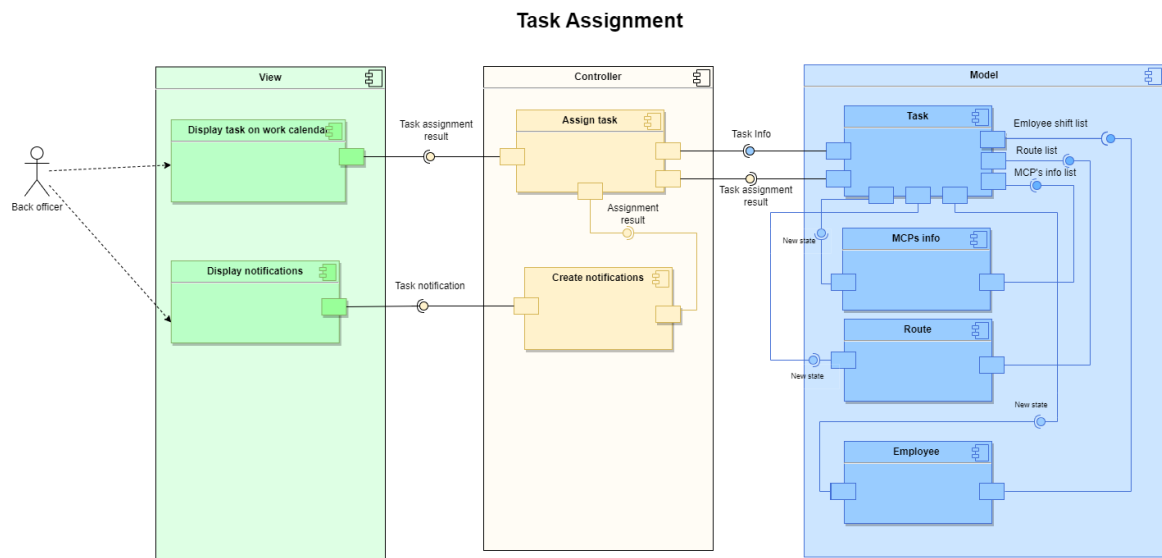
Task info có

- Hàm Get task với input là ID của một task, output là task ứng với ID đó.



3.2 Draw an implementation diagram for Task Assignment module

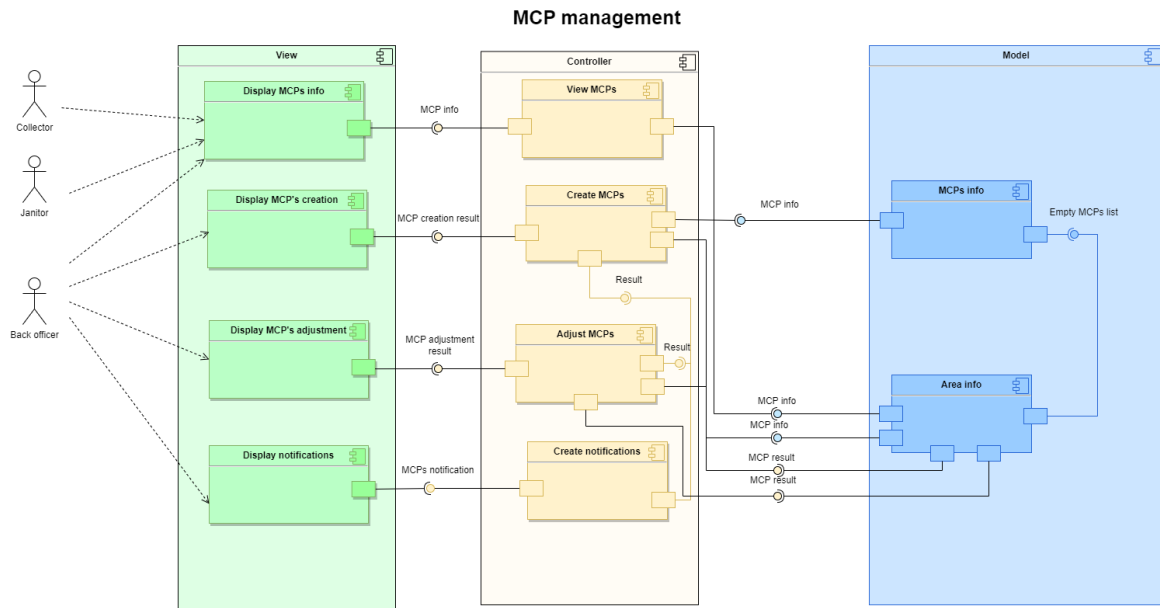
1. Component Diagram - Task Assignment



Mô tả

- Khi người dùng chọn chức năng gán công việc thì khối *Assign task* trong controller sẽ lấy thông tin về công việc cần gán từ khối *Task* trong model. Thông tin công việc của khối *Task* được tổng hợp từ các khối *MCPs info*, *Route*, *Employee* trong model. Sau khi người dùng hoàn thành việc gán công việc thì khối *Assign task* sẽ hiển thị nội dung công việc qua khối *Display task on work calendar* ở view và cập nhật thông tin công việc qua khối *Task* ở model
- Khối *Task* ở model sẽ cập nhật thông tin công việc về ca làm việc của người lao động, MCP và các tuyến đường qua các khối tương ứng *Employee*, *MCPs info*, *Route* ở model
- Sau đó khối *Assign task* sẽ gửi kết quả gán công việc qua khối *Create notifications* ở controller. Khối *Create notifications* sẽ hiển thị thông báo tương ứng với kết quả gán công việc qua khối *Display notifications* ở view

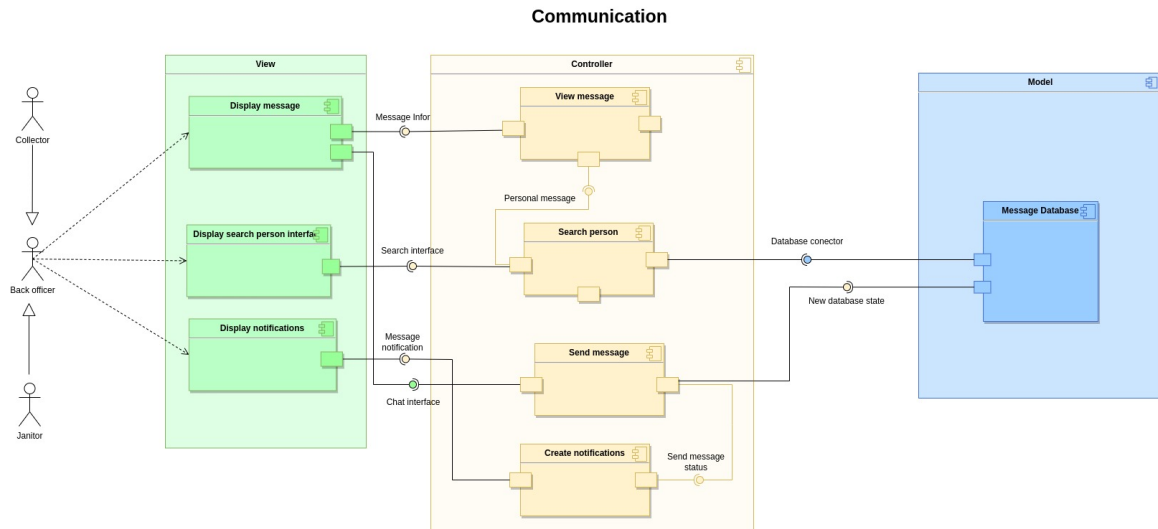
2. Component Diagram - MCP management



Mô tả

- Ta có model *Area info* chứa thông tin về các MCP trong một khu vực. *MCP's info* chứa thông tin về các MCP chưa được gán để xử lý, được cung cấp từ *Area info*. Khi người dùng muốn xem thông tin về MCP, khối controller *View MCP's* sẽ lấy thông tin tất cả các MCP trong một khu vực từ khối *Area Info*, sau đó cung cấp thông tin của MCP cần xem cho khối *Display MCP's info* để hiển thị lên.
- Khi người dùng muốn Thêm thông tin về một MCP, khối controller *Create MCP* sẽ lấy thông tin về các địa điểm cần gán để xử lý. Từ đó, sau khi xử lý, *Create MCP* cung cấp thông tin về kết quả thêm mới MCP cho khối *Display MCP's creation* để hiển thị lên màn hình, đồng thời gửi kết quả (thành công hay thất bại) về cho khối controller *Create notifications*. Tương tự với khối *Adjust MCP*, khối này sẽ gửi kết quả thay đổi thông tin của MCP về cho khối *Display MCP's adjust* để hiển thị lên cho người dùng và cũng gửi kết quả của việc thay đổi (thành công hay thất bại) về cho khối *Create notifications*, từ đó khối *Display notification* sẽ hiển thị lên thông báo kết quả của việc tạo mới/điều chỉnh. Đồng thời, cả hai khối controller này sẽ gửi kết quả điều chỉnh/tạo mới MCP về cho *Area info* để cập nhật lại.

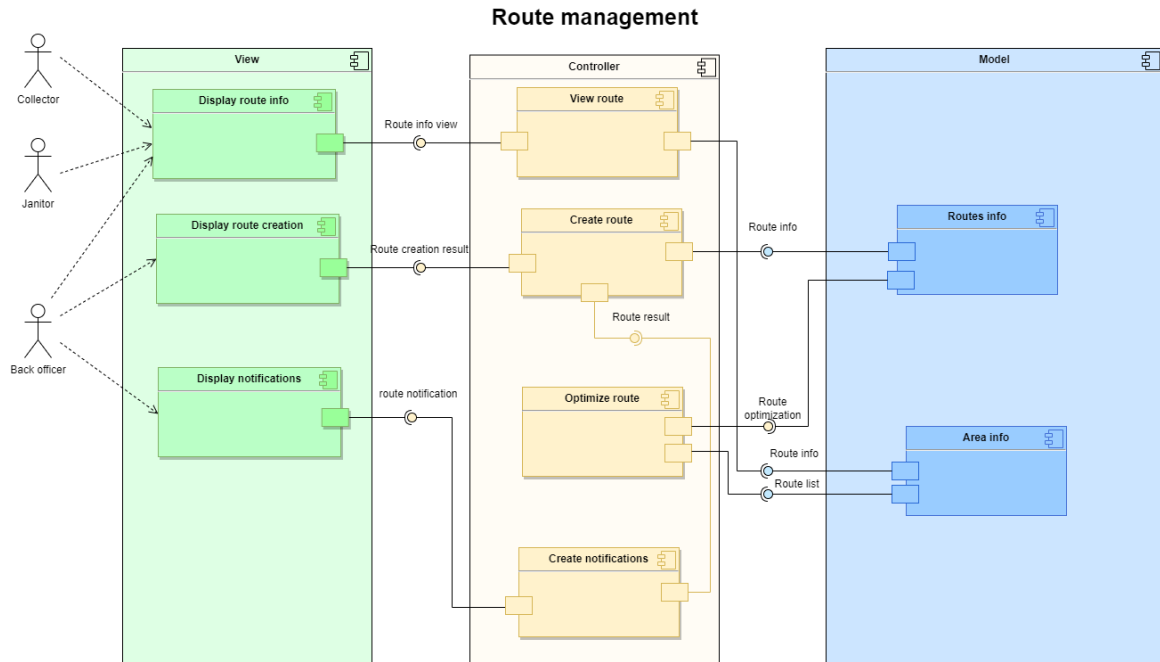
3. Component Diagram - Communication



Mô tả

- Khi người dùng sử dụng tính năng **Communication**, người dùng sẽ cần Tìm người mà mình muốn Gửi tin nhắn ở giao diện Tìm người. Khi đó, khối controller *Search person* sẽ load dữ liệu về tin nhắn từ Database thông qua interface Database connector mà model *Message Database* cung cấp.
- Khối controller *View Message* sẽ sử dụng interface Personal message được cung cấp để load lên đoạn hội thoại mà người dùng có với người được tìm kiếm, sau đó khối *Display Message* sẽ lấy Message Infor và hiển thị đoạn hội thoại lên màn hình.
- Sau đó, khối *Display Message* cũng cung cấp giao diện để nhập tin nhắn cho khối controller *Send Message*. Từ đó, *Send Message* có thể cập nhật tin nhắn mới vào Database và gửi kết quả của việc gửi tin nhắn về cho khối controller *Create notification* xử lý. Từ đó khối *Display notification* có thể hiển thị thông báo đến cả người nhận và người gửi.

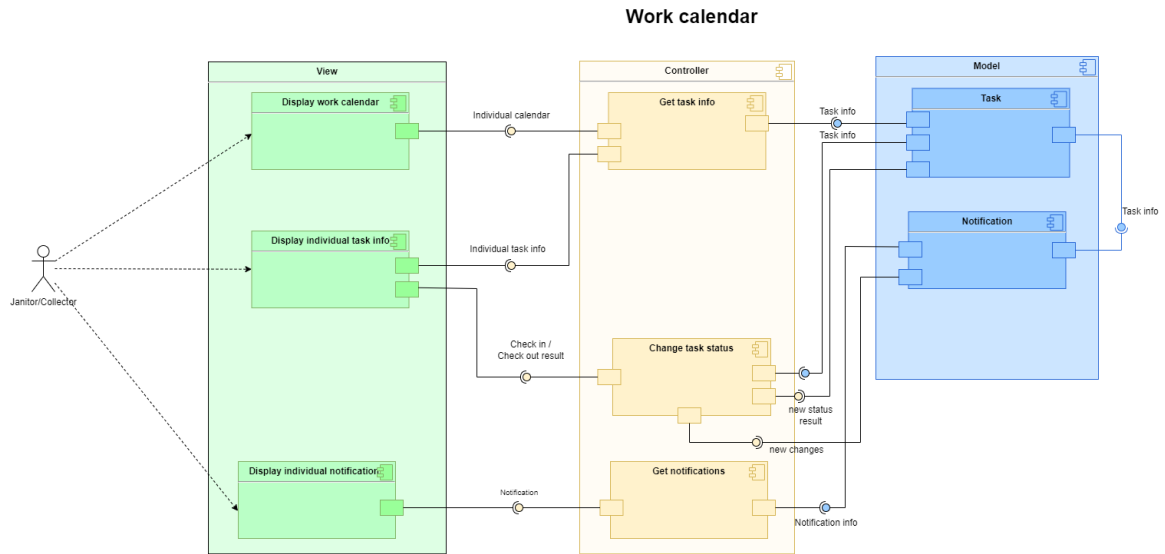
4. Component Diagram - Route management



Mô tả

- Tính năng Route Management giúp cho cả back-officer, janitor/collector quản lý lộ trình đường đi qua những điểm cần thu gom rác về một MCP nào đó. Hệ thống hiển thị giao diện Route management thông qua View *Route Info*, được Controller *View Route* lấy thông tin từ Model *Area Info* ở bên dưới Database (là khu vực cần quản lý thu gom rác về một MCP).
- Ngoài ra, hệ thống còn cho phép back-officer có thể tạo ra tuyến đường mới qua giao diện ở *Display route creation*. Controller *Create route* sẽ thực hiện tác vụ tạo ra tuyến đường mới với thông tin được lấy từ Model *Routes Info*. Hệ thống còn hỗ trợ tìm ra được tuyến đường tối ưu nhất (về mặt giao thông, quãng đường...) qua Controller *Optimize route*. Thông tin về những tuyến đường khả thi được lấy từ một danh sách các tuyến đường từ *Area info* (một khu vực có thể có nhiều tuyến đường khả thi) là chọn ra tuyến đường tối ưu nhất. Thông tin về tuyến đường đó sẽ được lưu lại bên dưới Model *Routes Info*.
- Khi có một tuyến đường mới được tạo ra từ Controller *Create route*, hệ thống sẽ gửi thông điệp tới một Controller mới tên là *Create notification*. Controller nào sẽ tạo ra một thông báo mới về tuyến đường đi cho người dùng và hiển thị ở *Display notification* cho người dùng biết.

5. Component Diagram - Work calendar



Mô tả

- Component Work Calendar mô tả work calendar dưới góc độ của janitor và collector. Janitor sẽ xem được thông tin về lịch trình làm việc của mình trên View *Display work calendar*. Khi đó, Controller *Get task info* sẽ lấy thông tin về task trong Model *Task* ở Database để hệ thống hiển thị Work calendar.
- Khi sử dụng tính năng Work Calendar, janitor và collector sẽ xem được thông tin chi tiết của một task cụ thể được hiển thị lên Work Calendar (ví dụ: giao rác tới một khu vực) qua View *Display individual task info*. Thông tin đó được Controller *Get task info* từ Model *Task* ở bên dưới phần Database. Trong quá trình làm việc, janitor/collector có thể đánh dấu một công việc nào đó được hoàn thành thông qua thao tác Check in/ Check out một task. Thao tác này sẽ gửi thông điệp đến Controller *Change task status* sẽ thay đổi trực tiếp trạng thái của một task (đã hoàn thành/ đang thực hiện/ chưa hoàn thành) ở phần Model *Task* bên dưới phần Database, đồng thời Controller *Change task status* cũng sẽ gửi dữ liệu thông báo đến khối *Notification* ở Model. Và hiển thị các thay đổi thông qua khối *Display individual task info* ở View.
- Khi có sự thay đổi về trạng thái của một task, hoặc task mới được tạo ra, thì Model *Notification*, cung cấp dữ liệu cho Controller *Get notification* và hiển thị thông báo đó qua phần *Display individual information* để người dùng có thể thấy được.

4 Task 4: Implementation – Sprint 1

4.1 Setting up. The team creates an online repository (github, bitbucket, etc) for version control. folders this stage, no need for a database to store all menu items, customers, etc. Data can be hard coded in code files.

Link GitHub: <https://github.com/loctvl842/uwc?fbclid=IwAR0AXKtJe1XXMB4Tozi7YlE61c7uTGSrbihFYnGeZQhrJuY80SzQSuFazA4>

- 4.2 Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files
- 4.3 Implement MVP1 – design an interface of either a Desktop-view central dashboard for Task Management for back-officers OR a Mobile-view Task assignment for Janitors and Collectors. Decide yourself what to include in the view. Design use a wireframe tool.